

# Bayes Optimal Shuffled Graph Classification: Applications in Statistical Connectomics

Joshua T. Vogelstein and Carey E. Priebe

**Abstract**—Graph classification algorithms often do not incorporate vertex label information in their classifiers. In this work, we investigate the extent to which discarding vertex labels can hinder classification performance, and for which random graph models it would be expected to matter. Via theory we demonstrate a collection of results. Specifically, if one “shuffles” the graphs prior to classification, the vertex label information is irretrievably lost, which can degrade misclassification performance (and does whenever the vertex labels have class-conditional signal). Thus, while one cannot hope to recover the labels, trying to recover the labels actually results in a consistent estimate of the optimal graph invariant. This approach therefore solves the question of “which invariant to use” for any graph classification problem, at least asymptotically. Via simulation we demonstrate that a finite (and small) number of training samples can be sufficient to achieve this bound. Finally, we apply this approach to a “connectome” classification problem (a connectome is the complete set of connections within a brain). Unshuffling the graphs indeed improves performance, although not over the best performance achievable composing a number of graph invariant and machine learning tools. Thus, given any unlabeled graph classification problem, the relative performance of an unshuffling approach might be difficult to predict with small sample sizes.

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



## 1 INTRODUCTION

CONSIDER the following idealized scenario. Let  $\mathbb{G} : \Omega \mapsto \mathcal{G}_n$  be a graph-valued random variable taking values  $G \in \mathcal{G}_n$ , where  $|\mathcal{G}_n| = 2^{\binom{n}{2}}$ . Let  $Y$  be a categorical random variable,  $Y : \Omega \mapsto \mathcal{Y} = \{y_1, \dots, y_c\}$ , where  $c < \infty$ , such that each graph has an associated class. Assume that graphs and classes are jointly sampled from some distribution,  $(\mathbb{G}, Y) \sim \mathbb{P}_{\mathbb{G}, Y} = \mathbb{P}$ . A graph classifier  $h : \mathcal{G}_n \mapsto \mathcal{Y}$  is any function that maps from graph space to class space. The *risk* of a classifier under 0–1 loss is the expected misclassification rate,  $L_{\mathbb{P}}(h) = \mathbb{E}_{\mathbb{P}}[h(G) \neq Y]$ . The optimal classifier is the classifier that minimizes risk:

$$h_*(G) = \operatorname{argmin}_{h \in \mathcal{H}} \mathbb{E}[h(G) \neq Y]. \quad (1)$$

where  $\mathcal{H}$  is the set of possible classifiers. Let  $L_* = L_{\mathbb{P}}(h_*)$  indicate minimal (optimal) risk.

Graph classification differs from classification of vector-valued random variables in several key aspects. First, the structure of a graph may encode information. Second, the vertex labels may or may not be observed. In unobserved scenarios, NP-hard problems rear their ugly heads [1].

## 2 GRAPH SHUFFLING

Let  $G$  and  $G'$  be isomorphic to another if and only if  $\exists Q(G) = G'$ , where  $Q : \mathcal{G}_n \mapsto \mathcal{G}_n$  is a vertex

permutation function. Similarly, let  $\mathbb{Q}$  be a permutation-valued random variable,  $\mathbb{Q} : \Omega \mapsto \mathcal{Q}_n$ , where  $\mathcal{Q}$  is the space of permutation functions on  $n$  vertices, and  $|\mathcal{Q}_n| = n!$ . Let  $\mathbb{P}_{\mathbb{Q}, \mathbb{G}, Y}$  be a *shuffled-graph/class* model.

An *unlabeled graph* is actually a set:  $\tilde{G} = \{Q(G) : \forall Q \in \mathcal{Q}_n\}$ , where  $\mathcal{Q}_n$  is the set of permutation functions on  $n$  vertices. Let  $\tilde{\mathcal{G}}_n$  be unlabeled graph space. A *shuffle channel*,  $\mathcal{C} : \mathcal{G}_n \mapsto \tilde{\mathcal{G}}_n$  is a channel that takes as input a graph, uniformly at random selects a permutation function  $Q \in \mathcal{Q}_n$ , and outputs an unlabeled graph. Let  $\tilde{h} : \tilde{\mathcal{G}}_n \mapsto \mathcal{Y}$  be an unlabeled graph classifier, and  $\tilde{L}_* = L_{\mathbb{P}}(\tilde{h})_*$  be the unlabeled optimal risk for the unlabeled optimal classifier,  $\tilde{h}_*$ , under 0–1 loss.

