Classifying brains according to mental properties

Joshua T. Vogelstein, R. Jacob Vogelstein, Carey Priebe
Johns Hopkins University

January 18, 2010

Abstract

1 Introduction

Suppose that you wanted to make the following argument:

Hypothesis 1. *A particular property of a brain causes a particular mental property.*

The "brain property" could be nearly anything: the genetic expression profile of a subset of neurons in a particular developmental stage, or microtubules becoming entangled, or columnar organization, or the bumps in our skulls, etc. Similarly, the "mental property" could be almost anything: intelligence, capacity for love, knowing the square-root of pi, etc. How might you go about *proving* Hypothesis 1 (H1)?

Here's how we would do it. First, we would change H1 to be *testable*, and change our desiderata from *proving* its accuracy to *collecting evidence* to support its claim. More specifically, we would make the following argument:

Hypothesis 2. *A change in particular property of a brain can predict a corresponding change in a particular mental property*[1].

Now, to collect evidence in support of H2, we need the following:

1. A way to describe any brain, b. Our description should live in the space of all possible descriptions of brains (or, at least those under consideration), $\mathcal{B}$.

2. A way to describe a mental property, m, which lives in the space of mental properties, $\mathcal{M}$. For instance, if the mental property under investigation is IQ, then $\mathcal{M}$ may be all possible scores on a particular IQ test.

3. A mapping from brain space to mental space, which enables us to relate m and b. Call this mapping, g, so $g : \mathcal{B} \mapsto \mathcal{M}$.

If we choose to take a statistical perspective, then both brains and minds are random variables. More specifically, let B be a random variable corresponding to a brain, so that any particular brain, b, is a sample from the space of brains, $\mathcal{B}$. The probability of B taking any value $b \in \mathcal{B}$ is given by the *marginal* distribution $F_B[b = B)$. Similarly, let M be a random variable corresponding to the mental property under investigation, so m is a sample from the space of mental properties, $\mathcal{M}$. The probability of M taking any value $m \in \mathcal{M}$ is given by the *prior* distribution $F_M(m = M)$. The mapping, g, tells us the relationship between any m and any b. Or, more formally, the mapping tells us about the *posterior* distribution of minds given brains, $F_{M|B}(m = M|b = B)$.

If we knew the *joint* distribution of minds and brains, $F_{BM}(b = B, m = M)$, and the marginal distribution of brains, F_B, then finding the mapping from brains to minds would be trivial: we would simply use Bayes rule to obtain the posterior, $F_{M|B} = F_{BM}/F_B$. 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 mapping, g_n, then takes a new brain and

[1] Note the relationship between this and *statistical-supervenience* [1]

the old *training data*, and makes predictions about the mental property m. Formally, $g_n : \mathcal{B} \times (\mathcal{B}, \mathcal{M})^n \mapsto \mathcal{M}$, so $g_n(b; \mathcal{D}_n) = \widehat{m}$.

This work contributes a perspective on a possible space of brains, $\mathcal{B}$, and several principled ways of choosing g_n. In so doing, we also describe a set of desiderata for both $\mathcal{B}$ and g_n to satisfy. We make no claim that our definitions or desiderata are novel or unique. Rather, the novel contribution (if any), is the overarching statistical perspective that unifies previously (potentially) disparate ideas into a single coherent framework.

2 A possible description of brains with certain desirable properties

Many possible spaces for the descriptions of brains exist. For instance, phrenologists thought that the "sulci and gyri" of the *skull* were sufficient to explain various mental properties, including intelligence. The space of brains they considered then, were all possible skull shapes. Clearly, such a space is not sufficient to compare the evidence support phrenological theory with an alternate theory, such as it is the sulci and gyri of the *brain* that determine those mental properties. So, can we enumerate a set of desiderata which we would use to select our brain-space? Here is a possible set:

1. $\mathcal{B}$ should be sufficiently large to be able account for whatever properties of the brain are casually related to the properties of cognition under investigation.
2. $\mathcal{B}$ should be sufficiently large to be able span multiple *levels of explanation*. That is, we might want to be able to compare a hypothesis about the role of genetic expression profiles with another hypothesis about default mode networks.
3. $\mathcal{B}$ should be only as large as necessary, and no larger. In particular, if we do not believe that total brain volume can *cause* a particular mental property, that it need not be included.
4. The properties of any particular brain, $b \in \mathcal{B}$, should be either measurable or estimatable, such that experimental observation may be used to obtain them.
5. $\mathcal{B}$ should admit algorithms that (are guaranteed to be able to) capture the relationship of interest.

