

Statistical Machine Learning Methods for High-dimensional Neural Population Data Analysis

Yuanjun Gao

Department of Statistics
Columbia University

TODO: add a diagram for statistical criticizing

TODO: add a page for spike train

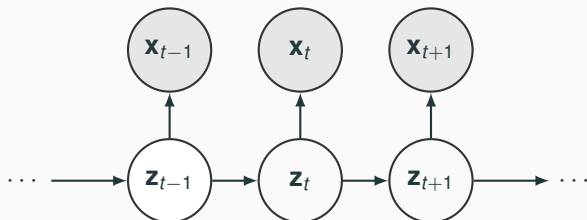
Table of Contents

- Neural Population Data Analysis with Latent Variable Models
 - Generalized count linear dynamical system
 - Linear dynamical neural population models through nonlinear embeddings
- Region of Interest Detection for Calcium Imaging Data
- Maximum Entropy Flow Networks

Table of Contents

- Neural Population Data Analysis with Latent Variable Models
 - Generalized count linear dynamical system
 - Linear dynamical neural population models through nonlinear embeddings
- Region of Interest Detection for Calcium Imaging Data
- Maximum Entropy Flow Networks

State space models

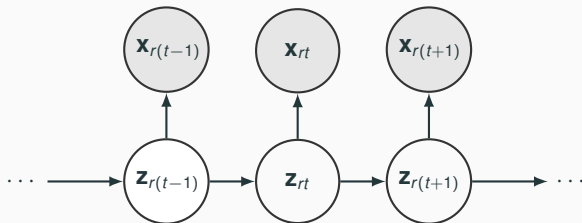


- $\mathbf{x}_t \in \mathbb{N}^n$: spike counts; $\mathbf{z}_t \in \mathbb{R}^m$: latent variables
- Joint distribution

$$p(\mathbf{x}, \mathbf{z}) = \underbrace{p(\mathbf{z}_1)}_{\text{Initial distribution}} \underbrace{\prod_{t=1}^{T-1} p(\mathbf{z}_{t+1} | \mathbf{z}_t)}_{\text{Transition model}} \underbrace{\prod_{t=1}^T p(\mathbf{x}_t | \mathbf{z}_t)}_{\text{Observation model}}$$

- Common input; Dynamical view of motor data (TODO: elaborate this line)

State space models: multiple trials



- $r = 1, \dots, R$: trial number
- $\mathbf{x}_{rt} \in \mathbb{N}^n$: spike counts; $\mathbf{z}_{rt} \in \mathbb{R}^m$: latent variables
- Joint distribution

$$p(\mathbf{x}, \mathbf{z}) = \prod_{r=1}^R \left[\underbrace{p(\mathbf{z}_{r1})}_{\text{Initial distribution}} \underbrace{\prod_{t=1}^{T-1} p(\mathbf{z}_{r(t+1)} | \mathbf{z}_{rt})}_{\text{Transition model}} \underbrace{\prod_{t=1}^T p(\mathbf{x}_{rt} | \mathbf{z}_{rt})}_{\text{Observation model}} \right]$$

Common parameterization and our extensions

- Common assumptions for latent dynamics: linear Gaussian dynamical system (LDS)

$$\mathbf{z}_1 \sim \mathcal{N}(\mu_1, Q_1)$$

$$\mathbf{z}_{t+1} | \mathbf{z}_t \sim \mathcal{N}(A\mathbf{z}_t, Q)$$

- Common observation models:

$$\mathbf{x}_t | \mathbf{z}_t \sim \underbrace{\mathcal{N}(C\mathbf{z}_t + d, \Sigma)}_{\text{model mismatch}} \text{ or } \underbrace{\text{Poisson}(\exp(C\mathbf{z}_t + d))}_{\text{equal dispersion}}$$

stringent assumptions

- Our extensions for observation model:
 - Generalized count distribution (GCLDS) (Gao et al. 2015)
 - Flexible nonlinear observation (fLDS) (Gao et al. 2016)

Table of Contents

- Neural Population Data Analysis with Latent Variable Models
 - Generalized count linear dynamical system
 - Linear dynamical neural population models through nonlinear embeddings
- Region of Interest Detection for Calcium Imaging Data
- Maximum Entropy Flow Networks

