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 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). The graph-matched 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 $\mathcal{V}$, where $|\mathcal{V}| = n < \infty$ is the number of vertices, and an edge set $\mathcal{E}$, where $|\mathcal{E}| \leq n^2$. 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. The cardinality of $\mathcal{G}_n$ is super-exponential in n. For example, when all graphs are assumed to be simple (that is, undirected binary edges without loops), then $|\mathcal{G}_n| = 2^{\binom{n}{2}} = d_n$. 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 $\sum_{G \in \mathcal{G}_n} \theta_{G|y} = 1$).

- *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.

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 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 $\text{Discrete}(G; \boldsymbol{\theta}'_y)$, 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 $\text{Discrete}(\widetilde{G}; \widetilde{\boldsymbol{\theta}}_y)$ 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 Implications of Shuffling

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 likelihood with a sum over all non-maximum posteriors:

$$\min_y \theta_{G|y}\pi_y \mapsto \sum_{y \in \mathcal{Y}'} \theta_{G|y}\pi_y \quad \text{where } \mathcal{Y}' = \{y\colon y \neq \operatorname*{argmax}_y \theta_{G|y}\}. \quad (12)$$

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:

$$h^\psi_* = \operatorname*{argmin}_{\psi \in \Psi, f^\psi \in \mathcal{F}^\psi} \mathbb{E}[f(\psi(\mathbb{G})) \neq Y], \quad (13)$$

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. (13) 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

$$h^\psi_*(G) = \operatorname*{argmax}_{y \in \mathcal{Y}} \widetilde{\theta}_{\psi(G)|y}\pi_y = \operatorname*{argmax}_{y \in \mathcal{Y}} \widetilde{\theta}_{\widetilde{G}|y}\pi_y = \widetilde{h}_*(G). \quad (14)$$

□

5 Learning Optimal Classifiers from Data

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. When the parameters are unknown (effectively always), we assume that 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 *labeled* graph classification the training data are $\mathcal{T}_s = \{\mathbb{Q}_i, \mathbb{G}_i, Y_i\}_{i \in [s]}$, where $[s] = \{1, \ldots, s\}$. For *shuffled* graph classification the training data are $\mathcal{T}'_s = \{\mathbb{G}'_i, Y_i\}_{i \in [s]}$, where $\mathbb{G}'_i = \mathbb{Q}_i(\mathbb{G}_i)$—thus, we are unable to observe useful vertex labels. For *unlabeled* graph classification the training data are again $\mathcal{T}'_s$. We assume that $\mathbb{P}_\mathbb{Q}$ is uniform, so that all label information is both unavailable and irrecoverable. Our task is to utilize training data to induce a classifier that approximates a Bayes classifier as closely as possible.

5.1 Bayes Plug-in Graph Classifiers

A *labeled* graph Bayes plugin classifier, $\hat{h}_s : \mathcal{G}_n \times \mathcal{T}_s \to \mathcal{Y}$, estimates the parameters $\{\boldsymbol{\theta}_y, \pi_y\}_{y \in \mathcal{Y}}$ using the training data $\mathcal{T}_s$, and then plugs those estimates into the labeled Bayes classifier, Eq. (8), resulting in

$$\hat{h}_s(G) = \operatorname*{argmax}_{y \in \mathcal{Y}} \hat{\theta}_{G|y}\hat{\pi}_y. \quad (15)$$

A *shuffled* graph Bayes plugin classifier, $\hat{h}'_s : \mathcal{G}_n \times \mathcal{T}'_s \to \mathcal{Y}$, estimates the parameters $\{\boldsymbol{\theta}'_y, \pi_y\}_{y \in \mathcal{Y}}$ using the training data $\mathcal{T}'_s$, and then plugs those estimates into the shuffled Bayes classifier, Eq. (9), resulting in

$$\hat{h}'_s(G) = \operatorname*{argmax}_{y \in \mathcal{Y}} \hat{\theta}'_{G|y} \hat{\pi}_y. \tag{16}$$

An *unlabeled* graph Bayes plugin classifier, $\hat{\widetilde{h}}_s : \mathcal{G}_n \times \mathcal{T}'_s \to \mathcal{Y}$, first determines in which unlabeled set each shuffled graph resides, using ψ as defined in Section 4.2. Then, it estimates the parameters $\{\widetilde{\theta}_{\psi(G')|y}\}$ and $\{\pi_y\}$ using the training data $\mathcal{T}'_s$. Finally, it plugs those estimates into the unlabeled Bayes classifier, Eq. (10), resulting in

$$\hat{\widetilde{h}}_s(G) = \operatorname*{argmax}_{y \in \mathcal{Y}} \hat{\widetilde{\theta}}_{\widetilde{G}|y} \hat{\pi}_y. \tag{17}$$

5.1.1 The Existence and Uniqueness of Universally Consistent Bayes Plug-in Graph Classifiers

For the above three classifiers, there exist estimators such that the classifiers exist, are unique, and moreover, are universally consistent, although the relative convergence rates and values that they converge to differ.

