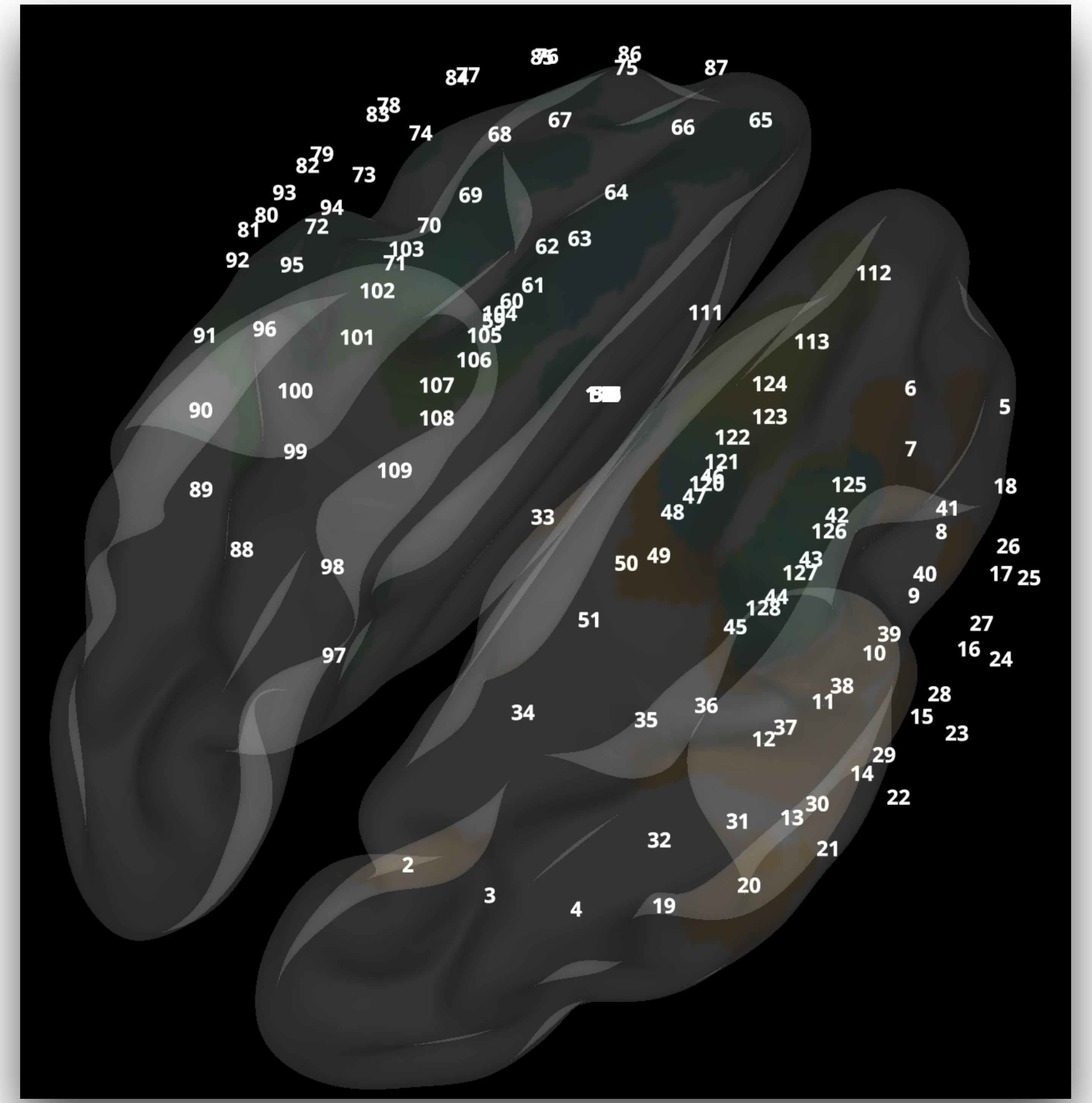
When and Where does our Working Memory Take Place?

