A Dynamic Bayesian Model for Detecting Neuronal Communities

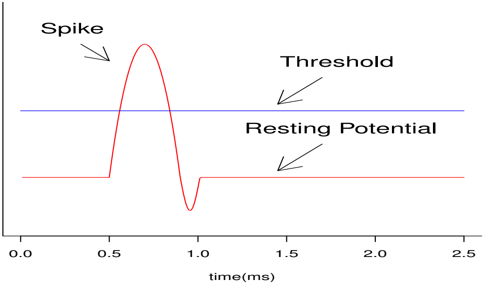
Babak Shahbaba[1]

Department of Statistics and Department of Computer Science, UCI

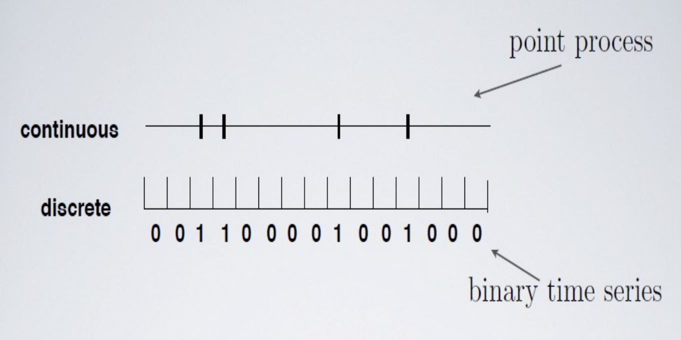
September, 2014

[1] Joint work with Sam Behseta, Bo Zhou, Hernando Ombao, David Moorman

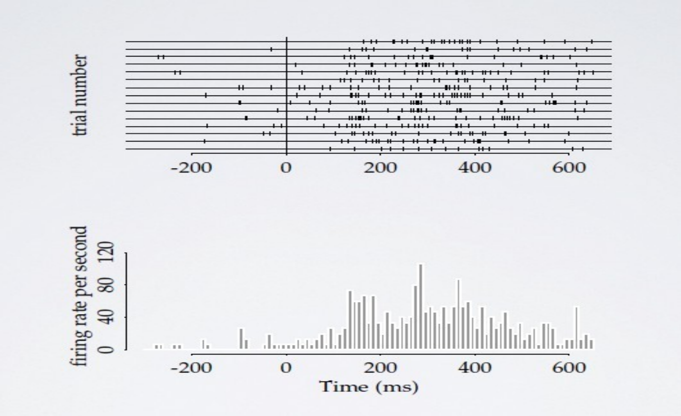
Spike Train



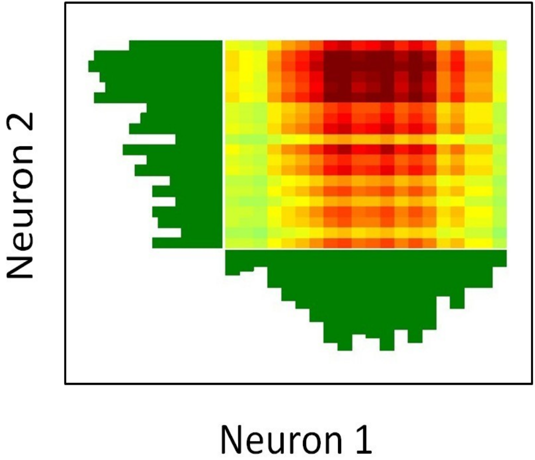
Spike Train


point process
continuous
discrete
0 0 1 1 0 0 0 0 1 0 0 1 0 0 0
binary time series