Let $\hat{L}_s = L(\hat{h}_s)$ be the risk of the induced *labeled* graph Bayes plugin classifier using the training data $\mathcal{T}_s$ to obtain maximum likelihood estimators for $\{\boldsymbol{\theta}_y, \pi_y\}_{y\in\mathcal{Y}}$. This yields

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

Proof: Because $\mathcal{G}_n$ and $\mathcal{Y}$ are both finite, the maximum likelihood estimates for the categorical parameters $\{\boldsymbol{\theta}_y, \pi_y\}_{y\in\mathcal{Y}}$ are guaranteed to exist and be unique [1]. Hence, the labeled graph Bayes plugin classifier is universally consistent to L_* (that is, it converges to L_* regardless of the true joint distribution, $\mathbb{P}_{\mathbb{Q},\mathbb{G},Y}$) [1]. □

Similarly, let $\hat{L}'_s = L(\hat{h}'_s)$ be the risk of the induced *shuffled* graph Bayes plugin classifier using the training data $\mathcal{T}'_s$ to obtain maximum likelihood estimators for $\{\boldsymbol{\theta}'_y, \pi_y\}_{y\in\mathcal{Y}}$. This yields

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

Proof: The above proof rests on the finitude of $\mathcal{G}_n$, which remains finite after shuffling (uniform or otherwise), and therefore, the above proof holds, replacing L_* with L'_*. □

Thus while one could merely plug the shuffled graphs into $\boldsymbol{\theta}'_y$, such a procedure is inadvisable. Specifically, the above procedure does not use the fact that all $\theta_{G'_i|y} = \theta_{G'_j|y}$ whenever $Q(G_i) = G_j$ for some $Q \in \mathcal{Q}$. Instead, consider the risk $\hat{\widetilde{L}}_s = L(\hat{\widetilde{h}}_s)$ of the induced *unlabeled* graph Bayes plugin classifier upon using the ψ function to map each shuffled graph to its corresponding unlabeled graph, and then obtaining maximum likelihood estimates of the unlabeled graph parameters, $\widetilde{\boldsymbol{\theta}}$.

Corollary 2. *$\hat{\widetilde{L}}_s \to \widetilde{L}_*$ as $s \to \infty$.*

Because $|\widetilde{\mathcal{G}}_n| \ll \mathcal{G}_n$ (by a factor of approximately $n!$), it follows that although both $\hat{h}'_s$ and $\hat{\widetilde{h}}_s$ are universally consistent, $\hat{h}'_s$ will generally perform worse. In fact, we have the following result:

Theorem 5. *$\mathbb{E}[\hat{L}'_s] \geq \mathbb{E}[\hat{\widetilde{L}}_s] \geq \mathbb{E}[\hat{L}_s]$ for all s.*

Proof: The first inequality follows from $\hat{\theta}_{G'|y}$ converging to $\theta'_{G|y}$ slower than $\hat{\widetilde{\theta}}_{\widetilde{G}|y}$ converges to $\widetilde{\theta}_{\widetilde{G}|y}$. Although $\theta'_{G|y}$ and $\widetilde{\theta}_{\widetilde{G}|y}$ encode identical information, the dimensionality of the two differs drastically. Specifically, $\widetilde{\theta}_{\widetilde{G}|y} \in \triangle_{\widetilde{d}_n}$ and $\theta'_{\widetilde{G}|y} \in \triangle_{d_n}$, and $\widetilde{d}_n \approx d_n/n!$.

