Clustering Neural Populations by Poisson Dynamic Factor Model

Ganchao Wei
University of Connecticut, Department of Statistics

UCONN

Introduction

- High-density silicon probes and large-scale calcium imaging methods allow neuroscientists to study neurons in the multi-population level.
- We are mostly interested in time-varying relationships within and between neural populations, which can usually be captured by low-dimensional latent state vectors. Both AR(1) and Gaussian process (GP) are widely used models for latent vectors.
- However, defining the populations is usually difficult. Moreover, single factor model provides a global linear model in a lower-dimensional subspace, which is not suitable for non-linear case.
- Here, we did clustering by Poisson dynamic factor model (PDFM), which (1) help to classify the neural population, and (2) describe global linear structure by local linear factor model.
- This method is useful for general multiple time series for counting data.

Models for Neural Population: PDFM

Poisson Dynamic Factor Model (PDFM)

Observation: a N-by-T matrix with counting data, i.e., $Y = (y_{it}) \in \mathbb{Z}_{\geq 0}^{N\times T}$ (N neurons, with counting observation up to T steps).

- To do clustering: separate population mean from the model.
- Reparametrize the model from linear dynamic system (LDS) model used in neuroscience.

Given the cluster indictor z_i for neuron i:

$$\boldsymbol{y_i} = (\boldsymbol{y}_{i1}, \dots, \boldsymbol{y}_{iT})' \sim Poi(\boldsymbol{\lambda}_i)$$
$$\log(\boldsymbol{\lambda}_i)\,|z_i = \boldsymbol{\mu}^{(z_i)} + \boldsymbol{X}^{(z_i)}\boldsymbol{c}_i$$

, where $\boldsymbol{\mu}^{(z_i)} = \left(\mu_1^{(z_i)}, \dots, \mu_T^{(z_i)}\right)' \in \mathbb{R}^T$, $\boldsymbol{X}^{(z_i)} = \left(\boldsymbol{x}_1^{(z_i)}, \dots, \boldsymbol{x}_T^{(z_i)}\right)' \in \mathbb{R}^{T\times p}$ and $\boldsymbol{c}_i \sim N_p(\boldsymbol{0}, \boldsymbol{I}_p)$. To ensure consistency, both $\mu_t^{(z_i)}$and $\boldsymbol{x}_t^{(z_i)}$ progression linearly with a Gaussian noise:

$$\mu_1^{(z_i)} \sim N(\mu_0, \Sigma_0) \qquad \mu_{t+1}^{(z_i)} \sim N\left(f^{(z_i)}\mu_t^{(z_i)} + g^{(z_i)}, \Sigma^{(z_i)}\right)$$
$$\boldsymbol{x}_1^{(z_i)} \sim N_p(\boldsymbol{x}_0, \boldsymbol{Q}_0) \qquad \boldsymbol{x}_{t+1}^{(z_i)} \sim N_p(\boldsymbol{A}^{(z_i)}\boldsymbol{x}_t^{(z_i)} + \boldsymbol{b}^{(z_i)}, \boldsymbol{Q}^{(z_i)})$$

Constraints for Identifiability

- Because of clustering, constraints on $\boldsymbol{c}_i$ is inappropriate.
- For simplicity, assume $\boldsymbol{A}^{(z_i)}$ and $\boldsymbol{Q}^{(z_i)}$ are diagonal.
- Let $\boldsymbol{X}'^{(z_i)}\boldsymbol{X}^{(z_i)}$ be diagonal with $p(p-1)/2$ constraints is enough, but the results are similar.

Population Mean

- Assume null model now.
- Can further include covariates, e.g., $\mu_t^{(z_i)} = \boldsymbol{d}_{it}'\boldsymbol{\beta}_t^{(z_i)}$, with $\boldsymbol{\beta}_t^{(z_i)} \sim N(\boldsymbol{\beta}_{t-1}^{(z_i)}, \boldsymbol{\Lambda}^{(z_i)})$ (Poisson-DGLM).

Approximate Marginal Likelihood

Let $\boldsymbol{\theta}^{(z_i)} = \left\{\boldsymbol{\mu}^{(z_i)}, \boldsymbol{X}^{(z_i)}, f^{(z_i)}, g^{(z_i)}, \Sigma^{(z_i)}, \boldsymbol{A}^{(z_i)}, \boldsymbol{b}^{(z_i)}, \boldsymbol{Q}^{(z_i)}\right\}$,

$$M_{\boldsymbol{\theta}^{(z_i)}}(\boldsymbol{y_i}) = P\left(\boldsymbol{y_i}\middle|\boldsymbol{\theta}^{(z_i)}\right) = \int P\left(\boldsymbol{y_i}\middle|\boldsymbol{\theta}^{(z_i)}, \boldsymbol{c_i}\right) P(\boldsymbol{c_i})\mathrm{d}(\boldsymbol{c_i})$$

For fast update (when doing clustering), do Gamma approximation.