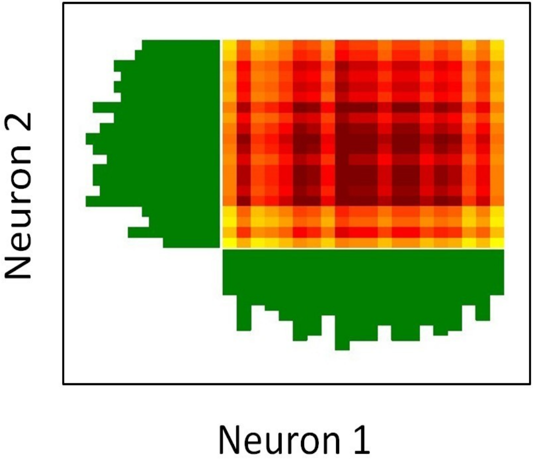
Spike Train and PSTH



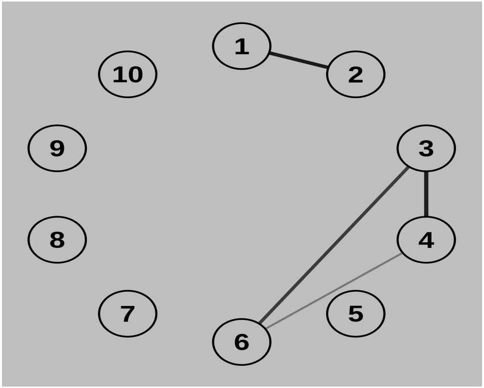
JPSTH– Independent Neurons



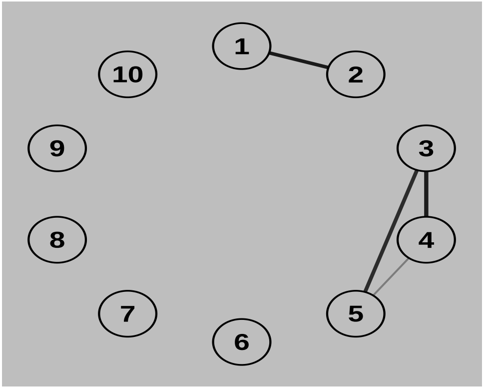
JPSTH– Synchronous Neurons



Neuronal Communities



Dynamic Neuronal Communities



Outline

- Modeling firing rates
- A stationary model for detecting synchrony
- A dynamic model for detecting synchrony
- A stationary model for detecting neuronal communities
- A dynamic model for detecting neuronal communities
- Simulation studies
- Experimental results

Modeling Firing Rates

- In our first attempt, we assumed that the firing rate for each neuron depends on an underlying latent variable, $u(t)$, which has a Gaussian process prior.

$$
\begin{aligned}
u(t) &\sim \mathrm{GP}(0, C) \\
C_{ij} &= Cov[u(t_i), u(t_j)] \\
&= \kappa^2 \exp[-\lambda(t_i - t_j)^2] + \delta_{ij}\xi^2
\end{aligned}
$$

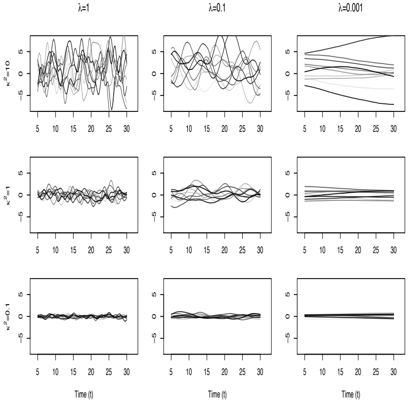
- We specify the spike probability, p_t, within time interval t in terms of $u(t)$ through the following transformation:

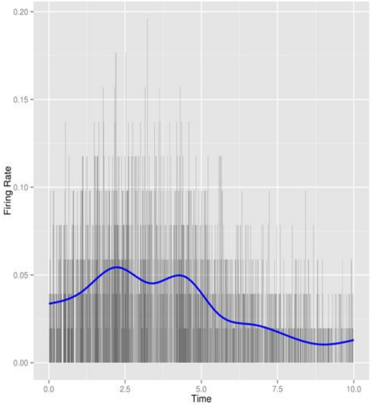
$$
p_t = \frac{1}{1 + \exp[-u(t)]}
$$

As $u(t)$ increases, so does p_t.

Modeling Firing Rates

- Our model can capture time-varying firing rates





- However, its extension to multiple neurons was not easy; therefore, we decided to use a slightly different model.

