Clustering Neural Populations by State-space Factor Models

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. Routinely, we do distance-based clustering at first. If not perfectly accurate → bias the latent structures.
- Use the state-space factor model (SSFM) to do clustering, which let the latent structure help with clustering and vice versa.
- This method can be used to cluster general multiple time series data, while extracting potential low-dimensional structures at the same time.

Models for Neural Population: SSFM

Neural spikes for multi-population are modeled by the **state-space factor model (SSFM)**.

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).

Given the cluster indictor z_i for neuron i, the generating model for each neuron spike is:

$$y_{it} \sim Poi(\lambda_{it})$$
$$\log(\lambda_{it})\,|z_i = d_i^{(z_i)} + \boldsymbol{c}_i^{(z_i)}\boldsymbol{x}_t^{(z_i)}$$
$$\left(d_i^{(z_i)}, \boldsymbol{c}_i^{\prime(z_i)}\right)^{\prime} \sim N_{p+1}\left(\boldsymbol{\mu}_{dc}^{(z_i)}, \boldsymbol{\Sigma}_{dc}^{(z_i)}\right)$$

, where $\boldsymbol{c}_i^{(z_i)} \in \mathbb{R}^p$ and $\boldsymbol{x}_t^{(z_i)} \in \mathbb{R}^p$. The latent vector $\boldsymbol{x}_t^{(z_i)}$ progresses linearly with a Gaussian noise:

$$\boldsymbol{x}_1^{(z_i)} \sim N_p(\boldsymbol{x}_0, \boldsymbol{Q}_0)$$
$$\boldsymbol{x}_{t+1}^{(z_i)}|\boldsymbol{x}_t^{(z_i)} \sim N_p(\boldsymbol{A}^{(z_i)}\boldsymbol{x}_t^{(z_i)} + \boldsymbol{b}^{(z_i)}, \boldsymbol{Q}^{(z_i)})$$

We can further model interactions between clusters by allowing non-zero elements in transition matrix across clusters.

Comment 1: Cluster-dependent $d_i^{(z_i)}$ and $\boldsymbol{c}_i^{(z_i)}$

- To make clustering possible, $d_i^{(z_i)}$ and $\boldsymbol{c}_i^{(z_i)}$ are both neuron- and cluster-dependent.
- In cluster k, the extended parameters $\boldsymbol{d}^{(k)} \in \mathbb{R}^N$ and $\boldsymbol{C}^{(k)} \in \mathbb{R}^{N\times p}$ will contain auxiliary parameters, i.e. $\left\{d_i^{(k)}, \boldsymbol{c}_i^{(k)} : z_i \neq k\right\}$ to help clustering.
- The prior $\boldsymbol{\mu}_{dc}^{(z_i)}$ and $\boldsymbol{\Sigma}_{dc}^{(z_i)}$ will help inference for these auxiliary parameters.

Comment 2: Constraints for Model Identifiability

- The model is over-parameterized, so that we need to put some constraints to ensure identifiability.
- In neuroscience, the fitted latent state receives special interests.
- Put constraints on $\boldsymbol{X}^{(k)} = \left(\boldsymbol{x}_1^{(k)}, \dots, \boldsymbol{x}_T^{(k)}\right) \in \mathbb{R}^{p\times T}$ directly, such that each row of $\boldsymbol{X}^{(k)}$ is centered around $\boldsymbol{0}$ and $\boldsymbol{X}^{(k)}\boldsymbol{X}^{\prime(k)} = \boldsymbol{I}_p$
- With further constraints for diagonal $\boldsymbol{A}^{(k)}$ and $\boldsymbol{Q}^{(k)}$, the model is identifiable.

Comment 3: Interpretations of Parameters

With the constraints above, the spiking feature of the neuron i is decomposed into three parts:

1) The baseline firing rate $d_i^{(z_i)}$
2) A set (p) of centered and orthonormal temporal patterns $\boldsymbol{X}^{(k)}$.
3) The "magnitude" of each temporal pattern $\boldsymbol{c}_i^{(z_i)}$.

All these 3 features will be used for clustering.