Motivation

- Doubly stochastic Poisson model implies **overdispersion**

$$\left. \begin{array}{l} \mathbf{z} \sim p(\mathbf{z}) \\ \mathbf{x} \sim \text{Poisson}(f(\mathbf{z})) \end{array} \right\} \Rightarrow \text{var}(\mathbf{x}) \geq E(\mathbf{x})$$

- Need a more flexible distribution to separate **firing rate variability** with **noise variability**.

$$\text{var}(\mathbf{x}) = \underbrace{\text{var}(E(\mathbf{x}|\mathbf{z}))}_{\text{firing rate variability}} + \underbrace{E(\text{var}(\mathbf{x}|\mathbf{z}))}_{\text{noise variability}}$$

Generalized count distribution family

- Generalized count (GC) distribution family

$$p_{\text{Poisson}}(x; \lambda) \propto \frac{\exp \{\log \lambda \cdot x\}}{x!}, \quad x \in \mathbb{N}$$

\Downarrow

$$p_{\text{GC}}(x; \theta, g(\cdot)) \propto \frac{\exp(\theta \cdot x + g(x))}{x!}, \quad x \in \mathbb{N}$$

where $\theta \in \mathbb{R}$, $g(\cdot) : \mathbb{N} \rightarrow \mathbb{R}$.

- Parameterizes **all** the count distributions **redundantly**.
- Given $g(\cdot)$, θ controls the expectation.
- $g(\cdot)$ controls the “shape” of the distribution.
Convex/concave $g(\cdot)$ implies over/under-dispersion.

Model formulation

- Linear dynamical systems with generalized count observation

$$\mathbf{z}_{r1} \sim \mathcal{N}(\mu_1, Q_1)$$

$$\mathbf{z}_{r(t+1)} | \mathbf{z}_{rt} \sim \mathcal{N}(A\mathbf{z}_{rt}, Q)$$

$$x_{rti} \sim \mathcal{GC}(c_i^T \mathbf{z}_{rt}, g_i(\cdot)), i = 1, \dots, n$$

- Practical considerations
 - Set $g_i(k) = -\infty$ for $k > K$ to facilitate computation;
 - Ridge penalty on the 2nd difference of $g_i(\cdot)$ to avoid overfitting;
 - Set $g_i(0) = 0$ without loss of generality.

Variational Bayes Expectation Maximization (VBEM)

- \mathbf{x} : data, \mathbf{z} : latent variables, θ : model parameters,
- Often hard to compute $p_{\theta}(\mathbf{x}) = \int p_{\theta}(\mathbf{x}, \mathbf{z}) d\mathbf{z}$ and $p_{\theta}(\mathbf{z}|\mathbf{x})$.
- Approximate the posterior by a **tractable** distribution family.

$$p_{\theta}(\mathbf{z}|\mathbf{x}) \approx q(\mathbf{z}) \in \mathcal{Q}$$

- Optimize a **lower bound of log likelihood**, or ELBO

$$\begin{aligned}\text{ELBO}(\theta, q) &= \int [\log p_{\theta}(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z})] q(\mathbf{z}) d\mathbf{z} \\ &= \log p_{\theta}(\mathbf{x}) - \text{KL}(q(\mathbf{z}) || p_{\theta}(\mathbf{z}|\mathbf{x})) \leq \log p_{\theta}(\mathbf{x})\end{aligned}$$

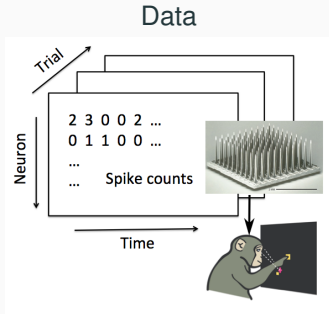
Variational Bayes Expectation Maximization (VBEM)

- VBEM: Optimize $\text{ELBO}(\theta, q) \leq \log p_{\theta}(\mathbf{x})$ iteratively
 - E-step: For a fixed θ , optimize q
 - M-step: For a fixed q , optimize θ
- VBEM for GCLDS
 - We set q to be multivariate Gaussian
 - We derive a looser but tractable ELBO
 - E-step: fast Laplace approximation initialization + dual optimization
 - M-step: convex optimization + analytical solution

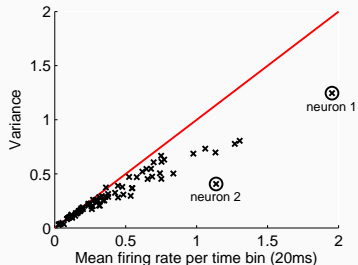
- For both simulated and real dataset, we compare GCLDS with PLDS (Poisson observation model)

