Neuronal Connectivity via Bayesian Tensor Factorization

Chris Mulligan

December 18, 2015

1 Introduction

In modern statistical neuroscience there is significant interest in discovering and modeling connections between individual neurons. The connections between neurons are useful for understanding how neurons process information and compute.

Recent advances in calcium imaging and multielectrode arrays give us access to the simultaneous spike train data for hundreds, or potentially even thousands, of neurons. Although with a few exceptions, such as C. elegens, this represents only a small fraction of the neurons in an organism, the techniques now employed are still quite valuable.

Parallel to this development in neuroscience, machine learning has seen a surge in development of, and interest in, matrix factorization techniques. While Non-negative matrix factorization (NMF) has been in use for many years, with the recent availability of very large dyadic datasets, many from internet companies like Netflix, there's been strong appreciation for NMF. More recently there's been interest in extending NMF from the 2 way regime of matrices to the M way regime of tensors. Along with these decomposition techniques, advances in inference algorithms, particularly Variational Inference and efficient MCMC algorithms, have allowed machine learning users to fit larger and more complex Bayesian models.

In this work we apply the techniques of probabilistic tensor factorization to the problem of neuron connectivity. While currently this technique does not provide strictly more accurate answers than other methods, it provides an interesting new avenue for exploratory data analysis and visualization.

There has been a significant amount of interest and prior work in neuron connectivity. Classically Cross-Correlation and Granger Causality based techniques were employed [1, 2] Generalized Linear Models [3]. More recently Generalized Linear Models, particularly L_1-regularized GLMs, have been applied with quantitative success, although the models don't lend themselves to interpretability as well as other techniques [3, 4]. A promising, though also problematic, literature is developing around network Hawkes processes, which can leverage some of the work in GLMs but offers advantages in inference and interpretability [5].

Perhaps the most similar work is recent applications of clustered factor analysis techniques to spike data [6], although there's other very current work applying factorization techniques to neural data [7, 8].

2 Bayesian Tensor Factorization

2.1 Model Definitions

We consider a standard setup with multiple simultaneously recorded spike trains. This consists of dataset of y_{it}, the observed spike count for neuron $i = 1, \ldots, N$ during time step $t = 1, \ldots, T$.

We will consider looking at neuron firings over a set of lags, some small number time steps over which we will infer the neuron interactions. Let $L \ll T$ be the maximum lag we consider.

Let $\underline{\mathbf{Y}}$ be a three way tensor of size $N \times N \times L$. Each element $y_{ijl} \in \mathbb{Z}_{\geq 0}$ is the count of times that neuron i fired l time steps after neuron j. This creates a triadic dataset, which describes the pairwise associations between neurons over a sliding window of up to L times, centered at each firing of the neuron.

As we see in the results in section 3.2, the formulation discussed here is not without its problems. In section 4 we propose an alternative model where $y_{ijl} \in [0, 1]$ is the conditional probability of neuron i firing l time steps after neuron j. While conceptually similar, it introduces additional mathematical and inferential difficultly.

2.2 Tensor Factorization

We use the Canonical Polyadic (CP) decomposition, because compared to the common alternative Tucker decomposition it is more interpretable and performs better with sparse counts [9, 10]. The K component decomposition decomposes the 3 way tensor into 3 latent factor matrices $\Theta^{(1)}, \Theta^{(3)}, \Theta^{(2)}$, with dimensions $d_m \times K$.

$$y_{ijl} \approx \hat{y}_{ijl} = \sum_{k=1}^{K} \theta^1_{ik} \theta^2_{jk} \theta^3_{lk}$$

Frequently $\hat{y}_{ijl}$ is called the "reconstruction" y_{ijl}, and represents the estimate based on the decomposition.

The columns of each of the Θ matrices represent a latent *component*, which hopefully will indicate a meaningful relationship between neurons. The $\Theta^{(1)}_k$ vector represents the neurons being "predicted", while the $\Theta^{(3)}_k$ vector represents the neurons being depended on, and $\Theta^{(2)}_k$ represents the lags of the dependency.

