On utilizing structure to classify brain-graphs according to mental properties

Abstract

abstract

1. Introduction

motivation The statistical analysis of populations of data that are well represented by networks is a rapidly developing field (**?**). In particular, many aspects of the world, including economics, telecommunications, social websites, and transportation grids, to name a few, are well characterized by networks, or *graphs*. While much work has been devoted to studying the statistics of individual graphs, less attention has been given to the analysis of populations of graphs. Our interest here is to develop a number of classifiers that operate directly on graphs. In our motivating example, the graphs correspond to brains of individuals, and mental properties of those individuals are the desired classifier output. Because the aspects of these brain-graphs relevant for the classification task at hand is often unknown, a priori, we build a number of different classifiers, each designed to be optimal under certain model assumptions. We show that classification performance increases as the number of training exemplars increases, and each algorithm outperforms the others when the data that is classified is generated from the assumed model. This suggests that these tools provide not only a novel mechanism for classifying graphs, but also a guide for understanding the mechanism underlying the different classes.

The rest of the manuscript is organized as follows....

2. Definitions

Our intention is to build classifiers the operate on graphs. To proceed, we rigorous define the mathematical objects under investigation.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

2.1. Mental Properties

By "mental property", we mean one of many possible mental properties, including intelligence level, knowledge of a certain fact or set of facts, skill level, etc. We will investigate the relationship between brains and a *single* mental property here, and leave multiple categorical classifications to future work. Let $\mathcal{M}$ be the space of all mental properties under investigation, so that $m \in \mathcal{M}$. We restrict our work here to two-way classifications, so $\mathcal{M} = \{0, 1\}$, corresponding to having/not having some property, or having an ability over some threshold, etc.

2.2. Brain-graphs

With this in mind, we propose the notion of a *brain-graph*. Specifically, we say that the brain may be well characterized as a labeled, attributed multigraph (which is a generalized notion of a or network). Formally, we define a brain-graph, $b \in \mathcal{B}$ as a triple, $\mathcal{B} = (\mathcal{V}, \mathcal{A}, \mathcal{X})$, defined by the following:

- The set of vertices (nodes), $V = \{V_i\}_{i\in[N]} \in \mathcal{V} \subseteq \mathbb{Z}$, where $V_i \in \{0, 1\}$ for $i \in [N] = \{1, 2, \ldots, N\}$, and $\mathbb{Z}$ is the set of integers, $\{0, 1, 2, \ldots\}$.
- The set of arcs (edges), $A = \{A_{ij}\}_{i,j\in[N]} \in \mathcal{E} \subseteq \mathbb{Z}^{N^2}$. Implicitly, the above definition allows for multiedges, i.e., $A_{ij} : \Omega \mapsto \mathbb{Z} \,\forall\, i, j$. For clarity, we restict this notion of multiedges to integer *edge weights*, not categorically different edges. Thus, the phrase "number of edges" will always refer to elements of the matrix A, and not number of edges between a particular pair of vertices.
- The set of vertex features, $X = \{X_i\}_{i\in[N]} \in \mathcal{X} \subseteq \mathbb{R}^{d\times N}$, and d is the countable dimensionality of the feature vectors. These vertex features may or may not be observed.

Throughout, we assume n is known a priori, and the vertices are *labeled*. Further, we assume that both edges and features are random variables, but only edges are observed. Thus, we denote a random brain-graph $b = (a, x)$, where $a = \{a_{ij}\}_{i,j\in[N]}$, and

$x = \{x_i^k\}_{i\in[N],k\in[d]}$. The class-conditional probability distributions are therefore be defined by $F_{B|M} = P[B|M] = P[A, X|M]$.