The second inequality follows from.... Therefore, $\hat{L}'_s$ is inadmissible under $0-1$ loss. Moreover, if vertex labels are available, $\hat{\widetilde{L}}_s$ is also inadmissible under $0-1$ loss. □

5.2 k Nearest Neighbor Universally Consistent Graph Classifiers

Corollary 2 demonstrates that one can induce a universally consistent classifier $\hat{\widetilde{h}}_s$ using Eq. (17). Theorem 5 further shows that the performance of $\hat{\widetilde{h}}_s$ dominates $\hat{h}'_s$ under $0-1$ loss. Yet, using $\hat{\widetilde{h}}_s$ is practically useless for two reasons. First, it requires solving s graph isomorphism problems. Unfortunately, there are no known algorithms for solving graph isomorphism problems with worst-case performance in only polynomial time [**?**]. Second, the number of parameters to estimate is super-exponential in n ($\widetilde{d}_n \approx 2^{n^2}/n!$), and acceptable performance will typically require $s \gg \widetilde{d}_n$. We can therefore not even store the parameter estimates for small graphs (e.g., $n = 30$), much less estimate them. We therefore consider an alternative strategy.

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. The universal consistency proof for k_sNN was extended to graph-valued data in [10], which we include here for completeness. Specifically, to compare labeled graphs, [10] considered a Frobenius norm distance

$$\delta(G_i, G_j) = \|A_i - A_j\|_F^2, \tag{18}$$

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

Theorem 6. *$\hat{L}^{\delta}_s \to L_*$ 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 $\hat{L}^{\delta'}_s$ 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, the below corollary follows immediately:

Corollary 3. *$\hat{L}_s^{\delta'} \to L'_*$ as $s \to \infty$.*

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

$$\widetilde{\delta}(G'_i, G'_j) = \min_{Q \in \mathcal{Q}_n} \left\| Q(A'_i) - A'_j \right\|_F^2, \tag{19}$$

where A'_i and A'_j are shuffled adjacency matrices. Let $\hat{L}_s^{\widetilde{\delta}}$ indicate the misclassification rate of the k_sNN classifier using the above graph-matched norm $\widetilde{\delta}$. Given an exact graph matching function—a function that actually solves Eq. (19)—we have the following result

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

Thus, given $\mathcal{T}'_s$, one could use either δ or $\widetilde{\delta}$ as part of a k_sNN classifier. Analogous to the Bayes plugin results above, for the k_sNN classifiers, we have

Theorem 7. *$\mathbb{E}[\hat{L}_s^{\delta'}] \geq \mathbb{E}[\hat{L}_s^{\widetilde{\delta}}] \geq \mathbb{E}[\hat{L}_s^{\delta}]$ for all s.*

Proof: The first equality follows from the following consideration. The k_sNN classifier using δ on shuffled graphs only benefits from graphs that have been shuffled identically, whereas the k_sNN using $\widetilde{\delta}$ benefits from all graphs within an equivalence class, regardless of how they have been shuffled. The second equality.....

Therefore, $\hat{L}_s^{\delta'}$ is inadmissible under $0-1$ loss. Moreover, if vertex label information is available, $\hat{L}_s^{\delta}$ is also inadmissible under $0-1$ loss.

□

Thus, when the training data consists of shuffled graphs, the best universally consistent classifier (of those considered herein) is a k_sNN that uses $\widetilde{\delta}$ as the distance metric. Other universally consistent classifiers either require estimating more parameters than there are molecules in the universe, or are inadmissible under $0-1$ loss. When vertex labels are available, none of the other universally consistent classifiers considered here outperforms the k_sNN classifier that uses the vertex labels.

5.3 Other Graph Invariants are Worse

The above theoretical results consider Bayes plug-in and k_sNN classifiers. Here we consider other classifiers. Specifically, let $\hat{L}_s^{\psi}$ be the misclassification rate for some classifier that operates on $\mathcal{T}'_s$, that is, only has access to shuffled graphs. Consider the set of seven graph invariants studied in [13]:

- **size:** number of edges in the graph
- **maximum degree:** maximum number of edges incident to a vertex
- **maximum eigenvalue:** maximum eigenvalue of adjacency matrix, an upper bound on the maximum average degree
- **scan statistic:** maximum size of neighborhood of a vertex
- **number of triangles:** a triangle is a collection of three vertices that are connected
- **clustering coefficient:** a measure of connected of the graph
- **average path length:** average number of edges one must traverse to travel from a vertex to any other vertex.

