Shuffled Graph Classification: Theory and Connectome Applications

Joshua T. Vogelstein and Carey E. Priebe

Abstract—In this work, we investigate the extent to which shuffling vertex labels can hinder classification performance, and for which random graph models one might expect this shuffling to be impactful. 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 classification performance (and often does). A specific graph-invariant classifier is shown to be Bayes optimal. Moreover, this classifier may be induced by training data in a consistent and efficient fashion. Unfortunately, both computational and sample size burdens make this "plugin" classifier impractical. A graph-matched Frobenius norm k_s nearest neighbor (GM-k_sNN) classifier, however, is also universally consistent, and expected to converge faster whenever "nearness" implies same class. Finally, we apply this approach to a connectome classification problem (a connectome is brain-graph where vertices correspond to (groups of) neurons and edges correspond to connections between them). An approximate GM-k_sNN classifier on the shuffled graphs performs better than a typical graph-invariant based k_sNN strategy, but not quite as well as the k_sNN on the labeled graphs. Thus, we demonstrate the practical utility of the theoretical derivations herein. Extending these results to weighted and (certain) attributed random graph models is straightforward.

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

1 Introduction

Representing data as graphs is becoming increasingly popular, as technological progress facilitates measuring "connectedness" in a variety of domains, including social networks, trade-alliance networks, and brain networks. While the theory of pattern recognition is deep [1], previous theoretical efforts regarding pattern recognition almost invariably assumed data are collections of vectors. Here, we assume data are collections of graphs (where each graph is a set of vertices and a set of edges connecting the vertices). For some data sets, the vertices of the graphs are *labeled*, that is, one can identify the vertex of one graph with a vertex of the others. For others, the labels are unobserved and/or assumed to not exist. We investigate the theoretical and practical implications of the absence of vertex labels.

These implications are especially important in the emerging field of "connectomics", the study of connections of the brain [2], [3]. In connectomics, one represents the brain as a graph (a brain-graph), where vertices correspond to (groups of) neurons and edges correspond to connections between them. In the lower part of the evolutionary hierarchy (e.g., worms and flies), many neurons have been assigned labels [4]. However, for even the simplest vertebrates, vertex labels are mostly unavailable when vertices correspond to neurons.

Thus, it seems classification of brain-graphs is likely to become increasingly popular. Although previous work has demonstrated some possible strategies of graph classification in both the labeled [5] and unlabeled [6] scenarios, relatively little work has compared the theoretical limitations of the two. We therefore develop a random graph model amenable to such theoretical investigations. The theoretical results lead to practical universally consistent graph classification algorithms. We demonstrate that these algorithms have desirable finite sample properties via simulation and synthetic data analysis.

2 Graph Classification Models

2.1 A labeled graph classification model

A (labeled) graph $G = (\mathcal{V}, \mathcal{E})$ consists of a vertex set, $\mathcal{V} = [n]$, where $n < \infty$ is the number of vertices and $[n] = \{1, \ldots, n\}$, and an edge set $\mathcal{E} \subseteq \binom{[n]}{2}$. Let $\mathbb{G} \colon \Omega \to \mathcal{G}_n$ be a graph-valued random variable taking values $G \in \mathcal{G}_n$, where $\mathcal{G}_n$ is the set of graphs on n vertices, and $|\mathcal{G}_n| = 2^{\binom{n}{2}} = d_n$. Let Y be a categorical random variable, $Y \colon \Omega \to \mathcal{Y} = \{y_0, \ldots, y_c\}$, where $c < \infty$. Assume the existence of a joint distribution, $\mathbb{P}_{\mathbb{G},Y}$ which can be decomposed into the product of a class-conditional distribution (likelihood) $\mathbb{P}_{\mathbb{G}|Y}$ and a class prior π_Y. Because n is finite, the class-conditional distributions $\mathbb{P}_{\mathbb{G}|Y=y} = \mathbb{P}_{\mathbb{G}|y}$ can be considered discrete distributions Discrete$(G; \boldsymbol{\theta}_y)$, where $\boldsymbol{\theta}_y$ is an element of the d_n-dimensional unit simplex $\triangle_{d_n}$ (satisfying $\theta_{G|y} \geq 0$ $\forall G \in \mathcal{G}_n$ and $\sum_{G \in \mathcal{G}_n} \theta_{G|y} = 1$).