2.3 Probabilistic Model

We can interpret the deterministic decomposition in a probabilistic lens, where the reconstruction is the expected value of a distribution for the actual observed

data. In this case we consider y a Poisson random variable, although other distributions are feasible, such as the Bernoulli conjectured in section 4. Thus:

$$y_{ijl} \sim Pois(\hat{y}_{ijl})$$

With this framework we can use maximum likelihood estimation using the poisson likelihood for the Θ matrices, which yields superior results to using squared error, and implicitly assuming a Gaussian likelihood. However, we can do even better than MLE with Poisson likelihood by using a Bayesian inference, following the work of Schein et al which generalizes Poisson Matrix Factorization to Tensors [11, 9, 12].

The conjugate prior for the Poisson distribution is the Gamma, and it has certain other nice properties for our purposes, therefore we impose the prior $\theta_{ik}^{(m)} \sim \text{Gamma}(\alpha, \alpha\beta^{(m)})$. Under this parameterization we take a very small α and small(ish) β, as seen in figure 1, which concentrates the probability mass near 0, but has a heavy tail. This prior naturally encourages the desired sparsity in the latent factors. The α value is set manually, and in the style of empirical Bayes β is estimated as part of the inference, since $1/\beta^{(m)}$ is the mean of $\Theta^{(m)}$.

2.4 Approximate Inference

The posterior distribution is:

$$P\left(\Theta^{(1:3)} \mid \underline{\mathbf{Y}}, \alpha, \beta^{(1:3)}\right)$$

As with most models, this posterior is analytically infeasible, and therefore must be approximated. We use standard variational inference techniques using a mean field approximation, based on the Python code released by Schein et

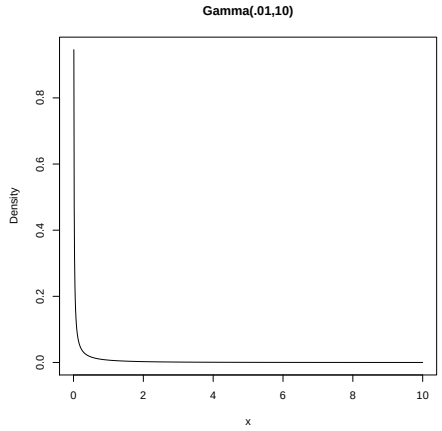

Figure 1: Gamma(.01, 10) Density

al [9]. In variational inference we posit a family of distributions Q, with parameters $S = \{\gamma_{d_m \cdot k \cdot m}, \delta_{d_m \cdot k \cdot m}\}$. Then solve the optimization problem $S^* = \text{argmin}_S \text{KL}(P||Q)$, to choose the member of Q as similar as possible to the posterior.

Here because the Q distribution factorizes completely the optimization can be done via coordinate ascent, using convergence of the evidence lower bound (ELBO) as the stopping criteria, where the ELBO is:

$$\mathcal{B}(S) = \mathbb{E}_Q \left[\log P(\underline{\mathbf{Y}}, \Theta^{(1:3)} | \alpha, \beta^{(1:3)}) \right] + H(Q)$$

We use independent Gamma variables for each latent factor, eg:

$$P\left(\theta_{ik}^{(1)} | \cdot\right) \approx Q\left(\theta_{ik}^{(1)} | S_{ik}^{(1)}\right) = \text{Gamma}\left(\theta_{ik}^{(1)} | \gamma_{ik}^{(1)}, \delta_{ik}^{(1)}\right)$$

As derived in Bayesian Poission Matrix Factorization [11] and Tensor Factorization [9], the update equations are, eg:

$$\gamma_{ik}^{(1)} = \alpha + \sum_{j,l} y_{ijl} \frac{\mathbb{G}_Q \left[\theta_{ik}^{(1)} \theta_{jk}^{(2)} \theta_{lk}^{(3)}\right]}{\sum_{k=1}^K \mathbb{G}_Q \left[\theta_{ik}^{(1)} \theta_{jk}^{(2)} \theta_{lk}^{(3)}\right]} = \alpha + \sum_{j,l} y_{ijl} \frac{\mathbb{G}_Q \left[\theta_{ik}^{(1)}\right] \mathbb{G}_Q \left[\theta_{jk}^{(2)}\right] \mathbb{G}_Q \left[\theta_{lk}^{(3)}\right]}{\sum_{k=1}^K \mathbb{G}_Q \left[\theta_{ik}^{(1)}\right] \mathbb{G}_Q \left[\theta_{jk}^{(2)}\right] \mathbb{G}_Q \left[\theta_{lk}^{(3)}\right]}$$

