Shuffled Graph Classification: Theory and Connectome Applications

Joshua T. Vogelstein
Johns Hopkins University, USA; Duke University, USA; Child Mind Institute, USA
Carey E. Priebe
John Hopkins University, USA

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. Applying these methods to classify sex and autism in two different human connectome classification tasks yields successful classification results in both applications.

Keywords: Statistical pattern recognition; Random graphs; Graph matching; Connectomics

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 (Devroye, Györfi, and Lugosi 1996), previous theoretical efforts regarding pattern recognition...
almost invariably assumed data were 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, which we define here as a graph where each vertex has a unique identifying attribute. For others, the labels are unobserved and/or assumed to not exist. Graph classification is the problem of determining to which class a graph belongs. We investigate the theoretical and practical implications of the absence of vertex labels in the context of graph classification.

These implications are especially important in the emerging field of “connectomics”, the study of connections of the brain (Hagmann 2005; Sporns 2010). 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 (White, Southgate, Thomson, and Brenner 1986). 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.

Several recent manuscripts have focused on classifying graphs without using vertex label information (Kashima and Inokuchi 2002; Saigo, Nowozin, Kadowaki, Kudo, and Tsuda 2008; Duin and Pkalskab 2012; Zare Borzeshi, Piccardi, Riesen, and Bunke 2012), and others have focused on methods that utilize vertex label information (Ketkar, Holder, and Cook 2009; Gibert, Valveny, and Bunke 2012). Some papers have addressed theoretical limitations of graph classification using label information (Vogelstein, Vogelstein, and Priebe 2009; Vogelstein, Gray, Vogelstein, and Priebe 2013). However, to our knowledge, no previous work has explicitly compared the theoretical and computational properties of the two problems. 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 two real brain-graph classification problem of significant scientific interest: sex classification from structural connectomes, and autism classification from functional connectomes.

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 $G : \Omega \to G_n$ be a labeled graph-valued random variable taking values $G \in G_n$, where $G_n$ is the set of labeled graphs on $n$ vertices.

The cardinality of $G_n$ is super-exponential in $n$. For example, when all labeled graphs are assumed to be simple (that is, undirected binary edges without loops), then $|G_n| = 2^{\binom{n}{2}} = d_n$. Let $Y$ be a categorical random variable, $Y : \Omega \to Y = \{y_0, \ldots, y_c\}$, where $c < \infty$. Assume the existence of a joint distribution, $P_{G,Y}$ which can be decomposed into the product of a class-conditional distribution (likelihood) $P_{G|Y}$ and a class prior $\pi_Y$. Because $n$ is finite, the class-conditional distributions $P_{G|y}$ can be considered discrete distributions $\text{Discrete}(G; \theta_y)$, where $\theta_y = (\theta_{G_1|y}, \ldots, \theta_{G_{dn}|y})$ is an element of the $d_n$-dimensional unit simplex $\Delta_{d_n}$ (satisfying $\theta_{G|i|y} \geq 0 \forall G \in G_n$ and $\sum_{G \in G_n} \theta_{G|i|y} = 1$).

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 $G, G' \in G_n$ to be isomorphic if and only if there exists a vertex permutation (shuffle) function $Q : G_n \to G_n$ such that $Q(G) = G'$.

Let $Q$ be a permutation-valued random variable, $Q : \Omega \to Q_n$, where $Q_n$ is the space of vertex permutation functions on $n$ vertices so that $|Q_n| = n!$.

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

Extending the above graph-classification model to include this vertex shuffling distribution yields $P_{Q,G,Y}$. 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

$$P_{Q,G,Y} = P_Q P_{G,Y} = P_Q P_{G|Y} \pi_Y = P_{Q(G)|Y} \pi_Y.$$  \hspace{1cm} (1)

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

Note that one can think of a labeled graph as a shuffled graph for which $Q$ 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 \( \tilde{G}_n \). The number of unlabeled graphs on \( n \) vertices is \( |\tilde{G}_n| = \tilde{d}_n \approx d_n/n! \) (see The On-Line Encyclopedia of Integer Sequences OETIS (2013), and references therein). An unlabeling function \( U : \mathcal{G}_n \rightarrow \tilde{G}_n \) is a function that takes as input a graph and outputs the corresponding unlabeled graph.

