

Shuffled Graph Classification: Theory and Connectome Applications

October 17, 2012

Abstract

We develop a formalism to address statistical pattern recognition of graph valued data. Of particular interest is the case of all graphs having the same number of uniquely labeled vertices. When the vertex labels are latent, such graphs are called *shuffled graphs*. Our formalism provides insight to trivially answer a number of open statistical questions including: (i) under what conditions does shuffling the vertices degrade classification performance and (ii) do universally consistent graph classifiers exist? The answers to these questions lead to practical heuristic algorithms with state-of-the-art finite sample performance, in agreement with our theoretical asymptotics.

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 (note that this is a special case of assuming vertices are labeled, where each vertex has a unique label). 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 tiers 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.

Classification of brain-graphs is therefore poised 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 universally consistent graph classification algorithms, and practical approximations thereof. We demonstrate that the approximate algorithm has desirable finite sample properties via a real brain-graph classification problem of significant scientific interest: sex classification.

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 $|\mathcal{E}| \leq n^2$.

Definition 1. Let $\mathbb{G}: \Omega \rightarrow \mathcal{G}_n$ be a labeled graph-valued random variable taking values $G \in \mathcal{G}_n$, where \mathcal{G}_n is the set of labeled graphs on n vertices.

The cardinality of \mathcal{G}_n is super-exponential in n . For example, when all labeled 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, $Y: \Omega \rightarrow \mathcal{Y} = \{y_0, \dots, y_c\}$, 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 $\text{Discrete}(G; \theta_y)$, where θ_y is an element of the d_n -dimensional unit simplex Δ_{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 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 $Q: \mathcal{G}_n \rightarrow \mathcal{G}_n$ such that $Q(G) = G'$. Let \mathbb{Q} be a permutation-valued random variable, $\mathbb{Q}: \Omega \rightarrow \mathcal{Q}_n$, where \mathcal{Q}_n is the space of vertex permutation functions on n vertices so that $|\mathcal{Q}_n| = n!$.

Definition 2. Let $\mathbb{G}' = \mathbb{Q}(\mathbb{G}): \Omega \rightarrow \mathcal{G}_n$ be a shuffled graph-valued random variable, that is, a labeled graph valued random variable that has been passed through a random shuffle channel \mathbb{Q} .

Extending the above graph-classification 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

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

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

Note that one can think of a labeled graph as a shuffled graph for which \mathbb{Q} is a point mass at $Q = I$, where I is the identity matrix.

2.3 An unlabeled graph classification model

The above shuffling view is natural whenever the vertices of the collection of graphs share a set of labels, but the labeling function is unknown. However, when the vertices of the collection of graphs have different labels, perhaps a different view is more natural.

An *unlabeled graph* \tilde{G} is the collection of graphs isomorphic to one another, that is, $\tilde{G} = \{Q(G)\}_{Q \in \mathcal{Q}_n}$. Let \tilde{G} be an element of the collection of graph isomorphism sets $\tilde{\mathcal{G}}_n$. The number of unlabeled graphs on n vertices is $|\tilde{\mathcal{G}}_n| = \tilde{d}_n \approx d_n/n!$ (see [7] and references therein). An *unlabeling function* $U: \mathcal{G}_n \rightarrow \tilde{\mathcal{G}}_n$ is a function that takes as input a graph and outputs the corresponding unlabeled graph.

Definition 3. Let $\tilde{\mathbb{G}} = U(\mathbb{G}): \Omega \rightarrow \tilde{\mathcal{G}}_n$ be an unlabeled graph-valued random variable, that is, a labeled graph-valued random variable that has been passed through an unlabeled channel. In other words, $\tilde{\mathbb{G}} = \{Q(\mathbb{G})\}_{Q \in \mathcal{Q}_n}$, and takes values $\tilde{G} \in \tilde{\mathcal{G}}_n$.