$$\delta_{ik}^{(1)} = \alpha\beta^{(1)} + \sum_{j,l} \mathbb{E}_Q \left[\theta_{ik}^{(1)} \theta_{jk}^{(2)} \theta_{lk}^{(3)}\right] = \alpha\beta^{(1)} + \sum_{j,l} \mathbb{E}_Q \left[\theta_{ik}^{(1)}\right] \mathbb{E}_Q \left[\theta_{jk}^{(2)}\right] \mathbb{E}_Q \left[\theta_{lk}^{(3)}\right]$$

This lends itself to numerous computational tricks for efficient inference, including memoizing the individual expectations and the denominator of γ, and not calculating any of the expectations for γ in the case that $y_{ijl} = 0$.

3 Results

3.1 Simple Synthetic Result

A small synthetic dataset consisting of 25 integrate and fire simulated neurons over 100,000ms was constructed with EnaS, simulating Multi-Electrode Array data [13]. Implanted within these neurons were approximately 5 very strong connections with varying lag times, and standard connections otherwise. From this raw data we construct our tensor $\underline{\mathbf{Y}}$ with $L = 20$ maximum lag. This tiny dataset yields a tensor with $25 \times 25 \times 20 = 12,500$ values.

We employ the tensor factorization procedure described here with $K = 6$ components and $\alpha = .01$. The components are visualized in figure 2. Each sub-figure consists of a component, and within the components the top left plot displays the weights on the triggered neurons, while the top right displays the weights on the triggering neurons, and the lower section visualizes the lags over which that interaction occurs.

Notice that in component 3 (labeled "G2", middle row, left side) neuron 5 depends strongly on neuron 21, with a lag of 2. In component 2 (labeled "G1", top right), there's a more complicated relationship where neurons 9, 11, 19, and 24 depend primarily on neuron 3, but also neuron 20, with a lag of 3.

There are several sensible ways of deriving a directed graph $\hat{W}$ from the factorization. Most intuitively, we can sum out the time dimension, and get $\hat{w}_{ij} = \sum_{l=1}^{L} \hat{y}_{ijl}$. Alternatively, we can examine the product of the first two latent dimensions, $\Theta^{(1)}\Theta^{(2)\mathrm{T}}$, visualized in figure 3.

From this data we can see that, given a very small network with clean and strong interactions this technique recovers those interactions. The latent components derived lend themselves well to visualization and interpretation. This suggests that this method is promising. However, this dataset is very small and very clean, so we need to proceed to more realistic settings.

3.2 Connectomics Challenges Results

We consider the "realistic" synthetic data provided by the Connectomics Challenge [14]. They simulate spiking neurons using the NEST simulator[15], with realistic dynamics and connectivity, then model calcium fluorescence and provide fluorescence data. The specific dataset of theirs I consider has $N = 100$ neurons, with $T \approx 180,000$ in a highly clustered network.

I begin by deconvolving the fluorescence to a spike train using the Vogelstein et al , as implemented in the PyFNND package [16, 17]

The dataset was constructed with $L = 150$ maximum lag, and fit with $K = 50$ components.

Unfortunate Graph Results

4 Future Work

References

[1] D. H. Perkel, G. L. Gerstein, and G. P. Moore, "Neuronal spike trains and stochastic point processes: Ii. simultaneous spike trains," *Biophysical journal*, vol. 7, no. 4, pp. 419–440, 1967.

[2] M. Kamiński, M. Ding, W. A. Truccolo, and S. L. Bressler, "Evaluating causal relations in neural systems: Granger causality, directed transfer function and statistical assessment of significance," *Biological cybernetics*, vol. 85, no. 2, pp. 145–157, 2001.