Via Monte Carlo [13] was unable to find a uniformly most powerful graph invariant (test statistic [**?**]) for a particular hypothesis testing scenario. The above results, however, indicate that there exists optimal classifiers (or test statistics) for any setting. Let $\hat{h}_s^{\hat{\pi}}$ be the *chance* classifier, that is

$$\hat{h}_s^{\hat{\pi}}(G) = \operatorname*{argmax}_{y \in \mathcal{Y}} \hat{\pi}_y, \tag{20}$$

and let $\hat{L}_s^{\hat{\pi}}$ be the misclassification rate for this classifier. From the above results, it follows that

Theorem 8. *$\hat{L}_s^{\hat{\pi}} \geq \hat{L}_s^{\psi} \geq \hat{L}'_s = \hat{\widetilde{L}}_s = \hat{L}_s^{\delta'} = \hat{L}_s^{\widetilde{\delta}} = L'_s = \widetilde{L}_s$ as $s \to \infty$.*

6 Real World Application

We buttress the above theoretical results via numerical experiments. Consider the following five classifiers:

- δ-1NN: A 1-nearest neighbor (1NN) with Frobenius norm distance on the *labeled* adjacency matrices.
- δ'-1NN: A 1NN with Frobenius norm distance on the *shuffled* adjacency matrices.
- $\widetilde{\delta}$-1NN: A 1NN with an *inexact* graph-matched Frobenius norm distance on the shuffled adjacency matrices. Because graph-matching is $\mathcal{NP}$-hard [11], we instead use an inexact graph matching approach based on the quadratic assignment formulation described in [12], which is only cubic in n.
- ψ-1NN: A 1NN with Euclidean distance using the seven graph invariants described above. Prior to computing the Euclidean distance, for each invariant, we rescale all the values to lie between zero and one (to normalized scale).
- $\hat{\pi}$: Classify all test graphs according to which ever class is more frequent in the training data.

Performance is assessed by misclassification rate.

6.1 Shuffled Connectome Classification

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.

Figure 1 confirms the above theoretical results in a particular finite sample regime. We apply the five algorithms discussed above to sub-samples of the connectome data, which shows approximate convergence rates for this data. Fortunately, this real data example supports the main theorems of this work. Specifically, the k_sNN classifier using δ on the *labeled* graphs achieves the lowest misclassification rate, as suggested must be by Theorem 7 (dashed gray line). Moreover, as suggested by Theorem 8, the k_sNN classifier using the inexact graph-matching Frobenius norm on the shuffled adjacency matrices, $\widetilde{\delta}$, performs best of all classifiers using only shuffled graphs (compare dashed black line with solid black and gray lines). On the other hand, while the k_sNN classifier using the Frobenius norm on shuffled graphs, δ', must eventually converge to L'_s, its convergence rate is quite slow, so the classifier using standard invariants ψ outperforms the simple δ' based k_sNN.

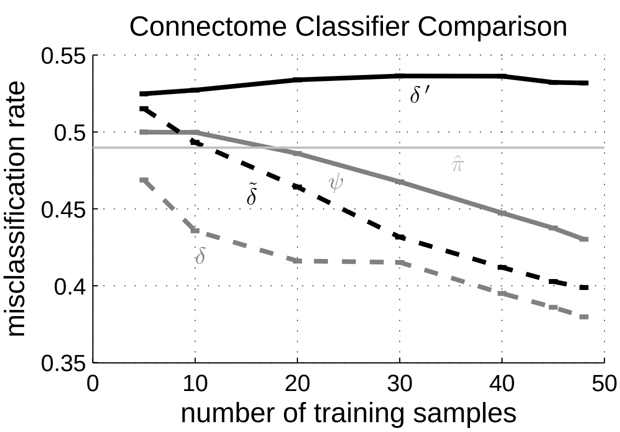

Fig. 1: Connectome misclassification rates for various classifiers. 2000 Monte Carlo sub-samples of the data were performed for each s, such that errorbars were neglibly small. Five classifiers were compared, as described in main text. Note that when s is larger than 20, as predicted by theory, we have $\hat{L}_s^{\hat{\pi}} > \hat{L}_s^{\psi} > \hat{L}_s^{\widetilde{\delta}} > \hat{L}_s^{\delta}$. Moreover, $\hat{L}_s^{\delta'} > \hat{L}_s^{\widetilde{\delta}} > \hat{L}_s^{\delta}$.

7 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, a Frobenius norm k_sNN classifier applied to the adjacency matrices may be used, which is also universally consistent (Corollary 3). Convergence rates may be considerably sped up by using a graph-matching Frobenius norm (Theorem 5). Because graph-matching is $\mathcal{NP}$-hard, we instead use an approximate graph-matching algorithm in practice (see [12] 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 ($s \geq 20$), 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

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\&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.