Note that the above desiderata are neither complete nor unique; rather, they provide (hopefully) a reasonable set of criteria for evaluating any proposed $\mathcal{B}$. 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 4-tuple, $\mathcal{B} = (\mathcal{V}, \mathcal{E}, \mathcal{X}_V, \mathcal{X}_E)$, defined by the following:

- The set of vertices (nodes), $V = \{V_i\}_{i \in [n_v]}$. If we assume, for instance, that neurons are the fundamental (atomic) unit of computation, then each vertex could correspond to a neuron. Regardless of what vertices represent, formally, we let $V_i \in \mathcal{V}_i = \{0, 1\}$ for $i \in [n_v] = \{1, 2, \ldots, n_v\}$, $n_v \leq \infty$, and $\mathcal{V} = \mathcal{V}_i^{n_v}$.
- The set of edges, $E = \{E_{ij}\}_{i,j \in [n_v]}$. Again, if we assume that neurons are the fundamental unit of computation, then each edge could correspond to a synapse. To simplify matters we may only consider the presence or absence of a synapse, in which case $E_{ij} \in \{0, 1\}$. Or, we may consider the effective strength of a synapse, in which case $E_{ij} \in \mathbb{Z}$, where $\mathbb{Z}$ is the set of integers, $\{0, 1, 2, \ldots\}$. Further, we could allow for the possibility of multiple "kinds" of synapses between any pair of neurons, such as chemical and electrical. In such a case, we have $E_{ijk} \in \mathcal{E}_{ijk} \subseteq [n_v]^2 \times [n_k]$, where $n_k \leq \infty$ is the maximum number of categorically different edges. In any case, we can define the space of edges as $\mathcal{E} = \mathcal{E}_{ijk}^{n_v^2 \times n_k}$.
- If vertices have features (or labels/attributes), then let X_i correspond to the feature vector for vertex i (features may correspond to *anything* about the vertex). Again, assuming vertices represent neurons, features may indicate neurotransmitter released, morphological properties, receptive fields, etc. Formally, $X_i \in \mathcal{X}_1 \subseteq \mathbb{R}^{d_V}$ for $i \in [n_v]$, and d_V is the dimensionality of the feature vectors, so $\mathcal{X}_V = \mathcal{X}_1^{n_v}$. It may be the case that certain features are measurable/observable, and others are hidden. If so, let $X_i = X_i^o \cup X_o^h$, and $X_i^o \cap X_i^h = \emptyset$.
- If edges also have features, then let X_{ij} correspond to the feature vector for edge (i, j). If edges are synapses, then edge features might include things like probability of release, post-synaptic potential shape, etc. Let $X_{ij} \in$

$\mathcal{X}_2 \subseteq \mathbb{R}^{d_E}$ for $(i,j) \in [n_v] \times [n_v]$, and d_E is the dimensionality of the edge features, so $\mathcal{X}_E = \mathcal{X}_2^{n_v^2}$. In the scenario where categorically different edges exist, let an additional index k indicates edge features for each category k, so $\mathcal{X}_E = \mathcal{X}_2^{n_v^2 \times n_k}$.

Given such a brain-space $\mathcal{B}$, a natural question is: does this notion of brain-graphs satisfy the above desiderata? Let's see:

1. The space of all possible brain-graphs does appear to be quite large, potentially incorporating many small and large details of brains.

2. Because each vertex V_i could correspond to a neuron, a column, a neuroanatomical region, etc., indeed, brain-graphs can span multiple levels of explanation. Even for a given brain, it is possible to let some V_i's represent neurons, and other V_i's represent neuroanatomical regions, even if this is a redundant characterization of the brain, in that the neurons are *within* the neuroanatomical region.

3. While the brain-graph space is large, it is not "all-encompassing." For instance, the space of all possible functions is not within brain-graph space. Thus, the brain-graph space seems to exclude at least some possibilities.

