Marginal Matching for the Neural "Permutation Problem"

Scott W Linderman

October 17, 2016

1 Introduction

The nematode *C. Elegans* is a unique model organism for neuroscientists in that its connectome, or neural wiring diagram, has been known for at least three decades (**?**). Despite this knowledge, an understanding of the functional significance of these synaptic connections has remained elusive. Recently, **?** measured the simultaneous activity of hundreds of neurons in head-fixed *C. Elegans*, providing an exciting opportunity to tackle this problem. With this data, we can attempt to infer how the one neuron influences its post-synaptic neighbors based on patterns of pre- and post-synaptic activity. However, these attempts are stifled by a major obstacle: the identity of the neurons, and hence the set of synaptic partners, is only partially known. Before we can infer the functional weights of the connections, we must first infer the neuron identities. To tackle this problem, we propose to find a labeling of neurons that approximately maximizes the marginal likelihood of the data, integrating over possible values of the functional connection weights.

2 Problem Statement

The adult hermaphrodite *C. Elegans* has $N = 302$ neurons, each of which is given a unique name, like AAVL or BWM. The "connectome" corresponds to a list of known synaptic connections that are present in every individual, which we may represent as a binary adjacency matrix, $\boldsymbol{A} \in \{0,1\}^{N \times N}$, where the entry $a_{n,n'} \triangleq a_{n \to n'}$ is 1 if there exists a directed synapse from neuron n to neuron n' and 0 otherwise. Since the synapses in *C. Elegans* are all gap junctions, these connections are undirected, making $\boldsymbol{A}$ symmetric.

While the connectome has been known for decades (**?**), the *weight*, or functional signficance, of those connections remains largely a mystery. We will represent the collection of functional weights as a matrix $\boldsymbol{W} \in \mathbb{R}^{N \times N}$, where $w_{n,n'} \triangleq w_{n \to n'}$ denotes the strength of the connection from neuron n to neuron n'. First, note that this matrix is not necessarily symmetric since the influence of a gap junction depends on the input resistance of the pre- and post-synaptic neurons. Second, note that this is overparameterized: the weighted adjacency matrix is actually given by the elementwise product $\boldsymbol{A} \odot \boldsymbol{W}$, rendering the values of $w_{n \to n'}$ meaningless where $a_{n \to n'} = 0$. Nevertheless, this parameterization will prove notationally convenient.

Our goal is to infer $\boldsymbol{W}$ from measurements of neural activity. First, consider the setting in which we record from a single worm and observe the matrix $\boldsymbol{Y} \in \mathbb{R}^{T \times M}$, where T is the number of time bins and $M \leq N$ is the number of observed neurons. For example, the entry $y_{t,m}$ may denote the relative change in fluorescence, i.e. the $\Delta F/F$, of neuron m at time t.

We map these M neurons onto the complete list of N neurons via a partial permutation,

$$\pi : \{1, \ldots, M\} \to \{1, \ldots, N\},$$

where π is an injective mapping from observed neurons to known neuron identities. In other words, a neuron identity can be assigned to at most one of the M observed neurons. We then define the partial permutation matrix, $\boldsymbol{P} \in \{0,1\}^{M \times N}$, where $P_{m,n} = \mathbb{I}[\pi(m) = n]$, and $\mathbb{I}[\cdot]$ is an indicator function that evaluates to one if its argument is true and to zero otherwise. Thus, $\boldsymbol{P}\boldsymbol{A}\boldsymbol{P}^\top$ is the $M \times M$ submatrix of $\boldsymbol{A}$ corresponding to the M observed neurons.

In general, the mapping π is not known *a priori*. We are typically only certain about the labels of, say, twenty or thirty of the neurons. Thus, to infer $\boldsymbol{W}$, we also need to infer π. The next section derives the marginal probability of a given permutation, a natural objective for empirical Bayesian optimization.

3 Marginal Probability of a Permutation

We begin by formulating a prior distribution on both $\boldsymbol{W}$ and π (which determines $\boldsymbol{P}$). The former will be taken to be a weak Gaussian distribution,

$$\boldsymbol{w}_n \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}),$$

where $\boldsymbol{w}_n$ is the n-th column of $\boldsymbol{W}$. For example, we could set $\boldsymbol{\mu} = \boldsymbol{0}$ and $\boldsymbol{\Sigma} = \sigma^2 \boldsymbol{I}$. We do not specify the prior on π in detail here, but note that it should depend on known and measured locations of the cells.

