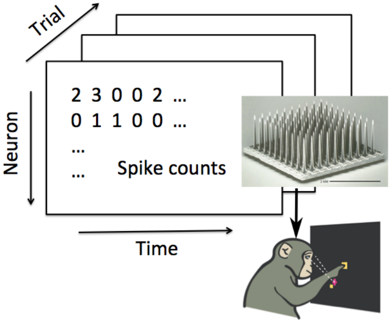
Statistical Machine Learning Methods for High-dimensional Neural Population Data Analysis

Yuanjun Gao

Department of Statistics
Columbia University

Overview



- Neuroscience + Big data = Opportunities!

Overview

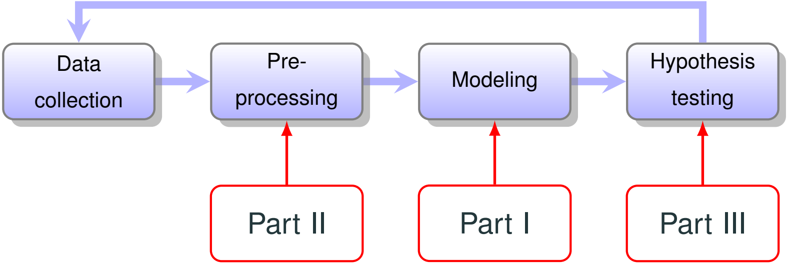
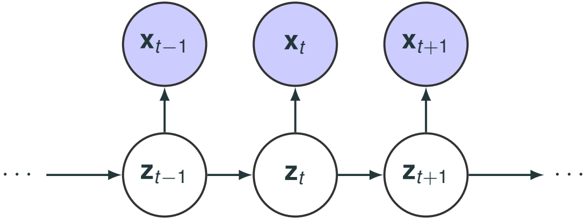

Data collection
Pre-processing
Modeling
Hypothesis testing
Part II
Part I
Part III

Table of Contents

Table of Contents

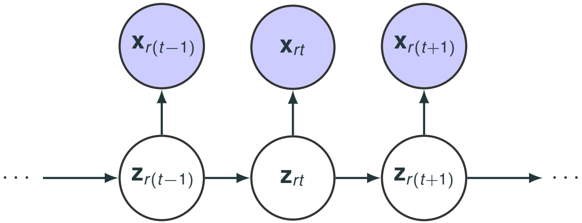
State space models



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

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

State space models: multiple trials



- $r = 1, ..., R$: 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:

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

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

Table of Contents

Motivation

- Doubly stochastic Poisson model implies overdispersion

$$\left.\begin{aligned} \mathbf{z} &\sim p(\mathbf{z}) \\ \mathbf{x} &\sim \text{Poisson}(f(\mathbf{z})) \end{aligned}\right\} \Rightarrow \begin{aligned} \text{var}(\mathbf{x}|\mathbf{z}) &= E(\mathbf{x}|\mathbf{z}) \\ \text{var}(\mathbf{x}) &\geq E(\mathbf{x}) \end{aligned}$$

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

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

Generalized count distribution family

- Generalized count (GC) distribution family

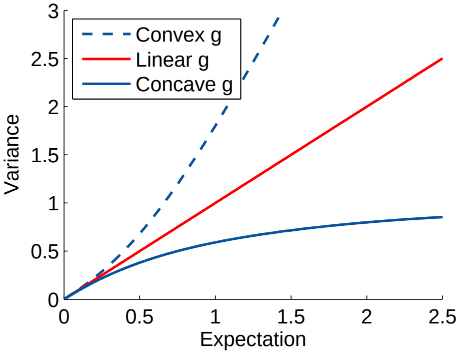
$$p_{\text{Poisson}}(x;\lambda) \propto \frac{\exp\{\log\lambda \cdot x\}}{x!}, \quad x \in \mathbb{N}$$

$$\Downarrow$$

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

  where $\theta \in \mathbb{R}$, $g(\cdot) : \mathbb{N} \to \mathbb{R}$.
- Parameterizes all the count distributions redundantly.
- Given $g(\cdot)$, θ controls the expectation.
- $g(\cdot)$ controls the “shape” of the distribution. Convex/concave/linear $g(\cdot)$ implies overdispersed/underdispered/Poisson distribution.

