Inferring Direct and Indirect Functional Connectivity Between Neurons From Multiple Neural Spike Train Data

Ben Shababo **Kui Tang** **Frank Wood**
Columbia University, New York, NY 10027, USA
{bms2156,kt2384}@columbia.edu, {fwood}@stat.columbia.edu

Abstract

Our project aims to model the functional connectivity of neural microcircuits. On this scale, we are concerned with how the activity of each individual neuron relates to other nearby neurons in the population. Though several models and methods have been implemented to infer neural microcircuit connectivity, these fail to capture unobserved influences on the microcircuit. In this paper, we address these hidden influences on the microcircuit by developing a model which takes into account functional connectivity between observed neurons over more than one time step. We then test this model by simulating a large population of neurons but only observing a subpopulation which allows us to compare our inferred indirect connectivity with the known direct connectivity of the total population. With a better understanding of the functional patterns of neural activity at the cellular level, we can begin to decode the building blocks of neural computation.

1 Introduction

1.1 Problem Description

As we learn more and more about the workings of the neuron and of specialized brain regions, the question increasingly becomes, how do these pieces sum to a whole? How do the patterns of connectivity give rise to vision, memory, motor function, and so on? Currently, a broad picture of the circuitry, or graphical connectivity, of the brain does not exist, but several projects are underway to organize the solution of this problem [11, 4]. Efforts to examine connectivity of the brain focus on scales ranging from brain regions each comprised of hundreds of millions of cells down to microcircuits of only a few cells. Further, some of these projects address structural connectivity and others functional connectivity [10, 9, 14, 5, 2].

In this project, we will focus on the functional connectivity of microcircuits: how firing activity of one neuron influences the firing of other nearby neurons. Importantly, functional connectivity does not always imply anatomical connectivity; it only implies that some set of neurons fire together in correlation. These jointly firing neurons may have a common input or be linked in a chain, rather than having monosynaptic connection.

1.2 Background

Several strategies have already been employed to infer the functional connectivity of microcircuits from calcium imaging and MEA data [8, 17, 1]. Of special interest to us and our approach are two recent Bayesian approaches. In [13], a pattern-growth algorithm is used to find frequent patterns of firing activity. These patterns define mutual information between neurons which they summarize in a dynamic Bayesian network. While their methodology presents a contribution to the study of

Bayesian networks, one limitation of this work in inferring the connectivity of microcircuits is that it only discovers relationships of excitation. In [12], network activity is modeled in terms of a collection of coupled hidden Markov chains, with each chain corresponding to a single neuron in the network and the coupling between the chains reflecting the networks connectivity matrix.

1.3 Unobserved Neurons as Indirect Inputs

Although the work to date has done much to address the problem of functional neural connectivity, there are still improvements to be made to current models. For example, current models do not address unobserved inputs to the system. In this paper, we account for these indirect influences on the observed neurons by extending the model of [12] so that it captures functional weights between neurons over multiple time steps, effectively extending the model back in time. Although Patnaik et. al. found higher-order causative chains in which a set of neurons interact through chains, cycles, and polychronous circuits, their model is limited by only caputring excitatory neural influences, and being constrained to observed neurons [13]. Our model captures both positive and negative influences and accounts for effects of neurons outside of the observed field.

2 Methods

2.1 Formal Model

We extend the parametric generative model proposed by [12] of joint spike trains on N neurons in discrete time. Mischencko et al. propose a model to infer the connectivity matrix W, where each entry w_{ij} encodes the influence of neuron j on the subsequent firing of neuron i. Their model can be decomposed into one part inferring neural spike train data from florescence imaging, and another part inferring W from spike train data. We focus on the latter.

We model neural spike trains as a discrete-time hidden Markov model. Denote by $n_i(t)$ whether neuron i fired at time t. We observe the firing of each neuron, $n_i(t), i = 1, ..., N$, at each discrete time step, such that $n_i(t) = 1$ when we observe a spike and $n_i(t) = 0$ when the neuron is silent. We model $n_i(t)$ as a Bernoulli random variable with parameter $f(J_i(t))$, where

$$J_i(t) = b_i + I_i(t) + \sum_{j=1}^{N} w_{ij} h_{ij}(t), \tag{1}$$

b_i accounts for the baseline firing rate of the neuron, and $I_i(t)$ accounts for indirect influences on neuron i from a fixed window of past time steps. The history term, h_{ij}, encodes the influence of neuron j on neuron i and is only dependent on the firing of j at time $t - \Delta$ where Δ is the size of each discrete time step. From [12], we model $h_{ij}(t)$ as an autoregressive function:

$$h_{ij}(t) = (1 - \Delta/\tau) h_{ij}(t - \Delta) + n_j(t - \Delta) + \sigma\sqrt{\Delta}\epsilon \tag{2}$$

where τ is the decay time constant, σ is the standard deviation of the noise and ϵ is a standard normal random variable representing noise. The parameterization of $h_{ij}(t)$ by Δ ensures that the physical parameters of the model scale with the timestep size, which is determined by temporal resolution of observed data and computational complexity of memory.