## 3 MODEL BASED GRAPH CLASSIFIERS

The joint distribution over graphs and classes may be decomposed into the product of a likelihood and prior term:  $\mathbb{P}_{\mathbb{G}, Y} = \mathbb{P}_{\mathbb{G}|Y} \pi_Y$ , where  $\mathbb{P}_{\mathbb{G}|Y}$  denotes the class-conditional distribution (likelihood), and  $\pi_Y$  denotes class prior probabilities.

A Bayes optimal graph classifier chooses the maximum a posteriori class [2]:

$$h_*(G) = \operatorname{argmax}_{y \in \mathcal{Y}} \mathbb{P}_{\mathbb{G}|y}(G) \pi_y, \quad (2)$$

where  $\mathbb{P}_{\mathbb{G}|y} = \mathbb{P}_{\mathbb{G}|Y=y}$  and  $\pi_y = \pi_{Y=y}$ . Let  $\tilde{\mathbb{P}} = \mathbb{P}_{\tilde{\mathbb{G}}, Y}$  indicate the joint distribution of unlabeled (or shuffled) graphs and classes. In other words,  $\tilde{\mathbb{P}}$  is the same as  $\mathbb{P}$  except that the graphs have been passed through a shuffle channel, so  $\mathbb{P}$  is a distribution over graphs, and  $\tilde{\mathbb{P}}$  is a distribution over sets of isomorphic graphs. A Bayes

• J.T. Vogelstein and C.E. Priebe are with the Department of Applied Mathematics and Statistics, Johns Hopkins University, Baltimore, MD 21218. E-mail: joshuaov@jhu.edu

optimal unlabeled graph classifier is:

$$\tilde{h}_*(\tilde{G}) = \operatorname{argmax}_{y \in \mathcal{Y}} \mathbb{P}_{\tilde{\mathbb{G}}|Y=y}(\tilde{G})\pi_y. \quad (3)$$

Extending the model to include the shuffling distribution yields  $\mathbb{P}' = \mathbb{P}_{\mathbb{Q}, \mathbb{G}, Y}$ . We assume throughout this work (with loss of generality) that the shuffling distribution is both *class independent* and *graph independent*; therefore, this joint model can be decomposed:

$$\mathbb{P}_{\mathbb{Q}, \mathbb{G}, Y} = \mathbb{P}_{\mathbb{Q}} \mathbb{P}_{\mathbb{G}|Y} \pi_Y. \quad (4)$$

A Bayes optimal shuffled graph classifier:

$$h'_*(G) = \operatorname{argmax}_{y \in \mathcal{Y}} \mathbb{P}_{\mathbb{G}|y}(G)\pi_y, \quad (5)$$

which is independent of the shuffling distribution.

Let  $L_*$ ,  $\tilde{L}_*$ , and  $L'_*$  be the Bayes risk, unlabeled Bayes risk, and shuffled Bayes risk under  $\mathbb{P}$ ,  $\tilde{\mathbb{P}}$ , and  $\mathbb{P}'$ , respectively. For brevity, we will use the shorthand  $\mathbb{P}_y = \mathbb{P}_{\mathbb{G}|Y=y}$ ,  $\tilde{\mathbb{P}}_y = \mathbb{P}_{\tilde{\mathbb{G}}|Y=y}$  and  $\mathbb{P}'_y = \mathbb{P}'_{\mathbb{G}|Y=y}$ .

### 3.1 $\mathbb{P}_y$ , $\tilde{\mathbb{P}}_y$ , and $\mathbb{P}'_y$ as Discrete Distributions

It may be illustrative to consider the class-conditional distributions of graphs as a discrete distribution:  $\mathbb{P}_y(G) = \text{Discrete}(G; \theta_y)$ , where  $\theta_y \in \Delta_d$ , where  $\Delta_d$  is the  $d$ -dimensional simplex (that is,  $\theta_{i|y} \geq 0$  for all  $i \in [d]$  and  $\sum_{i \in [d]} \theta_{i|y} = 1$ ,  $d = |\mathcal{G}_n|$  and  $[d] = \{1, 2, \dots, d\}$ ).

