A Neurocognitive Graph Theoretical Approach to Understanding the Relationship Between Minds and Brains

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

January 7, 2010

Current Paradigm The dominant paradigm of quantitative neuroscience in the 20^{th} century has been the "signal processing" framework []. Essentially, the brain is a box, that filters some stimulus, to produce some response (see Figure 1). This framework leads to the following goal:

Goal 1. *Learn the filter that the brain performs on stimuli to result in the actualized responses.*

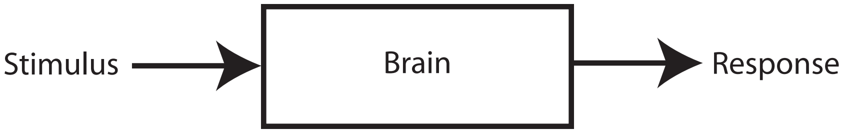

Figure 1: The signal-processing paradigm of quantitative neuroscience. The brain is a box that essentially *filters* the stimulus, outputting some response, which is often take to be multivariate time series (such as populations of spike trains or fMRI activity).

This paradigm has been appropriate, given the kind of data available to people investigating the brain. More specifically, the kind of data that has been most available has been time-series of signals related to neural activity []. Given that electrical engineers were largely the individuals obtaining and analyzing the data, it was natural to take a signal-processing approach. Often, the dynamic time-series data were used to estimate static parameters. In the previous decades, these communities have been more and more sophisticated models and algorithms to estimate these parameters []. Recently, issues such as parameter identifiability, consistency, bias, and model selection have been gaining traction as important desiderata for our models [].

Alternate Paradigm Here, we define a different goal, which suggests a complementary research paradigm to the current dogma:

Goal 2. *Construct a family of brain-models, $\mathcal{B}$, that is sufficient to provide* causal *explanations relating properties of minds with properties of brains.*

Note how the above goal is distinct in certain respects from the "filtering" goal. First, neither stimulus nor response is explicitly incorporated into this goal. While stimuli and response may be used as tools to obtain the parameters of $\mathcal{B}$, that is their only merit in this paradigm. Second, minds are explicitly incorporated into this goal. While spike trains or fMRI signal may indicate mental processes, they are likely not what is meant by "mind"; rather, they may be used as tools to infer mental states. Third, the inclusion of a class of models $\mathcal{B}$ suggests a statistical *static* framework, rather than a dynamics perspective. In addition to the above stated goal, we would like the family of models $\mathcal{P}$ to satisfy a number of desiderata:

1. $\mathcal{B}$ should be sufficiently general to account for whatever properties of the brain are casually related to the properties of cognition under investigation.

2. 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.

3. $\mathcal{B}$ should admit algorithms that (are guaranteed to be able to) capture the relationship of interest.
4. $\mathcal{B}$ should also admit *causal* studies, which entail modifying a particular $b \in \mathcal{B}$, to modify the corresponding mental property, $m \in \mathcal{M}$.

NeuroCognitive Graph Theory Here, we propose a novel approach, called NeuroCognitive Graph (NeCoG) theory, which we believe achieves the above goal and its associated desiderata. Specifically, we say that the brain may be well characterized as a labeled, attributed multigraph (which is a generalized notion of a or network[1]). Formally, we define a brain-graph, $b \in \mathcal{B}$ as a 4-tuple, $\mathcal{B} = (\mathcal{V}, \mathcal{E}, \mathcal{X}_V, \mathcal{X}_E)$, where

- $V_i \in \mathcal{V} \subseteq \mathbb{Z}$ for $i \in [V]$, is the set of vertices (nodes), where $[V] = \{1, 2, \ldots, V\}$
- $E_{ijk} \in \mathcal{E} \subseteq [V] \times [V] \times \mathbb{Z}$ for $(i, j, k) \in [V] \times [V] \times \mathbb{Z}$ is the set of (directed) edges, where $\mathbb{Z} = \{0, 1, \ldots\}$
- $\boldsymbol{X}_i \in \mathcal{X}_V \subseteq \mathbb{R}^{d_V}$ for $i \in [V]$ is the set of vertex attributes, and d_V is the dimensionality of the vertex attributes (potentially including vertex labels)
- $\boldsymbol{X}_{ijk} \in \mathcal{X}_E \subseteq \mathbb{R}^{d_E}$ for $(i, j, k) \in [V] \times [V] \times [K]$ is the set of edge attributes, d_E is the dimensionality of the edge attributes (potentially including edge labels), and k indexes edge categories, and $K \leq \infty$.