The joint distribution over unlabeled graphs and classes is therefore $\mathbb{P}_{\tilde{\mathbb{G}}, Y} = \mathbb{P}_{U(\mathbb{G}), Y} = \mathbb{P}_{U(\mathbb{Q}(\mathbb{G})), Y}$, which decomposes as $\mathbb{P}_{\tilde{\mathbb{G}}|Y} \pi_Y$. The class-conditional distributions $\mathbb{P}_{\tilde{\mathbb{G}}|y}$ over isomorphism sets (unlabeled graphs) can also be thought of as discrete distributions $\text{Discrete}(\tilde{G}; \tilde{\theta}_y)$ where $\tilde{\theta}_y \in \Delta_{\tilde{d}_n}$ are vectors in the \tilde{d}_n -dimensional unit simplex. Comparing shuffling and unlabeling for the independent and uniform shuffle distribution $\mathbb{P}_{\mathbb{Q}}$, we have $\{\theta'_{G|y} = \tilde{\theta}_{\tilde{G}|y}/|\tilde{G}| \text{ for all } G \in \tilde{G}\}$.

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 graph classifier, (ii) risk, (iii), the Bayes optimal classifier, and (iv) the Bayes risk.

3.1 Bayes Optimal Colored Graph Classifiers

A *labeled graph classifier* $h: \mathcal{G}_n \rightarrow \mathcal{Y}$ 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

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

where \mathcal{H} is the set of possible labeled graph classifiers. The *labeled graph Bayes risk* is given by $L_* = L(h_*)$, 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 $h: \mathcal{G}_n \rightarrow \mathcal{Y}$ (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

$$h'_* = \operatorname{argmin}_{h \in \mathcal{H}} L'(h), \quad (3)$$

where \mathcal{H} is again the set of possible labeled (or shuffled) graph classifiers. The *shuffled graph Bayes risk* is given by $L'_* = L'(h'_*)$, where L'_* implicitly depends on $\mathbb{P}_{\mathbb{Q}(\mathbb{G}), Y}$.

3.3 Bayes Optimal Unlabeled Graph Classifiers

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

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

The *unlabeled graph Bayes risk* is given by $\tilde{L}_* = \tilde{L}(\tilde{h}_*)$, where $\tilde{\mathcal{H}}$ is the set of possible unlabeled graph classifiers and \tilde{L}_* implicitly depends on $\mathbb{P}_{\tilde{\mathbb{G}}, Y}$.

3.4 Parametric Graph Classifiers

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

$$h_*(G) = \operatorname{argmax}_{y \in \mathcal{Y}} \theta_{G|y} \pi_y, \quad (5)$$

$$h'_*(G) = \operatorname{argmax}_{y \in \mathcal{Y}} \theta'_{G|y} \pi_y, \quad (6)$$

$$\tilde{h}_*(\tilde{G}) = \operatorname{argmax}_{y \in \mathcal{Y}} \tilde{\theta}_{\tilde{G}|y} \pi_y. \quad (7)$$

4 Under what conditions does shuffling the vertices degrade classification performance?

The result of either shuffling or unlabeled 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:

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

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

$$\begin{aligned} \tilde{L}_* &= \sum_{\tilde{G} \in \tilde{\mathcal{G}}_n} \min_y \tilde{\theta}_{\tilde{G}|y} = \sum_{\tilde{G} \in \tilde{\mathcal{G}}_n} \min_y \sum_{G \in \tilde{G}} \theta'_{G|y} = L'_* \\ &= \sum_{\tilde{G} \in \tilde{\mathcal{G}}_n} \min_y \sum_{G \in \tilde{G}} \theta_{G|y} \geq \sum_{\tilde{G} \in \tilde{\mathcal{G}}_n} \sum_{G \in \tilde{G}} \min_y \theta_{G|y} = L_* \end{aligned} \quad (8)$$

□

An immediate consequence of the above proof is that the inequality in the statement of Lemma 1 strict whenever the inequality in Eq. (8) is strict:

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

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

The above result demonstrates that even when the labels *do* carry some class-conditional signal, it may be the case that shuffling or unlabeled does not degrade performance. In other words, the following two statements are equivalent: (i) the labels contain information with regard to the classification task, and (ii) 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 $y \in \mathcal{Y}$. Uniform shuffling has the effect of “flattening” likelihoods within isomorphism sets, from θ_y to θ'_y , so that θ'_y satisfies $\{\theta'_{G|y} = \tilde{\theta}_{\tilde{G}|y}/|\tilde{G}| \forall G \in \tilde{G}\}$. 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:

$$\min_y \theta_{G|y} \pi_y \mapsto \sum_{y \in \mathcal{Y}'} \theta_{G|y} \pi_y \text{ where } \mathcal{Y}' = \{y: y \neq \operatorname{argmax}_y \theta_{G|y} \pi_y\}. \quad (9)$$

Prior to concluding this section, we remark that one can achieve Bayes optimal risk using graph invariants. 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 $Q \in \mathcal{Q}_n$ (note that an unlabeled function $U(G)$ is a special case of ψ). 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:

$$h_*^\psi = \operatorname{argmin}_{\psi \in \Psi, f^\psi \in \mathcal{F}^\psi} \mathbb{E}[f(\psi(G)) \neq Y], \quad (10)$$

where Ψ is the space of all possible invariants and \mathcal{F}^ψ is the space of classifiers composable with invariant ψ . The expectation in Eq. (10) is taken against $\mathbb{P}_{G,Y}$ or equivalently $\mathbb{P}_{Q(G),Y}$, since invariants are invariant. Let L_*^ψ denote the Bayes invariant risk.

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

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

$$\begin{aligned} h_*^\psi(G) &= \operatorname{argmax}_{y \in \mathcal{Y}} \tilde{\theta}_{\psi(G)|y} \pi_y \\ &= \operatorname{argmax}_{y \in \mathcal{Y}} \tilde{\theta}_{\tilde{G}|y} \pi_y = \tilde{h}_*(G). \end{aligned} \quad (11)$$

□

5 Do universally consistent graph classifiers exist?

Throughout this paper, we consider two distinct “flavors” of graph classifiers that we can estimate from the data: (i) Bayes plug-in and (iii) nearest neighbor. Below, we first introduce Bayes plug-in graph classifiers. In the following sections, we will discuss their asymptotic properties, as well as the asymptotic properties of k_s nearest neighbor graph classifiers.

5.1 Bayes Plug-In Graph Classifiers

Implementing the above optimal classifiers requires knowing the model parameters. When the parameters are unknown (effectively always), we assume that the data are sampled identically and independently from some unknown joint distribution: $(\mathbb{Q}_i(\mathbb{G}_i), Y_i) \stackrel{iid}{\sim} \mathbb{P}_{\mathbb{Q}, \mathbb{G}, Y}$. For *labeled* graph classification, $\mathbb{P}_{\mathbb{Q}}$ is assumed to be the identity function, therefore, $\mathcal{T}_s = \{(\mathbb{G}_i, Y_i)\}_{i \in [s]}$, because when graphs are labeled $\mathbb{Q}_i(\mathbb{G}_i) = \mathbb{G}_i$. For *shuffled* graph classification $\mathbb{P}_{\mathbb{Q}}$ is assumed to be uniform over the permutation matrices, so that all label information is both unavailable and irrecoverable. The training data are therefore $\mathcal{T}'_s = \{(\mathbb{G}'_i, Y_i)\}_{i \in [s]}$, where $\mathbb{G}'_i = \mathbb{Q}_i(\mathbb{G}_i)$. For *unlabeled* graph classification the training data are again \mathcal{T}'_s . Our task is to utilize training data to induce a classifier that approximates a Bayes classifier as closely as possible.