Similarly, the class-conditional distributions over isomorphic classes (unlabeled graphs) can be thought of as a discrete distribution:  $\mathbb{P}_{\tilde{\mathbb{G}}|y}(\tilde{G}) = \text{Discrete}(\tilde{G}; \eta_y)$ , where  $\eta_y \in \Delta_{\tilde{d}}$ , where  $\tilde{d} = |\tilde{\mathcal{G}}_n|$ . Note that in general  $\tilde{d} \ll d$ . Asymptotically,  $\tilde{d} = d/n!$ , with error monotonically decreasing and  $> 99\%$  accurate for  $n > 19$  (Ed Scheinerman, personal communication).

The shuffled graph distribution can similarly be represented by a discrete distribution:  $\mathbb{P}'_y(G) = \text{Discrete}(G; \theta'_y)$ , where  $\theta'_y \in \Delta_d$ . When  $\mathbb{P}_{\mathbb{Q}}$  is uniform,  $\theta'_y$  satisfies two additional constraints by definition: all shuffled graphs within the same isomorphism class are (i) equal to one another and (ii) proportional to the probability of that isomorphism class. Formally, (i)  $\{\theta'_{i|y} = \theta'_{j|y} \forall i, j : Q(G_i) = G_j \text{ for some } Q \in \mathcal{Q}_n\}$  and (ii)  $\{\theta'_{i|y} = \eta_{j|y}/|\tilde{G}_j| \forall i : G_i \in \tilde{G}_j\}$ .

## 4 SHUFFLING CAN DEGRADE OPTIMAL PERFORMANCE

The result of passing a graph through a shuffle channel can only degrade, but not improve Bayes risk. This is a restatement of the data processing lemma for this scenario. Specifically, [2] shows that the data processing lemma indicates that in the classification domain,  $L_X^* \leq L_{T(X)}^*$ , for any statistic  $T$  and data  $X$ . In our setting, this becomes:

**Theorem 1.**  $L_* \leq \tilde{L}_* = L'_*$ .

*Proof:* Assume  $|\mathcal{Y}| = 2$  and  $\pi_0 = \pi_1 = 1/2$ .

$$\begin{aligned} \tilde{L}_* &= \sum_{\tilde{G} \in \tilde{\mathcal{G}}_n} \min_y \tilde{\mathbb{P}}_y(\tilde{G}) = \sum_{\tilde{G} \in \tilde{\mathcal{G}}_n} \min_y \sum_{G' \in \tilde{\mathcal{G}}} \mathbb{P}_y(G') \\ &\geq \sum_{\tilde{G} \in \tilde{\mathcal{G}}_n} \sum_{G' \in \tilde{\mathcal{G}}} \min_y \mathbb{P}_y(G') = L_*. \end{aligned} \quad (6)$$

It is trivially true that  $\tilde{L}_* = L'_*$ .  $\square$

An immediate consequence of the above proof is that the inequality in Theorem 1 is a *strict inequality* whenever the inequality in Eq. 6 is strict:

**Theorem 2.**  $L_* < \tilde{L}_*$  whenever

$$\min_y \tilde{\mathbb{P}}_y(\tilde{G}) > \sum_{G' \in \tilde{\mathcal{G}}} \min_y \mathbb{P}_y(G')$$

A corollary of the above theorem is that even when the labels *do* carry some class-conditional signal, it may be the case that a shuffle channel does not degrade performance. In other words, to state that labels contain information is equivalent to stating that some graphs within an isomorphism class are class-conditionally more likely than others:  $\exists \theta_{i|y} \neq \theta_{j|y}$  and  $Q(G_i) = G_j$  for some  $G_i, G_j \in \mathcal{G}_n$  and  $Q \in \mathcal{Q}$ , and that  $\theta_{i|y} \neq \theta_{i|y'}$  or  $\theta_{j|y} \neq \theta_{j|y'}$ . Shuffling has the effect of “flattening” likelihoods within isomorphism classes,  $Q(\theta_y) = \theta'_y$  where  $\theta'$  satisfies  $\{\theta'_{i|y} = \eta_{j|y}/|\tilde{G}_j| \forall i : G_i \in \tilde{G}_j\}$ . But just because the shuffling changes class-conditional likelihoods does *not* mean that Bayes risk must also change.