	Mean	Variance	Likelihood
PLDS	✓	✗	✗
GCLDS	✓	✓	✓

Real data analysis: data



Variance and mean of spike counts



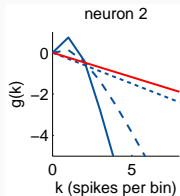
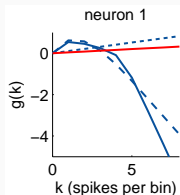
- Center-out reaching experiments
- Multi-electrode array recording
- Strong under-dispersion

Real data analysis: algorithms

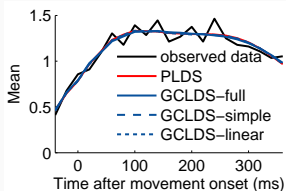
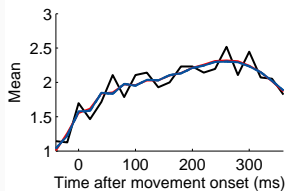
- Main algorithms to be compared
 - **PLDS**: Poisson observation
 - **GCLDS-full**: Generalized count observation, individual $g(\cdot)$ across neurons
- Two control cases for GCLDS
 - **GCLDS-linear**: truncated linear $g(\cdot)$ (truncated Poisson)
 - **GCLDS-simple**: $g(\cdot)$ shared across neurons (up to a linear function)

Real data analysis: single neuron fit

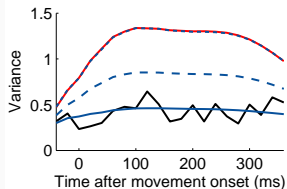
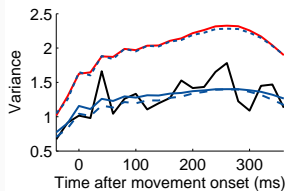
Fitted $g(\cdot)$



Fitted mean

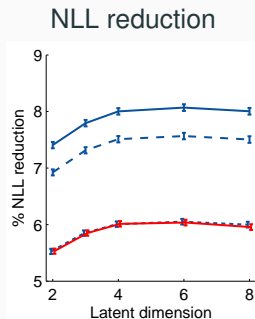
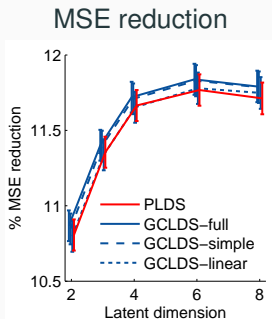


Fitted variance



Real data analysis: population fit

- Leave-one-neuron-out prediction



Conclusion and discussion

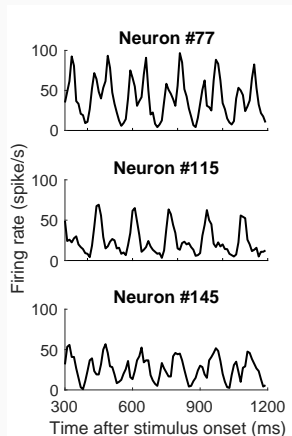
- Summary
 - Incorporated generalized count family into state space models.
 - Developed VBEM algorithm.
 - Observed superior fitted result on real neural data.
- Extensions
 - $g(\cdot)$ vary across time?
 - Share information of $g(\cdot)$ across neurons? (hierarchical model?)
 - Generative models for under-dispersion?

Table of Contents

- Neural Population Data Analysis with Latent Variable Models
 - Generalized count linear dynamical system
 - Linear dynamical neural population models through nonlinear embeddings
- Region of Interest Detection for Calcium Imaging Data
- Maximum Entropy Flow Networks

Motivation

- Neural activities lie in a low-dimensional **nonlinear manifold** rather than a **linear subspace**
- Flexible observation model makes the state space model more expressive



Model formulation: fLDS

- Linear dynamical systems with **nonlinear link** and count observation

$$\mathbf{z}_{r1} \sim \mathcal{N}(\mu_1, Q_1)$$

$$\mathbf{z}_{r(t+1)} | \mathbf{z}_{rt} \sim \mathcal{N}(A\mathbf{z}_{rt}, Q)$$

$$x_{rti} \sim \text{Poisson}(f_i(\mathbf{z}_{rt})) \text{ (PfLDS)}$$

$$\text{or } \mathcal{GC}(f_i(\mathbf{z}_{rt}), g_i(\cdot)) \text{ (GCfLDS)}$$

where f_i is a nonlinear function parameterized by a neural network

- Linear dynamics: simple, tractable, interpretable
- Nonlinear observation: flexibility

Inference algorithm: AEVB (high level idea)

- Auto-encoding Variational Bayes (AEVB)
- Learn a mapping (recognition model) from data to the approximate posterior distribution of latent variable.
- Jointly optimize the generative model parameters and recognition model parameters.
- Naturally incorporate stochastic optimization to handle large datasets.