Assume, for the moment, that we take the fundamental computational unit of the brain to be a point neuron. Then, each vertex is a neuron, and each edge is a synapse. Multi-edges potentially correspond to different edge magnitudes and/or categorically different edges (like chemical and electrical synapses). Vertex attributes could include neurotransmitter released, proteins expressed, morphological properties, receptive fields, etc. Edge attributes could include probability of release, post-synaptic potential shape, etc. This level of description, however, is not necessary. For instance, $\mathcal{V}$ might instead correspond to neuroanatomical regions, which admit very different notions for edges and attributes. This multi-scale aspect of $\mathcal{B}$ is an important advantage over other frameworks.

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 cognitive (mental) property, and $g \in \mathcal{G}$ be some function to learn.

- If $\mathcal{M} = \{0, 1\}$, then g is a two-way classifier: $g : \mathcal{B} \to \{0, 1\}$.
- If $\mathcal{M} = \{0, 1, \ldots, C\}$, then g is an $C-$way classifier: $g : \mathcal{B} \to \{0, 1, \ldots, C\}$.
- If $\mathcal{M} = \mathbb{R}^a$, then g is a (multivariate-) regressor: $g : \mathcal{B} \to \mathbb{R}^a$.

In general, solving the above problems—which means finding g—will depend on $F = F_{BM}$, the joint distribution of brains and minds. In practice, however, F is unknown, and therefore g must be estimated from the data. Assume we have a corpus of training data, $\mathcal{D}_n = \{(B_1, M_1), \ldots, (B_n, M_n)\}$, where n is the number of training samples. Our goal then is to compute $g_n : \mathcal{B} \times (\mathcal{B}, \mathcal{M})^n \to \mathcal{M}$, which takes as input an observed brain-graph b and n training paris $\{(b_1, m_1), \ldots, (b_n, m_n)\}$ and produces a prediction $\widehat{m} = g_n(b; \mathcal{D}_n)$. The goal is to find a g_n that minimizes some loss function, $L_F(g_n)$. For instance, when $|\mathcal{M}| = 2$, a potentially reasonable loss function is $L_F(g_n) = \mathbb{E}[P_F[g_n(B; \mathcal{D}_n) \neq M | \mathcal{D}_n]]$.

Finding a good g_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 g_n. Vogelstein et al. (2010) showed that a k_n nearest neighbor (knn) classifier is a universally consistent classifier (meaning, achieves the Bayes optimal performance), for F_{BM}, under a Frobenius norm distance metric. While this is a desirable property, our belief is that other g_n's may outperform the knn classifier on finite data sets. More specifically, while knn induces no bias whatever into g_n, the variance is large. Thus, by incorporating neuroscientific knowledge about these brain-graphs into g_n, it may be possible to only marginally increase the bias, but drastically reduce the variance, yielding improved performance.

One possible strategy is to propose a class of models, $\mathcal{P} = \{P_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \boldsymbol{\Theta} \subseteq \mathbb{R}^d\}$, whose dimensionality d could depend on the data. The goal then is to find a Minimally Sufficient Model (MSM; by analogy with minimally sufficient statistics), which is the brain-graph with the least parameters that is sufficient to explain the mental property under investigation. Classification/regression with g_n is then performed on $\widehat{\boldsymbol{\theta}}$, which lives in some space smaller than $\mathcal{B}$, thereby reducing the variance, without increasing bias too much (hopefully).

[1] I think binary graphs should probably go here, with extensions later.

Generative Model Consider the following generative model for attributed multigraphs:

- Let c be a *class identity*, where $c \in \mathcal{C} = \{0, 1, \ldots, C\}$
- Let $\boldsymbol{\theta}_c$ be the *parameters* for class c, where $\boldsymbol{\theta}_c \in \boldsymbol{\Theta} \subseteq \mathbb{R}^d$, for some $d = d_V + d_E \in \mathbb{Z}$, where the dimensionality of the parameters is implicitly a function of the data, $\mathcal{D}_n$.
- Let G be a *graph*, where $G(\boldsymbol{\theta}_c) \in (\mathcal{V}, \mathcal{E}, \mathcal{X}_V, \mathcal{X}_E)$, where for clarity, we restrict edges to be integer weights (i.e., only include a single category of edge attributes, but this may straightforwardly generalized)

To sample graphs from this generative model, assuming that C and V are given (the number of classes and vertices per graph, respectively), one can use the following procedure (generalizing to the unknown V case is straightforward and therefore omitted):

- sample $c \sim f_c$
- sample $\boldsymbol{\theta}_c \sim f_{\boldsymbol{\theta}}(\cdot|c)$
- for $i \in [V]$, sample $\boldsymbol{X}_i \sim f_{X_V}(\cdot|\boldsymbol{\theta}_c^V)$
- for $(i, j) \in [V] \times [V]$
  - sample $\boldsymbol{X}_{ij} \sim f_{X_E}(\cdot|\boldsymbol{\theta}_c^E)$
  - sample $E_{ij} \sim f_E(\cdot|\boldsymbol{X}_i, \boldsymbol{X}_j, \boldsymbol{X}_{ij})$