2.2 A shuffled graph classification model

In the above, it was implicitly assumed that the vertex labels were observed. However, in certain situations (such

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

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

as the motivating connectomics example presented in Section 1), this assumption is unwarranted. To proceed, we define two graphs $G, G' \in \mathcal{G}_n$ to be isomorphic if and only if there exists a vertex permutation (shuffle) function $Q \colon \mathcal{G}_n \to \mathcal{G}_n$ such that $Q(G) = G'$. Let $\mathbb{Q}$ be a permutation-valued random variable, $\mathbb{Q} \colon \Omega \to \mathcal{Q}_n$, where $\mathcal{Q}_n$ is the space of vertex permutation functions on n vertices so that $|\mathcal{Q}_n| = n!$. Extending the model to include this vertex shuffling distribution yields $\mathbb{P}_{\mathbb{Q},\mathbb{G},Y}$. 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 as

$$\mathbb{P}_{\mathbb{Q},\mathbb{G},Y} = \mathbb{P}_{\mathbb{Q}}\mathbb{P}_{\mathbb{G},Y} = \mathbb{P}_{\mathbb{Q}}\mathbb{P}_{\mathbb{G}|Y}\pi_Y. \tag{1}$$

As in the labeled case, the shuffled graph class-conditional distributions $\mathbb{P}_{\mathbb{Q}(\mathbb{G})|y}$ can be represented by discrete distributions Discrete$(G; \boldsymbol{\theta}'_y)$, where again $\boldsymbol{\theta}'_y \in \triangle_{d_n}$. When $\mathbb{P}_{\mathbb{Q}}$ is uniform on $\mathcal{Q}_n$, all shuffled graphs within the same isomorphism set are equally likely; that is $\{\theta'_{G_i|y} = \theta'_{G_j|y} \, \forall G_i, G_j \colon Q(G_i) = G_j \text{ for some } Q \in \mathcal{Q}_n\}$.

2.3 An unlabeled graph classification model

Let $\widetilde{\mathcal{G}}_n$ be the collection of isomorphism sets. An *unlabeled graph* $\widetilde{G}$ is an element of $\widetilde{\mathcal{G}}_n$. The number of unlabeled graphs on n vertices is $|\widetilde{\mathcal{G}}_n| = \widetilde{d}_n \approx d_n/n!$ (see [7] and references therein). An *isomorphism function* $U \colon \mathcal{G}_n \to \widetilde{\mathcal{G}}_n$ is a function that takes as input a graph and outputs the corresponding unlabeled graph. Let $\widetilde{\mathbb{G}} \colon \Omega \to \widetilde{\mathcal{G}}_n$ be an unlabeled graph-valued random variable taking values $\widetilde{G} \in \widetilde{\mathcal{G}}_n$. The joint distribution over unlabeled graphs and classes is therefore $\mathbb{P}_{\widetilde{\mathbb{G}},Y} = \mathbb{P}_{U(\mathbb{G}),Y} = \mathbb{P}_{U(\mathbb{Q}(\mathbb{G})),Y}$, which decomposes as $\mathbb{P}_{\widetilde{\mathbb{G}}|Y}\pi_Y$. The class-conditional distributions $\mathbb{P}_{\widetilde{\mathbb{G}}|y}$ over isomorphism sets (unlabeled graphs) can also be thought of as discrete distributions Discrete$(\widetilde{G}; \widetilde{\boldsymbol{\theta}}_y)$ where $\widetilde{\boldsymbol{\theta}}_y \in \triangle_{\widetilde{d}_n}$ are vectors in the $\widetilde{d}_n$-dimensional unit simplex. Comparing shuffling and unlabeling for the independent and uniform shuffle distribution $\mathbb{P}_{\mathbb{Q}}$, we have $\{\theta'_{G|y} = \widetilde{\theta}_{\widetilde{G}|y}/|\widetilde{G}| \text{ for all } G \in \widetilde{G}\}$.