In addition, we model the indirect inputs to n_i by summing the influences from all neurons, $n_j(t - s\Delta), s = 2, ..., S$, where S is the temporal limit on indirect influences. These higher order interactions are incorporated into the summed input to neuron $n_i(t)$ by adding the term,

$$I_i(t) = \sum_{s=2}^{S} \sum_{j=1}^{N} \beta_{ijs} n_j(t - s\Delta), \tag{3}$$

to the input function $J_i(t)$. Here, s is the number of time steps back and β_{ijs} is the weight of the indirect influence of $n_j(t - s\Delta)$ on $n_i(t)$. Along with inferring the direct connectivity matrix, W, we will also infer the three dimensional indirect connectivity matrix, β, and the instrinsic neuron parameters $\mathbf{b}$.

Following [12], we define the spiking probability as

$$f(J) = P\left(n > 0 | n \sim \text{Poiss}(e^J \Delta)\right) = 1 - \exp(-e^J \Delta) \tag{4}$$

which completes the setup for the generalized linear model. The joint probability of the entire graphical model is then

$$P(H, N|\theta) = \prod_{it} P(n_i(t)|\mathbf{h}_i(t), \mathbf{n}(t-2), ..., \mathbf{n}(t-S), \theta) P(\mathbf{h}_i(t)|, \mathbf{h}_i(t-1), \mathbf{n}(t-1), \theta). \quad (5)$$

2.2 Priors

Mishchencko et al. use two priors on the connectivity matrix, W, a sparseness prior and a prior which imposes "Dale's Law" which states that a neuron can only exert either an excitatory or an inhibitory influence on postsynaptic neurons [12]. We will include the sparseness prior in our model by adding an L1 regularization term to each w_{ij}. However, given that recent research has shown that neurotransmitter co-release is more common that once anticipated, we will not include a Dale's law prior in our model [15]. We also add L1 regularization to the indirect influences, extending the sparsity assumption. It should be noted that these priors are "improper priors," that is they are unnormalized.

3 Inference

3.1 Method & Strategy

We fit a maximum a posteriori estimate of the model through EM. For each neuron i, we represent the vector of incoming history terms from time t as $\mathbf{h}_i(t)$, and similarly for the vector of incoming weights. H denotes the set of all history terms, N denotes the set of all spike trains, and θ denotes the set of all parameters, $\{b_i, w_{ij}, \beta_{ijs} | i, j = 1 \dots N, s = 1 \dots S\}$ The expected value of the complete-data log-likelihood is given by

$$\begin{aligned} Q(\theta, \theta^{old}) &= E_{P(H|N,\theta^{old})} \left[\log P(H, N|\theta)\right] \\ &= \int \log \left[P(H, N|\theta)\right] P(H|N, \theta^{old}) dH \\ &= \sum_{it} \int \log \left[P(\mathbf{h}_i(t), n_i(t)|\theta^{old})\right] P(\mathbf{h}_i(t)|N, b_i, \mathbf{w}) d\mathbf{h}_i(t) \end{aligned}$$

where the last equality holds since the integrand can be separated into a sum of terms for each neuron-timestep (i, t).

Since each individual HMM chain in this coupled HMM model is d-separated from each other by the observed firing variables N and each neuron has its own set of parameters b_i, $\mathbf{w}_i$, and $\boldsymbol{\beta}_i$, we can perform EM over each neuron individually. This allows us to use MATLAB's parallel programming techniques to greatly speed up the algorithm. For the M-step we take advantage of MATLAB's optimization toolbox to maximize $Q(\theta, \theta^{old})$ over θ.