Note that we have partitioned $\boldsymbol{\theta}_c$ into $\boldsymbol{\theta}_c^V$ and $\boldsymbol{\theta}_c^E$. The probability of obtaining any graph, when using this procedure, is therefore given by:

$$P(G|C, V) = \left(\prod_{i,j \in [V]} f_E(E_{ij}|\boldsymbol{X}_i, \boldsymbol{X}_j, \boldsymbol{X}_{ij}) f_{X_E}(\boldsymbol{X}_{ij}|\boldsymbol{\theta}_c^E) \right) \left(\prod_{i \in [V]} f_{X_V}(\boldsymbol{X}_i|\boldsymbol{\theta}_c^V) \right) f_{\boldsymbol{\theta}}(\boldsymbol{\theta}_c|c) f_c(c) \qquad (1)$$

Evaluating Eq. 1 requires defining $f = \{f_c, f_{\boldsymbol{\theta}}, f_{X_v}, f_{X_E}, f_E\}$ (note that $f_{\boldsymbol{\theta}}$ implicitly depends on defining $\boldsymbol{\Theta} \subseteq \mathbb{R}^d$, which, in turn, requires a rule for deciding d based on the data). The most natural choice for f_c is a multinomial, that is, $f_c = MN(\vec{\pi})$, where $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_C\}$, $\pi_c > 0$, and $\sum_c \pi_c = 1$. Our task is then to choose the rest of f that yields: (i) models that display the properties of the data, and (ii) are statistically tractable, meaning that $\boldsymbol{\theta}$ may be estimated consistently from the data [2]. To proceed, we first write the posterior of interest. Importantly, it is often the case that some attributes and edges are hidden. Define $\boldsymbol{E} = \{E_{ij}\}_{i,j \in [V]}$, and let $\boldsymbol{E} = \boldsymbol{E}^h \cup \boldsymbol{E}^o$, where $\boldsymbol{E}^h$ corresponds to hidden edges, and $\boldsymbol{E}^o$ corresponds to observed edges. Similarly, let $\boldsymbol{X} = \boldsymbol{X}_V \cup \boldsymbol{X}_E$, and then $\boldsymbol{X} = \boldsymbol{X}^h \cup \boldsymbol{X}^o$. Finally, define $\boldsymbol{\theta} = \{\boldsymbol{\theta}_c\}_{c \in [C]}$. Thus, we are interested in estimating/maximizing:

$$P(\boldsymbol{\theta}, \boldsymbol{X}^h, \boldsymbol{E}^h|C, V, \boldsymbol{X}^o, \boldsymbol{E}^o) \propto \ldots \qquad (2)$$

Several strategies for maximizing Eq. 2 are possible. First, one could use a Gibbs sampling strategy, iteratively sampling $\boldsymbol{\theta}$, $\boldsymbol{X}^h$, and $\boldsymbol{E}^h$. Second, one could use an expectation maximization algorithm, recursively finding the expected values for $\boldsymbol{X}^h$ and $\boldsymbol{E}^h$, and then use them to estimate $\boldsymbol{\theta}$. The precise definition of f might necessitate approximating both these strategies. Alternately, greedy or variational approaches might be more efficient.

[2] The above generative framework generalizes and unifies previously proposed stochastic graph models, including Random Dot Product Graphs, stochastic block models, etc.

kidney and egg problem Imagine we are in the classification setting. It is quite possible that the difference between $P(B|M = 0)$ and $P(B|M = 1)$ could be purely a function of a fraction of edges having a different probability distribution. If so, one can define a set S containing all those edges: $S = \{(i,j)|(i,j) \in S\}$. The null hypothesis would be, $H_0 : \exists S$ such that $P(\{E_{ij}\}_{(i,j)\in S}|C = 0) \neq P(\{E_{ij}\}_{(i,j)\in S}|C = 1)$. Given the above class of models, we may write:

$$\begin{aligned} P(\{E_{ij}\}_{(i,j)\in S}|c) &= f(\cdot|\boldsymbol{X}, \boldsymbol{\theta}_c) \\ &= f_E(\cdot|\boldsymbol{X}) f_X(\cdot|\boldsymbol{\theta}_c) f_{\boldsymbol{\theta}}(\cdot|c) \\ &= f_E(\cdot|\boldsymbol{X}) f_{X^h}(\cdot|\boldsymbol{X}^o, \boldsymbol{\theta}_c) f_{\boldsymbol{\theta}}(\cdot|c) \end{aligned} \tag{3}$$

Limitations/extensions attributed vertices, hyperedges

Simulated applications

Concluding thoughts