This surprising (to us) result follows immediately upon realizing that posteriors can change without classification performance changing. The above results hold in the absence of equal priors, and are easily generalized to  $n$ -class classification problems. To see that, simply replace each minimum with a sum over all non-maxima:

$$\min_y \mathbb{P}_y(G) \mapsto \sum_{y \in \mathcal{Y}'} \mathbb{P}_y(G) \text{ where } \mathcal{Y}' = \{y : y \neq \operatorname{argmax}_y \mathbb{P}_y\}.$$

## 5 BAYES OPTIMAL GRAPH INVARIANT-BASED CLASSIFICATION AFTER SHUFFLING

A graph invariant is any function:  $\psi : \mathcal{G}_n \mapsto \mathbb{R}^{d'}$  for some  $d' < \infty$  such that  $\psi(G) = \psi(Q(G))$  for all  $G \in \mathcal{G}_n$  and  $Q \in \mathcal{Q}_n$ . A graph invariant based classifier,  $h^\psi$  first projects a graph into an invariant space and then classifies. The Bayes optimal graph invariant classifier minimizes risk over all invariants:

$$h_*^\psi = \operatorname{argmin}_{\psi \in \Psi, h^\psi \in \mathcal{H}^\psi} \mathbb{E}_{\mathbb{P}'}[h^\psi(\mathbb{G}) \neq Y], \quad (7)$$

where  $\Psi$  is the space of all possible invariants and  $\mathcal{H}^\psi$  is the space of classifiers using invariant  $\psi$ .  $L_*^\psi$  is the Bayes invariant risk. (NOTE TO CEP: I'm not sure that the above expectation is with respect to  $\mathbb{P}'$ , though it seems like it must be?)

**Theorem 3.**  $\tilde{L}_* = L_*^\psi$ .

*Proof:* Let  $f_{G'}(G)$  be a graph matching function, that is,  $f : \mathcal{G} \times \mathcal{G} \mapsto \{0, 1\}$ , taking value unity whenever  $G$  and  $G'$  are isomorphic to one another. Let  $\psi(G) = \mathbf{f}(G)$ , where  $\mathbf{f} = (f_{G_1}, \dots, f_{G_{\tilde{d}}})$ . In words,  $\phi(G)$  outputs a vector of length  $\tilde{d}$  with a single non-zero entry in the isomorphism class of  $G$ . Let  $h^\psi$  be an indicator function for the maximum a posteriori class of the corresponding isomorphism class. For instance, if  $G \in \tilde{\mathcal{G}}_j$ , then

$$h^\psi = \operatorname{argmax}_{y \in \mathcal{Y}} \eta_{j|y} \pi_y. \quad (8)$$

Thus, this graph invariant based classifier is the Bayes unlabeled graph classifier.  $\square$

## 6 A CONSISTENT AND EFFICIENT UNSHUFFLING-BASED CLASSIFIER

thm 4: use carey's f, prove consistency, because things are finite, law of large numbers. Since all estimates are of discrete probabilities, and there are a finite number of them, the law of large numbers implies convergence.

Section 4 shows that one cannot fruitfully “unshuffle” graphs: once they have been shuffled by a uniform shuffler, the label information is lost. Section 5 shows that if graphs have been uniformly shuffled, there is a relatively straightforward algorithm for optimally classifying. However, that classifier depends on knowing  $\mathbb{P}'$  a priori, a (typically) poor assumption. Instead of assume that we have  $\mathbb{P}'$ , we make a somewhat more reasonable assumption that a collection of data exists,  $\{(\mathbb{Q}_i, \mathbb{G}_i, Y_i)\}_{i \in [s]}$ , and each datum was independently sampled from the same distribution,  $(\mathbb{Q}_i, \mathbb{G}_i, Y_i) \stackrel{iid}{\sim} \mathbb{P}'$ . Sadly, we only observe some *training data*  $\mathcal{T}_s = \{\mathbb{G}'_i, Y_i\}_{i \in [s]}$ , where  $\mathbb{G}'_i = \mathbb{Q}_i(\mathbb{G}_i)$ , and are thusly unable to observe the vertex labels. Moreover,  $\mathbb{P}_{\mathbb{Q}}$  is uniform, such that all label information is both unavailable and irrecoverable. Our task is to induce a classifier,  $\hat{h}_s : \mathcal{G} \times (\mathcal{G} \times \mathcal{Y})^s \mapsto \mathcal{Y}$  that approximates  $\tilde{h}_*$  as closely as possible by utilizing the training data.