3. Simulation

To generate data to assess our brain-graph classifiers, we consider a species whose nervous system consists of the same (small) number of labeled neurons for each organism. *Caenorhabditis elegans* is believed to be such a species (Durbin, 1987). The hermaphroditic C. elegans' somatic nervous system consists of 279 interconnected neurons. While the graph with these neurons as vertices and edges defined by chemical synapses between neurons is not identical across individuals, it is reasonably consistent (Durbin, 1987). Furthermore, these animals exhibit a rich behavioral repertoire that depends on circuit properties (de Bono & Maricq, 2005). Thus, one may design an experiment by describing the joint distribution F_{BM} via class-conditional distributions $F_{B|M=m_j}$ for the C. elegans brain-graph for two mental properties of interest, m_0 and m_1, along with the prior probability of class membership $P[M = m_1]$. Here the mental property corresponds to the C. elegans exhibiting or not exhibiting a particular behavior (e.g., response to an odor).

To generate the data, we let the class-conditional random variable $A_{ij}|M = m_0$ be distributed Poisson$(E_{ij} + \eta)$, where E_{ij} is the number of chemical synapses between neuron i and neuron j according to (Varshney et al., 2009), with noise parameter $0 < \eta \ll 1$. The class-conditional random variable $A_{ij}|M = m_1$ is distributed Poisson$(E_{ij} + \varepsilon_{ij})$ for neurons $i, j \in \mathcal{E}_\Delta$, where $\mathcal{E}_\Delta$ is the set of edges deemed responsible for odor-evoked behavior according to (Chalasani et al., 2007), with signal parameter ε_{ij} uniformly sampled from $[-5, 5]$.

4. Classifiers

If we knew $F_{B|0}$ and $F_{B|1}$, then classification would be trivial, let $\widehat{m} = \arg\max_m F_{B|m}$. However, in practice these distributions are typically unknown. Therefore, we must estimate g from a corpus data. Assume we have collected N brain/mental pairs. Then, define the corpus of data as the collection of all such pairs: $\mathcal{D}_N = \{(b_1, m_1), \ldots, (b_n, m_n)\}$. The estimated classifier, $\widehat{g}$, then takes a new brain and the old *training data*, and makes predictions about the mental property m. Formally, $\widehat{g} : \mathcal{B} \times (\mathcal{B}, \mathcal{M})^n \mapsto \mathcal{M}$, so $\widehat{g}(b; \mathcal{D}_n) = \widehat{m}$. The goal is to find the classifier, $\widehat{g} \in \mathcal{G}$, that minimizes some loss function, L, given the data. For two-way classification, a potentially reasonable loss function is $L_F(\widehat{g}) = \mathbb{E}[P_F(\widehat{g}(B; \mathcal{D}_n) \neq M|\mathcal{D}_n)]$.

Below, we describe several different graph classification algorithms that we will apply to simulated data. Each algorithm is built explicitly with some model assumptions in mind. As the number of constraints increase, the bias of the classifier potentially increases. However, the variance certainly decreases. So, if we can find an algorithm that reduces the variance more than it increases the bias, then we win the bias-variance trade-off, and have obtained the best $\widehat{g}$ we can find. Further, comparing the various algorithmic performances' on real data will inform us with regard to which model assumptions are best supported by the data, potentially leading to greater insight and understanding of the underlying neurobiology.

5. k_n nearest neighbor (kNN)

The k_n nearest neighbor (kNN) algorithm was proven to be universally consistent when the data are vectors in $\mathbb{R}^d$ (Stone, 1977). More recently, Vogelstein et al. (2009) extended this proof to the space of attributed multigraphs (**?**). Universality implies that no matter what $F_{B|M}$ looks like, the kNN algorithm will asymptotically converge to the Bayes optimal estimator, that is, as $N \to \infty$, $\widehat{g} \to g_*$, where $g_* = \arg\max_{g\in\mathcal{G}} L_F(g)$.

The kNN algorithm on graphs operates as follows. Let $d(b_i, b_j) = \|a_i, a_j\|_F^2$, where a_i is the adjacency matrix for brain i. Given a new brain, b, compute $d(b, b_i)\, \forall i \in [n]$. Then, sort the brains and their corresponding mental properties according to their distances from b: $b_{(1)}, b_{(2)}, \ldots$. Define $S_0 = \sum_{i\in[N]} I\{d(b, b_{(i)} < d(b, b_{(k)}))|m_{(i)} = 0\}$, and similarly for S_1. Then,

$$\widehat{m} = g_{kNN}(b; \mathcal{D}_n) = \arg\max_{m=0,1} S_m \qquad (1)$$

Note that no model assumptions were made, so this algorithm will work regardless of $F_{B|M}$, assuming some rule to ensure that as $n \to \infty$, $k \to \infty$ and $k/n \to 0$. We chose $k = \sqrt{cn}$, where c was determined empirically to be 16.