[3] L. Paninski, "Maximum likelihood estimation of cascade point-process neural encoding models," *Network: Computation in Neural Systems*, vol. 15, no. 4, pp. 243–262, 2004.

[4] M. Zhao, A. Batista, J. P. Cunningham, C. Chestek, Z. Rivera-Alvidrez, R. Kalmar, S. Ryu, K. Shenoy, and S. Iyengar, "An l 1-regularized logistic model for detecting short-term neuronal interactions," *Journal of computational neuroscience*, vol. 32, no. 3, pp. 479–497, 2012.

[5] S. W. Linderman and R. P. Adams, "Scalable bayesian inference for excitatory point process networks," *arXiv preprint arXiv:1507.03228*, 2015.

[6] L. Buesing, T. A. Machado, J. P. Cunningham, and L. Paninski, "Clustered factor analysis of multineuronal spike data," in *Advances in Neural Information Processing Systems*, pp. 3500–3508, 2014.

[7] M. Rudolph and D. Blei, "The dirichlet-gamma filter for discovery of neural ensembles and their temporal dynamics," in *Poster at: Statistical Methods for Understanding Neural Systems*, 2015.

[8] J. Friedrich, D. Soudry, Y. Mu, J. Freeman, M. Ahrens, and L. Paninski, "Fast constrained non-negative matrix factorization for whole-brain calcium imaging data," in *Poster at: Statistical Methods for Understanding Neural Systems*, 2015.

[9] A. Schein, J. Paisley, D. M. Blei, and H. Wallach, "Bayesian poisson tensor factorization for inferring multilateral relations from sparse dyadic event counts," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1045–1054, ACM, 2015.

[10] T. G. Kolda and J. Sun, "Scalable tensor decompositions for multi-aspect data mining," in *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pp. 363–372, IEEE, 2008.

[11] A. T. Cemgil, "Bayesian inference for nonnegative matrix factorisation models," *Computational Intelligence and Neuroscience*, vol. 2009, 2009.

[12] B. Ermiş, E. Acar, and A. T. Cemgil, "Link prediction in heterogeneous data via generalized coupled tensor factorization," *Data Mining and Knowledge Discovery*, vol. 29, no. 1, pp. 203–236, 2015.

[13] H. Nasser, S. Kraria, and B. Cessac, "Enas: a new software for neural population analysis in large scale spiking networks," *BMC Neuroscience*, vol. 14, no. Suppl 1, p. P57, 2013.

[14] O. Stetter, D. Battaglia, J. Soriano, and T. Geisel, "Model-free reconstruction of excitatory neuronal connectivity from calcium imaging signals," *PLoS Comput. Biol*, vol. 8, no. 8, p. e1002653, 2012.

[15] M.-O. Gewaltig and M. Diesmann, "Nest (neural simulation tool)," *Scholarpedia*, vol. 2, no. 4, p. 1430, 2007.

[16] J. T. Vogelstein, A. M. Packer, T. A. Machado, T. Sippy, B. Babadi, R. Yuste, and L. Paninski, "Fast nonnegative deconvolution for spike train inference from population calcium imaging," *Journal of neurophysiology*, vol. 104, no. 6, pp. 3691–3704, 2010.

[17] A. Muldal, "PyFNND: A python implementation of fast non-negative deconvolution."