3 Bayes Optimal Graph Classifiers

We consider graph classification in the three scenarios described above: labeled, shuffled, and unlabeled. To proceed, in each scenario we define three mathematical objects: (i) a classifier, (ii) the Bayes optimal classifier, and (iii) the Bayes risk.

3.1 Bayes Optimal Labeled Graph Classifiers

A *labeled graph classifier* $h \colon \mathcal{G}_n \to \mathcal{Y}$ is any function that maps from labeled graph space to class space. The risk of a labeled graph classifier h under $0-1$ loss is the expected misclassification rate $L(h) = \mathbb{E}[h(\mathbb{G}) \neq Y]$, where the expectation is taken against $\mathbb{P}_{\mathbb{G},Y}$.

The *labeled graph Bayes optimal classifier* is given by

$$h_* = \operatorname*{argmin}_{h \in \mathcal{H}} L(h), \tag{2}$$

where $\mathcal{H}$ is the set of possible labeled graph classifiers.

The *labeled graph Bayes risk* is given by

$$L_* = \min_{h \in \mathcal{H}} L(h), \tag{3}$$

where L_* implicitly depends on $\mathbb{P}_{\mathbb{G},Y}$.

3.2 Bayes Optimal Shuffled Graph Classifiers

A *shuffled graph classifier* is also any function $h \colon \mathcal{G}_n \to \mathcal{Y}$ (note that the set of shuffled graphs is the same as the set of labeled graphs). However, by virtue of the input being a shuffled graph as opposed to a labeled graph, the shuffled risk under $0-1$ loss is given by $L'(h) = \mathbb{E}[h(\mathbb{Q}(\mathbb{G})) \neq Y]$, where the expectation is taken against $\mathbb{P}_{\mathbb{Q}(\mathbb{G}),Y}$.

The *shuffled graph Bayes optimal classifier* is given by

$$h'_* = \operatorname*{argmin}_{h \in \mathcal{H}} L'(h), \tag{4}$$

where $\mathcal{H}$ is again the set of possible labeled (or shuffled) graph classifiers. The *shuffled graph Bayes risk* is given by

$$L'_* = \min_{h \in \mathcal{H}} L'(h), \tag{5}$$

where L'_* implicitly depends on $\mathbb{P}_{\mathbb{Q}(\mathbb{G}),Y}$.

3.3 Bayes Optimal Unlabeled Graph Classifiers

An *unlabeled* graph classifier $\widetilde{h} \colon \widetilde{\mathcal{G}}_n \to \mathcal{Y}$ is any function that maps from unlabeled graph space to class space. The risk under $0-1$ loss is given by $\widetilde{L}(\widetilde{h}) = \mathbb{E}[\widetilde{h}(\widetilde{\mathbb{G}}) \neq Y]$, where the expectation is taken against $\mathbb{P}_{\widetilde{\mathbb{G}},Y}$.

The *unlabeled graph Bayes optimal classifier* is given by

$$\widetilde{h}_* = \operatorname*{argmin}_{\widetilde{h} \in \widetilde{\mathcal{H}}} L(\widetilde{h}), \tag{6}$$

The *unlabeled graph Bayes risk* is given by

$$\widetilde{L}_* = \min_{\widetilde{h} \in \widetilde{\mathcal{H}}} L(\widetilde{h}), \tag{7}$$

where $\widetilde{\mathcal{H}}$ is the set of possible unlabeled graph classifiers and $\widetilde{L}_*$ implicitly depends on $\mathbb{P}_{\widetilde{\mathbb{G}},Y}$.

3.4 Parametric Classifiers

The three Bayes optimal graph classifiers can be written explicitly in terms of their model parameters:

$$h_*(G) = \operatorname*{argmax}_{y \in \mathcal{Y}} \theta_{G|y}\pi_y, \tag{8}$$

$$h'_*(G) = \operatorname*{argmax}_{y \in \mathcal{Y}} \theta'_{G|y}\pi_y, \tag{9}$$

$$\widetilde{h}_*(\widetilde{G}) = \operatorname*{argmax}_{y \in \mathcal{Y}} \widetilde{\theta}_{\widetilde{G}|y}\pi_y. \tag{10}$$