6. Edge independent assumption

In the above, $F_{B|M}$ was not explicitly specified, as the kNN algorithm is model-free and non-parametric. Here, we assume that each edge is independent, conditioned on the class:

$$P[B|M] = P[A|M] = \prod_{i,j\in[N]} P[A_{ij}|M] \qquad (2)$$

To utilize such a model, a precise distribution for $P[A_{ij}|M]$ must be specified. A natural choice is a

Poisson distribution, thus:

$$P[A_{ij} = a_{ij}|M = m] = \text{Poisson}(a_{ij}; \lambda_{ij,m}) = \frac{\lambda_{ij,m}^{a_{ij}} e^{\lambda_{ij,m}}}{a_{ij}!} \tag{3}$$

Note that this is a generalization of an Erdos-Renyi graph, where each binary edge is independent but sampled from the same Bernoulli distribution with probability p. The above implies that the class conditional distributions are governed entirely by $\Lambda_m = \{\lambda_{ij,m}\}_{ij\in[N]}$. Thus, the class conditional difference is defined entirely by the difference matrix, $\Lambda_\Delta = |\Lambda_0 - \Lambda_1|$. Let $\mathcal{E}_\Delta$ be the set of non-zero edges, and the number of non-zero edges in Λ_Δ is k, that is $|\mathcal{E}_\Delta| = k$.

7. Coherence

In the following, we consider three related notions of "coherence" that operate on this difference matrix, and depict them in Figure 1. Assume, for simplicity, that $|\mathcal{E}_\Delta| = k$, that is, only k of the N^2 possible edges are different between $F_{B|0}$ and $F_{B|1}$. Note that there are $\binom{N^2}{k}$ possible difference choices for $\mathcal{E}_\Delta$. We say that $\mathcal{E}_\Delta$ is *coherent* if all edges in $\mathcal{E}_\Delta$ have *both* end points in some set $\mathcal{V}_\Delta \ll \mathcal{V}$, that is, $\forall(i,j) \in \mathcal{E}_\Delta$, $i,j \in \mathcal{V}_\Delta$. An example coherent difference matrix, Λ_Δ, is shown in Figure 1, right panel. Note that non-zero elements lie along a reduced set of rows and columns. Incoherent is the opposite extreme, which in edges in $\mathcal{E}_\Delta$ are sampled uniformly from $\binom{N^2}{k}$, as shown if Figure 1, left panel. Semicoherent is an intermediate state, in which all edges have at least one end point from the reduced set of vertices, $\mathcal{V}_\Delta$, but not necessarily both, as shown in the middle column of Figure 1. Coherent and semicoherent correspond to different but related "kidney and egg" models (**?**), as depicted in Figure 2. It could be said that for incoherent difference matrices, there is no "eggistence".

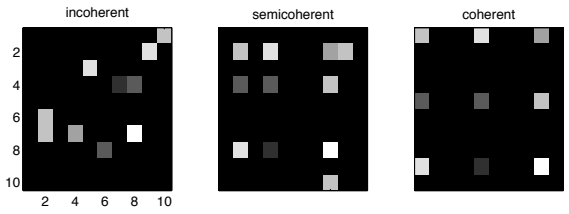

Figure 1. Schematic for different notions of coherence operating on the difference matrix, Λ_Δ: coherent (left panel), semicoherent (middle panel) and incoherent (left panel). Darker pixels are closer to zero. In each, both N and $k = 10$, corresponding to the presumed sparsity level of typical brain-graphs.