A Bayes plugin unlabeled graph-classifier estimates the likelihood and prior terms and plugs them in to Eq. (3)

$$\hat{h}_s(G) = \operatorname{argmax}_{y \in \mathcal{Y}} \hat{\mathbb{P}}_{\tilde{\mathcal{G}}|Y=y}(G) \hat{\pi}_y \quad (9)$$

**Theorem 4.**  $\hat{h}_s \rightarrow \tilde{h}^*$  as  $s \rightarrow \infty$

*Proof:* Given consistent estimators for  $\tilde{\mathbb{P}}_y(G)$  and  $\pi_Y$ , the Bayes plugin classifier is also consistent [2]. Formally, if  $\hat{\mathbb{P}}_y(G) \rightarrow \mathbb{P}_y(G)$  and  $\hat{\pi}_Y \rightarrow \pi_Y$  as  $s \rightarrow \infty$ , then  $\hat{h}_s \rightarrow \tilde{h}_*$  as  $s \rightarrow \infty$ . Because  $\mathcal{G}_n$  and  $\mathcal{Y}$  are both finite, the empirical histograms,  $\{\hat{\mathbb{P}}_{\tilde{\mathcal{G}}|y}\}_{y \in \mathcal{Y}}$  and  $\hat{\pi} = (\hat{\pi}_{y_0}, \dots, \hat{\pi}_{y_c})$ , are guaranteed to converge to the true histograms by law of large numbers. Thus,  $\hat{h}_s \rightarrow \tilde{h}^*$  as  $s \rightarrow \infty$ .  $\square$

## 7 A PRACTICAL APPROACH TO UNLABELED GRAPH CLASSIFICATION

Although Eq. 9 yields consistency from Theorem 4, however, utilizing this is practically hopeless. In the

unlabeled case, this requires solving an infinite number of NP-hard problems. Specifically, using Eq. ?? requires  $\mathbf{f}$  from Section 5, which is a graph matching function. Graph matching is known to be NP-hard [?]. Although there is some hope because most graph matching problems can be solved in polynomial time [?], that hope is lost upon discovering that the constant is crazy large [?]. We therefore consider an approximate approach as follows.

We assume  $(\mathbb{Q}, \mathbb{G}, Y), (\mathbb{Q}_1, \mathbb{G}_1, Y_1), \dots, (\mathbb{Q}_s, \mathbb{G}_s, Y_s) \stackrel{iid}{\sim} \mathbb{P}'$ , and we only observe  $\mathbb{G}', \mathcal{T}_s$ , where  $\mathcal{T}_s = \{(\mathbb{G}'_i, Y_i)\}_{i \in [s]}$  and  $\mathbb{G}'_i = \mathbb{Q}_i(\mathbb{G}_i)$ . Given these training data  $\mathcal{T}_s$ , we proceed as follows.

For each class,  $y \in \mathcal{Y}$ , choose an exemplar (or prototype [?]) shuffled graph,  $\mathbb{G}'_{y+} \in \{\mathbb{G}'_i\}_{i: y_i=y}$ . Then, try to “unshuffle” each shuffled graph in each class to its corresponding prototype using an inexact graph matching algorithm,  $\hat{f}_y = \hat{f}_{\mathbb{G}'_{y+}}$ . Although many inexact graph matching algorithms are available, we select the quadratic assignment problem (QAP) based approach developed in [3], which is cubic in  $n$  and outperforms the previous state-of-the-art on all benchmarks tested. This yields  $\{\hat{\mathbb{G}}_i\}$ , which we can then use to estimate the *labeled* likelihoods,  $\mathbb{P}_{\mathbb{G}|Y=y}$ .