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

The joint distribution over unlabeled graphs and classes is therefore \( \mathbb{P}_{\bar{G},Y} = \mathbb{P}_{U(\mathcal{G}),Y} = \mathbb{P}_{U(Q(\mathcal{G})),Y} \), which decomposes as \( \mathbb{P}_{\bar{G}|Y} \pi_Y \). The class-conditional distributions \( \mathbb{P}_{\bar{G}|y} \) over isomorphism sets (unlabeled graphs) can also be thought of as discrete distributions \( \text{Discrete}(\bar{G}; \tilde{\theta}_y) \) 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 \( \mathbb{P}_Q \), we have \( \{\theta'_{G|y} = \tilde{\theta}_{G|y}/|\bar{G}| \text{ for all } G \in \bar{G}\} \).

### 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 four mathematical objects: (i) a graph classifier, (ii) risk, (iii) the Bayes optimal classifier, and (iv) the Bayes risk.

#### 3.1 Bayes Optimal Labeled 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 \( L(h) = \mathbb{E}[h(G) \neq Y] \), where the expectation is taken against \( \mathbb{P}_{G,Y} \). The labeled graph Bayes optimal classifier is given by

\[
h_* = \arg\min_{h \in \mathcal{H}} L(h),
\]
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 $P_{G,Y}$.

### 3.2 Bayes Optimal Shuffled Graph Classifiers

A shuffled graph classifier is also any function $h: G_n \rightarrow \mathcal{Y}$ (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 $L'(h) = E[h(Q(G)) \neq Y]$, where the expectation is taken against $P_{Q(G),Y}$. The shuffled graph Bayes optimal classifier is given by

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

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 $P_{Q(G),Y}$.

### 3.3 Bayes Optimal Unlabeled Graph Classifiers

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

$$\tilde{h}_* = \arg\min_{\tilde{h} \in \tilde{\mathcal{H}}} L(\tilde{h}).$$

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

### 3.4 Parametric Graph Classifiers

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

$$h_*(G) = \arg\max_{y \in \mathcal{Y}} \theta_{G|y} \pi_y,$$

$$h'_*(G) = \arg\max_{y \in \mathcal{Y}} \theta'_{G|y} \pi_y,$$

$$\tilde{h}_*(\tilde{G}) = \arg\max_{y \in \mathcal{Y}} \tilde{\theta}_{\tilde{G}|y} \pi_y.$$
4. Under What Conditions Does Shuffling the Vertices Degrade Classification 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, Devroye, Györfi, and Lugosi (1996) showed 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 $\pi_0 = \pi_1 = 1/2$.

$$\tilde{L}_* = \sum_{\tilde{G} \in \tilde{G}_n} \min_y \tilde{\theta}_{\tilde{G}|y} = \sum_{\tilde{G} \in \tilde{G}_n} \min_y \sum_{G \in G} \theta_{G|y} = L'_*$$

$$= \sum_{\tilde{G} \in \tilde{G}_n} \min_y \sum_{G \in G} \theta_{G|y} \geq \sum_{\tilde{G} \in \tilde{G}_n} \sum_{G \in G} \min_y \theta_{G|y} = L_*.$$

An immediate consequence of the above proof is that the inequality in the statement of Lemma 1 is 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

$$\min_y \tilde{\theta}_{\tilde{G}|y} > \sum_{G \in \tilde{G}} \min_y \theta_{G|y}.$$
cation problems. To see this, ignoring ties, simply replace each minimum likelihood with a sum over all non-maximum posteriors:

$$\min_y \theta_{G|y} \pi_y \mapsto \sum_{y \in Y'} \theta_{G|y} \pi_y$$

where

$$Y' = \{y: y \neq \arg\max_y \theta_{G|y} \pi_y\}. \quad (9)$$

Prior to concluding this section, we remark that one can achieve Bayes optimal risk using graph invariants. A graph invariant on $G_n$ is any function $\psi$ such that $\psi(G) = \psi(Q(G))$ for all $G \in G_n$ and $Q \in Q_n$ (note that an unlabeling function $U(G)$ is a special case of $\psi$). 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:

$$h^\psi_* = \arg\min_{\psi \in \Psi, f^\psi \in F^\psi} \mathbb{E}[f(\psi(G)) \neq Y], \quad (10)$$

where $\Psi$ is the space of all possible invariants and $F^\psi$ is the space of classifiers composable with invariant $\psi$. The expectation in Eq (10) is taken against $P_{G,Y}$ or equivalently $P_{Q(G),Y}$, since invariants are invariant. Let $L^\psi_*$ denote the Bayes invariant risk.