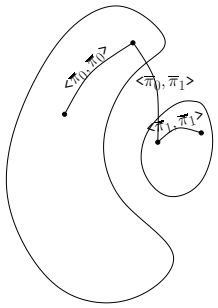

Figure 2. Kidney and egg schematic. If the only difference between classes are the parameters governing edges within the "egg", then the difference matrix, Λ_Δ is coherent. If the parameters governing edges between the kidney and the egg also differ between the two classes, then Λ_Δ is semicoherent. If Λ_Δ is incoherent, no egg exists.

8. Approximate Naive Bayes Classifier

The Naive Bayes classifier makes no additional assumptions other than edge independence. To use this classifier in practice, we first compute the maximum likelihood estimate of the parameters $m = 0$:

$$\widehat{\lambda}_{ij,0} = \frac{1}{n_0} \sum_{l|m_l=0} a_{ij}^l \tag{4}$$

where n_0 is the number of training samples in class 0, and a_{ij}^l is the number of edges between vertices i and j for brain l. The same procedure may be used for $m = 1$ as well, to obtain $\widehat{\Lambda}_0$ and $\widehat{\Lambda}_1$. Then, given a new brain with adjacency matrix, a, choose $\widehat{m}$:

$$\widehat{m} = g_{iid}(b; \mathcal{D}_n) = \arg\max_{m=0,1} \prod_{ij\in[N]} \text{Poisson}(a_{ij}; \widehat{\lambda}_{ij,m}) \tag{5}$$

In practice, it is sometimes the case that $a_{ij}^l = 0 \, \forall ij \in [N]$ and $m_l = 0$ or 1, but then for the new data, $a_{ij} > 0$. In such a scenario, $\widehat{\lambda}_{ij,m} = 0$, so $P[a_{ij} > 0|\widehat{\lambda}_{ij,m}] = 0$. This results in a zero probability for such a graph, which is an undesirable artifact of a small sample size. To mediate this effect, we could impose a prior on each Λ_m, such that even in the small sample size case, 0 probabilities are less frequent. A reasonable choice would be a Gamma distribution, which is the conjugate prior of the Poisson distribution. However, the Gamma distribution also has 2 hyperparameters, and we do not have a reasonable way of choosing them in real data. Thus, instead, we manually impose the following rule: if $\widehat{\lambda}_{ij,m} = 0$, then let $\widehat{\lambda}_{ij,m} = 1/(2n_m)$, i.e., $\widehat{\lambda}_{ij,m}$ is half of what it would be if exactly one class m training observation exhibited an edge between i and j. Because of this hack,

this algorithm is an approximate Naives Bayes Plug-in classifier. Note that performance of the approximate naive Bayes is independent of the signal coherence.

9. Edge independent and reduced canonical subspace contains signal

The above algorithm operates on the full parameter space, Λ. Each class conditional parameter lives in a N^2 dimensional space: $\Lambda_m \in \mathbb{R}_+^{N^2}$. However, if it is the case that only some of the edges are governed by different distributions for the two classes, that is $|\mathcal{E}_\Delta| = k < N^2$, then all those edges not in $\mathcal{E}_\Delta$ can be safely ignored upon classifying. Thus, an improved classifier, that only operates on this low dimensional canonical subspace (LDCS), is:

$$\widehat{m} = g_{LDCS}(b; \mathcal{D}_n) = \arg\max_{m=0,1} \prod_{ij \in \mathcal{E}_\Delta} \text{Poisson}(a_{ij}; \widehat{\lambda}_{ij,m}) \tag{6}$$

However, a priori, both $\mathcal{E}_\Delta$ and k are unknown. Therefore, some dimensionality reduction technique is required. Unfortunately, an exhaustive search for all possible $\mathcal{E}_\Delta$ is typically computationally intractable, as it requires a search over $\sum_{k=2}^{N-1} \binom{N}{k}$ possible subspaces. Therefore, some approximations are required, to obtain $\widehat{g}_{LDCS}$.