A multi-level sub-graph analysis of brain functional connectivities.

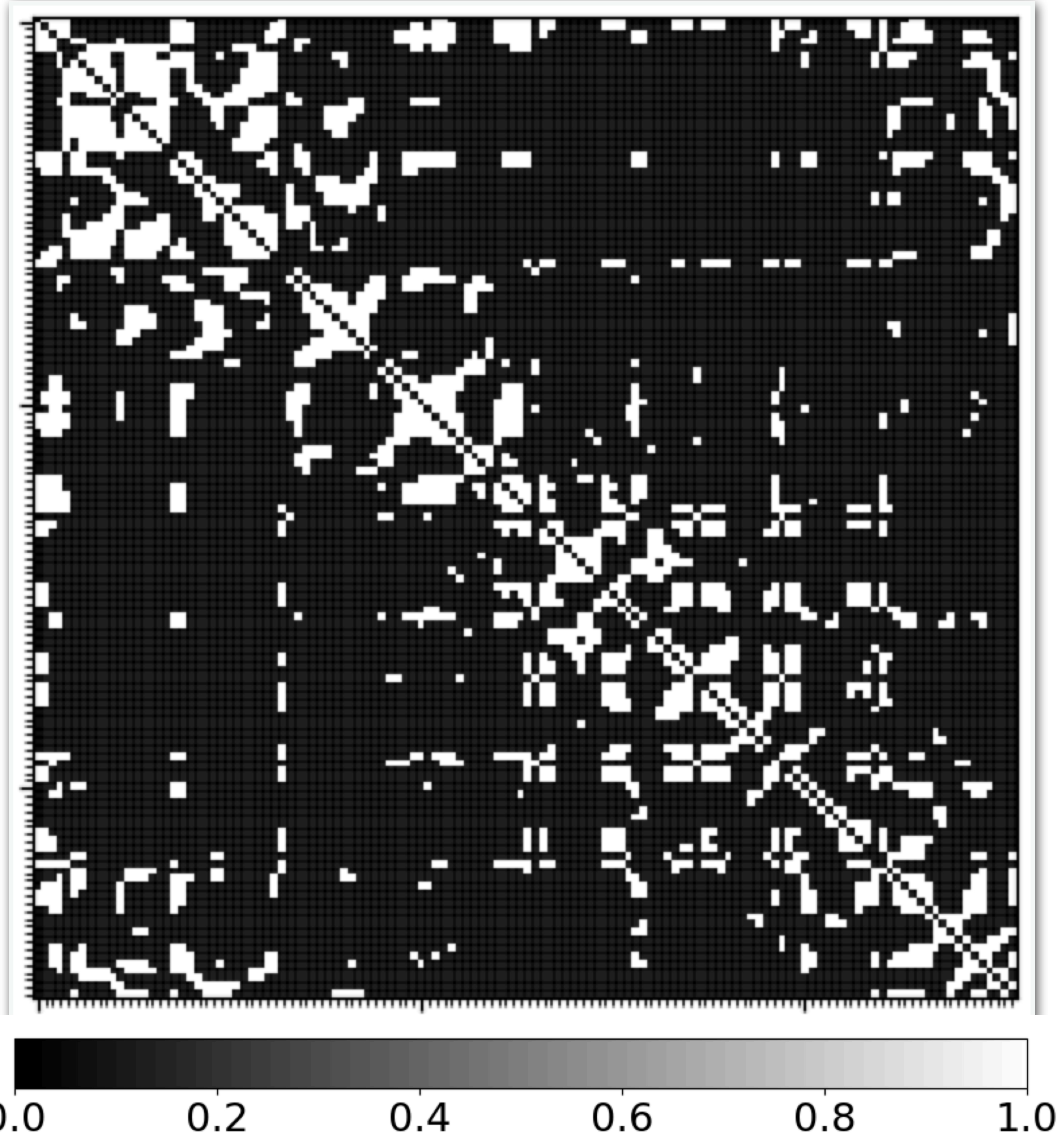
Maoran Xu, Leo Duan
Department of Statistics, University of Florida



EEG Experiment and Working Memory

Analyzing Brain Connectivity

Task: compare the connectocomes among work loads over time.