Stationary Model for Detecting Synchrony

- For multiple neurons, we model the joint firing probability of spike trains at time t, $\mathrm{P}_r(Y_{1t} = y_{1t}, \dots, Y_{nt} = y_{nt})$, subject to the following simplex constraint:

$$\sum_{(y_{1t},\dots,y_{nt})\in(0,1)^n} \mathrm{P}_r(Y_{1t} = y_{1t}, \dots, Y_{nt} = y_{nt}) = 1, \quad t = 1, 2, \dots, T$$

- We use a continuous latent variable u_{it} and a threshold τ_{ij}

$$Y_{ij} = \mathbb{1}_{(-\infty,\tau_{ij}]}(u_{ij}) \quad = \quad \begin{cases} 1, & \text{if } u_{ij} \leq \tau_{ij} \\ 0, & \text{otherwise.} \end{cases}$$

$$(u_{1t}, \dots, u_{nt})^T \quad \sim \quad N(0, \Sigma)$$

- The support of the joint distribution is the Cartesian cross of the real lines partitioned into 2^n quadrants.

Covariance Matrix

- We specify the the covariance matrix,

$$\Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{12} & \sigma_{22} & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1n} & \sigma_{2n} & \cdots & \sigma_{nn} \end{bmatrix},$$

by decomposing it as follows:

$$\Sigma = \begin{bmatrix} \sigma_{11}^{\frac{1}{2}} & 0 & \cdots & 0 \\ 0 & \sigma_{22}^{\frac{1}{2}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{nn}^{\frac{1}{2}} \end{bmatrix} \begin{bmatrix} 1 & \rho_{12} & \cdots & \rho_{1n} \\ \rho_{12} & 1 & \cdots & \rho_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{1n} & \rho_{2n} & \cdots & 1 \end{bmatrix} \begin{bmatrix} \sigma_{11}^{\frac{1}{2}} & 0 & \cdots & 0 \\ 0 & \sigma_{22}^{\frac{1}{2}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{nn}^{\frac{1}{2}} \end{bmatrix}$$

- We use ρ_{ij} for inference regarding the connection between neurons i and j.

Gaussian Process Prior on the Thresholds

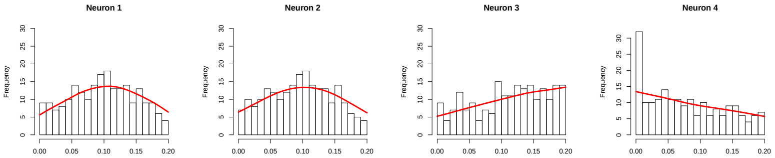
- For each neuron, we assume the following Gaussian process prior for the *threshold* parameters:

$$
\begin{aligned}
(\tau_{i1}, ..., \tau_{iT}) &\sim \mathrm{GP}(0, C_i) \\
C_i|_{j,k} &= \mathrm{Cov}(\tau_{ij}, \tau_{ik}) \\
&= \kappa_i^2 \exp[-\lambda_i (t_i - t_j)^2] + \delta_{jk} \xi_i^2
\end{aligned}
$$

- Alternatively, a Brownian motion prior can be used to improve computational efficiency:

$$C_i|_{j,k} = \mathrm{Cov}(\tau_{ij}, \tau_{ik}) = \theta_i \min(j, k)$$

Illustration



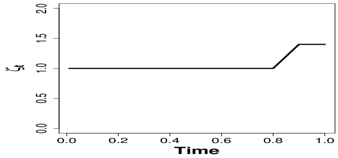
Parameter	Posterior Estimate
ρ_{12}	**0.75 (0.66,0.84)**
ρ_{13}	0.02 (-0.12,0.15)
ρ_{14}	0.12 (-0.01,0.25)
ρ_{23}	-0.05 (-0.19,0.12)
ρ_{24}	0.03 (-0.13,0.17)
ρ_{34}	**-0.46 (-0.61,-0.32)**

Limitations

- The above method has several shortcomings:
  - It is static (stationary) so it cannot capture the dynamics of neural networks.
  - Also, the method does not identify neuronal communities.