A *labeled* graph Bayes plugin classifier, $\hat{h}_s : \mathcal{G}_n \times (\mathcal{G} \times \mathcal{Y})^s \rightarrow \mathcal{Y}$, estimates the parameters $\{\theta_y, \pi_y\}_{y \in \mathcal{Y}}$ using the training data $\mathcal{T}_s = \{(\mathbb{G}_i, Y_i)\}_{i \in [s]}$, and then plugs those estimates into the labeled Bayes classifier, Eq. (5), resulting in

$$\hat{h}_s(G) = \operatorname{argmax}_{y \in \mathcal{Y}} \hat{\theta}_{G|y} \hat{\pi}_y, \quad (12)$$

where the dependency on the training data is implicit in the $\hat{h}_s(G)$ notation.

A *shuffled* graph Bayes plugin classifier, $\hat{h}'_s : \mathcal{G}_n \times (\mathcal{G} \times \mathcal{Y})^s \rightarrow \mathcal{Y}$, estimates the parameters $\{\theta'_y, \pi_y\}_{y \in \mathcal{Y}}$ using the training data $\mathcal{T}'_s = \{(\mathbb{G}'_i, Y_i)\}_{i \in [s]}$, and then plugs those estimates into the shuffled Bayes classifier, Eq. (6), resulting in

$$\hat{h}'_s(G) = \operatorname{argmax}_{y \in \mathcal{Y}} \hat{\theta}'_{G|y} \hat{\pi}_y. \quad (13)$$

An *unlabeled* graph Bayes plugin classifier, $\hat{\tilde{h}}_s : \tilde{\mathcal{G}}_n \times (\tilde{\mathcal{G}}_n \times \mathcal{Y})^s \rightarrow \mathcal{Y}$, first determines in which unlabeled set each shuffled graph resides, using ψ as defined in Section 5. Then, it estimates the parameters $\{\tilde{\theta}_{\psi(G')|y}\}_{y \in \mathcal{Y}}$ and $\{\pi_y\}_{y \in \mathcal{Y}}$ using the training data \mathcal{T}'_s . Finally, it plugs those estimates into the unlabeled Bayes classifier, Eq. (7), resulting in

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

For brevity, we will sometimes refer to the above three induced classifiers as simply “classifiers”. Moreover, the sequence of classifiers (for example, $\{h_s\}_{s \rightarrow \infty}$) we will also refer to as a “classifier”.

5.2 Bayes Plug-in Graph Classifiers are Universally Consistent

The three parametric classifiers, Eqs. (12)–(14), admit classifier estimators that 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 $\{\theta_y, \pi_y\}_{y \in \mathcal{Y}}$. Note that \hat{L}_s is a random variable, as it is a function of the random training data \mathcal{T}_s . This yields

Lemma 4. $\hat{L}_s \xrightarrow{P} L_*$ as $s \rightarrow \infty$.

Proof. Because \mathcal{G}_n and \mathcal{Y} are both finite, the maximum likelihood estimates for the categorical parameters $\{\theta_y, \pi_y\}_{y \in \mathcal{Y}}$ 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, $\mathbb{P}_{\mathcal{Q}, \mathcal{G}, \mathcal{Y}}$) [1]. \square

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 $\{\theta'_y, \pi_y\}_{y \in \mathcal{Y}}$. This yields

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

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

Thus while one could merely plug the shuffled graphs into θ'_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 $\hat{\tilde{L}}_s = L(\hat{\tilde{h}}_s)$ 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, $\hat{\theta}$.

Corollary 2. $\hat{\tilde{L}}_s \xrightarrow{P} \tilde{L}_*$ as $s \rightarrow \infty$.

Because $|\tilde{\mathcal{G}}_n| \ll |\mathcal{G}_n|$ (by a factor of approximately $n!$), it follows that classifying by first projecting the graphs into a lower dimensional space should yield improved performance. Specifically, we have the following result:

Lemma 5. $\hat{\tilde{h}}_s$ dominates \hat{h}'_s for shuffled graph data.

Proof. Consider the scalar $\tilde{\theta}_{\tilde{G}|y}$ decomposed into the vector $(\tilde{\theta}_{G_1|y}, \dots, \tilde{\theta}_{G_{|\tilde{\mathcal{G}}|}|y})$, where each $\tilde{\theta}_{G_i|y} = \tilde{\theta}_{\tilde{G}|y} / |\tilde{\mathcal{G}}|$.