4 Theoretical Results

4.1 Shuffling Can Degrade Optimal Performance

The result of either shuffling or unlabeling a graph can only degrade, but not improve Bayes risk. This is a restatement of the data processing lemma for this scenario. Specifically, [1] shows that the data processing lemma indicates that in the classification domain $L^*_X \leq L^*_{T(X)}$ for any transformation T and data X. In our setting, this becomes:

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

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

$$\begin{aligned}\widetilde{L}_* &= \sum_{\widetilde{G} \in \widetilde{\mathcal{G}}_n} \min_y \widetilde{\theta}_{\widetilde{G}|y} = \sum_{\widetilde{G} \in \widetilde{\mathcal{G}}_n} \min_y \sum_{G \in \widetilde{G}} \theta'_{G|y} = L'_* \\ &= \sum_{\widetilde{G} \in \widetilde{\mathcal{G}}_n} \min_y \sum_{G \in \widetilde{G}} \theta_{G|y} \geq \sum_{\widetilde{G} \in \widetilde{\mathcal{G}}_n} \sum_{G \in \widetilde{G}} \min_y \theta_{G|y} = L_*. \quad (11)\end{aligned}$$

□

An immediate consequence of the above proof is that the inequality in the statement of Theorem 1 strict whenever the inequality in Eq. (11) is strict:

Theorem 2. *$L_* < \widetilde{L}_* = L'_*$ if and only if there exists $\widetilde{G}$ such that*

$$\min_y \widetilde{\theta}_{\widetilde{G}|y} > \sum_{G \in \widetilde{G}} \min_y \theta_{G|y}.$$

The above result demonstrates that even when the labels *do* carry some class-conditional signal, it may be the case that shuffling or unlabeling does not degrade performance. In other words, to state that labels contain information is equivalent to stating that some graphs within an isomorphism set are class-conditionally more likely than others: $\exists\, \theta_{G_i|y} \neq \theta_{G_j|y}$ where $Q(G_i) = G_j$ for some $G_i, G_j \in \mathcal{G}_n$, $Q \in \mathcal{Q}_n$, and $y \in \mathcal{Y}$. Shuffling has the effect of "flattening" likelihoods within isomorphism sets, from $\boldsymbol{\theta}_y$ to $\boldsymbol{\theta}'_y$, so that $\boldsymbol{\theta}'_y$ satisfies $\{\theta'_{G|y} = \widetilde{\theta}_{\widetilde{G}|y}/|\widetilde{G}| \, \forall \colon G \in \widetilde{G}\}$. But just because the shuffling changes class-conditional likelihoods does *not* mean that Bayes risk must also change. This result follows immediately upon realizing that posteriors can change without classification performance changing. The above results are easily extended to consider non-equal class priors and c-class classification problems. To see this, ignoring ties, simply replace each minimum likelihod with a sum over all non-maximum posteriors:

$$\min_y \theta_{G|y}\pi_y \mapsto \sum_{y \in \mathcal{Y}'} \theta_{G|y}\pi_y \text{ where } \mathcal{Y}' = \{y \colon y \neq \operatorname*{argmax}_y \theta_{G|y}\}.$$

4.2 Bayes Optimal Graph Invariant Classification After Shuffling

A graph invariant on $\mathcal{G}_n$ is any function ψ such that $\psi(G) = \psi(Q(G))$ for all $G \in \mathcal{G}_n$ and $Q \in \mathcal{Q}_n$. A graph invariant classifier is a composition of a classifier with an invariant function, $h^\psi = f^\psi \circ \psi$. The Bayes optimal graph invariant classifier minimizes risk over all invariants:

$$h^\psi_* = \operatorname*{argmin}_{\psi \in \Psi, f^\psi \in \mathcal{F}^\psi} \mathbb{E}[f(\psi(\mathbb{G})) \neq Y], \qquad (12)$$

where Ψ is the space of all possible invariants and $\mathcal{F}^\psi$ is the space of classifiers composable with invariant ψ. The expectation in Eq. (12) is taken against $\mathbb{P}_{\mathbb{G},Y}$ or equivalently $\mathbb{P}_{\mathbb{Q}(\mathbb{G}),Y}$, since invariants are invariant. Let L^ψ_* denote the Bayes invariant risk.