Illustrating Limitations of Static Models

- Consider the following example, where the probability of co-firing is the product of marginal probabilities times ζ, representing excessive firing rate beyond random:



- Using our method, the 95% probability interval for correlation is $[-0.039, 0.066]$.

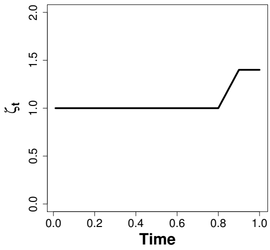
- The method of Kass and Kelly (2011) and Shahbaba et al. (2014) also report non-significant results.

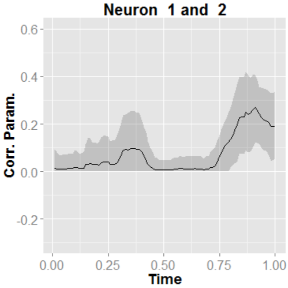
A non-stationary model

- We can modify the correlation matrix to make it non-stationary, while preserving its positive-definiteness:

$$
\begin{aligned}
\Sigma_t &= \begin{bmatrix} \sigma_{11} & \sigma_{12} I_1^t I_2^t & \cdots & \sigma_{1n} I_1^t I_n^t \\ \sigma_{12} I_2^t I_1^t & \sigma_{22} & \cdots & \sigma_{2n} I_2^t I_n^t \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1n} I_n^t I_1^t & \sigma_{2n} I_n^t I_2^t & \cdots & \sigma_{nn} \end{bmatrix} \\
&= \begin{bmatrix} I_1^t & 0 & \cdots & 0 \\ 0 & I_2^t & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & I_n^t \end{bmatrix} \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{12} & \sigma_{22} & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1n} & \sigma_{2n} & \cdots & \sigma_{nn} \end{bmatrix} \begin{bmatrix} I_1^t & 0 & \cdots & 0 \\ 0 & I_2^t & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & I_n^t \end{bmatrix} \\
&+ \begin{bmatrix} \sigma_{11}(1 - I_1^t) & 0 & \cdots & 0 \\ 0 & \sigma_{22}(1 - I_2^t) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{nn}(1 - I_n^t) \end{bmatrix}.
\end{aligned}
$$

- The resulting model can capture the dynamics of the cross-neuronal interactions





- However, it still cannot detect neuronal communities.

Chinese Restaurant Process

- In order to detect subsets (communities) of correlated neurons, we use Ewens sampling formula to allocate neurons into K partitions without pre-specifying K
- Given the previous $j-1$ assignments, we assign the j^{th} neuron either to an existing subset, M, with probability

$$\frac{|M|}{\alpha + j - 1}$$

or to an empty subset (i.e., the neuron starts a new partition) with probability

$$\frac{\alpha}{\alpha + j - 1}$$

Chinese Restaurant Process

- This defines a prior distribution on a partition, $\{z_1, z_2, \dots, z_n\}$, where z_i assigns the i^{th} neuron to one of the K partitions.

- Given the partition, we rewrite the correlation matrix, R, as follows:

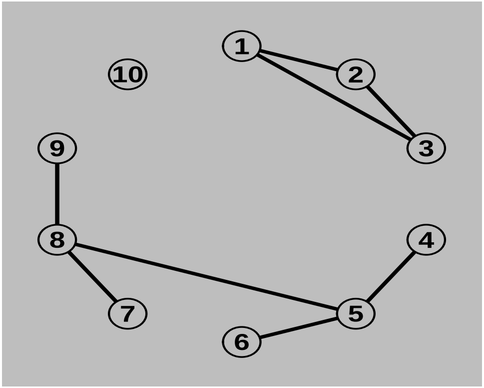
$$R = \begin{bmatrix} 1 & \rho_{12}\mathbb{1}_{\{z_1=z_2\}} & \cdots & \rho_{1n}\mathbb{1}_{\{z_1=z_n\}} \\ \rho_{12}\mathbb{1}_{\{z_1=z_2\}} & 1 & \cdots & \rho_{2n}\mathbb{1}_{\{z_2=z_n\}} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{1n}\mathbb{1}_{\{z_1=z_n\}} & \rho_{2n}\mathbb{1}_{\{z_2=z_n\}} & \cdots & 1 \end{bmatrix}$$

