
Bayesian latent structure discovery from multi-neuron recordings: Supplementary Material

Scott W. Linderman
Columbia University
swl2133@columbia.edu

Ryan P. Adams
Harvard University and Twitter
rpa@seas.harvard.edu

Jonathan W. Pillow
Princeton University
pillow@princeton.edu

1 Model comparison via Predictive Log Likelihood

How can we compare different network models in a principled manner? Predictive log likelihood of held-out time bins is insufficient since it only depends directly on \mathbf{A} and \mathbf{W} . The network prior does aid in the estimation of \mathbf{A} and \mathbf{W} , but this is only an indirect effect, and it may be small relative to the effect of the data. Instead, we hold out neurons rather than time bins. Accurate network models play a crucial role in predicting held-out neurons' activity, since the distribution of incident connections to the held-out neuron are informed solely by the network model.

Formally, we estimate the probability of a held-out neuron's spike train $\mathbf{s}_{n^*} = [s_{1,n^*}, \dots, s_{T,n^*}]$, given the observed spike trains. We integrate over the latent variables and parameters underlying the observed spike train, as well as those underlying the new spike train, using Monte Carlo. Let $\mathbf{Z} = \{\{\mathbf{w}_n, \mathbf{a}_n, \nu_n, \mathbf{u}_n, \mathbf{v}_n\}_{n=1}^N, \boldsymbol{\theta}\}$, and let $\mathbf{z}_{n^*} = \{\nu_{n^*}, \mathbf{w}_{n^*}, \mathbf{a}_{n^*}, \mathbf{u}_{n^*}, \mathbf{v}_{n^*}\}$. Then,

$$p(\mathbf{s}_{n^*} | \mathbf{S}) \approx \int p(\mathbf{s}_{n^*} | \mathbf{z}_{n^*}, \mathbf{S}) p(\mathbf{z}_{n^*} | \mathbf{Z}) p(\mathbf{Z} | \mathbf{S}) d\mathbf{z}_{n^*} d\mathbf{Z} \approx \frac{1}{Q} \sum_{q=1}^Q p(\mathbf{s}_{n^*} | \mathbf{z}_{n^*}^{(q)}, \mathbf{S}),$$
$$\mathbf{z}_{n^*}^{(q)} \sim p(\mathbf{z}_{n^*} | \mathbf{Z}^{(q)}), \quad \mathbf{Z}^{(q)} \sim p(\mathbf{Z} | \mathbf{S}).$$

The samples $\{\mathbf{Z}^{(q)}\}_{q=1}^Q$ are posterior samples generated our MCMC algorithm given \mathbf{S} . While a proper Bayesian approach would impute \mathbf{s}_{n^*} , for large N this approximation suffices. For each sample, we draw a set of latent variables and connections for neuron n^* given the parameters $\mathbf{Z}^{(q)}$. These, combined with the spike train, enable us to compute the likelihood of \mathbf{s}_{n^*} .

2 Further MCMC details

Gibbs sampling the parameters of the network model

- *Independent Model* Under an independent model, the neurons do not have latent variables so all we have to sample are the global parameters, $\boldsymbol{\theta}$. If the independent model applies to the adjacency matrix, then $\boldsymbol{\theta} = \bar{\rho}$. The model is conjugate with a beta prior. If the independent model applies to the weights, then $\boldsymbol{\theta} = \{\bar{\mu}, \bar{\Sigma}\}$, and the model is conjugate with a normal inverse-Wishart prior.
- *Stochastic Block Model (SBM) updates:* If a stochastic block model is used for either the adjacency matrix or the weights, then it is necessary to sample the class assignments from their conditional distribution. We iterate over each neuron and update its assignment given the rest by sampling from the conditional distribution. For example, if \mathbf{u}_n governs a stochastic block model for the adjacency matrix, the conditional distribution of the label for neuron n is given by,

$$p(\mathbf{u}_n = c | \{\mathbf{u}_{m \neq n}\}, \mathbf{A}, \boldsymbol{\theta}) \propto \pi_c \prod_{m=1}^N p(a_{m \rightarrow n} | \rho_{c_m \rightarrow}) p(a_{n \rightarrow m} | \rho_{c \rightarrow c_m}),$$

where $\theta = \{\pi, \{\rho_{c \leftarrow c'}\}\}$. For stochastic block models of the weight matrix, W , the conditional distribution depends on $w_{n \rightarrow m}$ and $w_{m \rightarrow n}$ instead.

Given the class assignments and the network, the parameters $\rho_{c \leftarrow c'}$, $\mu_{c \leftarrow c'}$, $\Sigma_{c \leftarrow c'}$, and π are easily updating according to their conditional distributions, assuming π and $\rho_{c \rightarrow c'}$ are given conjugate Dirichlet and beta priors, respectively.

- **Latent location updates:** We resample the locations using hybrid Monte Carlo (HMC) [1]. Since the latent variables are continuous and unconstrained, this method is quite effective. In addition to the locations, the latent distance model is parameterized by a location scale, η . Given the locations and an inverse gamma prior, the inverse gamma conditional distribution can be computed in closed form.

The remaining parameters include the log-odds, γ_0 , if the distance model applies to the adjacency matrix. This can be sampled alongside the locations with HMC. For a latent distance model of weights, the baseline mean and variance, (μ_0, σ^2) , are conjugate with a normal inverse-gamma prior.

Observation parameter updates The observation parameter updates depend on the particular distribution. Bernoulli observations have no parameters. In the binomial model, ν_n corresponds to the maximum number of possible spikes — this can often be set a priori, but it must upper bound the maximum observed spike count. For negative binomial spike counts, the shape parameter ν_n can be resampled as in Zhou et al. [2]. One possible extension is to introduce a transition operator that switches between binomial and negative binomial observations in order to truly capture both over- and under-dispersion.

We implemented our code in Python using Cython and OMP to parallelize the Pólya-gamma updates. This is available at <https://github.com/slinderman/pyglm>.

Number of MCMC iterations For both the synthetic and retinal ganglion cell results presented in the main paper, we ran our MCMC algorithm for 1000 iterations and used the last 500 samples to approximate posterior expectations. These limits were set based on monitoring the convergence of the joint log probability. Even after convergence, the weighted adjacency matrix continues to vary from sample to sample, reflecting genuine posterior uncertainty. In some cases, the samples of the stochastic block model seem to get stuck in local modes that are difficult to escape. This is a challenge with coclustering models like these, and more sophisticated transition operators could be considered, such as collapsing over block parameters in order to update block assignments.

References

- [1] Radford M. Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, pages 113–162, 2010.
- [2] Mingyuan Zhou, Lingbo Li, Lawrence Carin, and David B Dunson. Lognormal and gamma mixed negative binomial regression. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1343–1350, 2012.