Now, although it may seem as we projected our graphs from a space of cardinality  $\tilde{d}$  to  $\tilde{d} \times n!$ , given the labeled graphs, a variety of graph classification algorithms are available. We choose to use the independent edge graph classifier developed in [4]. Under the independent edge assumption, the likelihood of an edge is given by  $\mathbb{P}[A_{uv}|Y=y] = \mathbb{P}_{uv|y}$ , yielding a likelihood factorization:  $\mathbb{P}_y = \prod_{uv} \mathbb{P}_{uv|y}$ . Thus, the class-conditional distribution is given by a matrix,  $P_y \in (0, 1)^{n \times n}$ . The labeled graphs,  $\{\mathbb{G}_i\}$  can be used to estimate these likelihood matrices,  $\{P_y\}$ , as described in [4] using L-estimators to bound the estimates from the boundaries. Now, we can use a Bayes plugin labeled graph classifier via the class-conditional posterior estimates:

$$\hat{y} = \operatorname{argmax}_{y \in \mathcal{Y}} \hat{\rho}_y \text{ where } \hat{\rho}_y = \hat{\mathbb{P}}_{uv|y}(\hat{f}_y(G)) \hat{\pi}_y. \quad (10)$$

To gain some intuition with regard to why this approach would be successful, consider the following simple yet illustrative example. Let each  $\tilde{\mathbb{P}}_y$  puts all its mass on a single but distinct isomorphism class, so  $\eta_{j|y_j} = 1$  and  $\eta_{j'|y_j} = 0 \forall j' \neq j$ . Thus, if  $\hat{f}_y$  worked, then every edge present in a class would have unity likelihood, and all other edges would have zero likelihood. Thus, a single class will have a posterior estimate equal to unity, which will therefore be the maximum a posteriori class. This simple example can be extended without modification to allow each class have a non-overlapping set of non-zero likelihoods. Moreover, reverting to the simple two-class and two non-zero likelihood scenario, consider a different generalization.

## 8 SIMULATED EXPERIMENT

To demonstrate the practicality of an isomorphism-based unlabeled graph classifier, we conduct the following simulated experiment. Sample  $s + 1$  triplets identically and independently from the joint shuffler/graph/class model,  $(Q_i, G_i, Y_i) \stackrel{iid}{=} \mathbb{P}' = \mathbb{P}_Q \mathbb{P}_{G|Y} \pi_Y$ , where  $\mathbb{P}_Q$  is uniform, and  $\pi_Y$  is Bernoulli so  $\pi_0 = \pi_1 = 1/2$ . The edges are independent so the likelihood factorizes as above. Specifically, we let  $\mathbb{P}_{uv|0} = \text{Bernoulli}(p_{uv|y})$ , where  $p_{uv|0} \stackrel{indep}{\sim} \text{Uniform}(0, 0.7)$ , and  $p_{uv|1} = p_{uv|0} + 0.3$ .

Figure 1 shows the performance of this Bayes plugin classifier as a function of the number of training samples. As we had hoped, performance monotonically increases towards optimal performance (gray dot), even though the graph matching algorithm we used was approximate, and only  $\mathcal{O}(n^3)$  instead of exponential.

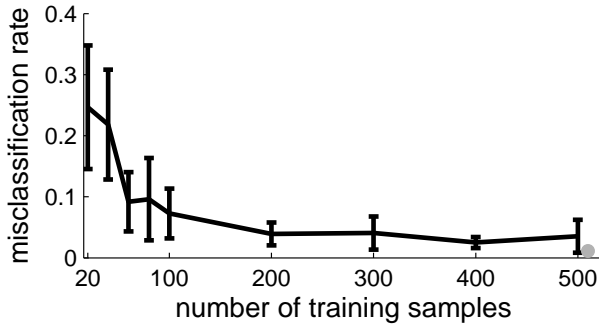


Fig. 1. Inexact graph matching can be used to approximate a consistent shuffled graph classifier. Data in this simulation was sampled from the independent edge model described above. For each number of training samples, we tested using 5000 test samples, and repeated 10 times. The gray dot indicates Bayes optimal performance. (NOTE TO CEP: actually, i forgot to QAP the test data to each training class in this example. i think it would converge much “faster” if i included that step. that is, faster in  $s$ , but now testing requires performing 2 QAPs, whereas before, it did not, so it might actually take longer. we will see soon, as i’m running that now.)