4. For a space to admit algorithms guaranteed to capture the relationship of interest, one must prove limiting results. For instance, Stone proved in 1977 [2] that the k_n nearest neighbor algorithm is guaranteed to converge to the Bayes optimal solution. In 2010, Vogelstein et al. [1] proved a similar result holds for brain-graphs.

Note one significant omission in the above is any *dynamic* notion, that is, the above brain-graph description is entirely *static*[**?**]. This fact is easily remedied by introducing an index t into the space, i.e., $\mathcal{B}_t$, so the entire space can be time-varying. This more general notion of *dynamic* brain-graphs will be addressed in future work.

Thus, it seems that brain-graphs indeed satisfy the above criteria. This is not to say that brain-graphs are the only space one could define to satisfy these criteria, merely that brain-graphs are sufficient. So, given such a space of brains, the next question is: "how can one choose a mapping between brains and minds?"

3 Possible approaches to choosing mappings with desirable properties

Given $\mathcal{B}$, what can we do with it? As stated above, our goal is to relate these models to properties of cognition. More specifically, let $\mathcal{M}$ characterize the space of the mental (cognitive) property, and $g \in \mathcal{G}$ be some mapping to learn. Then $g : \mathcal{B} \to \{0, 1, \ldots, C\}$ is a $C-$way classifier. As discussed above, solving these problems—which means finding g—will depend on the joint distribution of brains and minds, $F = F_{BM}$. Because F_{MB} is typically unknown, g must be estimated, using some training data, $\mathcal{D}_n$, to obtain $g_n(b; \mathcal{D}_n)$. The particular g_n should be the one that minimizes some loss function, $L_F(g_n)$, over the space of all possible g_n's, $\mathcal{G}$. For instance, when $|\mathcal{M}| = 2$, $\mathcal{G}$ is all possible two-way classifiers, and a potentially reasonable loss function is $L_F(g_n) = \mathbb{E}[P_F(g_n(B; \mathcal{D}_n) \neq M | \mathcal{D}_n)]$.

Thus, given a mental property, a decision about how to represent it, $\mathcal{M}$, and a loss function, L, our task is to find a good algorithm, g_n. Two complementary strategies are possible: model-free and model-based. Model-free algorithms have the advantage that no model need be specified. Unfortunately, these models often provide very little intuition (if any) about the underlying causes of the relationship of interest, and therefore lack *interpretability*. On the other hand, model-based algorithms can provide rather simple interpretations. However, model-based algorithms require defining a space of models sufficient to capture the relevant aspects of $\mathcal{B}$. Classes of models sufficiently large to span $\mathcal{B}$ often have large variance, and suffer from over-fitting issues. Note that many standard algorithms including linear, quadratic, and support vector based classifiers/regressors *implicitly* define a model, and are therefore not strictly "model-free".

3.1 Model-free algorithms

Model-free algorithms often operate on interpoint distance space, as opposed to the explicit data space [3]. More formally, given any two brain-graphs, b^1 and b^2, first define an interpoint (pseudo-) distance metric: $\rho : \mathcal{B} \times \mathcal{B} \mapsto [0, \infty)$. This reduces the problem from operating in $\mathcal{B}$ to operating in $\mathbb{R}_+$. Because the data collected is often corrupted by noise, it is typical to also introduce a smoothing function: $s : \mathcal{B} \mapsto \mathcal{B}$. For the brain-graph scenario, this may correspond to inferring unobserved edges. Thus, the smoothing-derived (pseudo-) distance metric, ρ' is defined as: $\rho' = \rho(s(b^1), s(b^2))$.

Perhaps the prototypical model-free algorithm is the k_n nearest neighbor (kNN) algorithm. Vogelstein et al. (2010) showed that a kNN classifier is a universally consistent classifier (meaning, achieves the Bayes optimal performance), for any F_{BM}, under a Frobenius norm distance metric. In other words, they let defined $\rho(\cdot) = \|\cdot\|_F$, and s was simply the identity (that is, no smoothing). Simulations showed that for only a few hundred simulated sample data points, this kNN achieved misclassification error rate below 10%. In practice, it is often the case that other model-free algorithms outperform (in accuracy) any particular kNN, including class cover catch digraphs, decision trees, and various ensemble approaches such as random forests [4]. Unfortunately, scant theoretical works is available to provide proofs of universal consistency for these alternate model-free algorithms [**?**].