In summary, the cluster parameters of cluster k are $\boldsymbol{\Theta}_k = \left\{\boldsymbol{d}^{(k)}, \boldsymbol{C}^{(k)}, \boldsymbol{\mu}_{dc}^{(k)}, \boldsymbol{\Sigma}_{dc}^{(k)}, \boldsymbol{X}^{(k)}, \boldsymbol{A}^{(k)}, \boldsymbol{b}^{(k)}, \boldsymbol{Q}^{(k)}\right\}$, with prior $\boldsymbol{H}$. The generating process is denoted as $\boldsymbol{Y}_i = (Y_{i1}, \dots, Y_{iT})^{\prime} \sim SSFM(\boldsymbol{\Theta}_{z_i})$.

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 $\boldsymbol{\pi}$
$\Theta_1, \dots \Theta_k \overset{\text{i.i.d.}}{\sim} \boldsymbol{H}$	given $K = k$
$\boldsymbol{Y}_i = (Y_{i1}, \dots, Y_{iT})^{\prime} \sim SSFM(\boldsymbol{\Theta}_{z_i})$ 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

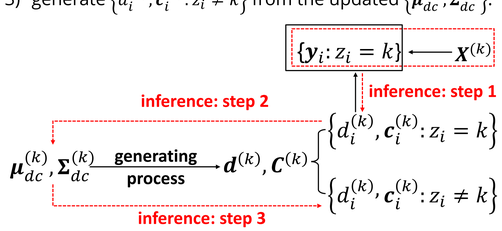
Sample posteriors by MCMC.

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

- Constrained $\boldsymbol{X}^{(k)}$: 1) draw unconstrained sample by the Laplace-approximation and then 2) project the sample to the constraint space by singular value decomposition (SVD).
- Due to unimodality and Markovian structure, the posterior mode can be found efficiently in $O(T)$.

Update auxiliary parameters in $\boldsymbol{d}^{(k)} \in \mathbb{R}^N$ and $\boldsymbol{C}^{(k)}$.

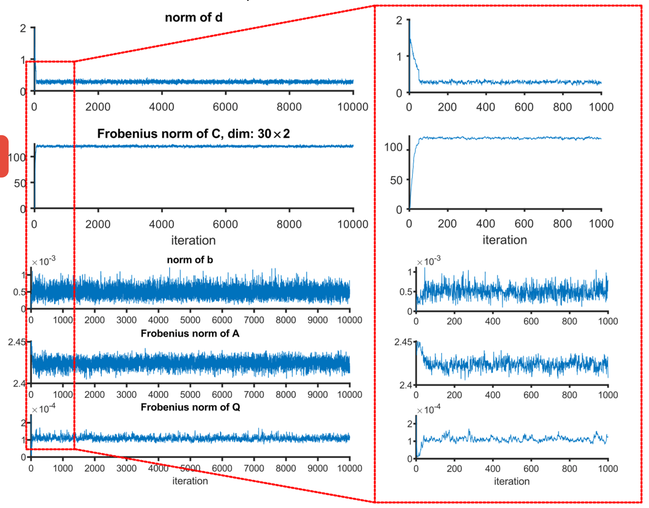
1) update $\left\{d_i^{(k)}, \boldsymbol{c}_i^{(k)} : z_i = k\right\}$ by NUTS.
2) update $\left\{\boldsymbol{\mu}_{dc}^{(k)}, \boldsymbol{\Sigma}_{dc}^{(k)}\right\}$ by Gibbs sampler based on $\left\{d_i^{(k)}, \boldsymbol{c}_i^{(k)} : z_i = k\right\}$.
3) generate $\left\{d_i^{(k)}, \boldsymbol{c}_i^{(k)} : z_i \neq k\right\}$ from the updated $\left\{\boldsymbol{\mu}_{dc}^{(k)}, \boldsymbol{\Sigma}_{dc}^{(k)}\right\}$.