Non-stationary Model for Detecting Synchrony

- We generalize the above model by allowing the partitions change over time,

$$R_t = \begin{bmatrix} 1 & \rho_{12}\mathbb{1}_{\{z_{1t}=z_{2t}\}} & \cdots & \rho_{1n}\mathbb{1}_{\{z_{1t}=z_{nt}\}} \\ \rho_{12}\mathbb{1}_{\{z_{1t}=z_{2t}\}} & 1 & \cdots & \rho_{2n}\mathbb{1}_{\{z_{2t}=z_{nt}\}} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{1n}\mathbb{1}_{\{z_{1t}=z_{nt}\}} & \rho_{2n}\mathbb{1}_{\{z_{2t}=z_{nt}\}} & \cdots & 1 \end{bmatrix}$$

Simulation 1– Stationary Model



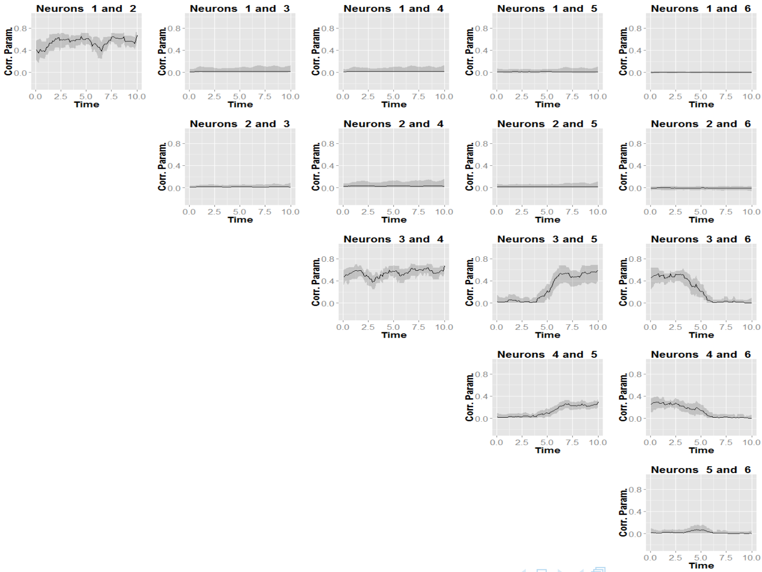
Simulation 1– Stationary Model

1	**0.45(0.31,0.59)**	**0.52(0.33,0.69)**	0.02(-0.16,0.19)	-0.01(-0.17,0.14)	0.00(-0.13,0.14)	0.04(-0.10,0.21)	0.00(-0.11,0.12)	0.01(-0.12,0.15)	0.00(-0.10,0.11)
0.45(0.31,0.59)	1	**0.47(0.31,0.64)**	0.03(-0.12,0.16)	0.00(-0.12,0.11)	0.00(-0.13,0.12)	-0.02(-0.17,0.13)	0.01(-0.15,0.16)	0.02(-0.12,0.17)	0.00(-0.12,0.11)
0.52(0.33,0.69)	**0.47(0.31,0.64)**	1	0.00(-0.13,0.14)	0.01(-0.11,0.15)	0.00(-0.15,0.14)	0.02(-0.11,0.16)	0.01(-0.10,0.15)	0.03(-0.11,0.18)	0.01(-0.11,0.16)
0.02(-0.16,0.19)	0.03(-0.12,0.16)	0.00(-0.13,0.14)	1	**0.42(0.27,0.55)**	0.05(-0.0.10,0.22)	-0.03(-0.18,0.11)	0.01(-0.15,0.17)	-0.04(-0.21,0.12)	-0.02(-0.17,0.12)
-0.01(-0.17,0.14)	0.00(-0.12,0.11)	0.01(-0.11,0.15)	**0.42(0.27,0.55)**	1	**0.44(0.30,0.59)**	0.03(-0.12,0.20)	**0.51(0.35,0.66)**	-0.03(-0.18,0.11)	0.00(-0.14,0.15)