3.2 E-Step with Sequential Monte Carlo

The main challenge of the E-step is to update the joint distribution of the model by finding values for the hidden sequences $\mathbf{h}_i$. For this model there is no analytic update to the values of H, so we employ sampling to approximate distributions over each $\mathbf{h}_i(t)$. Since the transition probabilities, (2), are linear-Gaussian distributions, but the emission probabilities, (5), are not, we cannot use a standard Kalman filter [3]. Instead, we adapt a sequential Monte Carlo approach used in [16, 12] to estimate the latent history terms by by producing a sampled approximation of $Q(\theta, \theta^{old})$. We use a standard forward particle filter [3] and a marginal backward smoother [6] to obtain our samples.

We initialize the particle filter by sampling M particles from a normal distribution centered at 0 with standard deviation $\sigma\sqrt{\Delta}$ and each particle having uniform weight. At each timestep we draw particles from the *prior proposal* $P(\mathbf{h}_i^{(m)}(t)) = P(\mathbf{h}_i(t)|\mathbf{h}_i^{(m)}(t-1))$ which is a just the normal distribution given by (2). We update the particle weights using the recurrence (omitting θ for brevity)

$$p_f^{(m)}(t) = p_f^{(m)}(t-1) P(n_i(t)|\mathbf{h}_i(t)), \quad (6)$$

where $p_f^{(m)}(t)$ is the forward probability, or particle weight, for that sample.

As the particle filter evolves, many particles will have low likelihoods in the next timeslice, driving many weights to zero. To prevent degeneracy, we employ stratified resampling such that we draw new samples when the effective number of particles,

$$N_{eff} = |\mathbf{p}_f(t)|^{-1}, \tag{7}$$

is less than $M/2$, where $\mathbf{p}_f(t)$ denotes a vector of all particle weights for one timestep [16]. That is, we draw M new samples from our current set of samples with replacement with probabilities proportional to their current weights, and reset weights to uniform.

Once all samples and forward weights $p_f^{(m)}$ have been computed, we run the backwards marginal filter recurrences beginning at $t = T$, where the filtering and smoothing distributions are identical:

$$r^{(m,m')}(t, t-1) = p_b^{(m)} \frac{p_f^{(m)} P(\mathbf{h}_i^{(m)}(t) | \mathbf{h}_i^{(m')}(t-1))}{\sum_{m'} p_f^{(m')}(t-1) P(\mathbf{h}_i^{(m)}(t) | \mathbf{h}_i^{(m')}(t-1))} \tag{8}$$

$$p_b^{(m')}(t-1) = \sum_{m=1}^{M} r^{(m,m')}(t, t-1) \tag{9}$$

giving the approximate distribution

$$P(\mathbf{h}_i(t) | N, \theta^{old}) = \sum_{m'=1}^{M} p_b^{(m')} \delta \left[\mathbf{h}_i(t) - \mathbf{h}_i^{(m')}(t) \right] \tag{10}$$

where δ is the Dirac delta function.

3.3 M-Step

In the M step, we update θ such that

$$\theta = \arg\max_{\theta} \left\{ Q(\theta, \theta^{old}) \right\}. \tag{11}$$

We performed the M step using MATLAB's optimization toolbox with the objective function $Q(\theta, \theta^{old})$ subject to the following constraints:

1. $0 \leq b_i \leq 5$
2. $-5 \leq \mathbf{w}_{ij}, \beta_{ijs} \leq 5$

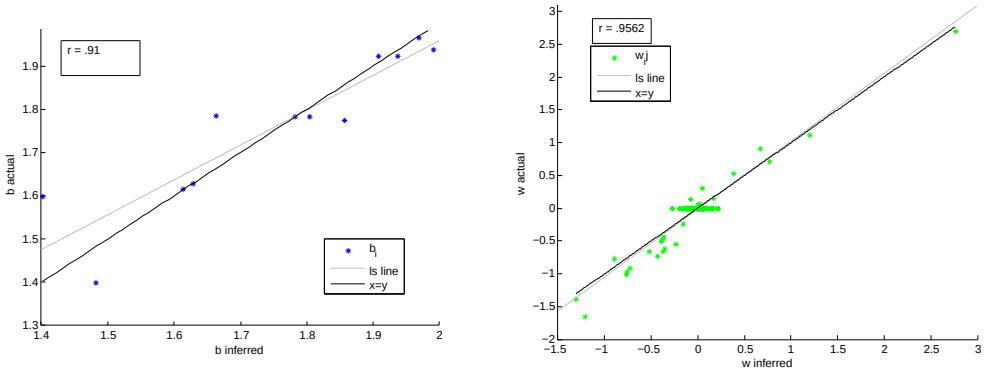
Furthermore, we included an L1-regularizer on both W and β with $\lambda_W = 4$ and $\lambda_\beta = 1$. The results is a sparse prior on these parameters and changes Q such that