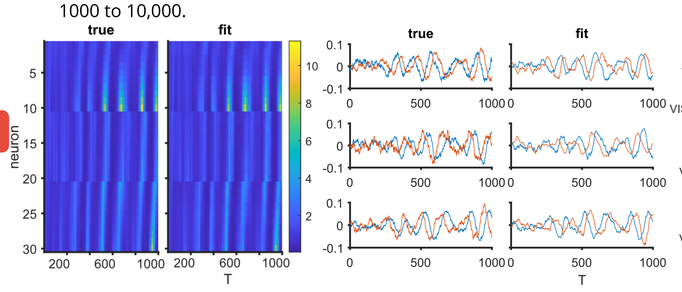
Simulations

Simulation 1: Neurons with Known Labels

3 clusters, 10 neuron in each cluster. The dimension of latent vectors is $p = 2$ and recording length is $T = 1000$. Run MCMC for 10,000 iterations. Some trace plots:

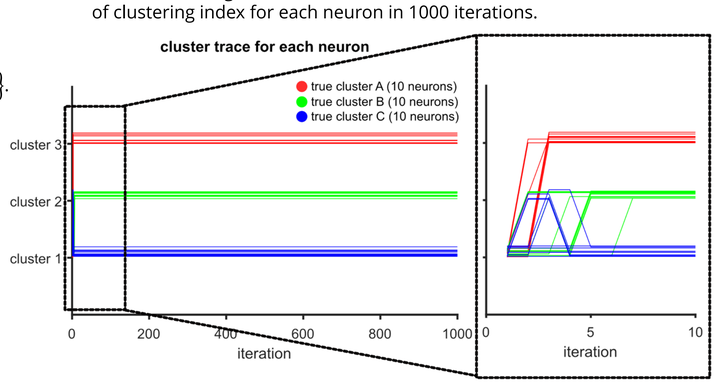


The averages of fitted mean firing rate and latent sate, from iteration 1000 to 10,000.



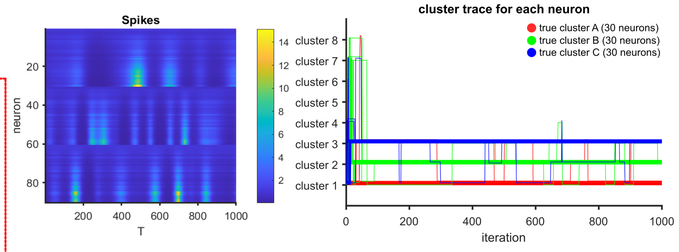
Simulation 2: Neurons with Unknown Labels

The same settings as simulation 1 but with unknown labels. The trace of clustering index for each neuron in 1000 iterations.



Simulation 3: A More Challenging Setting

30 neurons in each cluster. In each cluster, some neurons (left panel: tops within each cluster) have weak signals (hard to cluster).



Application

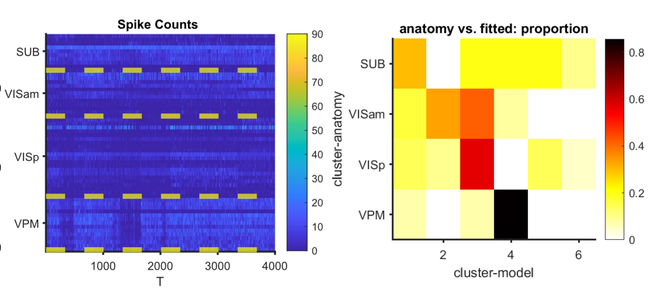
57 neurons from 4 anatomical sites:

- Subiculum (**SUB**): part of the hippocampus involved in spatial navigation/memory
- 2 visual areas (**VISp** and **VISam**)
- a part of thalamus (**VPM**): involved in sensation/movement

Hard to cluster:

- Activity in all these areas depends a bit on the movement of the animal.
- Each area has different types of neurons within it, e.g. excitatory vs. inhibitory (~20-30%).

Use ~30 min recordings for clustering (bin size = 0.5s). Set $p = 4$. The average results from iteration 1000 to 3000.



Acknowledgements

I thank my advisors **Dr. Ian H. Stevenson** and **Dr. Xiaojing Wang** for detailed and constructive discussions, comments and suggestions. Thank the Allen Institute for sharing the neuropixels data.

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).