Theorem 3. $\widetilde{L}_* = L^\psi_*$.

Proof: Let ψ indicate in which equivalence set G resides; that is, $\psi(G) = \widetilde{G}$ if and only if $G \in \widetilde{G}$. Then

$$h^\psi_*(G) = \operatorname*{argmax}_{y \in \mathcal{Y}} \widetilde{\theta}_{\psi(G)|y}\pi_y = \operatorname*{argmax}_{y \in \mathcal{Y}} \widetilde{\theta}_{\widetilde{G}|y}\pi_y = \widetilde{h}_*(G). \qquad (13)$$

□

4.3 A universally consistent unshuffling classifier

Section 4.1 shows that one cannot fruitfully "unshuffle" graphs: once they have been shuffled by a uniform shuffler, any label information is lost. Section 4.2 shows that if graphs have been uniformly shuffled, there is a relatively straightforward algorithm for optimal classification. However, that classifier depends on knowing the parameters, $\widetilde{\boldsymbol{\theta}} = \{\widetilde{\boldsymbol{\theta}}_y\}_{y \in \mathcal{Y}}$ and $\boldsymbol{\pi} = \{\pi_y\}_{y \in \mathcal{Y}}$. Instead we consider the data are sampled identically and independently from some unknown joint distribution: $(\mathbb{Q}_i, \mathbb{G}_i, Y_i) \overset{iid}{\sim} \mathbb{P}_{\mathbb{Q},\mathbb{G},Y}$. For shuffled graph classification we observe only the *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 useful vertex labels. Moreover, $\mathbb{P}_\mathbb{Q}$ is uniform, so that all label information is both unavailable and irrecoverable. Our task is to utilize training data to induce a classifier $\hat{h}_s \colon \mathcal{G}_n \times (\mathcal{G}_n \times \mathcal{Y})^s \to \mathcal{Y}$ that approximates $\widetilde{h}_*$ as closely as possible.

An unlabeled graph Bayes *plugin* classifier estimates the likelihood and prior terms and plugs them in to Eq. (10):

$$\hat{h}_s(\widetilde{G}) = \operatorname*{argmax}_{y \in \mathcal{Y}} \hat{\widetilde{\theta}}_{\widetilde{G}|y}\hat{\pi}_y. \qquad (14)$$

Let $\hat{L}_s = L(\hat{h}_s)$ be the risk of the induced classifier using maximum likelihood to obtain the plugin estimate for Eq. (14).

Theorem 4. *$\hat{L}_s \to \widetilde{L}_*$ as $s \to \infty$.*

Proof: Because $\widetilde{\mathcal{G}}_n$ and $\mathcal{Y}$ are both finite, their respective maximum likelihood estimates are guaranteed consistent by the law of large numbers. Hence, the unlabeled graph Bayes plugin classifier is also consistent to $\widetilde{L}_*$ [8]. Note that this classifier is universally consistent, meaning that it converges to $\widetilde{L}_*$ regardless of the true joint distribution, $\mathbb{P}_{\mathbb{Q}(\mathbb{G}),Y}$. □

4.4 k Nearest Neighbor Universally Consistent Shuffled Graph Classifiers

Although Eq. (14) yields consistency from Theorem 4, utilizing this is practically hopeless as it requires solving s computationally difficult graph isomorphism problems, and acceptable performance will typically require $s \gg \widetilde{d}_n$. Specifically, using Eq. (14) requires first enumerating all $\widetilde{d}_n$ isomorphism sets, then determining in which isomorphism set the to-be-classified graph and each of the training graphs reside. This approach is therefore, in general, impractical because the number of parameters to estimate, $\widetilde{d}_n$, is too large. We therefore consider some modifications.