Inference algorithm: AEVB (algorithm)

- Decompose ELBO by trials

$$\text{ELBO}(\theta, q) = \sum_{r=1}^R \int [\log p_{\theta}(\mathbf{x}_r, \mathbf{z}_r) - \log q(\mathbf{z}_r)] q(\mathbf{z}_r) d\mathbf{z}_r$$

- Map data \mathbf{x}_r to $q(\mathbf{z}_r)$ by a parameterized function

$$q(\mathbf{z}_r) = q_{\phi}(\mathbf{z}_r; \mathbf{x}_r) = \mathcal{N}(\mu_{\phi}(\mathbf{x}_r), \Sigma_{\phi}(\mathbf{x}_r))$$

- Learn both θ and ϕ by optimizing ELBO

$$\text{ELBO}(\theta, \phi) = \sum_{r=1}^R \int [\log p_{\theta}(\mathbf{x}_r, \mathbf{z}_r) - \log q_{\phi}(\mathbf{z}_r; \mathbf{x}_r)] q_{\phi}(\mathbf{z}_r; \mathbf{x}_r) d\mathbf{z}_r$$

- Do stochastic optimization with gradient of a single trial

Inference algorithm: AEVB (important details)

- Specific parameterization of the recognition model

$$q(\mathbf{z}_r) = q_\phi(\mathbf{z}_r; \mathbf{x}_r) = \mathcal{N}(\mu_\phi(\mathbf{x}_r), \Sigma_\phi(\mathbf{x}_r))$$

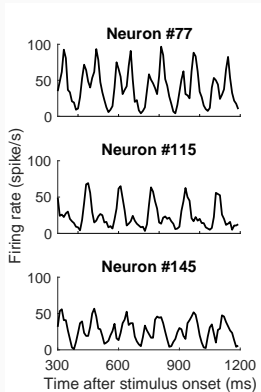
- Block tri-diagonal precision matrix that agrees with Markovian structure
 - Potentially useful to perform filtering in an online fashion
- Reparameterization trick for stochastic optimization
 - Easy implementation
 - Low variance

Experiments

	Mean	Variance	Likelihood	Concise representation
PLDS	✓	✗	✗	✗
GCLDS	✓	✓	✓	✗
PfLDS	✓	✗	✗	✓
GCfLDS	✓	✓	✓	✓

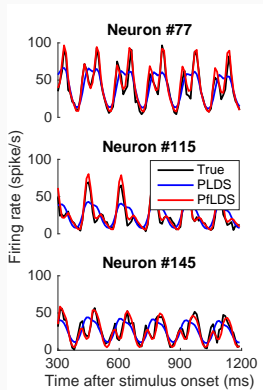
Real data analysis: primate visual cortex

Firing rate



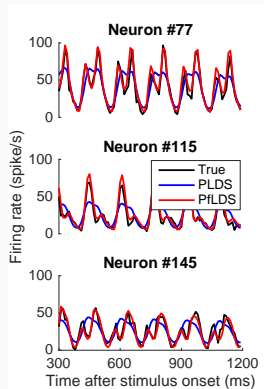
Real data analysis: primate visual cortex

Firing rate



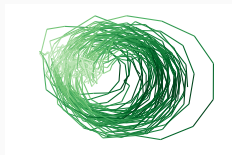
Real data analysis: primate visual cortex

Firing rate

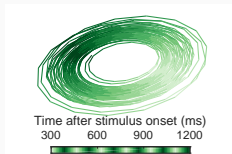


Latent projection

PLDS

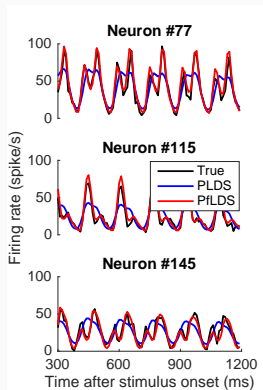


PfLDS



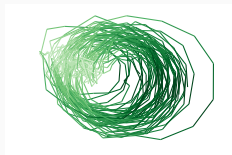
Real data analysis: primate visual cortex