Note that each $\tilde{\theta}_{G_i|y} = \theta'_{G_i|y}$. Yet, the estimators, $\tilde{\theta}_{G_i|y}$ and $\hat{\theta}'_{G_i|y}$ are not equal, because the former can borrow strength from all shuffled graphs within the same unlabeled graph, but the latter does not. Assuming without loss of generality that the class priors are equal and known, the above domination claim is equivalent to stating that for each G ,

$$\mathbb{P}[\operatorname{argmax}_{y \in \mathcal{Y}} \hat{\tilde{\theta}}_{G|y} \neq \operatorname{argmax}_{y \in \mathcal{Y}} \tilde{\theta}_{G|y} | \mathcal{T}'_s] \leq \mathbb{P}[\operatorname{argmax}_{y \in \mathcal{Y}} \hat{\theta}'_{G|y} \neq \operatorname{argmax}_{y \in \mathcal{Y}} \theta'_{G|y} | \mathcal{T}'_s]. \quad (15)$$

Because $\tilde{\theta}_{G|y} = \theta'_{G|y}$, the only difference between the two sides of the above inequality is the estimators. We know that the estimators have the following distributions:

$$s_{\tilde{\mathcal{G}}} \hat{\tilde{\theta}}_{G|y} \sim \operatorname{Binomial}(\tilde{\theta}_{G|y}, s_{\tilde{\mathcal{G}}}) \quad (16a)$$

$$s_G \hat{\theta}'_{G|y} \sim \operatorname{Binomial}(\tilde{\theta}_{G|y}, s_G), \quad (16b)$$

where $s_{\tilde{G}}$ is the number of observations of any $G \in \tilde{G}$ in the training data, and s_G is the number of observations of G in the training data. From this, we see that for each G , $\hat{\theta}_{G|y}$ will have a tighter concentration around the truth due to its borrowing strength, because $s_{\tilde{G}} \geq s_G$, so our result holds. \square

5.3 k_s Nearest Neighbor Graph Classifiers are Universally Consistent

Corollary 2 demonstrates that one can induce a universally consistent classifier \hat{h}_s using Eq. (14). Lemma 5 further shows that the performance of \hat{h}_s dominates \hat{h}'_s . Yet, using \hat{h}_s is practically useless for two reasons. First, it requires solving s graph isomorphism problems. Unfortunately, there are no algorithms for solving graph isomorphism problems with worst-case performance known to be in only polynomial time [8]. Second, the number of parameters to estimate is super-exponential in n ($\tilde{d}_n \approx 2^{n^2}/n!$), and acceptable performance will typically require $s \gg \tilde{d}_n$. We can therefore not even store the parameter estimates for small graphs (e.g., $n = 30$), much less estimate them. This motivates consideration of an alternative strategy.

A k_s nearest-neighbor (k NN) classifier using Euclidean norm distance is universally consistent to L_* for vector-valued data as long as $k_s \rightarrow \infty$ with $k_s/s \rightarrow 0$ as $s \rightarrow \infty$ [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_s NN was extended to graph-valued data in reference [10], which we include here for completeness. Specifically, to compare labeled graphs, reference [10] considered a Frobenius norm distance

$$\delta(G_i, G_j) = \|A_i - A_j\|_F^2, \quad (17)$$

where A_i is the adjacency matrix representation of the labeled graph, G_i . Let \hat{h}_s^δ denote the Frobenius norm k_s NN classifier on *labeled* graphs using δ , and let \hat{L}_s^δ indicate the misclassification rate for this classifier. Reference [10] showed:

Lemma 6. $\hat{L}_s^\delta \xrightarrow{p} L_*$ as $s \rightarrow \infty$.

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

Let \hat{h}'_s denote the Frobenius norm k_s NN classifier on *shuffled* graphs using δ' , and let \hat{L}'_s indicate the misclassification rate for this classifier. From the above lemma and Corollary 1, the below follows immediately:

Corollary 3. $\hat{L}'_s \xrightarrow{p} L'_*$ as $s \rightarrow \infty$.