- Assume conditional independency: $M_{\boldsymbol{\theta}^{(z_i)}}(\boldsymbol{y_i}) = \prod_{t=1}^T P\left(y_{it}\middle|\boldsymbol{\theta}_t^{(z_i)}\right)$
- Denote $\lambda_t^{(z_i)} = \exp(\mu_t^{(z_i)} + \boldsymbol{x}_t'^{(z_i)}\boldsymbol{c}_i)$, which is $\lambda_t^{(z_i)} \sim lognormal(\mu_t^{(z_i)}, \boldsymbol{x}_t'^{(z_i)}\boldsymbol{x}_t^{(z_i)})$
- Approximate lognormal by Gamma distribution, i.e. $\lambda_t^{(z_i)} \sim Gamma\left(\left(\boldsymbol{x}_t'^{(z_i)}\boldsymbol{x}_t^{(z_i)}\right)^{-1}, \boldsymbol{x}_t'^{(z_i)}\boldsymbol{x}_t^{(z_i)} \cdot e^{\mu_t^{(z_i)}}\right) = Gamma(a_{it}, b_{it})$.
- By conjugacy,

$$P\left(y_{it}\middle|\boldsymbol{\theta}_t^{(z_i)}\right) = \int P\left(y_{it}\middle|\lambda_t^{(z_i)}\right) P\left(\lambda_t^{(z_i)}\right)\mathrm{d}\left(\lambda_t^{(z_i)}\right) = NB(y_{it}|r_{it}, p_{it})$$

, where $r_{it} = a_{it}$ and $p_{it} = 1/(1+b_{it})$

- The approximation is good when a_{it} is not super small, which is nearly guaranteed when doing clustering.

In summary, $\boldsymbol{Y}_i \sim PDFM(\boldsymbol{\theta}_{z_i})$, with prior for $\boldsymbol{\theta}_{z_i}$ as $\boldsymbol{H}$

Models for Clustering: MFM

- In practice, it's impossible to know the **number of clusters**.
- Dirichlet process mixtures (DPM) model?
- **Wrong!** The number of neural populations is finite but unknown.
- Put Prior on number of cluster directly → **mixture of finite mixtures (MFM) model**.
- Can easily integrate the field knowledge about number of clusters into the model.

$K \sim p_k$	where p_k is a p.m.f. on $\{1,2,\dots\}$
$\boldsymbol{\pi} = (\pi_1, \dots \pi_k) \sim Dirichilet_k(\gamma, \dots, \gamma)$	given $K = k$
$Z_1, \dots, Z_N \overset{\text{i.i.d.}}{\sim} \boldsymbol{\pi}$	given $\boldsymbol{\pi}$
$\boldsymbol{\theta}_1, \dots \boldsymbol{\theta}_k \overset{\text{i.i.d.}}{\sim} \boldsymbol{H}$	given $K = k$
$\boldsymbol{Y}_i = (Y_{i1}, \dots, Y_{iT})' \sim SSFM(\boldsymbol{\theta}_{z_i})$ given $\boldsymbol{\theta}_{1:K}$ and $Z_{1:N}$	independently for $i = 1, \dots, N$,

In the following implementations, I simply put the geometric prior on $K \sim Geometric(r)$, with $r = 0.2$.

Inference

Sample posteriors by MCMC.

To sample SSFM-related parameters efficiently, instead of using particle MCMC, do normal approximation with Gibbs sampler.

Due to unimodality and Markovian structure, the posterior mode can be found efficiently in $O(T)$.

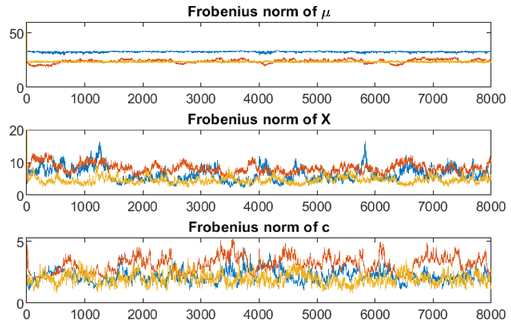
The MFM-related parameters are updated by the analog of partition-based algorithm in DPM.