$$\begin{aligned}
\theta &= \arg\max_{\theta} \left\{ \int \log \left[P(H, N | \theta) \right] P(H | N, \theta^{old}) P(\theta^{old}) dH \right\} \\
&= \arg\max_{\theta} \left\{ \sum_{it} \int \log \left[P(\mathbf{h}_i(t), n_i(t) | \theta^{old}) \right] P(\mathbf{h}_i(t) | N, b_i, \mathbf{w}) d\mathbf{h}_i(t) - \lambda_W \sum_{ij} |w_{ij}| - \lambda_\beta \sum_{ijs} |\beta_{ijs}| \right\}
\end{aligned}$$

4 Experiments

4.1 Data Source & Simulation

We tested our model on actual multiple neuronal spike train data as well as on simulated spike trains. The actual data have been provided by the Buszaki Lab and was recorded simultaneously from 87 prefrontal cortex neurons of a behaving rat over the course of roughly 40 minutes. This data has been pre-processed so that we begin with the spike trains which are in the expected form of $n_i(t)$, that is binary random variables.



(a) Baselines (b) Connectivity Weights

Figure 1: Inferred vs. true b_i and w_{ij} values for fully observed set of 12 neurons.

The simulated neuronal spike train data was generated from our model using equations (1), (2), and (5) and with $.02 \leq \sigma \leq .25$, $\tau = .02$, and $\Delta = .01$. However, it is very important to note that we did not include the term $I_i(t)$ in simulation, since this value only carries meaning when dealing with a subpopulation of neurons. Values for $\mathbf{b}$ were chosen from from $\mathcal{N}(1.64, .2)$ and W from $\mathcal{E}(.5)$ for excitatory connections and $\mathcal{E}(2.3)$ for inhibitory connections - where $\mathcal{E}(\lambda)$ signifies an exponential distribution with mean λ. W was then altered such that only about 10% of values were non-zero.

4.2 Results

4.2.1 Validation

Before investigating the effects of our novel $\boldsymbol{\beta}$ terms, we validated our algorithm by inferring only the $\mathbf{b}$ and W parameters of a completely observed set of 12 neurons. Because we observe the full population, the $\boldsymbol{\beta}$ terms have no meaning and are omitted.

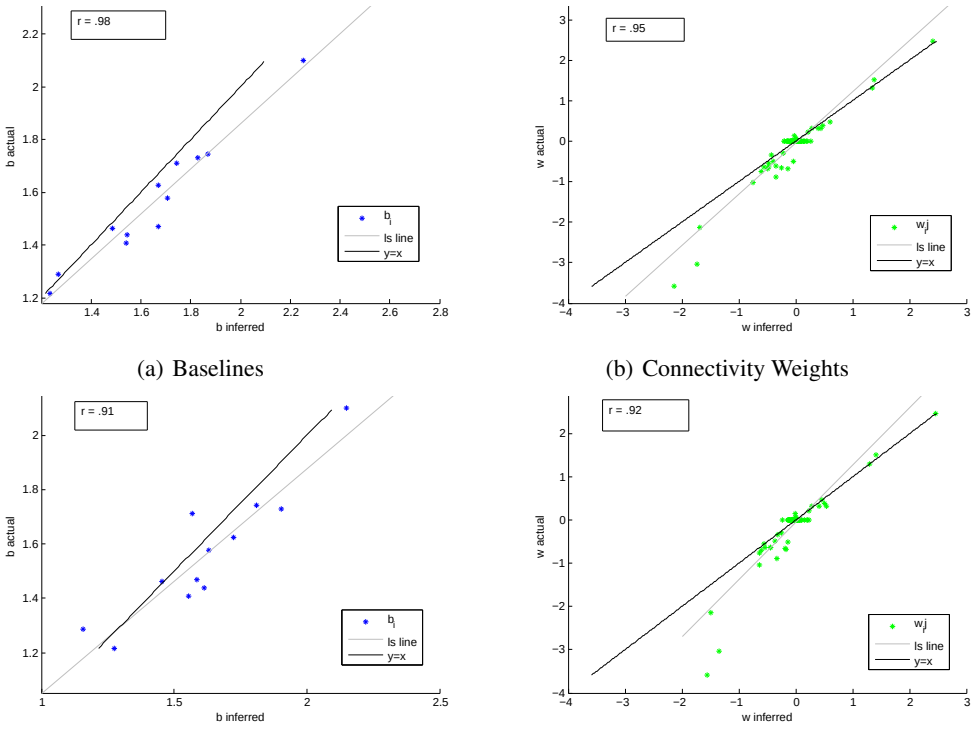
As shown in Figure 1, we were able to closely retrieve the parameters and therefore are confident that our algorithm works. However, it should be noted that here as well as in other experiments we generally underestimated inhibitory weights. This makes sense because firing is sparse in these networks and inferring inhibitory connections is dependent on coincident firing between an excitatory neuron, and inhibitory neuron, and a shared target neuron. Furthermore, it is difficult to determine an inhibitory weight unless a target neuron is firing tonically. Nonetheless, strong inhibitory connections are still well represented in our results.