Firing rate

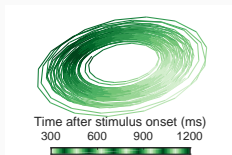


Latent projection 1-step-ahead prediction

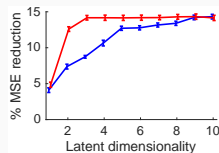
PLDS



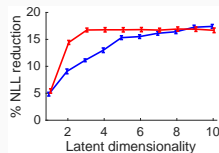
PfLDS



MSE reduction

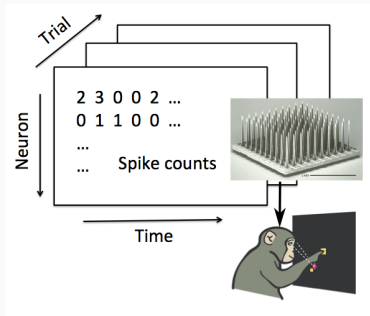


NLL reduction

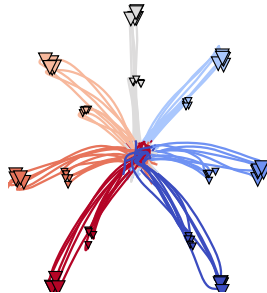


Real data analysis: Primate motor cortex

Data



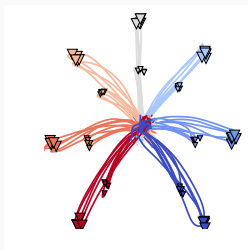
Reaching trajectory



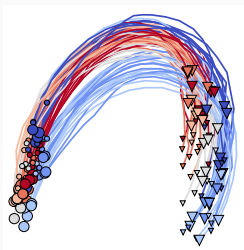
Real data analysis: Primate motor cortex

- Latent projection with 2 latent dimensions

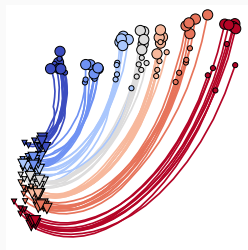
Reaching trajectory



PLDS

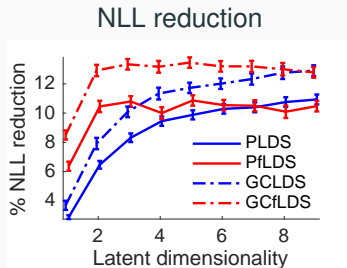
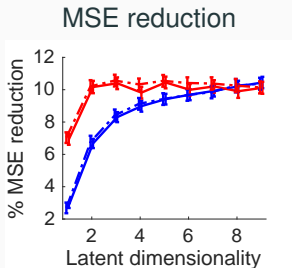


PfLDS



Real data analysis: Primate motor cortex

- One-step-ahead predictive performance



Conclusion and discussion

- Summary
 - Incorporated nonlinear observation into state space models.
 - Developed AEVB algorithm (flexible and scalable).
 - Obtain concise latent representation.
- Future work
 - Better stochastic optimization scheme
 - Interpretable nonlinearity
 - Application on more complex datasets

Table of Contents

- Neural Population Data Analysis with Latent Variable Models
 - Generalized count linear dynamical system
 - Linear dynamical neural population models through nonlinear embeddings
- Region of Interest Detection for Calcium Imaging Data
- Maximum Entropy Flow Networks

Introduction: calcium imaging data

TODO: incorporate a video?

Model formulation: idea

- $X \in \mathbb{R}^{N \times T}$ represents the calcium imaging data, where each column is a (vectorized) frame that contains N pixels
- Decompose X into a product of K spatial component and temporal component

$$X = D \cdot A^T + \text{noise}$$

- $D = [D_1, \dots, D_K] \in \mathbb{R}^{N \times K}$ represents the neuron shapes
- $A = [A_1, \dots, A_K] \in \mathbb{R}^{T \times K}$ is the neural activities
- Further exploit structure of the components (localized neuron shapes)

Model formulation: objective

- Structured matrix factorization

$$\begin{aligned} & \underset{D, A}{\text{minimize}} && \|X - DA^T\|_2^2 + f_D(D), \\ & \text{subject to} && D_k \in \mathcal{D}_w^+; k = 1, \dots, K, \\ & && \|A_k\|_2 \leq c_k, \end{aligned} \tag{1}$$