Next we specify an autoregressive model for the observed activity,

$$\begin{aligned} y_{t,m} &= \boldsymbol{x}_t^\top \boldsymbol{P}(\boldsymbol{w}_{\pi(m)} \odot \boldsymbol{a}_{\pi(m)}) + \epsilon_{t,m}, \\ \epsilon_{t,m} &\sim \mathcal{N}(0, \eta^2_{\pi(m)}), \end{aligned}$$

where $\boldsymbol{w}_{\pi(m)} \in \mathbb{R}^N$ and $\boldsymbol{a}_{\pi(m)} \in \{0,1\}^N$ are the $\pi(m)$-th columns of $\boldsymbol{W}$ and $\boldsymbol{A}$, respectively, and $\boldsymbol{x}_t \in \mathbb{R}^M$ is a deterministic function of the preceding activity,

$$x_{t,m} = f(y_{1,m}, \ldots, y_{t-1,m}).$$

Note that this model ignores the activity of the $(N - M)$ unobserved neurons.

Now we can write the joint probability of the model,

$$p(\boldsymbol{Y}, \boldsymbol{W}, \pi \,|\, \boldsymbol{A}) = p(\boldsymbol{W})\, p(\pi) \prod_{t=1}^{T} \prod_{m=1}^{M} p(y_{t,m} \,|\, \boldsymbol{x}_t, \boldsymbol{W}, \pi, \boldsymbol{A}).$$

Since both the prior on $\boldsymbol{w}_n$ and the likelihood for $y_{t,m}$ are linear and Gaussian in $\boldsymbol{w}_n$, the conditional distribution of $\boldsymbol{w}_n$ is Gaussian as well. Let $\boldsymbol{X} \in \mathbb{R}^{T \times M}$ be a matrix with rows given by $\boldsymbol{x}_t^\top$. Fixing π, we have,

$$\begin{aligned} p(\boldsymbol{w}_{\pi(m)} \,|\, \boldsymbol{Y}, \pi, \boldsymbol{A}) &\propto \mathcal{N}(\boldsymbol{w}_{\pi(m)} \,|\, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \prod_{t=1}^{T} \mathcal{N}(y_{t,m} \,|\, \boldsymbol{x}_t^\top \boldsymbol{P}(\boldsymbol{w}_{\pi(m)} \odot \boldsymbol{a}_{\pi(m)}), \eta^2_{\pi(m)}) \\ &\propto \mathcal{N}(\boldsymbol{w}_{\pi(m)} \,|\, \widetilde{\boldsymbol{\mu}}_{\pi(m)}, \widetilde{\boldsymbol{\Sigma}}_{\pi(m)}), \end{aligned}$$

where

$$\widetilde{\boldsymbol{\Sigma}}_{\pi(m)} = \left[\boldsymbol{\Sigma}^{-1} + \eta_{\pi(m)}^{-2}(\boldsymbol{P}^\top \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{P}) \odot (\boldsymbol{a}_{\pi(m)}\boldsymbol{a}_{\pi(m)}^\top)\right]^{-1},$$
$$\widetilde{\boldsymbol{\mu}}_{\pi(m)} = \widetilde{\boldsymbol{\Sigma}}_{\pi(m)}\left[\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + \eta_{\pi(m)}^{-2}(\boldsymbol{P}^\top \boldsymbol{X}^\top \boldsymbol{y}_{:,m}) \odot \boldsymbol{a}_{\pi(m)}\right].$$

Moreover, since the model is conditionally conjugate, we can evaluate the *marginal* probability of the observed data and a permutation, integrating over the corresponding weights of the network. We have,

$$\begin{aligned}
p(\boldsymbol{Y}, \pi \mid \boldsymbol{A}) &= \int p(\boldsymbol{Y}, \boldsymbol{W}, \pi \mid \boldsymbol{A}) \, \mathrm{d}\boldsymbol{W} \\
&\propto p(\pi) \prod_{m=1}^{M} \int p(\boldsymbol{y}_m \mid \boldsymbol{w}_{\pi(m)}, \boldsymbol{a}_{\pi(m)}) \, p(\boldsymbol{w}_{\pi(m)}) \, \mathrm{d}\boldsymbol{W} \\
&= p(\pi) \prod_{m=1}^{M} \frac{|\boldsymbol{\Sigma}|^{-1/2} \exp\left\{-\frac{1}{2}\boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\right\}}{\left|\widetilde{\boldsymbol{\Sigma}}_{\pi(m)}\right|^{-1/2} \exp\left\{-\frac{1}{2}\widetilde{\boldsymbol{\mu}}_{\pi(m)}^\top \widetilde{\boldsymbol{\Sigma}}_{\pi(m)}^{-1}\widetilde{\boldsymbol{\mu}}_{\pi(m)}\right\}} \\
&= p(\pi) \prod_{m=1}^{M} p(\boldsymbol{y}_m \mid \pi, \boldsymbol{A}).
\end{aligned}$$