4.2.2 Indirect Weights

In the case of simulated data, we have an interesting opportunity to test the quality of the indirect component of our model. We can simulate a large population neurons and only use a subset as the observed neurons in our model. We can then compare our inferred indirect weights with the known direct weights of the simulated data. Also, we can compare the results of the $\mathbf{b}$ and W parameters of the same data with $\boldsymbol{\beta}$ inferred and not inferred.

When we performed inference on a subpopulation of 12 neurons (out of 25 total) and omitted the $\boldsymbol{\beta}$ parameters we still found a high correlation between the actual and inferred values of $\mathbf{b}$ and W, however we consistently overestimated the values of $\mathbf{b}$. When we included the $\boldsymbol{\beta}$ parameters, we found that the correlation remained more or less the same for both $\mathbf{b}$ and W, but we much more accurately inferred the $\mathbf{b}$ values Figure 2. These results make sense because without indirect weights, any excitatory input into the system (excitatory inputs are more likely in our simulation and in actual nervous systems) would have to be summed into the $\mathbf{b}$ parameters. When we include indirect weights, these input can be put in their proper place, making estimations of $\mathbf{b}$ more accurate.

We also designed a network such that there was a series of strong excitatory connections from an observed neuron, through several unobserved neurons, and back into the observed population. When we performed inference over this network, we were able to find relatively strong β value that cor-



(a) Baselines

(b) Connectivity Weights

(c) Baselines with Indirect Weights

(d) Connectivity Weights with Indirect Weights

Figure 2: Inferred vs. true b_i and w_{ij} values for an observed set of 12 out of 25 total neurons. The top row shows inference without indirect weights. The bottom row shows the accuracy improvement when indirect weights are added to the model. See text for details.

responded with this unobserved path of influence on the appropriate time scale. For example, the network contained the chain of weights $w_{13,4} = 1.9$, $w_{1,13} = 2.0$. This results in an indirect connection from neuron 4 to neuron 1 through the unobserved neuron 13. We found the corresponding values $\beta_{1,4,1} = .33$, $\beta_{1,4,2} = .34$, and $\beta_{1,4,3} = .17$. (Compare these values to the average over $\beta_{1js} = .009$ and variance of .016.) The subscripts here mean that neuron 4 is influencing neuron 1 indirectly over 2-4 time steps.

However, we did not find a representative β value in every instance where it would seem likely. This could either be a flaw in our model or a result of more complex interaction in the network that are difficult to analyze upon inspection. Also, the correlation of inferred to actual $\mathbf{b}$ values was lower than in most of our experiments, $r = .49$.

4.2.3 Real Data

When took spike trains recorded from 87 rat prefrontal cortex, binned them such that $\Delta = .010$ and performed inference over a subpopulation of 12 neurons. Our results here were more ambiguous, returning a handful of fairly weak inhibitory neurons among other neurons with no strong weights onto other observed neurons. For $\boldsymbol{\beta}$, we found that $\beta_{max} = .1234$, $\beta_{min} = -.0594$, $\beta_{mean} = -.00062$, and $\beta_{var} = .0003$.

5 Conclusions & Future Directions

While we feel confident that our algorithm and methodology are sound and we see some improvement in inference when we include $\boldsymbol{\beta}$, there is clearly more work to be done to fit indirect connectivity into the current model. Though we improved the $\mathbf{b}$ values, one would expect the values to be originally overestimated and then reduced instead of the opposite. This is because with indirect weights accounted for, their weights would have to be included in $\mathbf{b}$. We plan to look into this logic more in further experiments. Also, we would like to investigate creating a prior for $\boldsymbol{\beta}$ using methods similar to the pattern-growth algorithms used in [13].