3.2 Model-based algorithm

The other possible strategy is to propose a class of models, $\mathcal{P} = \{P_\theta : \theta \in \Theta \subseteq \mathbb{R}^d\}$, that describe the data (note that the choice of how to determine the dimensionality d of the model specifies whether the model is parametric, semiparametric, or nonparametric). Like the class of brain-graphs, $\mathcal{B}$, the class of models, $\mathcal{P}$, should satisfy several desiderata:

1. Lower dimensional Θ are preferred over higher dimensional Θ, *ceteris parabus*.
2. As the number of data points approaches infinity, so should the dimensionality of the model, that is: $n \rightarrow \infty \implies d \rightarrow \infty$.
3. θ should be at least *generically identifiable* [5].
4. Estimators for θ should be consistent [6].

Model-based approaches therefore have a multi-step process: (i) define $\mathcal{P}$, (ii) fit the model by finding $\widehat{\boldsymbol{\theta}} \in \Theta$ that minimizes some loss function, and (iii) find/compute $g_n(\widehat{\boldsymbol{\theta}})$ that minimizes some other loss function[2]. Below, we describe several possible model classes, $\mathcal{P}$, with increasing complexity. For each, a heuristic for how to fit the model is provided. Given that these are model-based approaches, each class of models admits a posterior distribution, $P(m|\widehat{\boldsymbol{\theta}})$. Thus, g_n will always provide the maximum a posteriori estimate of m, or some close approximation to it.

3.2.1 Edge independent models

The first class of models we consider greatly simplifies the brain-space. Specifically, we make the following simplifying assumptions:

1. the number of vertices is known and fixed for all brain-graphs
2. each edge is independent and distributed according to the same parametric family of distributions
3. vertex and edge features do not contain any useful information with regard to M

Given these three assumptions, each brain is completely characterized by its adjacency matrix. Let e^l indicate the adjacency matrix for brain l, that is $e^l = \{e_{ij} | e_{ij} \in e^l\} \overset{iid}{\sim} P_E(\cdot|\theta)$. If each edge identically distributed, then θ is a scalar. Otherwise, θ is a vector, with up to n_v^2 parameters. Because $\mathcal{E}$ is the space of multi-edges, each each e_{ij}^l potentially takes any integer value. Thus, any discrete probability mass function is possible. To satisfy the above desiderata, however, exponential family distributions are preferred, as they admit identifiable and consistent estimators. Furthermore, if a Bayesian perspective is taken, exponential family distributions admit conjugate priors, meaning that maximum a posteriori estimates may be obtained, assuming an appropriate prior is available. For instance, a reasonable model may be that edges are sampled from a Poisson distribution, and the parameters of the Poisson come from a Gamma distribution. The prior can incorporate knowledge, for instance, that brain-graphs tend to be rather sparse [**?**].

For example, assume that we have classes $0, 1, \ldots, C$, and edges can take any integer value. Then, $\theta = \{\lambda_0, \lambda_1, \ldots, \lambda_C\}$, corresponding to the expected weight of any edge in class c. Further assume that the hyperparameters $\{\alpha, \beta\}$ are known, and the same across classes. To estimate λ_c, we have:

[2] Is it interesting to note that neither paradigm actually operates directly on the data? The model-free approach operates on ρ, and the model-based approach operates on $\widehat{\boldsymbol{\theta}}$.