This provides a "score" that we can attempt to optimize over the space of permutations. In the recordings of **?**, a subset of $L < M$ neurons have already been labeled, making the problem slightly easier. Still, the number of such permutations is $(M-L)!$, and, to our knowledge, this poses an intractable optimization problem. Instead, we propose a simplified version of this problem that can be solved exactly.

4 An Approximate Solution via Weighted Bipartite Matching

We will attempt to find a permutation by solving a relaxed version of the maximum marginal likelihood problem above. Specifically, we will formulate the permutation inference problem as a weighted bipartite matching problem, for which the Hungarian algorithm (**?**) provides an exact solution in cubic time.

We construct a bipartite graph with $M-L$ nodes on the left and $N-L$ nodes on the right. The left nodes correspond to the neurons for which we seek labels, and the right nodes correspond to the possible neuron identities. For each pair of nodes m on the left and n on the right, connect them with an edge of weight, $e_{m,n}$. The Hungarian algorithm provides a matching (a subset of $M-L$ edges such that each left node is connected to exactly one right node and each right node has at most one edge) such that the sum of edge weights is maximized. Furthermore, it does so in $O((M-L)^2(N-L)^2)$ time. The question is, how should we set the edge weights?

It is not clear how to set the edge weights to correspond to the objective function above since the marginal probability depends on an entire assignment, $\pi(1), \ldots, \pi(M)$. For the matching problem, we want an objective function that just depends on a single assignment, $\pi(m)$. A simple approximation is to set,

$$e_{m,n} = p(\pi(m) = n) \, p(\boldsymbol{y}_m \mid \pi(m) = n, \widetilde{\boldsymbol{A}}(m,n)) \prod_{\ell=1}^{L} p(\boldsymbol{y}_\ell \mid \pi(m) = n, \widetilde{\boldsymbol{A}}(m,n)),$$

excusing the abuse of notation. Here, we assume the prior, $p(\pi)$ factorizes such that we can evaluate the marginal, $p(\pi(m) = n)$. There are two approximations:

1. We only consider the marginal probability of only the *labeled* neurons, $\ell = 1, \ldots, L$, and one unlabeled neuron, m.
2. We define a "pseudo-adjacency matrix" $\widetilde{\boldsymbol{A}}(m, n)$ that zeros out all entries except for the $(L+1) \times (L+1)$ submatrix corresponding to the labeled neurons and observed neuron m, which we are assigning to label $\pi(m) = n$. More formally, we define $\widetilde{\boldsymbol{A}}(m, n)$ with entries $\widetilde{a}_{i \to j}(m, n)$ as follows,

$$\widetilde{a}_{i \to j}(m, n) = \begin{cases} a_{i \to j} & \text{both } i \text{ and } j \text{ are either labeled or equal to } n \\ 0 & \text{o.w.} \end{cases}$$

This pseudo-adjacency matrix leads to the following interpretation of the edge weight: $e_{m,n}$ is the marginal probability of the activity of the labeled neurons after including a single extra neuron, m, with label $\pi(m) = n$. While this ignores a significant amount of information (i.e. the $M - L - 1$ other neurons, it successfully renders the edge weights independent of one another. Alternatively, we could set the other edges of $\widetilde{\boldsymbol{A}}(m, n)$ to one rather than zero in order to include all other possible interactions, even though many would truly be zero.

5 A Greedy Approach

For this problem, the $O(M^2 N^2)$ complexity of the Hungarian algorithm may be unacceptable. A simpler approach is to just do a greedy fit. Using the same edge weights as above, we could iterate over each unlabeled neuron, m, and each possible assignment, n, evaluate the weight of the corresponding edges, and add the assignment $\pi(m) = n$ for the highest weighted edge. After making this assignment, we add m to the set of labeled neurons and repeat. The complexity of this algorithm is only $O(M^2 N)$. However, note that evaluating the edge weights is, in the worst case, $O(M^3)$ as well. Yikes!