A k_s nearest-neighbor classifier using Euclidean norm distance is universally consistent to L_* for vector-valued data as long as $k_s \to \infty$ with $k_s/s \to 0$ as $s \to \infty$ [9]. This non-parametric approach circumvents the need to estimate many parameters in high-dimensional settings such as graph-classification. [10] extended this proof to the graph domain, which we include here for completeness. Specifically, to compare labeled graphs, they considered a Frobenius norm distance

$$\delta(G_i, G_j) = \|A_i - A_j\|_F^2 , \qquad (15)$$

where A_i is the adjacency matrix representation of the labeled graph, G_i. Letting $\hat{L}_{k_s NN}$ indicate the misclassification rate for the Frobenius norm k_sNN classifier, [10] showed:

Theorem 5. *$\hat{L}_{k_s NN} \to L_*$ as $s \to \infty$.*

Proof: Because both $\mathcal{G}$ and $\mathcal{Y}$ have finite cardinality, the law of large numbers ensures that eventually as $s \to \infty$, the plurality of nearest neighbors to a test graph will be identical to the test graph. □

Let $\hat{L}'_{k_s NN}$ indicate the misclassification rate of the Frobenius-norm k_sNN on *shuffled* graphs. From the fact that the number of shuffled graphs is *equal to* the number of labeled graphs, $|\mathcal{G}'| = |\mathcal{G}|$, the below corollary follows immediately:

Corollary 1. *$\hat{L}'_{k_s NN} \to \widetilde{L}_*$ as $s \to \infty$.*

The number of unlabeled graphs is vastly less than the number of labeled or shuffled graphs, $|\mathcal{G}'| \approx |\mathcal{G}|/n!$. Therefore, given that we observed only labeled or shuffled graphs, but not unlabeled graphs, we consider the "graph-matched Frobenius norm" distance

$$\delta'(G_i, G_j) = \operatorname*{argmin}_{Q \in \mathcal{Q}_n} \|Q(G_i) - G_j\|_F^2 . \qquad (16)$$

Let $\hat{L}'_{GM-k_s NN}$ indicate the misclassification rate of the k_sNN classifier using the above graph-matched norm. Given an exact graph matching function—a function that actually solves Eq. (16)—we have the following result

Corollary 2. *$\hat{L}'_{GM-k_s NN} \to \widetilde{L}_*$ as $s \to \infty$.*

Because $|\widetilde{\mathcal{G}}| \ll |\mathcal{G}|$, it must be that the rate of convergence for the graph-matched norm k_sNN is faster than the pure Frobenius norm k_sNN. We prove an even stronger result: the expected misclassification rate for any *finite* number of samples is lower upon using the graph-matched norm than using the pure Frobenius norm, assuming the same sequence $\{k_s\}_{s \in \mathbb{Z}}$ is used for both:

Theorem 6. *$\mathbb{E}[L'_{GM-k_s NN}] < \mathbb{E}[L'_{k_s NN}]$ for all s.*

Proof: The expected number of shuffled graphs that are identical to a test shuffled graph G' in each class is equal to $\theta_{G'|y} s_y$, where s_y is the number of graphs in class y. Thus, the expected number of mistakes is simply

$$\mathbb{E}\Big[\sum_{i \in [s]} y \neq \hat{y}\Big] = \min_y \widetilde{\theta}_{G|y} s_y . \qquad (17)$$

Let the unlabeled graph corresponding to $\widetilde{G}$ be G'. The number of mistaken unlabeled graphs is $\approx \mathbb{E}\Big[\sum_{i \in [s]} y \neq \hat{y}\Big]/n!$. □

5 Numerical Results

The above theoretical results suggest that, given a collection of graphs, a graph-matched Frobenius norm k_sNN classifier will achieve optimal performance, given sufficient data. To investigate the convergence rates in practical terms, we conduct a couple numerical experiments comparing the performance of four classifiers:

- **Labeled-1NN:** A 1-nearest neighbor (1NN) with Frobenius norm distance on the labeled adjacency matrices.
- **GM-1NN:** A 1NN with an *approximately* graph-matched Frobenius norm distance on the shuffled adjacency matrices. Approximate because graph-matching is NP-hard [11]. Therefore, we instead use an inexact graph matching approach based on the quadratic assignment formulation described in [12], which is only cubic in n.
- **ϕ-1NN:** A 1NN with Euclidean distance on four graph invariants: size, max-degree, a greedy approximation to maximum-average-degree, and scan-statistic (see [13] for details). Prior to computing the Euclidean distance, for each invariant, we rescale all the values to lie between zero and one.
- **Chance:** Classify all graphs according to which ever class is more frequent in the training data.