$$\widehat{\lambda}_c = \underset{\lambda_c \geq 0}{\operatorname{argmax}} \prod_{l|m^l=c} P(\lambda_c|b^l) = \underset{\lambda_c \geq 0}{\operatorname{argmax}} \left(\prod_{l|m^l=c} P(b^l|\lambda_c) \right) P(\lambda_c) = \underset{\lambda_c \geq 0}{\operatorname{argmax}} \prod_{\substack{l|m^l=c \\ (i,j)\in E^l}} P_E(e^l_{ij}|\lambda_c) P(\lambda_c)$$

$$= \underset{\lambda_c \geq 0}{\operatorname{argmax}} \prod_{\substack{l|m^l=c \\ (i,j)\in E^l}} \text{Poisson}(e^l_{ij};\lambda_c)\text{Gamma}(\lambda_c;\alpha,\beta)$$

$$= \underset{\lambda_c \geq 0}{\operatorname{argmax}} \prod_{\substack{l|m^l=c \\ (i,j)\in e^l}} \text{Gamma}(\lambda_c;\alpha + \sum_{\substack{l|m^l=c \\ (i,j)\in e^l}} e^l_{ij}, \beta + \sum_{l\in[n]} |e^l|). \tag{1}$$

Once we obtain $\widehat{\theta} = \{\widehat{\lambda}_0, \widehat{\lambda}_1, \ldots, \widehat{\lambda}_C\}$, given a new brain, b, classifying b is trivial:

$$\widehat{m} = g_n(b; \mathcal{D}_n, \widehat{\theta}) = \underset{c\in[C]}{\operatorname{argmax}}\{P_E(b|\widehat{\lambda}_c)\} = \underset{c\in[C]}{\operatorname{argmax}} \prod_{(i,j)\in e} \text{Poisson}(e_{ij};\widehat{\lambda}_c). \tag{2}$$

This strategy, of course, can be easily generalized in a number of ways. First, λ could be a function of edge identity, instead of the same across all edges. Second, the hyperparameters could be different across classes. Third, the hyperparameters can be estimated using, for instance, cross-validation. Note that for numerical reasons, we typically use log posteriors, requiring a summation instead of multiplication of many potentially small numbers.

Unfortunately, while tractable and simple, this model does not satisfy all of our above desiderata with respect to $\mathcal{P}$. In particular, this is a *parametric* model, meaning that the number of parameters is necessarily *finite* ($\leq n_v^2$). Thus, while this model can act as a naïve base model, more sophisticated models are desirable.

3.2.2 Edge conditionally independent model

In Section 3.2.1, we defined a very simple model, completely neglecting all features, and assuming the each edge is independent. For many data sets, making either of these assumptions (and all the more so for both of these assumptions) will not yield satisfactory descriptions of the data [**?**]. Here, we relax the last two assumptions. Rather, the probability of each edge is only a function of the feature vectors for the two vertices that define the edge, and the feature vector of the edge. In this model, edges are not independent, but they are *conditionally independent* given the features. Formally, denote vertex features by $x_v = (x_1, \ldots, x_{n_v})$ and denote edge features by $x_e = (x_{1,1}, \ldots, x_{1,n_v}, x_{2,1}, \ldots x_{n_v,n_v})$, so the total feature vector for a brain-graph is $x = (x_v, x_e)$. Then, the brain-graph is characterized entirely by $b = (e, x)$, and for any brain-graph, one can write:

$$P(b^l|\theta) = P(e^l, x^l|\theta) = P_E(e^l|x^l)P_X(x^l|\theta) \tag{3}$$

Assuming that all edges and features are observed, and we have classes $\{0, 1, \ldots, C\}$, yielding conditional parameters, $\{\theta_0, \theta_1, \ldots, \theta_C\}$, then we can estimate the parameters for this model, for any particular class c:

$$\widehat{\theta}_c = \underset{\theta_c\in\Theta}{\operatorname{argmax}} \prod_{l|m^l=c} P(\theta_c|e^l, x^l) = \underset{\theta_c\in\Theta}{\operatorname{argmax}} \prod_{l|m^l=c} P(e^l, x^l|\theta_c)P_\theta(\theta_c)$$

$$= \underset{\theta_c\in\Theta}{\operatorname{argmax}} \prod_{l|m^l=c} P_E(e^l|x^l)P_X(x^l|\theta_c)P_\theta(\theta_c) = \underset{\theta_c\in\Theta}{\operatorname{argmax}} \prod_{l|m^l=c} P_X(x^l|\theta_c)P_\theta(\theta_c). \tag{4}$$

To solve Eq. 4, we must only specify both P_X and P_θ, but not P_E. However, the nature of P_E will impact the other decisions, so we therefore specify it as well. Because the data under consideration are integer weighted edges, a natural choice for P_E is a Poisson distribution, with some rate, $\lambda \geq 0$:

$$P_E(e_{ij}|x_i, x_j) = \text{Poisson}(e_{ij}; f(\langle x_i, x_j\rangle)) \tag{5}$$

where $\langle\cdot,\cdot\rangle$ indicates a dot product, and f is some "link" function. Note that the dependence on edge features, x_{ij}, has been dropped. The link function here plays the role of mapping the dot product into the support of Poisson rates, which is non-negative (similar to the link function in a generalized linear model [7]). We consider two possibilities for x.