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

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

$$h^\psi_*(G) = \arg\max_{y \in Y} \tilde{\theta}_{\psi(G)|y} \pi_y = \arg\max_{y \in Y} \tilde{\theta}_{\tilde{G}|y} \pi_y = \tilde{h}_*(G). \quad (11)$$

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 (ii) 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 $k$ 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: $(Q_i(G_i), Y_i) \overset{\text{iid}}{\sim} P_{Q,G,Y}$. For labeled graph
classification, \( P_Q \) is assumed to be the identity function, therefore, \( T_s = \{ (G_i, Y_i) \}_{i \in [s]} \), because when graphs are labeled \( Q_i(G_i) = G_i \). For shuffled graph classification \( P_Q \) is assumed to be uniform over the permutation matrices, so that all label information is both unavailable and irrecoverable. The training data are therefore \( T'_s = \{ (G'_i, Y_i) \}_{i \in [s]} \), where \( G'_i = Q_i(G_i) \). For unlabeled graph classification the training data are again \( T'_s \) (even though technically an unlabeled graph is a set of labeled graphs, for unlabeled graph classification, we only observe a single sample, which is sufficient to define the entire equivalence class). 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, \( \hat{h}_s : G_n \times (G \times \mathcal{Y})^s \rightarrow \mathcal{Y} \), estimates the parameters \( \{ \theta_y, \pi_y \}_{y \in \mathcal{Y}} \) using the training data \( T_s = \{ (G_i, Y_i) \}_{i \in [s]} \), and then plugs those estimates into the labeled Bayes classifier, Eq (5), resulting in

\[
\hat{h}_s(G) = \arg \max_{y \in \mathcal{Y}} \hat{\theta}_{G|y} \hat{\pi}_y, \tag{12}
\]

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

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

\[
\hat{h}'_s(G) = \arg \max_{y \in \mathcal{Y}} \hat{\theta}'_{G|y} \hat{\pi}_y. \tag{13}
\]

An unlabeled graph Bayes plugin classifier, \( \tilde{h}_s : G_n \times (G_n \times \mathcal{Y})^s \rightarrow \mathcal{Y} \), first determines in which unlabeled set each shuffled graph resides, using \( \psi \) as defined in Eq (11). Then, it estimates the parameters \( \{ \tilde{\theta}_G(G')_{|y} \}_{y \in \mathcal{Y}} \) and \( \{ \pi_y \}_{y \in \mathcal{Y}} \) using the training data \( T'_s \). Finally, it plugs those estimates into the unlabeled Bayes classifier, Eq (7), resulting in

\[
\tilde{h}_s(G) = \arg \max_{y \in \mathcal{Y}} \hat{\tilde{\theta}}_{G=U(G)|y} \hat{\pi}_y. \tag{14}
\]

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 $s \to \infty$.

**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 (Devroye, Györfi, and Lugosi 1996). Hence, the labeled graph Bayes plugin classifier is universally consistent to $L_*$ (that is, it converges to $L_*$ regardless of the true joint distribution, $P_{Q,G,Y}$) (Devroye, Györfi, and Lugosi 1996). ■

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 5.** $\hat{L}'_s \xrightarrow{p} L'_*$ as $s \to \infty$.

**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 $L'_*$.

Thus while one could merely plug the shuffled graphs into $\theta'_y$, 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 $\hat{L}_s = L(\hat{h}_s)$ of the induced unlabeled graph Bayes plugin classifier upon using the $\psi$ 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 6.** $\hat{L}_s \xrightarrow{p} \tilde{L}_*$ as $s \to \infty$.