- \mathcal{D}_w^+ : non-negative vectors whose nonzero values is within a $w \times w$ window
- $f_D(D)$ regularizes the neuron shape (discussed later)
- $\|A_k\|_2 \leq c_k$ avoids degenerate solution

Greedy algorithm

- Scan the each frame of the video with a small Gaussian kernel
- At iteration k , given the current residue (unexplained by existing ROI)
 - **Greedy identification**: Identify the location p_k where the Gaussian kernel explains most of the data (across time)
 - **Shape fine tuning**: Locally optimize the spatial and temporal component
 - **Residue update**: Subtract the newly identified ROI

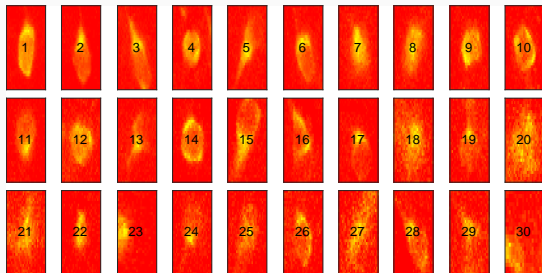
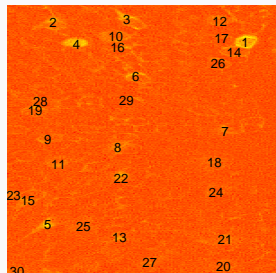
Shape fine tuning

- Given current residue R , an identified center pixel p_k , denote S_k as a $w \times w$ window centered at p_k

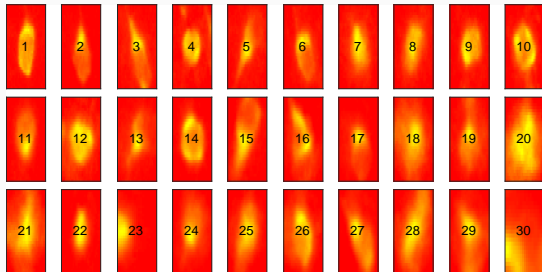
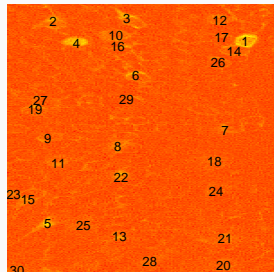
$$\begin{aligned} & \underset{D_k, A_k}{\text{minimize}} && \|R - D_k A_k^T\|^2 + f(D_k), \\ & \text{subject to} && D_{kp} \geq 0, p \in S_k, \\ & && D_{kp} = 0, p \notin S_k, \\ & && \|A_k\|_2 \leq c_k, \end{aligned} \tag{2}$$

- $f(D_k) = \sum_{i=1}^3 \lambda_i f_i(D_k)$
 - $f_1(D_k) = \sum_p \tau(p, p_k) |D_{kp}|$ encourages sparsity
 - $f_2(D_k) = \sum_p (D_{kp} - G_{p_k})^2$ encourage Gaussian shape
 - $f_3(D_k) = \sum_{p_1 \text{ and } p_2 \text{ are neighbors}} (D_{kp_1} - D_{kp_2})^2$ encourages smoothness
- Optimize D_k and A_k by block coordinate descent

Real data analysis: sample patch, no shape regularization



Real data analysis: sample patch, shape regularization



Conclusion and discussion

- Summary
 - Formulating calcium imaging ROI detection as a structure matrix factorization problem
 - Greedy algorithm with shape regularization
 - Fast ROI detection algorithm
- Future work
 - More spatial and temporal structure
 - Overlapping neuron
 - Online ROI detection
 - Motion correction, background elimination

Table of Contents

- Neural Population Data Analysis with Latent Variable Models
 - Generalized count linear dynamical system
 - Linear dynamical neural population models through nonlinear embeddings
- Region of Interest Detection for Calcium Imaging Data
- Maximum Entropy Flow Networks

Introduction: maximum entropy principle

- **Maximum entropy principle:** Subject to some given prior knowledge (moment/support constraints), the distribution that makes **minimal additional assumptions** is that which has the **largest entropy** of any distribution obeying those constraints
- Useful for model formulation and hypothesis testing.
- **Problem:** hard to learn and sample from maximum entropy distribution when constraints are complicated

Gibbs distribution

Normalizing flow

MEFN formulation

Learning by augmented Lagrangian method

Application: Texture modeling

Conclusion

Acknowledgement