References

[1] Paulo Aguiar and Miguel Rodrigues. Neuronal connectivity inference from spike trains using an empirical probabilistic causality measure. *BMC Neuroscience*, 10 (Suppl 1):P169, 2009. ISSN 1471-2202. doi: 10.1186/1471-2202-10-S1-P169. URL http://www.biomedcentral.com/1471-2202/10/S1/P169.

[2] Sourabh Bhattacharya, Moon-Ho Ringo Ho, and Sumitra Purkayastha. A bayesian approach to modeling dynamic effective connectivity with fmri data. *NeuroImage*, 30(3): 794 – 812, 2006. ISSN 1053-8119. doi: 10.1016/j.neuroimage.2005.10.019. URL http://www.sciencedirect.com/science/article/pii/S1053811905008001.

[3] C.M. Bishop. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[4] Jason W. Bohland, Caizhi Wu, Helen Barbas, Hemant Bokil, Mihail Bota, Hans C. Breiter, Hollis T. Cline, John C. Doyle, Peter J. Freed, Ralph J. Greenspan, Suzanne N. Haber, Michael Hawrylycz, Daniel G. Herrera, Claus C. Hilgetag, Z. Josh Huang, Allan Jones, Edward G. Jones, Harvey J. Karten, David Kleinfeld, and Rolf Kötter. A proposal for a coordinated effort for the determination of brainwide neuroanatomical connectivity in model organisms at a mesoscopic scale. *PLoS Computational Biology*, 5(3):1–9, 03 2009.

[5] Ann-Shyn Chiang, Chih-Yung Lin, Chao-Chun Chuang, Hsiu-Ming Chang, Chang-Huain Hsieh, Chang-Wei Yeh, Chi-Tin Shih, Jian-Jheng Wu, Guo-Tzau Wang, Yung-Chang Chen, Cheng-Chi Wu, Guan-Yu Chen, Yu-Tai Ching, Ping-Chang Lee, Chih-Yang Lin, Hui-Hao Lin, Chia-Chou Wu, Hao-Wei Hsu, Yun-Ann Huang, Jing-Yi Chen, Hsin-Jung Chiang, Chun-Fang Lu, Ru-Fen Ni, Chao-Yuan Yeh, and Jenn-Kang Hwang. Three-dimensional reconstruction of brain-wide wiring networks in drosophila at single-cell resolution. *Curr Biol*, 21(1):1–11, Jan 2011. doi: 10.1016/j.cub.2010.11.056.

[6] Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10(3):197–208, July 2000. ISSN 0960-3174. doi: 10.1023/A:1008935410038. URL http://dx.doi.org/10.1023/A:1008935410038.

[7] Wulfram Gerstner and Werner M. Kistler. *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press, 2002.

[8] Sebastian Gerwinn, Jakob H Macke, and Matthias Bethge. Bayesian inference for generalized linear models for spiking neurons. *Front Comput Neurosci*, 4:12, 2010. doi: 10.3389/fncom.2010.00012.

[9] Viren Jain, H Sebastian Seung, and Srinivas C Turaga. Machines that learn to segment images: a crucial technology for connectomics. *Curr Opin Neurobiol*, 20(5):653–66, Oct 2010. doi: 10.1016/j.conb.2010.07.004.

[10] Seymour Knowles-Barley, Nancy J Butcher, Ian A Meinertzhagen, and J Douglas Armstrong. Biologically inspired em image alignment and neural reconstruction. *Bioinformatics*, 27(16):2216–23, Aug 2011. doi: 10.1093/bioinformatics/btr378.

[11] Daniel S Marcus, John Harwell, Timothy Olsen, Michael Hodge, Matthew F Glasser, Fred Prior, Mark Jenkinson, Timothy Laumann, Sandra W Curtiss, and David C Van Essen. Informatics and data mining tools and strategies for the human connectome project. *Front Neuroinform*, 5:4, 2011. doi: 10.3389/fninf.2011.00004.

[12] Yuriy Mishchencko, Joshua T. Vogelstein, and Liam Paninski. A bayesian approach for inferring neuronal connectivity from calcium fluorescent imaging data. Submitted to the Annals of Applied Statistics, 2011.

[13] Debprakash Patnaik, Srivatsan Laxman, and Naren Ramakrishnan. Discovering excitatory relationships using dynamic bayesian networks. *Knowledge and Information Systems*, 29:273–303, 2011. ISSN 0219-1377. URL http://dx.doi.org/10.1007/s10115-010-0344-6. 10.1007/s10115-010-0344-6.