Performance is assessed by misclassification rate.

5.1 Simulation

We conduct the following simulated experiment. Generate s samples identically and independently from the joint shuffler/graph/class model, $(\mathbb{Q}_i, \mathbb{G}_i, Y_i) \overset{iid}{=} \mathbb{P}_{\mathbb{Q}} \mathbb{P}_{\mathbb{G}|Y} \pi_Y$, where $\mathbb{P}_{\mathbb{Q}}$ is uniform, and π_Y is Bernoulli so $\pi_0 = \pi_1 = 1/2$. We assume a binary independent edge model, so the likelihood factorizes

$$\theta_{G|y} = \prod_{(u,v) \in \mathcal{E}} \text{Bernoulli}(p_{uv|y}). \qquad (18)$$

To choose the $p_{uv|y}$'s, we sample $p_{uv|1} \overset{indep.}{\sim} \text{Uniform}(0, 0.7)$, and $p_{uv|0} = p_{uv|1} + 0.3$. We let $k_s = 1$ for $s < 10^6$ and then $k_s = 1/s$ thereafter.

We compare the the held-out misclassification rate as a function of the number of training samples for four different classifiers:

Figure 1 shows the relative performance of each. As expected, the Labeled-1NN classifier performs best,

As we had hoped, performance monotonically increases towards optimal performance (gray dot), even though the graph matching algorithm we used was approximate.

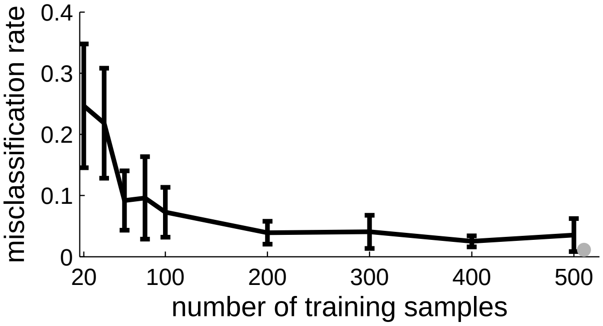

Fig. 1: Inexact graph matching can be used to approximate a consistent shuffled graph classifier. Data in this simulation was sampled from the independent edge model described above. For each number of training samples, we tested using 5000 test samples, and repeated 10 times. The gray dot indicates 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.)

5.2 Shuffled Connectome Classification

Emboldened by the simulated performance of our unlabeled graph classifier, we decided to try it on a real-world application. A "connectome" is a brain-graph in which vertices correspond to (groups of) neurons, and edges correspond to connections between them. Diffusion Magnetic Resonance (MR) Imaging and related technologies are making the acquisition of MR connectomes routine [14]. 49 subjects from the Baltimore Longitudinal Study on Aging comprise this data, with acquisition and connectome inference details as reported in [15]. Each connectome yields a 70 vertex simple graph (binary, symmetric, and hollow 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. The performance of a 1NN algorithm is reported in Table 1. When using the vertex labels, a "labeled-1NN" achieves 37% misclassification rate. Chance performance (only using the estimated prior) on this data is 49%. These two numbers provide bounds on performance. We then pass all the graphs through a shuffle channel, and implement GM-1NN using the approximately graph-matched Frobenius norm. This approach yields 41% misclassification rate, slightly worse than the labeled graph case. Finally, we compare the performance of our graph-matched 1NN algorithm with a more "standard" graph-invariant based algorithm, referred to as $\phi(G)$-1NN.

This results in 43% misclassification rate, slightly worse than the performance of our graph-matched 1NN.

TABLE 1: MR Connectome Leave-One-Out Misclassification Rates. Labeled-1NN refers to the 1NN classifier using . GM-k_sNN refers to the approximately graph-matched Frobenius norm k_sNN on the shuffled graphs. $\phi(G)$-1NN refers to using the Euclidean norm distance k_sNN on the four graph-invariants described in the main text. Chance is the Bayes plugin classifier using only the prior probabilities.