To proceed, we compute the difference matrix: $\widehat{\Lambda}_\Delta = |\widehat{\Lambda}_0 - \widehat{\Lambda}_1|$. Large elements of $\widehat{\Lambda}_\Delta$indicate that the two class conditional distributions differ drastically, and small elements of $\widehat{\Lambda}_\Delta$indicate that the class conditional distributions are nearly the same. We then collapse $\widehat{\Lambda}_\Delta$into a vector, and then sort it from highest to lowest, and plot the result in Figure 3 (left panel). To select the LDCS, $\widehat{\mathcal{E}}_\Delta$, we use the profile likelihood procedure outlined by (**?**), and find the first "elbow" (right panel). All elements of $\widehat{\Lambda}_\Delta$to the left of the elbow comprise the approximate LDCS, $\widehat{\mathcal{E}}_\Delta$. Note that if the number of edges in $\mathcal{E}_\Delta$ were known, a priori, then this step would be omitted, and we would just select the k canonical dimensions for which $\widehat{\Lambda}_\Delta$ was largest.

Note that many alternate dimensionality reduction techniques are available. For instance, one could first compute the eigenvectors of $\widehat{\Lambda}_\Delta$, and then use the profile likelihood method (or some other method) to decide how many eigenvectors to include. The disadvantage of this approach is a loss of interpretability, as the low-dimensional subspace is no longer canonical. One could recover canonicality by thresholding the eigenvectors that are kept, and only keeping the edges that are above this threshold. However, this two-step canonical subspace search is somewhat less elegant than the more direct method described above. Another strategy for choosing the appropriate LDCS would be a greedy forward search, in which we build a series of classifiers, each adding the next largest element of $\widehat{\Lambda}_\Delta$, and computing misclassification rate, stopping when the rate stops decreasing. Note that none of these LDCS techniques utilize information about the coherence of Λ_Δ, and thus their performance will be independent of coherence.

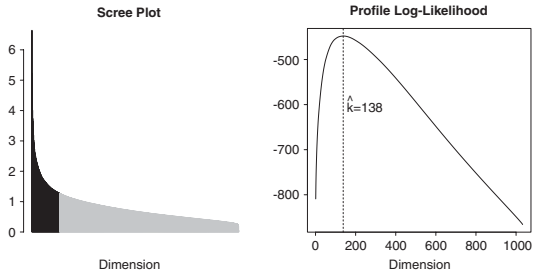

Figure 3. elbow finding

10. Edge independent and semi-coherent

Now, assume that we have reason to believe that the difference matrix is semicoherent. This corresponds to the scenario in which certain vertices are more likely to change edge parameters than others. In such a scenario, we can compute the "degree" of each vertex, which we define as the sum of edge weights associated with that vertex:

$$D_i = \sum_{j \in [N]} A_{ij} \tag{7}$$

Let $D_{i,m}$ indicate the degree of vertex i in class m, and define $\mathbf{D}_m = \{D_{1,m}, \ldots, D_{N,m}\}$ be the vector of degrees in class m. Define the degree difference vector as the absolute value of the class conditional degree vectors: $\mathbf{D}_\Delta = |\mathbf{D}_0 - \mathbf{D}_1|$. We can estimate $\mathbf{D}_\Delta$ from the data, and build a classifier:

$$\widehat{m} = g_{\widehat{D}}(b; \mathcal{D}_n) = \arg\max_{m=0,1} \left\| \widehat{\mathbf{D}}_\Delta - \mathbf{D}_\Delta \right\|_2^2 \tag{8}$$

Alternately, we can first reduce the dimensionality by using the same means described above, and then compute the distance only on the reduced dimensions.

11. Edge independent and coherence

If the edges are independent, but $\mathcal{E}_\Delta$ is coherent, then we can use this information to build a classifier with this in mind. In particular, we can compute the "scan statistic", ψ_i for each edge.

12. Edge conditionally independent

13. Results

The main result is the each of the above classifiers performance improves as a function of the amount of data (see Figure **??**).

Further, in our simulation, the class conditional differences are coherent, so classifiers utilizing coherence structure outperform those that do not (Figure **??**).

Finally, classifiers that reduce the dimensionality of the problem outperform their full-dimensionality counterparts.