[14] Deepak Ropireddy and Giorgio A Ascoli. Potential synaptic connectivity of different neurons onto pyramidal cells in a 3d reconstruction of the rat hippocampus. *Front Neuroinform*, 5:5, 2011. doi: 10.3389/fninf.2011.00005.

[15] E.A. Thomas and J.C. Bornstein. Inhibitory cotransmission or after-hyperpolarizing potentials can regulate firing in recurrent networks with excitatory metabotropic transmission. *Neuroscience*, 120(2):333 – 351, 2003. ISSN 0306-4522. doi: 10.1016/S0306-4522(03)00039-3. URL http://www.sciencedirect.com/science/article/pii/S0306452203000393.

[16] Joshua T. Vogelstein, Brendon O. Watson, Adam M. Packer, Rafael Yuste, Bruno Jedynak, and Liam Paninski. Spike inference from calcium imaging using sequential monte carlo methods. *Biophysical Journal*, 97(2):636 – 655, 2009. ISSN 0006-3495. doi: 10.1016/j.bpj.2008.08.005. URL http://www.sciencedirect.com/science/article/pii/S0006349509003117.

[17] Daniel Yasumasa Takahashi. Connectivity inference between neural structures via partial directed coherence. *Journal of applied statistics*, 34(10):1259–1273, 2007. doi: 10.1080/02664760701593065.

Appendix

5.1 Hyperparameters

We set $.02 \leq \sigma \leq .25$, $\tau = .02$ [7], and $\Delta = .01$ for both the simulation and inference, though we did vary both σ and τ and found our inference robust to those changes.

5.2 Derivatives for M-step

Evaluating Q is expensive as it requires a complete pass over the data—up to 15,000 timesteps. Analytical derivatives - the gradient and Hessian - are essential; fortunately, they are not difficult to derive. Denote by θ_1, θ_2 any scalar parameters. Then

$$\begin{aligned}
\frac{\partial Q}{\partial \theta_1} &= (1 - n_i(t)) \left(-\exp\left\{-e^{J_i(t)}\right\} \Delta \frac{\partial J}{\partial \theta_1} \right) \\
&\quad + n_i(t) \frac{\exp\left\{-e^{J_i(t)}\Delta + J_i(t)\right\} \Delta \frac{\partial J}{\partial_\theta 1}}{1 - \exp\left\{-e^{J_i(t)}\Delta\right\}} \\
\frac{\partial^2 Q}{\partial \theta_1 \partial \theta_2} \Big/ \frac{\partial J}{\partial \theta_1} \frac{\partial J}{\partial \theta_2} &= \exp\left\{-e^{J_i(t)\Delta} + J_i(t)\right\} \Delta \\
&\quad + \frac{\exp\left\{-2e^{J_i(t)}\Delta + 2J_i(t)\right\} \Delta^2}{1 - \exp\left\{-e^{J_i(t)\Delta}\right\}} \\
&\quad + \exp\left\{-e^{J_i(t)\Delta} + 2J_i(t)\right\} \Delta^2
\end{aligned}$$