$x \in \mathbb{R}^d_+$ If x lives in some space such that $\langle x_i, x_j \rangle \geq 0$ for all $x_i, x_j \in \mathcal{X}_V$, then f can be a simple identity function. One example of such a space is the unit hypercube [**?**]. The parameters governing this distribution, $\vec{\mu}$, σ, and α control the frequency, variability and correlation of x, respectively.

sampling x
estimating parameters
conjugate prior
(waiting to get the pre-print on hypercube distributions).

$x \in \mathbb{R}^d$ However, if x's live in some more general space, such as $\mathbb{R}$, then f can be an exponential function. In such a scenario, x can be sampled from a multivariate Gaussian distribution, that is $P_X(\cdot|\theta) = \mathcal{N}(\cdot; \mu, \Sigma)$. Ignoring the prior P_θ, we can use standard tools to compute the maximum likelihood estimate (MLE) of θ_c:

$$\widehat{\mu}_c = \underset{\mu_c}{\operatorname{argmax}} \prod_{l|m^l=c} \mathcal{N}(x^l; \mu_c, \Sigma_c) = \frac{1}{n_c} \sum_{l|m^l=c} x_l \tag{6}$$

$$\widehat{\Sigma}_c = \underset{\Sigma_c}{\operatorname{argmax}} \prod_{l|m^l=c} \mathcal{N}(x^l; \mu_c, \Sigma_c) = \frac{1}{n_c} \sum_{l|m^l=c} (x^l - \widehat{\mu}_c)(x^l - \widehat{\mu}_c)^\mathsf{T}. \tag{7}$$

Alternately, the Normal-Inverse-Wishart distribution is the conjugate prior of the Gaussian distribution, so we could incorporate prior information. In practice, incorporating prior information will typically be required, as $\theta_c \in \mathbb{R}^d$ is far too high-dimensional to estimate from typical data sets, since $d \geq n_v^2 + n_v$. Thus, we can incorporate strong prior knowledge in a number of ways. First, we assume that vertex and edge features are independent, meaning Σ_c is block-diagonal. We can make Σ_c more "blocky" by assuming vertex features are independent of one another, and similarly for edge features. This yields an independent covariance matrix for each feature vector. Then, we can assume that each feature vector has a diagonal covariance matrix. More strongly, we can assume each covariance matrix is a scalar times the identity matrix. If we are feeling really saucy, we could even assume that the covariance matrices are known.

Choosing $\widehat{d}$ For both the above choices, this model can satisfy all the above criteria for $\mathcal{P}$, assuming that d is allowed to increase to infinity. Both semiparameter and nonparameter approaches could be used to choose $\widehat{d}$. Cross-validation, AIC, BIC, or many other methods may be used to determine the optimal $\widehat{d}$ from the data.

Classifying Letting $\theta = \{\theta_0, \ldots, \theta_C\}$, once $\widehat{\theta}_c \in \mathbb{R}^{\widehat{d}}$ is estimated for each $c \in [C]$, we classify a new brain $b = (e, x)$ according to:

$$\begin{aligned}\widehat{m} = g_n(b; \mathcal{D}_n, \theta) &= \underset{c\in[C]}{\operatorname{argmax}} P(b|\theta_c) = \underset{c\in[C]}{\operatorname{argmax}} P_E(e|x) P_X(x|\theta_c) P_\theta(\theta_c) \\ &= \underset{c\in[C]}{\operatorname{argmax}} \left(\prod_{i,j\in[n_v]} P_E(e_{ij}|x_i, x_j, x_{ij}) P_X(x_{ij}|\widehat{\theta}_{c_{ij}}) \right) \left(\prod_{i\in[n_v]} P_X(x_i|\widehat{\theta}_{c_i}) \right) P_\theta(\widehat{\theta}_c),\end{aligned} \tag{8}$$

where we have assumed that $P_E(x_i|\theta_c) = P_E(x_i|\theta_{c_i})$ for all i.