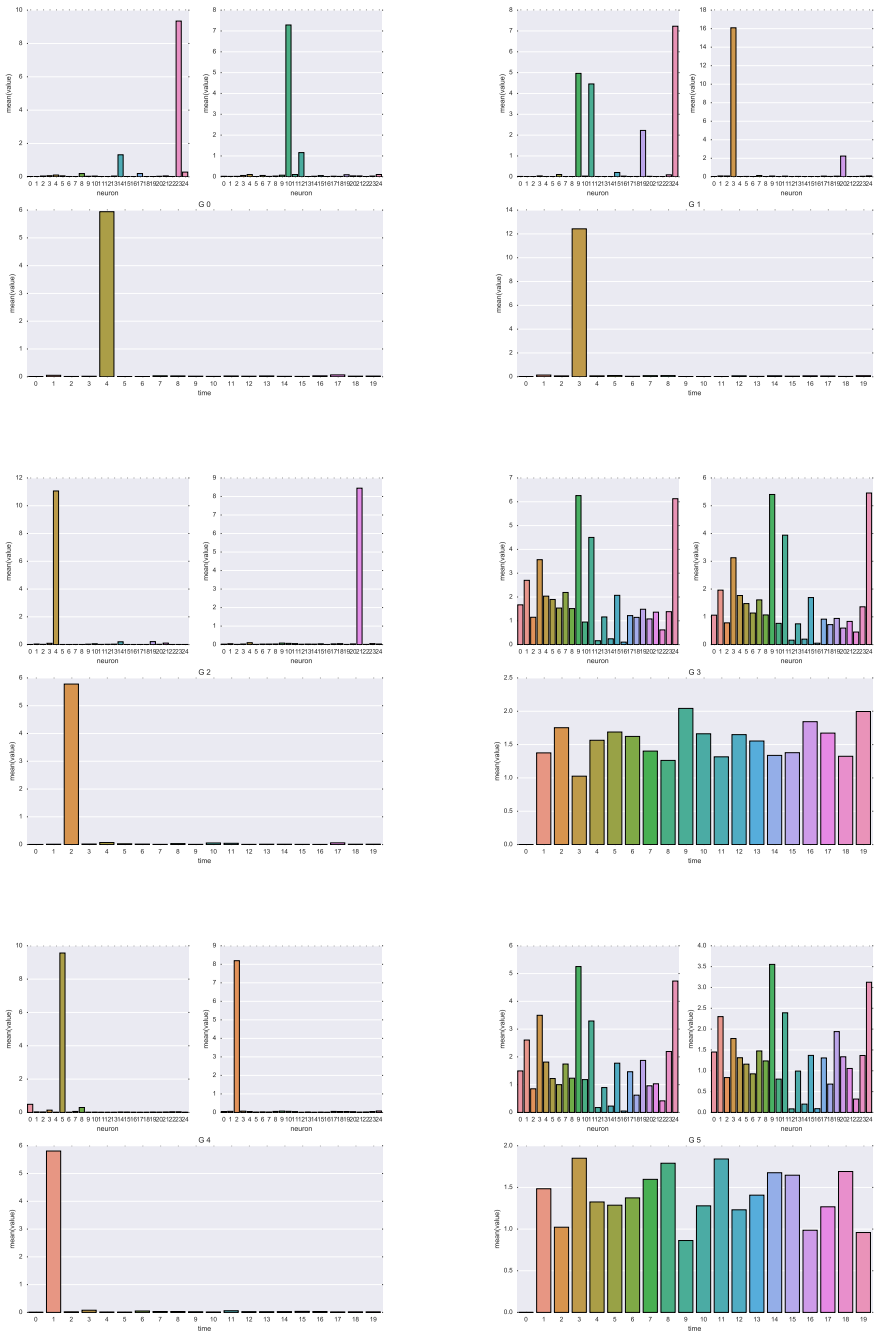

Figure 2: Our 6 fitted components for the Simple Synthetic Data

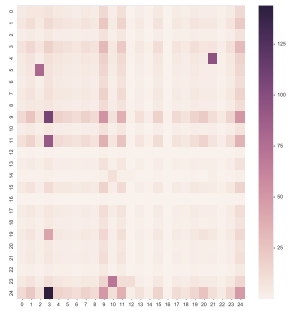

Figure 3: Simple Synthetic Data connectivity graph inference

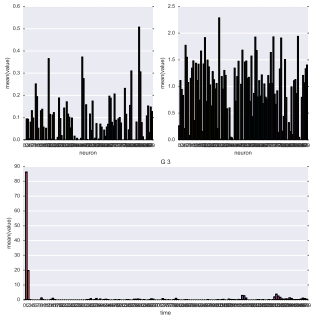

Figure 4: Component 3, one of several which captures neuron correlations in time 0