Generalized count distribution family



- Probability mass function for GC distribution family

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

Model formulation

- Linear dynamical systems with generalized count observation

$$\begin{aligned} \mathbf{z}_{r1} &\sim \mathcal{N}(\mu_1, Q_1) \\ \mathbf{z}_{r(t+1)} | \mathbf{z}_{rt} &\sim \mathcal{N}(A\mathbf{z}_{rt}, Q) \\ x_{rti} &\sim \mathcal{GC}(c_i^T \mathbf{z}_{rt}, g_i(\cdot)), i = 1, ..., n \end{aligned}$$

- 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;
    penalty $= \lambda \sum_{k=1}^{K-1} (g_i(k-1) - 2g_i(k) + g_i(k+1))^2$.

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

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

Variational Bayes Expectation Maximization (VBEM)

- VBEM: Optimize 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

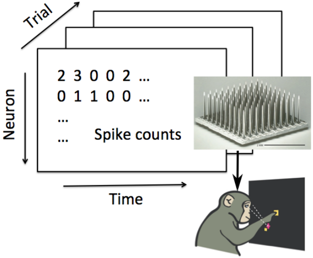
Experiments

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

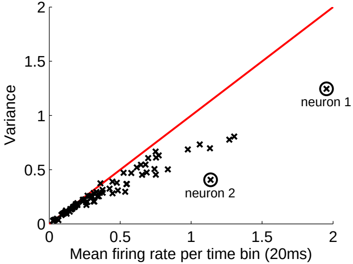
	Mean	Variance	Likelihood
PLDS	✓	✗	✗
GCLDS	✓	✓	✓

Real data analysis: data

Data



Variance and mean of spike counts

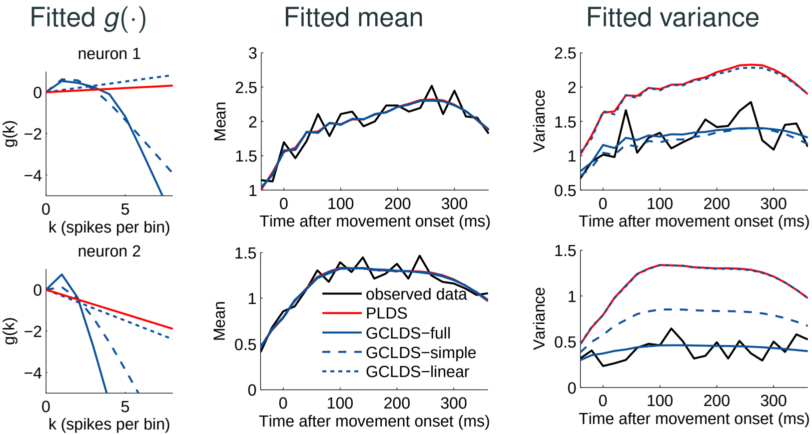


- Center-out reaching experiments
- Multi-electrode array recording
- Strong under-dispersion

Real data analysis: algorithms

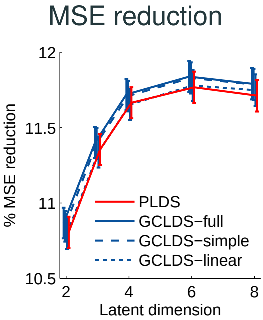
- Main algorithms to be compared
  - PLDS: Poisson observation
  - GCLDS-full: Generalized count observation, individual $g(\cdot)$ across neurons
- Two control cases for GCLDS
  - GCLDS-linear: truncated linear $g(\cdot)$ (truncated Poisson)
  - GCLDS-simple: $g(\cdot)$ shared across neurons (up to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

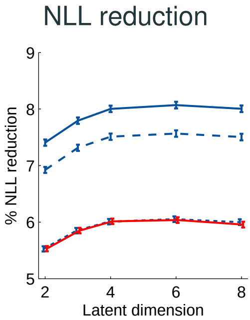
Real data analysis: single neuron fit



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 results on real neural data.
- Future work
  - Time-varying dispersion structure.
  - Hierarchical model that share information of $g(\cdot)$ across neurons.
  - Generative models for under-dispersion.
- Gao Y, Buesing L, Shenoy KV, Cunningham JP (2015) High-dimensional neural spike train analysis with generalized count linear dynamical systems. NIPS 2015.