Unobserved features, but known parameters In the above, we implicitly assumed that all the edges and features were observed. In practice, however, this is often not the case. Assume, for the moment, that e *and* θ are known, but x is not known. Then, we have:

$$\begin{aligned}\widehat{x} &= \underset{x\in\mathcal{X}}{\operatorname{argmax}} P(x|e;\theta) = \underset{x\in\mathcal{X}}{\operatorname{argmax}} P(e|x;\theta) P_X(x|\theta) = \underset{x\in\mathcal{X}}{\operatorname{argmax}} P_E(e|x) P_X(x|\theta) \\ &= \underset{x\in\mathcal{X}}{\operatorname{argmax}} \prod_{i,j\in[n_v]} P_E(e_{ij}|x_i, x_j) P_X(x_i|\theta_i) P_X(x_j|\theta_j) \\ &= \underset{x\in\mathcal{X}}{\operatorname{argmax}} \sum_{i,j\in[n_v]} \ln P_E(e_{ij}|x_i, x_j) + \ln P_X(x_i|\theta_i) + \ln P_X(x_j|\theta_j)\end{aligned} \tag{9}$$

where the first equality follows from applying Bayes rule twice and dropping terms that do not depend on x, and the second equality comes from our model assumptions. Substituting Eq. (5) and letting P_X be a Gaussian, Eq. (9) simplifies:

$$\begin{aligned}\widehat{x} &= \underset{x\in\mathcal{X}}{\operatorname{argmax}} \sum_{i,j\in[n_v]} e_{ij} - \exp\{x_i^\intercal x_j\} - \ln(e_{ij}!) - \sum_{k=i,j} \ln\left((2\pi)^{d/2}|\Sigma_k|^{1/2}\right) - \frac{1}{2}(x_k-\mu_k)^\intercal \Sigma_k^{-1}(x_k-\mu_k) \\ &= \underset{x\in\mathcal{X}}{\operatorname{argmax}} \sum_{i,j\in[n_v]} -\exp\{x_i^\intercal x_j\} - \sum_{k=i,j} \frac{1}{2}(x_k-\mu_k)^\intercal \Sigma_k^{-1}(x_k-\mu_k) \end{aligned} \tag{10}$$

where the second equality comes from dropping terms independent of x. Seems to me like everything is now concave in x, so we can use gradient ascent to obtain $\widehat{x}$. I haven't checked yet though, and it seems a little too good to be true.

Unobserved features, and unknown parameters In practice, often neither the features x, nor the parameters, θ, are known a priori. Thus, our goal is to jointly infer x and θ. Three strategies are "natural": (i) an expectation-maximization (EM) strategy, in which one alternates computing the expected value of x^h, and using that expected value to maximize the estimate of θ, (ii) a Gibbs sampling approach, in which we recursively sample from each to build up the joint distribution, (iii) an approximate EM approach where instead of computing the full expectation, we only compute the MAP estimate. Assuming we can obtain $\widehat{x}$, we can plug it in to update $\widehat{\theta}$, and recurse.

EM strategy To use the EM strategy, assume for the moment that edges are all observed, but features are not. Therefore, the EM approach proceeds in two steps:

$$\begin{aligned}&\textbf{E step: Compute } Q(\theta,\theta') = E_{P(x|e;\theta')} \ln P(e,x|\theta) = \int P(x|e;\theta') \ln P(e,x|\theta) dx \\ &\textbf{M step: Compute } \widehat{\theta} = \underset{\theta}{\operatorname{argmax}}\, Q(\theta,\theta')\end{aligned}$$

Note that $P(e,x|\theta)$ factorizes:

$$\begin{aligned}P(e,x|\theta) &= P(e|x;\theta)P(x|\theta) = P(e|x)P(x|\theta) = \prod_{ij} P(e_{ij}|x_i,x_j)P(x_i,x_j|\theta) \\ &= \prod_{ij} P(e_{ij}|x_i,x_j)P(x_i|\theta_i)P(x_j|\theta_j)\end{aligned} \tag{11}$$