Treed-SVD Decomposition

The Tree-Spanning Stiefel Space

For a given graph $G=(V, E, A)$ with adjacency matrix A, we view it as a common signal-plus-noise graph

$$A = P\Lambda P^T + \mathscr{E},$$

where P is an orthonormal matrix and every column of P has a maximum of 2^{l-1} unique values. Λ is a diagonal matrix indicating the weight of each level.

We call such matrix space a Tree-Spanning Stiefel Space

$$\mathcal{V}_{\mathcal{T}}^{d\times n} = \mathcal{T}_d \cap \mathcal{V}^{d\times n}.$$

where

$$\mathcal{V}^{d\times n} = \{\boldsymbol{P} \in \mathbb{R}^{n\times d} : \boldsymbol{P}^\intercal \boldsymbol{P} = \boldsymbol{I}_d\}$$

and

$$\mathcal{T}_d = \{\boldsymbol{P} \in \mathbb{R}^{n\times d} : P_{i,l} = P_{i',l} \text{ if } i, i' \in \mathcal{N}_{l,s}\ \forall l, s\}.$$

The Tree-Spanning Stiefel Space reduces the dimensionality while keeping the independence between different scales.

The Computing Algorithm

- Estimating the structure:

1. Compute the eigenvector of corresponding to the largest eigenvalue, and partition by the sign as $\mathcal{T}_{k+1}$.
2. Maximize the conditional likelihood

$$L(P_{k+1}, \lambda_{k+1} | \tilde{\boldsymbol{A}}_{1:k}, \boldsymbol{A}) = \exp\{-\|(\boldsymbol{A} - \tilde{\boldsymbol{A}}_{1:k}) - \lambda_{k+1} P_{k+1} P_{k+1}^\intercal\|_F\}.$$