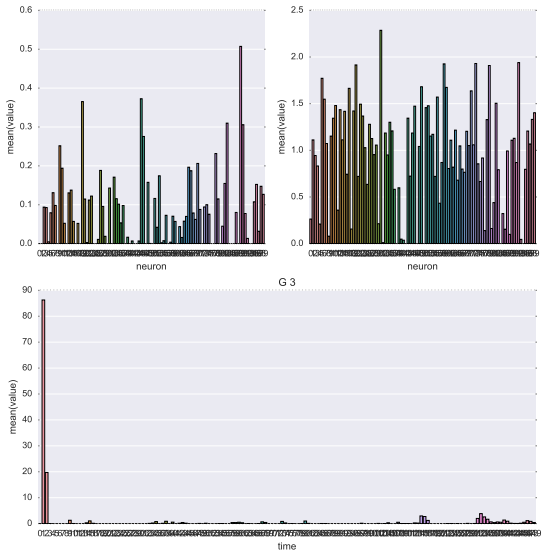

Figure 5: 3: one of several which captures neuron correlations in time 0

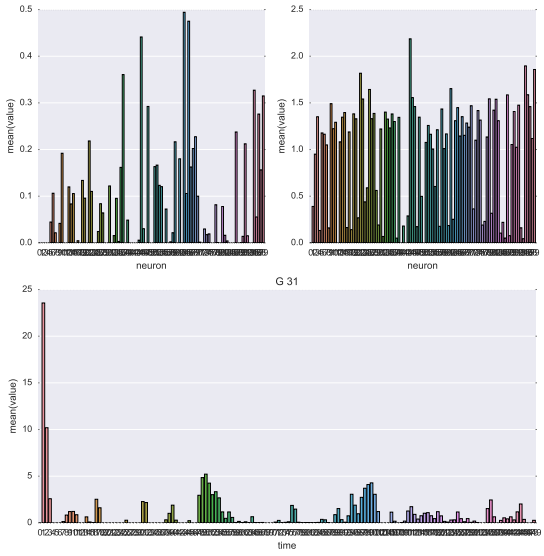

Figure 6: 31: Interesting temporal pattern, and sparse i vector

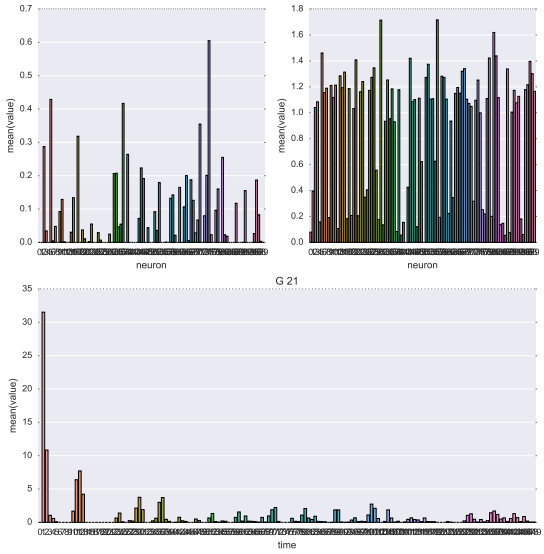

Figure 7: 21: Note sparsity of neurons, but relative flatness in time

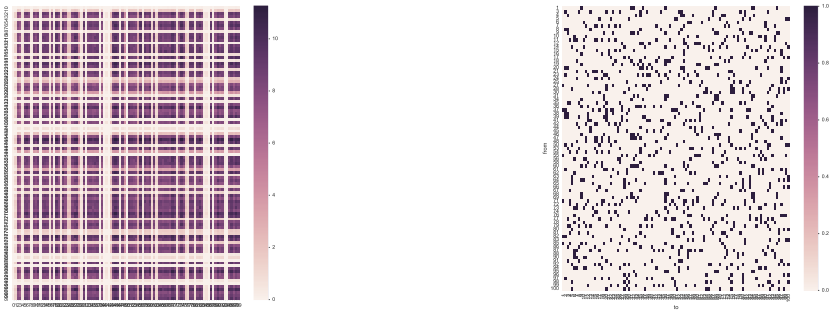

Figure 8: Left: Inferred with same method. Right: Actual