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 7.** $\hat{h}_s$ dominates $\hat{h}'_s$ for shuffled graph data.
Proof. Consider the scalar $\tilde{\theta}_{\tilde{G}|y}$ decomposed into the vector $(\tilde{\theta}_{G_1|y}, \ldots, \tilde{\theta}_{G_{\tilde{|G}|}|y})$, where each $\tilde{\theta}_{G_i|y} = \tilde{\theta}_{\tilde{G}|y} / |\tilde{G}|$. Note that each $\tilde{\theta}_{G_i|y} = \theta'_{G_i|y}$. Yet, the estimators, $\hat{\tilde{\theta}}_{G_i|y}$ 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 $G$,

$$P[\arg\max_{y \in Y} \tilde{\theta}_{G_i|y} \neq \arg\max_{y \in Y} \theta'_{G_i|y}] \leq P[\arg\max_{y \in Y} \hat{\tilde{\theta}}_{G_i|y} \neq \arg\max_{y \in Y} \hat{\theta}'_{G_i|y}] .$$

(15)

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

$$s_{\tilde{G}} \tilde{\theta}_{G_i|y} \sim \text{Binomial}(\tilde{\theta}_{G_i|y}, s_{\tilde{G}})$$

(16a)

$$s_G \hat{\theta}_{G_i|y} \sim \text{Binomial}(\tilde{\theta}_{G_i|y}, s_G)$$

(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 $G$, $\tilde{\theta}_{G_i|y}$ will have a tighter concentration around the truth due to its borrowing strength within the equivalence class, because $s_{\tilde{G}} \geq s_G$, so our result holds.

5.3 $k$ Nearest Neighbor Graph Classifiers Are Universally Consistent

Corollary 6 demonstrates that one can induce a universally consistent classifier $\hat{h}_s$ using Eq (14). Lemma 7 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 (Fortin 1996). Second, the number of parameters to estimate is super-exponential in $n$ ($d_n \approx 2^{n^2/n!}$), and acceptable performance will typically require $s \gg d_n$. 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 $k_s \rightarrow$
∞ with $k_s/s \to 0$ as $s \to \infty$ (Stone 1977). This non-parametric approach circumvents the need to estimate many parameters in high-dimensional settings such as graph-classification. The universal consistency proof for $k$NN was extended to graph-valued data in Vogelstein, Vogelstein, and Priebe (2009), which we include here for completeness. Specifically, to compare labeled graphs, Vogelstein, Vogelstein, and Priebe (2009) 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 $\hat{h}_s^\delta$ denote the Frobenius norm $k$NN classifier on labeled graphs using $\delta$, and let $\hat{L}_s^\delta$ indicate the misclassification rate for this classifier. Vogelstein, Vogelstein, and Priebe (2009) showed:

**Lemma 8.** $\hat{L}_s^\delta \xrightarrow{p} L_s^\delta$ as $s \to \infty$.

**Proof.** Because both $G$ and $Y$ 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 $\hat{h}_s'^\delta$ denote the Frobenius norm $k$NN classifier on shuffled graphs using $\delta'$, and let $\hat{L}_s'^\delta$ indicate the misclassification rate for this classifier. From the above lemma and Corollary 5, the below follows immediately:

**Corollary 9.** $\hat{L}_s'^\delta \xrightarrow{p} L_s'^\delta$ as $s \to \infty$.

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 $\hat{h}_s'^\delta'$ indicate the misclassification rate of the $k$NN classifier using the above graph-matched norm $\delta'$ shuffled graphs, and let $\hat{L}_s'^\delta'$ 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 10.** $\hat{L}_s'^\delta' \xrightarrow{p} L_s'^\delta'$ as $s \to \infty$.

Thus, given shuffled data $\mathcal{T}_s'$, one could consider either $\hat{h}_s^\delta$ or $\hat{h}_s'^\delta$.
Interestingly, when the data are labeled graphs, $\mathcal{T}_s$, one can outperform $\hat{h}_s^\delta$ 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 $k$NN classifier using $\delta'$ labeled graphs, and let $\hat{L}_s^{\delta'}$ indicate the misclassification rate for this classifier. We therefore state without proof:

**Lemma 11.** Neither $\hat{h}_s^{\delta'}$ nor $\hat{h}_s^\delta$ 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 $k$NN that uses $\delta'$ 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 $k$NN 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 Pao, Coppersmith, and Priebe (2011): size, max degree, max eigenvalue, scan statistic, number of triangles, clustering coefficient, and average path length. Via Monte Carlo, Pao, Coppersmith, and Priebe were unable to find a uniformly most powerful graph invariant test statistic (Priebe, Coppersmith, and Rukhin 2010) for a particular hypothesis testing scenario on shuffled graphs. The above results, however, indicate that there exists optimal classifiers (or test statistics) for any unlabeled or shuffled graph setting. To proceed, let $\hat{h}_s^\pi$ be the chance classifier, that is

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

(19)

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

**Lemma 12.** In expectation,

$$\hat{L}_s^\pi \geq \hat{L}_s' = \hat{L}_s = \hat{L}_s^\psi = \hat{L}_s^{\delta'} = \hat{L}_s'^{\delta'} \text{ as } s \to \infty.$$
Table 1. Order of computational properties for training the various shuffled graph classifiers.

| name          | notation | time  | space                      |
|---------------|----------|-------|----------------------------|
| chance        | \( \hat{h}_a^s \) | s     | 1                          |
| invariant     | \( \hat{h}_a^w \) | \( e^n s \) | \( \min(2^{(\frac{1}{2})}/n!, s) \) |
| labeled Bayes plug-in | \( \hat{h}_s^\psi \) | s     | \( \min(2^{(\frac{1}{2})}, s) \) |
| unlabeled Bayes plug-in | \( \hat{h}_s \) | \( e^n s \) | \( \min(2^{(\frac{1}{2})}/n!, s) \) |
| labeled kNN   | \( \hat{h}_s^\delta \) | \( n^2 s^2 \) | \( n^2 s \) |
| unshuffled kNN| \( \hat{h}_s^\delta' \) | \( e^n s^2 \) | \( n^2 s \) |

5.4 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 \( \approx 30 \)). Of those, the labeled \( k \)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 “heuristic” classifiers (that is, classifiers that currently lack certain desirable theoretical properties).

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 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 (Vogelstein et al. 2011). Denote this classifier \( \hat{h}_s^\delta' \).

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.
We consider two connectome datasets. The first is a “structural” connectome, estimated from diffusion magnetic resonance (MR) imaging. Imaging diffusion MR has become routine (Hagmann et al. 2010). We acquired 49 subjects from the Baltimore Longitudinal Study on Aging for this analysis, with acquisition and connectome inference details as reported in Gray et. al. (2010). 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. This dataset was used because we have previous results demonstrating useful classification results (Vogelstein et al. 2013).

A second dataset that we used is based on “functional” MR. We used a subset of The Autism Brain Imaging Data Exchange (ABIDE) consisting of 763 subjects, 403 of which were diagnosed with autism, and 360 matched controls (Di Martino et al. 2013). To estimate these functional connectomes, given a time-series of MR activity per subject, we estimate the correlation between all pairs of regions, as defined by the Harvard-Oxford atlas, which has 112 regions (Desikan et al. 2006). We threshold all correlations below 0.5 and discard the diagonal values, yielding a 112 vertex simple graph for each subject.

Consider the following five classifiers:

- Labeled: $\delta$-kNN: A $k$-nearest neighbor ($k$NN) with Frobenius norm distance on the labeled adjacency matrices.
- Shuffled: $\delta'$-kNN: A $k$NN with Frobenius norm distance on the shuffled adjacency matrices.
- Unshuffled: $\tilde{\delta}$-kNN: A $k$NN with an approximate graph-matched Frobenius norm distance on the shuffled adjacency matrices, as described above. Because graph-matching is NP-hard (Garey and Johnson 1979), we instead use an inexact graph matching approach based on the quadratic assignment formulation described in Vogelstein et al. (2011), which only requires $O(n^3)$ time.
- Invariant: $\psi$-kNN: A $k$NN 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.
- Chance: $\hat{\pi}$: Use the chance classifier defined above.