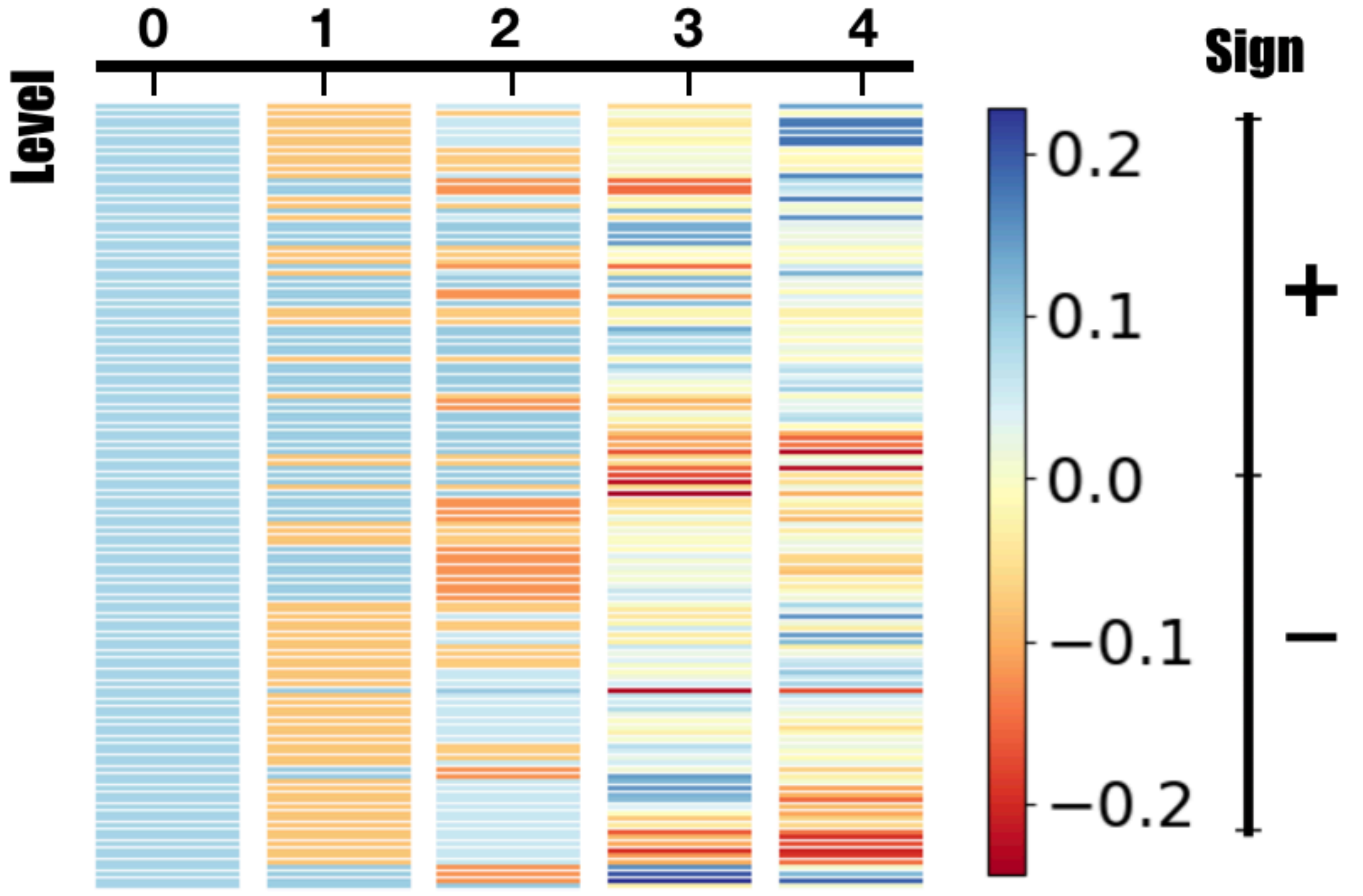
by solving the equivalent quadratic programming:

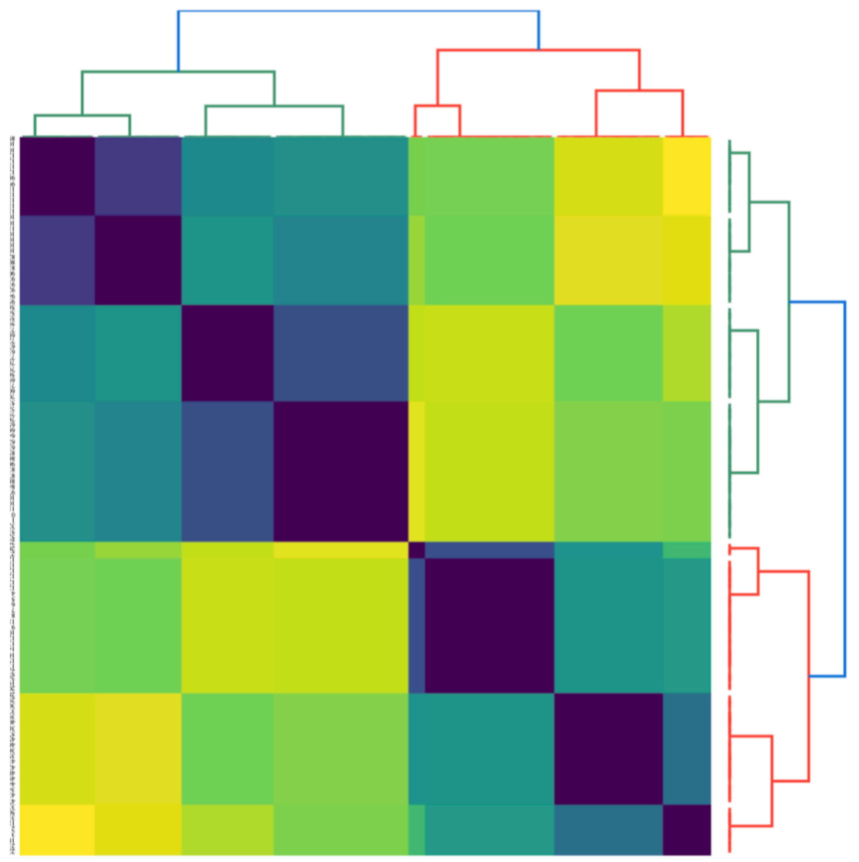
$$\max_{P_{k+1}} P_{k+1}^\intercal \left(\boldsymbol{A} - \tilde{\boldsymbol{A}}_{1:k}\right) P_{k+1}$$
$$\text{subject to } \boldsymbol{P} \in \mathcal{V}_{\mathcal{T}}^{(k+1)\times n},$$