Given shuffled graph data \mathcal{T}'_s , however, other distance metrics appear more “natural” to us. For example, consider the “graph-matched Frobenius norm” distance:

$$\delta'(G'_i, G'_j) = \min_{Q \in \mathcal{Q}_n} \|Q(A'_i) - A'_j\|_F^2, \quad (18)$$

where A'_i and A'_j are shuffled adjacency matrices. Let \hat{h}'_s indicate the misclassification rate of the k_s NN classifier using the above graph-matched norm δ' *shuffled* graphs, and let \hat{L}'_s indicate the misclassification rate for this classifier. Given an exact graph matching function—a function that actually solves Eq. (18)—we have the following result:

Corollary 4. $\hat{L}'_s \xrightarrow{p} L'_*$ as $s \rightarrow \infty$.

Thus, given shuffled data \mathcal{T}'_s , one could consider either \hat{h}_s^δ or $\hat{h}_s^{\delta'}$.

Interestingly, when the data are labeled graphs, \mathcal{T}_s , one can outperform \hat{h}_s^δ by *shuffling*, that is, by apparently destroying the label information. Consider an example in which $\theta = \theta'$, such that no information is in the labels. In such scenarios, shuffling can effectively borrow strength from different labeled graphs that are within the same unlabeled graph set. Let $\hat{h}_s^{\delta'}$ indicate the misclassification rate of the k_s NN classifier using δ' labeled graphs, and let $\hat{L}_s^{\delta'}$ indicate the misclassification rate for this classifier. We therefore state without proof:

Lemma 7. *Neither $\hat{h}_s^{\delta'}$ nor \hat{h}_s^δ dominates when data are labeled graphs.*

Thus, when the training data consists of shuffled graphs, the best universally consistent classifier (of those considered herein) is a k_s NN that uses δ' as the distance metric. Other universally consistent classifiers that we considered either require estimating more parameters than there are molecules in the universe, or are inadmissible under 0 – 1 loss. When vertex labels are available, no classifier dominates.

5.4 Comparing Asymptotic Performances

The above theoretical results consider Bayes plug-in and k_s NN classifiers. Here we consider other classifiers. Specifically, let \hat{L}_s^ψ 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 [11]: size, max degree, max eigenvalue, scan statistic, number of triangles, and average path length. Via Monte Carlo, [11] was unable to find a uniformly most powerful graph invariant (test statistic [12]) for a particular hypothesis testing scenario with unlabeled graphs. The above results, however, indicate that there exists optimal classifiers (or test statistics) for any unlabeled or shuffled graph setting. To proceed, let $\hat{h}_s^{\hat{\pi}}$ be the *chance* classifier, that is

$$\hat{h}_s^{\hat{\pi}}(G) = \operatorname{argmax}_{y \in \mathcal{Y}} \hat{\pi}_y, \tag{19}$$

and let $\hat{L}_s^{\hat{\pi}}$ be the misclassification rate for this classifier. Moreover, let \hat{L}_s^ψ be the risk of the invariant classifier that is equivalent to the unlabeled Bayes plug-in classifier (see Lemma 3). From the above results, it follows that:

Lemma 8. *In expectation,*

$$\hat{L}_s^{\hat{\pi}} \geq \hat{L}'_s = \hat{L}_s = \hat{L}_s^\psi = \hat{L}_s^{\delta'} = \hat{L}'_s \text{ as } s \rightarrow \infty.$$

5.5 Comparing Computational Properties

While asymptotic results can be informative and insightful, understanding the computational properties of the different classifiers can be as (or even more) informative for real applications. Table 1 compares the space and time complexity of the various classifiers considered above. Only the k NN classifiers have the property that they do not require more space than there are atoms in the universe (for any n bigger than ≈ 30). Of those, the labeled k_s NN classifier does not require time exponential in the number of vertices. Therefore, we only found one type of classifier with performance guarantees that has both polynomial space and time. Unfortunately, the finite sample performance of this classifier is abysmal. This motivates constructing approximate classifiers.