## 9 UNLABELED CONNECTOME CLASSIFICATION

Inspired by the simulated performance of our unlabeled graph classifier, we decided to try it on a real-world application. A “connectome” is a graph in which vertices correspond to biological neural units, and edges correspond to connections between the units. Diffusion Magnetic Resonance (MR) Imaging and related technologies are making the acquisition of MR connectomes routine [5]. 49 subjects from the Baltimore Longitudinal Study on Aging comprise this data, with acquisition and connectome inference details as reported in [6]. Each connectome yields a  $70 \times 70$  element binary adjacency matrix. Associated with each graph is class label based

on the gender of the individual (24 males, 25 females). Because the vertices are labeled, we can compare the results of having the labels and not having the labels. A  $k_n$  nearest neighbor ( $knn$ ) classifier is universally consistent, that is, guaranteed to achieve optimal performance in the limit [7], and therefore seems more appropriate than an independent edge model. Performance is evaluated with leave-one-out misclassification rate and reported in Table 1. When using the vertex labels, a standard  $knn$  achieves 20% misclassification rate. Chance performance (only using the estimated prior) on this data is 49%. These two numbers provide bounds on performance. When all graphs are passed through a shuffle channel, we first try to unshuffle the graphs using the above mentioned QAP algorithm. Given the unshuffled graphs, performance changes to 45%, not particularly impressive. The performance of the independent edge model based Bayes plugin classifier for unlabeled graphs is similarly unimpressive. We therefore develop a hybrid approach in which the independent edge model is assumed, and parameters are estimated using the vertex labels. Given these estimates, we can use the QAP algorithm to match each test graph to the two likelihood matrices, and then use the Bayes plugin classifier. This approach yields a 31% misclassification rate. In contrast, a “standard” graph invariant based approach, which computes the graph invariants from [8], and plugs them into various machine learning algorithms (including the winner [9]), yields misclassification rates as low as 25%.

TABLE 1  
MR Connectome Leave-One-Out Misclassification Rates

N/A-QAP	1-QAP	48-QAP	1NN-GI
20%	31%	45%	25%

## 10 DISCUSSION

In this work, we have address both the theoretical and practical limitations of classifying graphs with and without including labels. Specifically, we show that shuffling the vertex labels results in an irretrievable situation, with a possible degradation of classification performance, and a necessary degradation if the vertex labels contained class-conditional signal. Moreover, although one cannot hope to recover the vertex labels, estimating them yields an asymptotically optimal classifier. This suggest that efforts to estimate the vertex labels may yield useful classification results, outperforming “standard” graph-invariant based classifiers. Via simulation we show that an approximate graph matching algorithm converges to the optimal performance with only about 500 training samples for a particular independent edge random graph model. Finally, we demonstrate with connectome data that estimating the vertex labels can be useful, but that there remains room to grow to exceed misclassification performance of a carefully chosen graph invariant  $\circ$  machine learning based approach on this

data. These connectome data, much like other collections of graphs, can also be equipped with both vertex and edge attributes. As such, we hope to extend the results herein to consider the more general cases.

## ACKNOWLEDGMENTS

## REFERENCES

- [1] D. Conte, P. Foggia, C. Sansone, and M. Vento, "THIRTY YEARS OF GRAPH MATCHING," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 18, no. 3, pp. 265–298, 2004.
- [2] L. Devroye, L. Györfi, and G. Lugosi, *A probabilistic theory of pattern recognition*. New York: Springer-Verlag, 1996.
- [3] J. T. Vogelstein and Priebe, "conroy's QAP," *Submitted for publication*, 2011.
- [4] J. T. Vogelstein and C. E. Priebe, "ind edge class paper," *Submitted for publication*, 2011.
- [5] 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," *J Neurosci Methods*, vol. 194, no. 1, pp. 34–45, 2010. [Online]. Available: [http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list\\_uids=20096730](http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=20096730)
- [6] W. R. Gray, J. T. Vogelstein, and R. J. Vogelstein, "Mr. Cap," *Submitted for publication*, 2011.
- [7] C. J. Stone, "Consistent Nonparametric Regression," *The Annals of Statistics*, vol. 5, no. 4, pp. 595–620, Jul. 1977. [Online]. Available: <http://projecteuclid.org/euclid.aos/1176343886>
- [8] H. Pao, G. A. Coppersmith, and C. E. Priebe, "Statistical Inference on Random Graphs : Comparative Power Analyses via Monte Carlo," 2010.
- [9] K. Crammer, M. Dredze, and F. Pereira, "Exact Convex Confidence-Weighted Learning," *Journal of Multivariate Analysis*, vol. 99, no. 5, pp. 1–8, 2008. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0047259X07000784>

PLACE  
PHOTO  
HERE

**Joshua T. Vogelstein** is a spritely young man, engorged in a novel post-buddhist metaphor.

PLACE  
PHOTO  
HERE

**Carey E. Priebe** Buddha in training.