Performance is assessed by misclassification rate. Note that a $k$NN classifier has $k$ increasing with sample size; therefore, $k$NN for any fixed sample size is a special case of the $k$NN. Moreover, the 1NN has certain desirable properties, like bounded expected error, which makes it an especially interesting special case (Devroye, Györfi, and Lugosi 1996).
Figure 1 (left panel) 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 these data. For each case, we only show the 1NN classifier. Fortunately, this real data example supports the main lemmas of this work. Specifically, the $k$NN classifier using $\delta$ on the labeled graphs (dashed gray line) achieves the lowest misclassification rate for all $s$, which one would expect if labels contain sufficient class signal. Moreover, the $k$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 $k$NN classifier using the Frobenius norm on shuffled graphs, $\delta'$, must eventually converge to $L'$, its convergence rate is quite slow, so the classifier using standard invariants $\psi$ outperforms the simple $\delta'$ based $k$NN.

Figure 1 (right panel) shows the classification results from the ABIDE functional connectomes. For these data, we fit $k$NN classifiers for a sequence of $k$’s on the entire dataset, and then chose a $k$ with good performance. We then subsampled the data 100 times, each time randomly choosing 400 subjects. Classification performance was assessed via leave-one-out performance using the $k$ selected on the whole data set. Thus, the precise quantitative results might be a bit overly optimistic; the point of this analysis, however, was to determine the effect of shuffling on the various classifiers. As evidenced by the errorbar plots, the non-shuffled functional connectomes (Labeled) enabled classification significantly better than chance, whereas none of the algorithms on the shuffled graphs performed better than chance (indicated by the black horizontal line). Thus, in this example, it seems as if shuffling destroyed all the class-conditional signal (of which there was not very much it seems).

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 6). While this unshuffling approach dominates the more naïve approach (Lemma 7), it is intractable in practice due to the difficulty of graph matching and the large number of isomorphism sets. Instead, a Frobenius norm $k$NN classifier applied to the adjacency matrices may be used, which is also universally consistent (Corollary 10). Surprisingly, none of the considered classifiers dominate the other for labeled data (Lemma 11), yet asymptotically, we can order shuffled graph classifiers (Lemma 12).

Because graph-matching is $\mathcal{NP}$-hard, we instead use an approximate graph-matching algorithm in practice (see Vogelstein et al. 2011 for details). Applying these $k$NN classifiers to a problem of considerable scientific interest – classifying human structural MR connectomes – we find that even with a relatively small sample size ($s \geq 20$), the approximately graph-matched $k$NN algorithm performs nearly as well as the $k$NN algorithm using vertex labels, and slightly better than a $k$NN algorithm applied to a set of graph invariants proposed previously (Pao, Coppersmith, and Priebe 2011). This suggests that the asymptotics might apply even for very small sample sizes. Thus, this theoretical insight has led us to improved practical classification performance. Moreover, applying these methodologies to another dataset of considerable and clinical interest, the ABIDE autism dataset, we discover that our classifier can classify subjects as autistic or not only upon using the vertex labels. When we discard the vertex labels, all algorithms are at chance.

As a reviewer pointed out, we have focused entirely on the case where all graphs have the same number of vertices. The results, however, trivially
Shuffled Graph Classification

extend to the case where different graphs have different numbers of vertices. In particular, one can “zero-pad” all edges in all but the largest graph, and then apply all the above results. Moreover, if edges are weighted with a discrete set of different weights, many of the above results still hold. Extensions to continuously weighted or (certain) attributed graphs are relatively straightforward.

References

DESIKAN, R.S., SÉGONNE, F., FISCHL, B., QUINN, B.T., DICKERSON, B.C., BLACKER, D., . . . KILLIANY, R.J. (2006), “An Automated Labeling System for Subdividing the Human Cerebral Cortex on MRI Scans into Gyral Based Regions of Interest”, NeuroImage 31(3), 968–980.

DEVROYE, L., GYÖRFI, L., and LUGOSI, G. (1996), A Probabilistic Theory of Pattern Recognition, New York: Springer.

DI MARTINO, A., YAN, C.G., LI, Q., DENIO, E., CASTELLANOS, F.X., ALAERTS, K., . . . MILHAM, M.P. (2013), “The Autism Brain Imaging Data Exchange: Towards a Large-Scale Evaluation of the Intrinsic Brain Architecture in Autism”, Molecular Psychiatry, advance online publication June 2013, doi10.1038/mp.2013.78.

DUIN, R.P.W., and PAKALSKA, E. (2011), “The Dissimilarity Space: Bridging Structural and Statistical Pattern Recognition”, Pattern Recognition Letters, 33, 826–832.