6 Real World Application

We buttress the above theoretical results via numerical experiments. The asymptotic results combined with the computational complexities of the above described algorithm suggest that none of the proposed algorithms have all the properties we effectively require for real world applications, in particular, polynomial

Table 1: Order of computational properties for training the various shuffled graph classifiers.

name	notation	time	space
chance	$\hat{h}_s^{\hat{\pi}}$	s	1
invariant	\hat{h}_s^{ψ}	$e^n s$	$\min(2^{\binom{n}{2}}/n!, s)$
labeled Bayes plug-in	\hat{h}'_s	s	$\min(2^{\binom{n}{2}}, s)$
unlabeled Bayes plug-in	\tilde{h}_s	$e^n s$	$\min(2^{\binom{n}{2}}/n!, s)$
labeled k_s NN	$\hat{h}_s^{\delta'}$	$n^2 s^2$	$n^2 s$
unshuffled k_s NN	$\hat{h}'_s{}^{\delta'}$	$e^n s^2$	$n^2 s$

space and time complexity, as well as reasonable convergence rates. We therefore propose a different algorithm, which lacks universal consistency, but can be run on real data with good hope for reasonable performance. In particular, we modify $\hat{h}'_s{}^{\delta'}$, the *unshuffled* k NN classifier. Instead of requiring this classifier to actually solve the graph matching problem, Eq. (18), we use a recently proposed state-of-the-art approximate cubic time algorithm [16]. Denote this classifier $\hat{h}_s^{\tilde{\delta}}$.

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 [13]. 49 subjects from the Baltimore Longitudinal Study on Aging comprise this data, with acquisition and connectome inference details as reported in [14]. Each connectome yields a 70 vertex simple graph (binary, symmetric, and hollow adjacency matrix). Associated with each graph is class label based on the sex 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.

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.
- $\tilde{\delta}$ -1NN: A 1NN with an *approximate* graph-matched Frobenius norm distance on the shuffled adjacency matrices, as described above. Because graph-matching is \mathcal{NP} -hard [15], we instead use an inexact graph matching approach based on the quadratic assignment formulation described in [16], which only requires $\mathcal{O}(n^3)$ time.
- 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.
- $\hat{\pi}$: Use the chance classifier defined above.

Performance is assessed by leave-one-out misclassification rate.

Figure 1 reifies 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 lemmas of this work. Specifically, the k_s NN classifier using δ on the *labeled* graphs (dashed gray line) achieves the lowest misclassification rate

for all s , which one would expect if labels contain appropriate class signal. Moreover, the k_s NN classifier using the inexact graph-matching Frobenius norm on the shuffled adjacency matrices, $\tilde{\delta}$, 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_s NN 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_s NN.

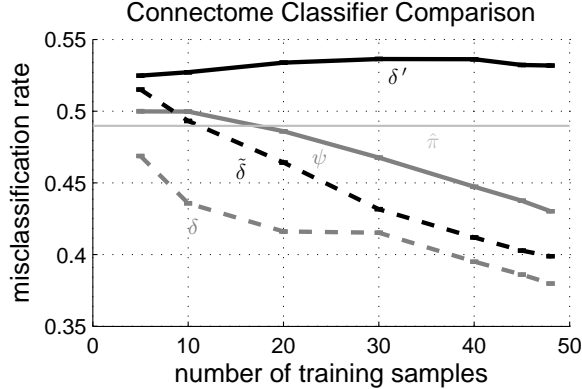


Figure 1: Connectome misclassification rates for various classifiers. 2000 Monte Carlo sub-samples of the data were performed for each s , such that errorbars were negligibly small. Five classifiers were compared, as described in main text. Note that when s is larger than 20, as predicted by theory,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