and set

$$\lambda_{k+1} = P_{k+1}^T A P_{k+1}.$$

Repeat step 1 and 2 until the sub-tree is not separable.





Change Point Detection

We model the change point detection task as a combination of Hidden Markov Model and clustering problem. Suppose there are K hidden states, where the center for the state is $\mu^{(k)}$. A_t belongs to state z_t.

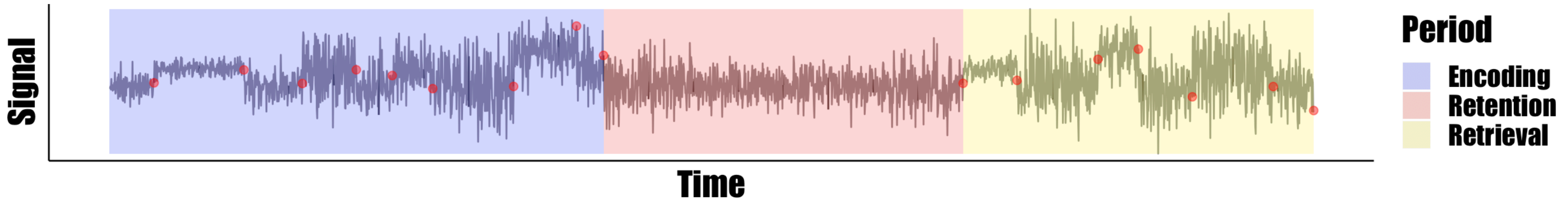
The Hidden Markov Clustering

1. Conditioned on $\{z_t\}_t$, update $\{\mu^{(k)}\}_k$ using Treed-SVD algorithm.
2. Conditioned on $\{\mu^{(k)}\}_k$, update the latent assignment one-at-a-time, from the categorical distribution with

$$pr(z_t = k \mid z_{t-1}, z_{t+1}) = \frac{p_{z_{t-1},k} p_{k,z_{t+1}} \exp\{-\|\boldsymbol{A}_t - \boldsymbol{\mu}^{(k)}\|\}}{\sum_{l=1}^{K} p_{z_{t-1},l} p_{l,z_{t+1}} \exp\{-\|\boldsymbol{A}_t - \boldsymbol{\mu}^{(l)}\|\}}$$

3. Conditioned on $\{z_t\}_t$, update the transition probabilities by

$$(p_{k,1}, \ldots, p_{k,K}) \sim \mathrm{Dir}\{\alpha + \sum_{t=1}^{T} \mathbf{1}(z_{t-1} = k, z_t = 1), \ldots, \alpha + \sum_{t=1}^{T} \mathbf{1}(z_{t-1} = k, z_t = K)\}$$



The End

Thank you for listening.

Contact me: maoranxu at ufl dot edu