Simulations

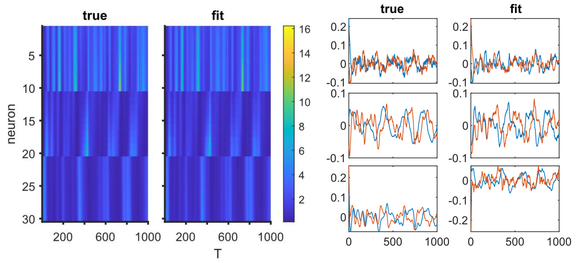
The data is simulated using the standard LDS model: $\log \boldsymbol{\lambda}_i = \delta_i \boldsymbol{1}_T + \boldsymbol{G}^{(z_i)}\boldsymbol{d}_i$, where $\delta_i \in \mathbb{R}$ and $\boldsymbol{d}_i \in \mathbb{R}^q$. By transformation, we can show that $p = q + 1$.

Simulation 1: Neurons with Known Labels

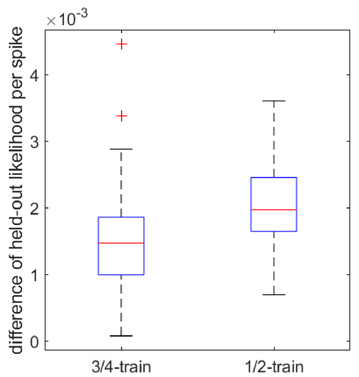
3 clusters, 10 neuron in each cluster. The dimension of latent vectors is $p = 1$ ($q = 2$) and recording length is $T = 1000$. Run MCMC for 8000 iterations. The trace of Frobenius/ Euclidean norm of $\mu^{(k)}, X^{(k)}$ and C for 3 clusters



The convergence achieved fast with somewhat strong autocorrelation. The averages of fitted mean firing rate and latent state (after transformation to LDS parametrization), from iteration 500 to 1000.



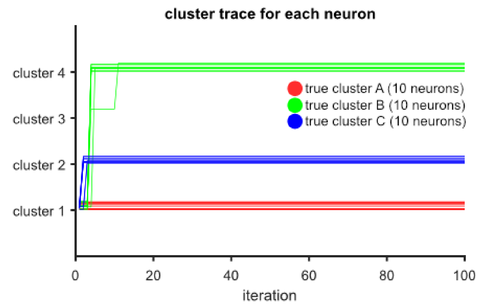
The following boxplot shows the difference of held-out likelihood per spike between (1) PDFM for 3 clusters with $p = 1$ for each and (2) PDFM for 1 cluster with $p = 5$, with 3/4 and 1/2 speckled training



- Doing PDFM cluster-wise (mixture of PDFM) > single PDFM
- The less the data, the more significant the benefit

Simulation 2: Neurons with Unknown Labels

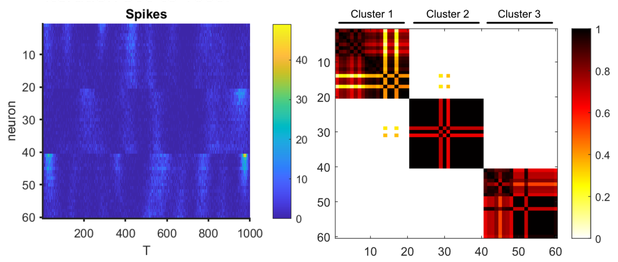
The same settings as simulation 1 but with unknown labels. The trace of clustering index for each neuron in 100 iterations.



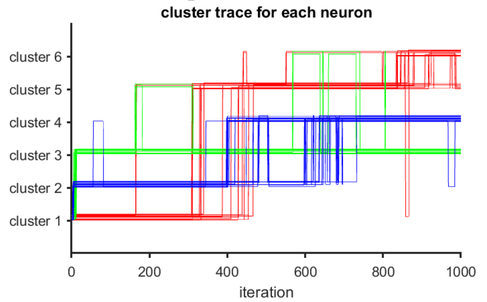
Simulation 3: Same waveform, different amplitude

- 20 neurons in each cluster
- Same waveform, different amplitude
- Most have weak signals

. The spiking counts and the posterior similarity matrix, from iteration 500 to 1000.



The trace of clustering index



Acknowledgements

I thank my advisors **Dr. Ian H. Stevenson** and **Dr. Xiaojing Wang** for detailed and constructive discussions, comments and suggestions.

References

Macke, J. H. et al. Empirical models of spiking in neural populations. *Adv. Neural Inf. Process. Syst.* **24**, (2011).
Miller, J. W. & Harrison, M. T. Mixture models with a prior on the number of components. *J. Am. Stat. Assoc.* **113**, 340 (2018).