7 Discussion

In this work, we address both the theoretical and practical limitations of classifying shuffled graphs, relative to labeled and unlabeled graphs. Specifically, first we construct the notion of shuffled graphs and shuffled graph classifiers in a parallel fashion with labeled and unlabeled graphs/classifiers, as we were unable to find such notions in the literature. Then, we show that shuffling the vertex labels results in an irretrievable situation, with a possible degradation of classification performance (Lemma 1). Even if the vertex labels contained class-conditional signal, Bayes performance may remain unchanged (Lemma 2). Moreover, although one cannot recover the vertex labels, one can obtain a Bayes optimal classifier by solving a large number of graph isomorphism problems (Lemma 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 (Corollary 2). While this unshuffling approach dominates the more naïve approach (Lemma 5), it is intractable in practice due to the difficulty of graph matching and the large number of isomorphism sets. Instead, a Frobenius norm k_s NN classifier applied to the adjacency matrices may be used, which is also universally consistent (Corollary 4). Surprisingly, none of the considered classifiers dominate the other for labeled data (Lemma 7), yet asymptotically, we can order shuffled graph classifiers (Lemma 8).

Because graph-matching is \mathcal{NP} -hard, we instead use an approximate graph-matching algorithm in practice (see [16] for details). Applying these k_s NN classifiers to a problem of considerable scientific interest—classifying human MR connectomes—we find that even with a relatively small sample size ($s \geq 20$), the approximately graph-matched k_s NN algorithm performs nearly as well as the k_s NN algorithm *using* vertex labels, and slightly better than a k_s NN algorithm applied to a set of graph invariants proposed previously [11]. This suggests that the asymptotics might apply even for very small sample sizes. Thus, this theoretic-

cal insight has led us to improved practical classification performance. Extensions to weighted or (certain) attributed graphs are straightforward.

Acknowledgment

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

References

- [1] L. Devroye, L. Györfi, G. Lugosi, and L. Györfi, *A Probabilistic Theory of Pattern Recognition*. 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 to IEEE PAMI*, 2011.
- [6] R. P. W. Duin, E. Pkalska, and E. Pkalskab, “The dissimilarity space: Bridging structural and statistical pattern recognition,” *Pattern Recognition Letters*, vol. in press, no. April, May 2011. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0167865511001322><http://www.sciencedirect.com/science/article/pii/S0167865511001322><http://dx.doi.org/10.1016/j.patrec.2011.04.019>
- [7] “Number of graphs on n unlabeled nodes.” [Online]. Available: <http://oeis.org/A000088>
- [8] S. Fortin, “The Graph Isomorphism Problem,” *Technical Report, University of Alberta, Dept of CS*, 1996. [Online]. Available: <http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.32.9419>
- [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] H. Pao, G. A. Coppersmith, C. E. Priebe, H. P. Ao, G. A. C. Oppersmith, and C. E. P. Riebe, “Statistical inference on random graphs: Comparative power analyses via Monte Carlo,” *Journal of Computational and Graphical Statistics*, pp. 1–22, 2010. [Online]. Available: <http://pubs.amstat.org/doi/abs/10.1198/jcgs.2010.09004>
- [12] C. E. Priebe, G. A. Coppersmith, and A. Rukhin, “You say graph invariant, I say test statistic,” *Statistical Computing Statistical Graphics Newsletter*, vol. 21, no. 2, pp. 11–14, 2010.

- [13] 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
- [14] W. R. Gray, J. A. Bogovic, J. T. Vogelstein, B. A. Landman, J. L. Prince, and R. J. Vogelstein, “Magnetic resonance connectome automated pipeline: an overview.” *IEEE pulse*, vol. 3, no. 2, pp. 42–8, Mar. 2010. [Online]. Available: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6173097>
- [15] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences)*. W. H. Freeman, 1979. [Online]. Available: <http://www.amazon.com/Computers-Intractability-NP-Completeness-Mathematical-Sciences/dp/0716710455>
- [16] J. T. Vogelstein, J. C. M. Conroy, L. J. Podrazik, S. G. Kratzer, D. E. Fishkind, R. J. Vogelstein, and C. E. Priebe, “(Brain) Graph Matching via Fast Approximate Quadratic Programming,” *Arxiv preprint*, 2011.