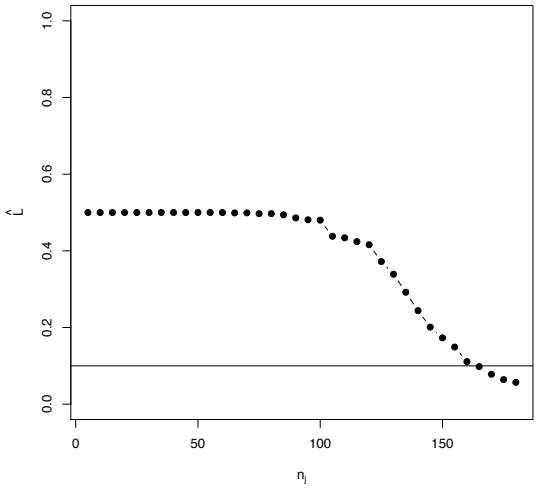

Figure 4. C. elegans graph classification simulation results. $\widehat{L}_F^{1000}(g_n)$ is plotted as a function of class-conditional training sample size n_j, suggesting that for $\varepsilon = 0.1$ we can determine that $\mathcal{M} \overset{\varepsilon}{\sim}_F \mathcal{B}$ holds with 99% confidence with just a few hundred training samples generated from F_{BM}. Each dot depicts an estimate for $L_F(g_n)$; standard errors are $(L_F(g_n)(1 - L_F(g_n))/1000)^{1/2}$. E.g., $n_j = 180$; $k_n = 53$; $\widehat{L}_F^{1000}(g_n) = 0.057$; standard error less than 0.01. We reject $H_0 : L_F(g^*) \geq 0.10$ at $\alpha = 0.01$. $L_F(g^*) \approx 0$ for this simulation.

14. Discussion

Acknowledgments

References

Chalasani, Sreekanth H, Chronis, Nikos, Tsunozaki, Makoto, Gray, Jesse M, Ramot, Daniel, Goodman, Miriam B, and Bargmann, Cornelia I. Dissecting a circuit for olfactory behaviour in caenorhabditis elegans. *Nature*, 450(7166):63–70, Nov 2007. doi: 8/nature06292. URL http://dx.doi.org/8/nature06292.

de Bono, Mario and Maricq, Andres Villu. Neuronal substrates of complex behaviors in c. elegans. *Annu Rev Neurosci*, 28:451–501, 2005. doi: 10.1146/annurev.neuro.27.070203.144259. URL http://dx.doi.org/10.1146/annurev.neuro.27.070203.144259.

Durbin, R. M. *Studies on the Development and Organisation of the Nervous System of Caenorhabditis elegans*. PhD thesis, University of Cambridge, 1987.

Stone, C.J. Consistent nonparametric regression. *The annals of statistics*, 5(4):595–620, 1977.

Varshney, L.R., Chen, B.L., Paniagua, E., Hall, D.H., and Chklovskii, D.B. Structural Properties of the Caenorhabditis elegans Neuronal Network. *ArXiv*, 2009.

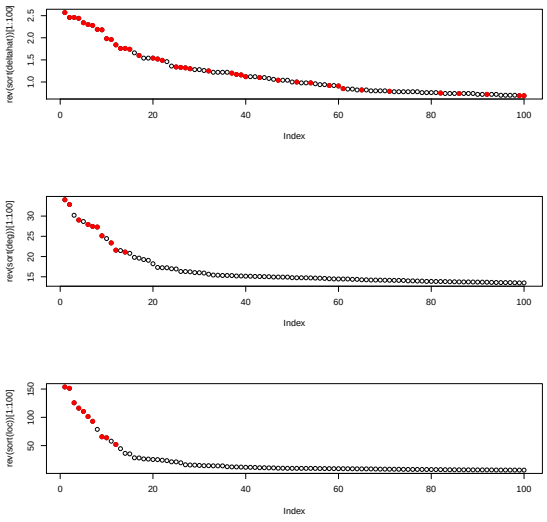

Figure 5. we can approximate low dimensional subspaces