0.00(-0.13,0.14)	0.00(-0.13,0.12)	0.00(-0.15,0.14)	0.05(-0.0.10,0.22)	**0.44(0.30,0.59)**	1	0.03(-0.11,0.18)	0.01(-0.14,0.16)	0.04(-0.11,0.20)	0.01(-0.14,0.17)
0.04(-0.10,0.21)	-0.02(-0.17,0.13)	0.02(-0.11,0.16)	-0.03(-0.18,0.11)	0.03(-0.12,0.20)	0.03(-0.11,0.18)	1	**0.46(0.29,0.61)**	0.05(-0.08,0.21)	0.02(-0.13,0.17)
0.00(-0.11,0.12)	0.01(-0.15,0.16)	0.01(-0.10,0.15)	0.01(-0.15,0.17)	**0.51(0.35,0.66)**	0.01(-0.14,0.16)	**0.46(0.29,0.61)**	1	**0.44(0.28,0.60)**	0.00(-0.12,0.13)
0.01(-0.12,0.15)	0.02(-0.12,0.17)	0.03(-0.11,0.18)	-0.04(-0.21,0.12)	-0.03(-0.18,0.11)	0.04(-0.11,0.20)	0.05(-0.08,0.21)	**0.44(0.28,0.60)**	1	0.00(-0.13,0.13)
0.00(-0.10,0.11)	0.00(-0.12,0.11)	0.01(-0.11,0.16)	-0.02(-0.17,0.12)	0.00(-0.14,0.15)	0.01(-0.14,0.17)	0.02(-0.13,0.17)	0.00(-0.12,0.13)	0.00(-0.13,0.13)	1

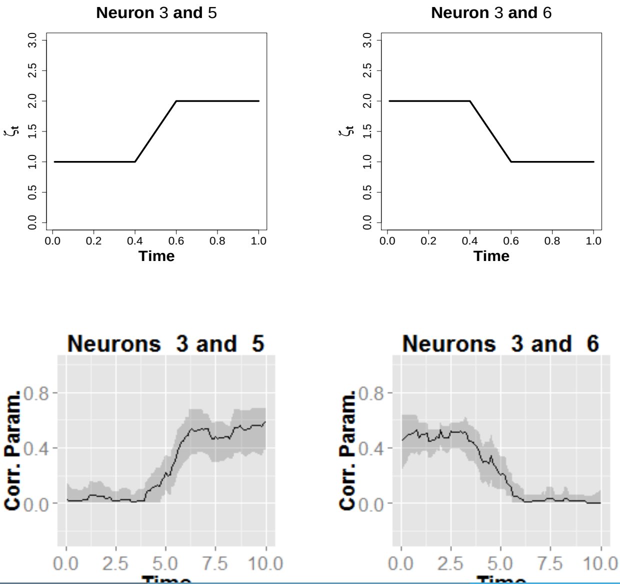
Simulation 2– Dynamic Model

- We first generate time-varying marginal firing probabilities for each neuron; the joint probabilities of the two neurons is the product of marginal firing probabilities times an extra term, ζ
- For the first community, there are two neurons (indexed by 1 and 2) with constant correlation structure ($\zeta = 2$)
- For the second community, there are 4 neurons (indexed by 3, 4, 5 and 6), where neuron 3 and 4 have constant correlation ($\zeta = 2$)
- The correlations between neuron 3 and neurons 5 and 6 change over
- Neurons 7 to 10 are not involved in the network ($\zeta = 1$)