FORTIN, S. (1996), “The Graph Isomorphism Problem”, Technical Report, University of Alberta, Department of Computer Science.

GAREY, M.R., and JOHNSON, D.S. (1979), Computers and Intractability: A Guide to the Theory of NP-Completeness, New York: W.H. Freeman.

GIBERT, J., VALVENY, E., and BUNKE, H. (2012), “Graph Embedding in Vector Spaces by Node Attribute Statistics”, Pattern Recognition 45(9), 3072–3083.

GRAY, W.R., BOGOVIC, J.A., VOGELSTEIN, J.T., LANDMAN, B.A., PRINCE, J.L., and VOGELSTEIN, R.J. (2010), “Magnetic Resonance Connectome Automated Pipeline: An Overview”, IEEE Pulse 3(2), 42–48.

HAGMANN, P. (2005), “From Diffusion MRI to Brain Connectomics”, PhD thesis, Institut de traitement des signaux, University of Lausanne, Switzerland.

HAGMANN, P., CAMMOUN, L., GIGANDET, X., GERHARD, S., ELLEN GRANT, P., WEDEEN, V., . . . SPORNS, O. (2010), “MR Connectomics: Principles and Challenges”, Journal Neuroscience Methods, 194(1), 34–45.

KASHIMA, H., and INOKUCHI, A. (2002), “Kernels for Graph Classification”, in Proceedings of the ICDM Workshop on Active Mining, Maebashi, Japan, 2002.

KETKAR, N.S., HOLDER, L.B., and COOK, D.J. (2009), “Empirical Comparison of Graph Classification Algorithms”, IEEE Symposium on Computational Intelligence and Data Mining, pp. 259–266.

OEIS (2013), “The On-Line Encyclopedia of Integer Sequences: Number of Graphs on n Unlabeled Nodes”, http://oeis.org/A000088.

PAO, H., COPPERSMITH, G.A., and PRIEBE, C.E (2011), “Statistical Inference on Random Graphs: Comparative Power Analyses Via Monte Carlo”, Journal of Computational and Graphical Statistics, 20, 295–416.

PRIEBE, C.E., COPPERSMITH, G.A., and RUKHIN, A. (2010), “You Say Graph Invariant, I Say Test Statistic”, Statistical Computing Statistical Graphics Newsletter, 21(2), 11–14.

SAIGO, H., NOWOZIN, S., KADOWAKI,T.T., KUDO, T., and TSUDA, K. (2008), “gBoost: A Mathematical Programming Approach to Graph Classification and Regression”, Machine Learning 75(1), 69–89.
SPORNS, O. (2010), *Networks of the Brain*, Cambridge MA: The MIT Press.
STONE, C.J. (1977), “Consistent Nonparametric Regression”, *The Annals of Statistics* 5(4), 595–620.

VOGELSTEIN, J.T., CONROY, J.C., PODRAZIK, L.J., KRATZER, S.G., FISHKIND, D.E., VOGELSTEIN, R.J., and PRIEBE, C.E. (2011), “Fast Inexact Graph Matching with Applications in Statistical Connectomics”, [http://arxiv.org/abs/1112.5507](http://arxiv.org/abs/1112.5507).

VOGELSTEIN, J.T., GRAY, W.R., VOGELSTEIN, R.J., and PRIEBE, C.E. (2013), “Graph Classification Using Signal Subgraphs: Applications in Statistical Connectomics”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7), 1539–1551.

VOGELSTEIN, J.T., VOGELSTEIN, R.J., and PRIEBE, C.E (2009), “Are Mental Properties Supervenient on Brain Properties?” *Nature Scientific Reports* 1(100), 11.

WHITE, J.G., SOUTHGATE, E., THOMSON, J.N., and BRENNER, S. (1986), “The Structure of the Nervous System of the Nematode Caenorhabditis Elegans”, *Philosophical Transactions of Royal Society London Series B, Biological Sciences* 314(1165), 1–340.

ZARE BORZESHI, E., PICCARDI, M., RIESEN, K., and BUNKE, H. (2012), “Discriminative Prototype Selection Methods for Graph Embedding”, *Pattern Recognition* 46(6), 1648-1657.