Further, $P(x|e;\theta)$ also factorizes:

$$P(x|e;\theta) = \prod_{ij} P(x_i,x_j|\vec{e}_{ij},\theta_{ij}) \tag{12}$$

where $\vec{e}_{ij} = \{\boldsymbol{e}_{i\cdot}, e_{\cdot i}, e_{j\cdot}, e_{\cdot j}\}$, where we introduce notation $e_{i\cdot} = [e_{i,1},\ldots,e_{i,n_v}]$, and $\theta_{ij} = [\theta_i,\theta_j]$. Therefore, we can expand Q:

$$Q(\theta,\theta') = \iint \prod_{i,j\in[n_v]} P(x_i,x_j|\vec{e}_{ij};\theta_{ij})[\ln P(e_{ij}|x_i,x_j) + \ln P(x_i|\theta_i) + \ln P(x_j|\theta_j)]dx_i dx_j \tag{13}$$

If $P(x|\theta)$ is Gaussian, then we can approximate $P(x_i,x_j|\vec{e}_{ij};\theta_{ij})$ as a Gaussian, and maybe the same for $P(e_{ij}|x_i,x_j)$, and then maybe something good happens?

Kidney and egg like special case only a small set of vertices change their features. fit the model assuming two classes. compute $D(P_X(x_i|\theta_0)||P_X(x_i|\theta_1))$ for all i. rank them, and call them $x_{(1)},\ldots,x_{(n_v)}$. fit the model assuming that every vertex except $x_{(1)}$ comes from a one class model, but fit $x_{(1)}$ to a two-class model. cross-validate to see how well that works. re-rank vertices using the same metric, based on this new model. repeat, adding another vertex to the two-class case, until we stop getting improvements.

4 Results

To evaluate the various above described algorithms, we use each approach on simulated data. In particular, XXX here we place the simulation from our supervenience paper XXX:

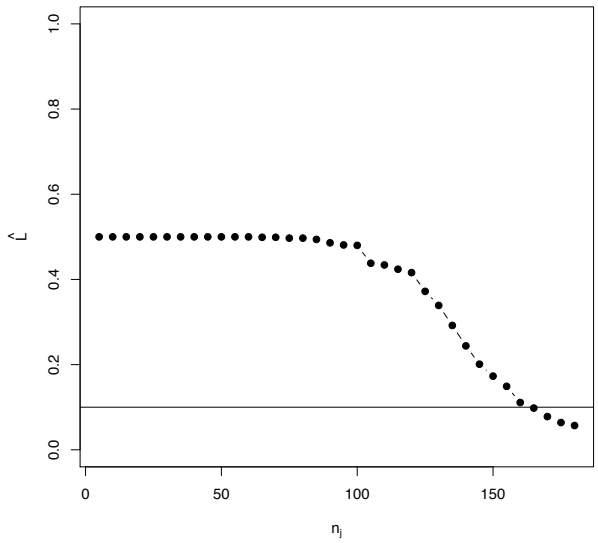

Figure 1: fig gets additional lines for additional algorithms now.

5 Discussion

Acknowledgments This research was partially supported by the NSA Research Program in Applied Neuroscience

References

[1] C. E. P. Joshua T. Vogelstein, R. Jacob Vogelstein, "Are mental properties supervenient on brain properties," *in preparation*.

[2] C. Stone, "Consistent nonparametric regression," *The annals of statistics*, vol. 5, no. 4, pp. 595–620, 1977.

[3] J. Maa, D. Pearl, and R. Bartoszynski, "Reducing multidimensional two-sample data to one-dimensional interpoint comparisons," *The Annals of Statistics*, pp. 1069–1074, 1996.

[4] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[5] E. S. Allman, C. Matias, and J. A. Rhodes, "Identifiability of parameters in latent structure models with many observed variables," *ANNALS OF STATISTICS*, vol. 37, p. 3099, 2009.

[6] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*. Springer, 1996.

[7] P. McCullagh and J. Nelder, *Generalized Linear Models*. Chapman and Hall, 1989.