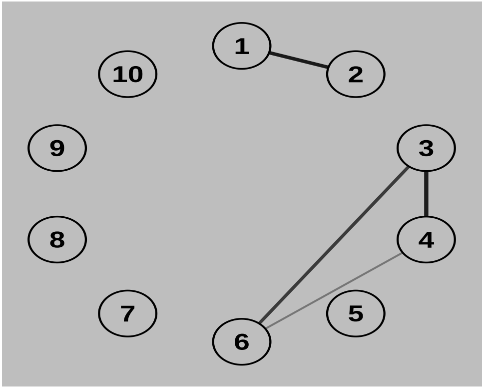
Simulation 2– Dynamic Model



Simulation 2– Dynamic Model

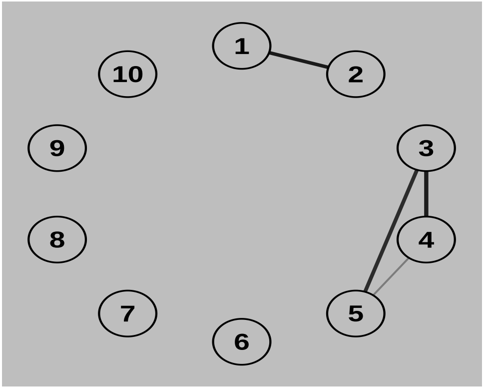


Simulation 2– Dynamic Model



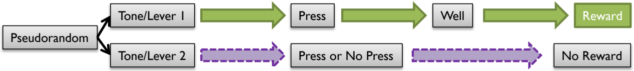
First Half

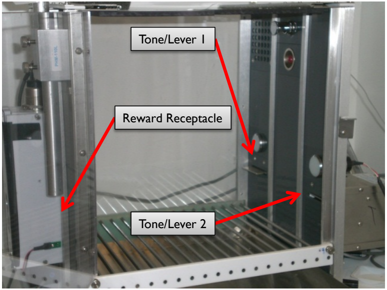
Simulation 2– Dynamic Model



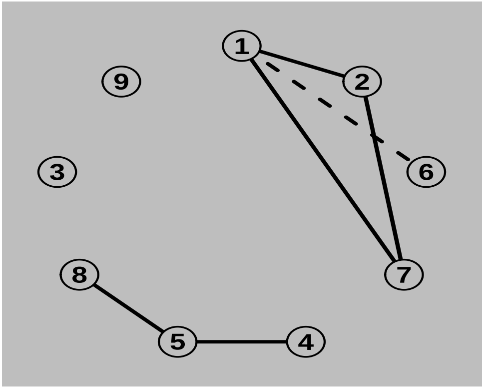
Second Half

Experimental Data



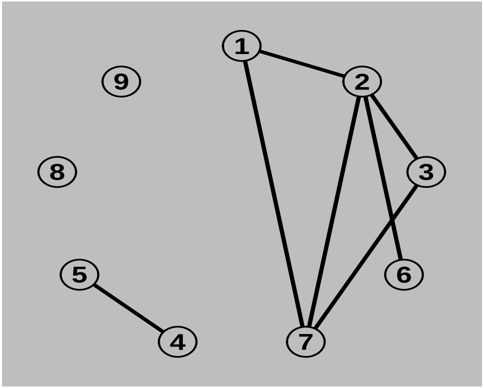


Experimental Data



Rewarded

Experimental Data



Non-rewarded

Discussion

- Our method has a number of advantages:
  - It captures the time-varying dependencies among neurons
  - It is useful for testing differences in cross-neuronal correlations in activity between different experimental conditions
  - It can be applied to a moderate to large number of neurons, and the computational complexity increases with the number of interacting neurons only
  - It can be applied to different data from a broad spectrum including continuous-valued time series that have some latent structure

Discussion

- To improve our method, we need to
  - properly and adaptively adjust bin-widths
  - allow for possible changes in the direction of relationships (from positive to negative and vice versa)
  - reduce computational cost

Acknowledgement

- Collaborators:
  - Sam Behseta (CSUF)
  - Hernando Ombao (UCI)
  - David Moorman (UMass Amherst)
  - Bo Zhou (UCI)
  - Shiwei Lan (UCI)
- Grants:
  - NSF: IIS-1216045
  - NIH: R01-AI107034

Thank You!