Labeled-1NN	GM-1NN	$\phi(G)$-1NN	Chance
37%	41%	43%	49%

6 Discussion

In this work, we address both the theoretical and practical limitations of classifying shuffled graphs, relative to labeled and unlabeled graphs. Specifically, we show that shuffling the vertex labels results in an irretrievable situation, with a possible degradation of classification performance (Theorem 1). Even if the vertex labels contained class-conditional signal, Bayes performance may remain unchanged (Theorem 2). Moreover, although one cannot hope to recover the vertex labels, one can obtain a Bayes optimal classifier by solving a large number of graph isomorphism problems (Theorem 3). This resolves a theoretical conundrum: is there a set of graph invariants that can yield a universally consistent graph classifier? When the generative distribution is unavailable, one can induce a consistent and efficient "unshuffling" classifier by using a graph-matching strategy (Theorem 4). Unfortunately, this is intractable in practice due to the difficulty of graph matching and the large number of isomorphism sets. Instead, Frobenius norm k_sNN classifier applied to the adjacency matrices may be used, which is also universally consistent (Corollary 1). Convergence rates may be considerably sped up by using a graph-matching Frobenius norm (Theorem 6). Because graph-matching is NP-hard, we instead use an approximate graph-matching algorithm in practice (see **??** for details). Applying these k_sNN classifiers to a problem of considerable scientific interest—classifying human MR connectomes—we find that even with a relatively small sample ($s = 49$), the approximately graph-matched

k_sNN algorithm performs nearly as well as the k_sNN algorithm *using* vertex labels, and slightly better than a k_sNN algorithm applied to a set of graph invariants proposed previously [13]. Thus, this theoretical insight has led us to improved practical classification performance. Extensions to weighted or (certain) attributed graphs are straightforward.

Acknowledgments

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

References

[1] L. Devroye, L. Györfi, G. Lugosi, and L. Gyorfi, *A Probabilistic Theory of Pattern Recognition (Stochastic Modelling and Applied Probability)*. New York: Springer, 1996. [Online]. Available: http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20\&amp;path=ASIN/0387946187

[2] P. Hagmann, "From diffusion MRI to brain connectomics," Ph.D. dissertation, Institut de traitement des signaux, 2005.

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

[4] J. White, E. Southgate, J. N. Thomson, and S. Brenner, "The structure of the nervous system of the nematode caenorhabditis elegans." *Philosophical Transactions of Royal Society London. Series B, Biological Sciences*, vol. 314, no. 1165, pp. 1–340, 1986.

[5] J. T. Vogelstein, W. R. Gray, R. J. Vogelstein, and C. E. Priebe, "Graph Classification using Signal Subgraphs: Applications in Statistical Connectomics," *Submitted for publication*, 2011.

[6] R. P. Duin and E. Pkalskab, "The dissimilarity space: bridging structural and statistical pattern recognition," *Pattern Recognition Letters*, vol. in press, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865511001322

[7] "The On-Line Encyclopedia of Integer Sequences — A000088." [Online]. Available: http://oeis.org/A000088

[8] L. Devroye, L. Gyorfi, and G. Lugosi, *A probabilistic theory of pattern recognition*. New York: Springer-Verlag, 1996.

[9] C. J. Stone, "Consistent Nonparametric Regression," *The Annals of Statistics*, vol. 5, no. 4, pp. 595–620, Jul. 1977.

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

[11] M. R. Garey and D. S. Johnson, *Computer and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[12] J. T. Vogelstein, J. C. Conroy, L. J. Podrazik, S. G. Kratzer, R. J. Vogelstein, and C. E. Priebe, "A Quadratic Assignment Problem Approach to Graph Matching: Applications in Statistical Connectomics," *Submitted for publication*, 2011.

[13] H. Pao, G. Coppersmith, and C. Priebe, "Statistical inference on random graphs: Comparative power analyses via Monte Carlo," *Journal of Computational and Graphical Statistics*, pp. 1–22, 2010. [Online]. Available: http://pubs.amstat.org/doi/abs/10.1198/jcgs.2010.09004

[14] 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

[15] W. R. Gray, J. T. Vogelstein, and R. J. Vogelstein, "Mr. Cap," *Submitted for publication*, 2011.

PLACE PHOTO HERE

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

PLACE PHOTO HERE

Carey E. Priebe Buddha in training.