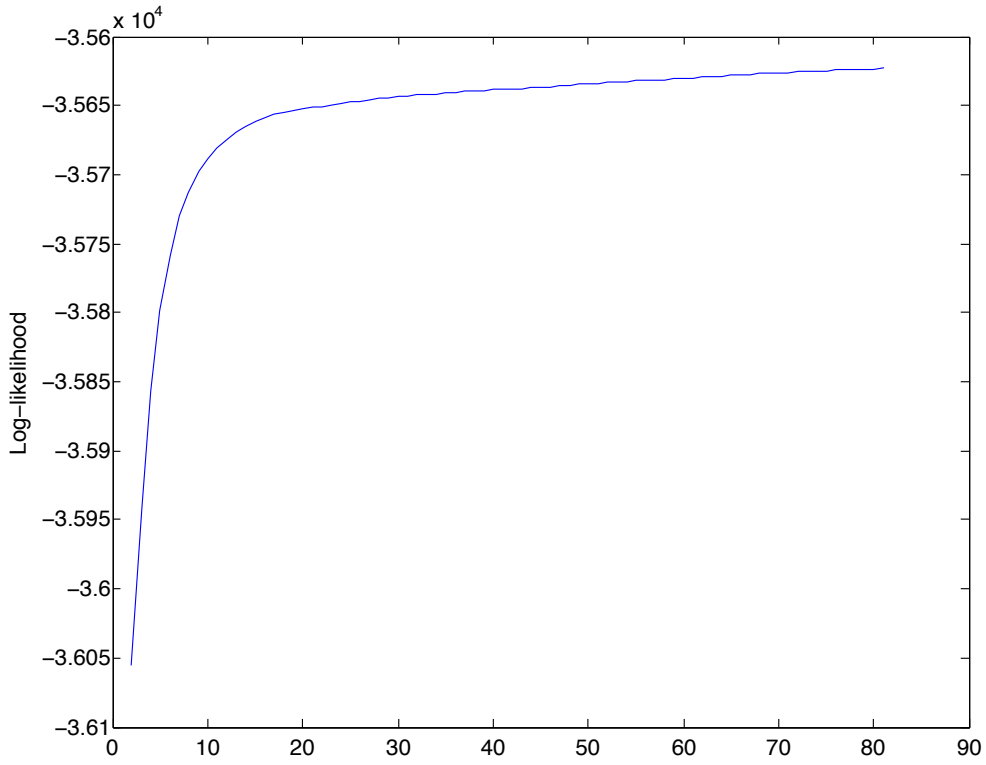


Figure 3: The log likelihood increases and converges.

We omit the $\frac{\partial^2 J}{\partial \theta_1^2}$, since J is linear in each variable, so the second derivatives are zero. To compute both the gradient and Hessian, parameter, we simply multiply the general forms above by each one of:

$$\frac{\partial J_i(t)}{\partial b_i} = 1 \tag{12}$$

$$\frac{\partial J_i(t)}{\partial w_{ij}} = h_{ij}(t) \tag{13}$$

$$\frac{\partial J_i(t)}{\partial \beta_{ijs}} = n_i(t-s) \tag{14}$$

5.3 Diagnostic Plots

See Figures 3 and 4.

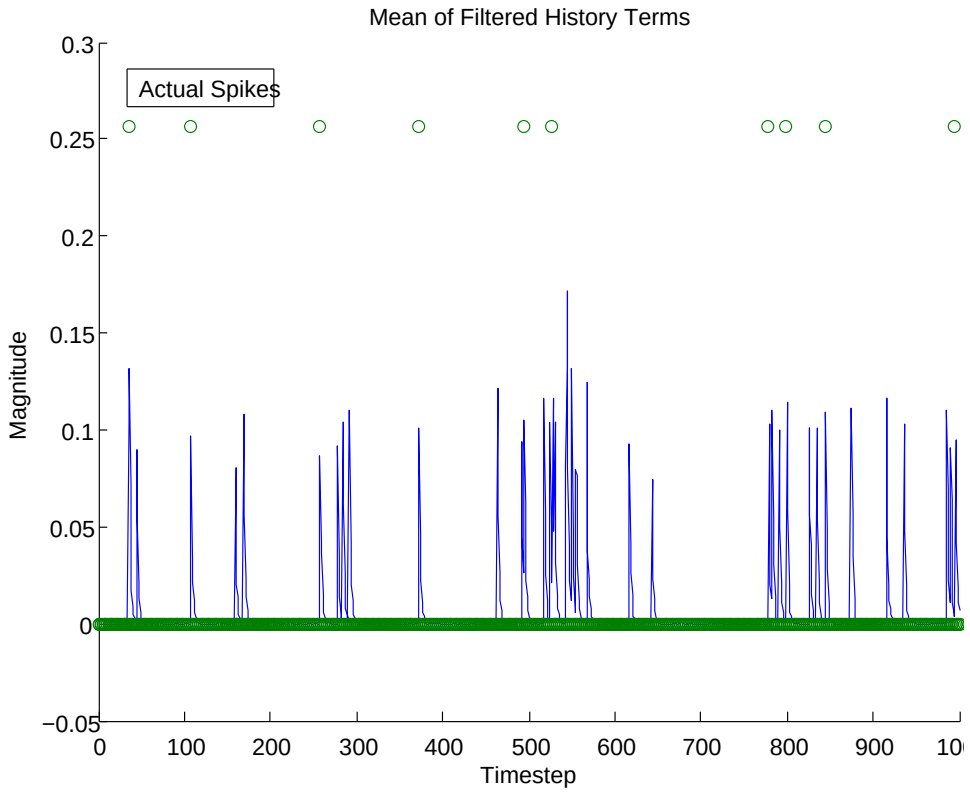


Figure 4: Output of one iteration of the SMC sampler. Green dot indicates a neural spike; blue lines indicate the mean of the filtered history terms on a single neuron. Since the model is probabilistic, not all large history terms induce a spike, and not all spikes force a large history term.