Table of Contents

I. Neural Population Data Analysis with Latent Variable Models

- Generalized count linear dynamical system
- Linear dynamical neural population models through nonlinear embeddings

II. Region of Interest Detection for Calcium Imaging Data

III. 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

$$
\begin{aligned}
\mathbf{z}_{r1} &\sim \mathcal{N}(\mu_1, Q_1) \\
\mathbf{z}_{r(t+1)} | \mathbf{z}_{rt} &\sim \mathcal{N}(A\mathbf{z}_{rt}, Q) \\
x_{rti} &\sim \text{Poisson}(f_i(\mathbf{z}_{rt})) \text{ (PfLDS)} \\
&\quad \text{or } \mathcal{GC}(f_i(\mathbf{z}_{rt}), g_i(\cdot)) \text{ (GCfLDS)}
\end{aligned}
$$

  where f_i is a nonlinear function parameterized by a neural network
- Linear latent dynamics: simple, tractable, interpretable
- Nonlinear observation: flexible

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.
- Tractable for a large class of graphical models

Inference algorithm: AEVB (algorithm)

- Decompose ELBO by trials

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

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

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

- Learn both θ and ϕ by stochastic gradient descent on ELBO

$$\begin{aligned}\nabla \mathrm{ELBO}(\theta, q_\phi) &\approx R \times \nabla \int \left[\log \frac{p_\theta(\mathbf{x}_r, \mathbf{z}_r)}{q_\phi(\mathbf{z}_r | \mathbf{x}_r)} \right] q_\phi(\mathbf{z}_r | \mathbf{x}_r) d\mathbf{z}_r \\ &\approx R \times \text{an unbiased estimator of gradient}\end{aligned}$$

Structure of the recognition model

- Generative model:

$$p_\theta(\mathbf{z}_r|\mathbf{x}_r) \propto \underbrace{p_\theta(\mathbf{z}_{r1}) \prod_{t=1}^{T-1} p_\theta(\mathbf{z}_{t(t+1)}|\mathbf{z}_{rt})}_{Gaussian} \prod_{t=1}^{T} \underbrace{p_\theta(\mathbf{x}_{rt}|\mathbf{z}_{rt})}_{Complicated}$$

- Recognition model, product-of-Gaussian form:

$$q_\phi(\mathbf{z}_r|\mathbf{x}_r) \propto \underbrace{q_\phi(\mathbf{z}_{r1}) \prod_{t=1}^{T-1} q_\phi(\mathbf{z}_{r(t+1)}|\mathbf{z}_{rt})}_{Gaussian} \prod_{t=1}^{T} \underbrace{q_\phi(\mathbf{z}_{rt}|\mathbf{x}_{rt})}_{Gaussian}$$

  Approximates a complicated factor with a Gaussian factor dependent on the data in a complicated way.

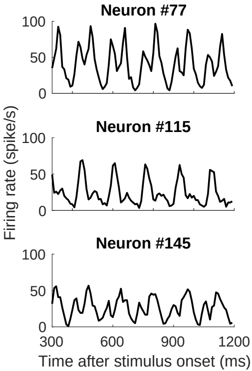
- Jointly Gaussian distribution with block tri-diagonal precision matrix. Maintaining the Markovian structure.

Experiments

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

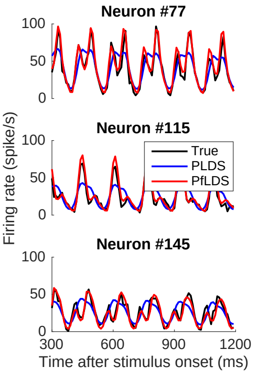
Real data analysis: primate visual cortex

Firing rate



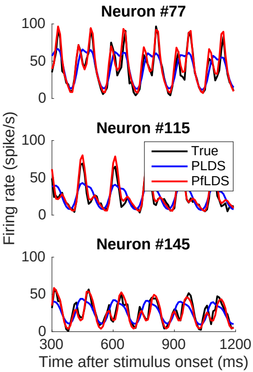
Real data analysis: primate visual cortex

Firing rate

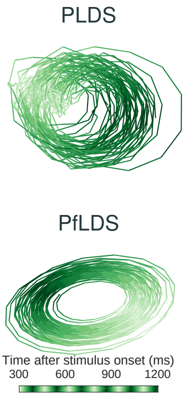


Real data analysis: primate visual cortex

Firing rate

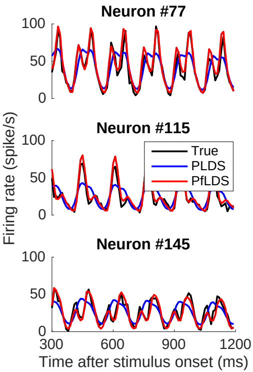


Latent projection



Real data analysis: primate visual cortex

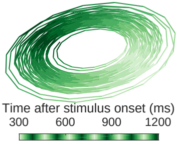
Firing rate



Latent projection

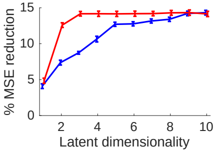
PLDS

PfLDS

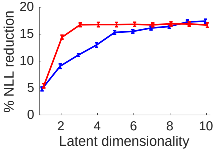


1-step-ahead prediction

MSE reduction

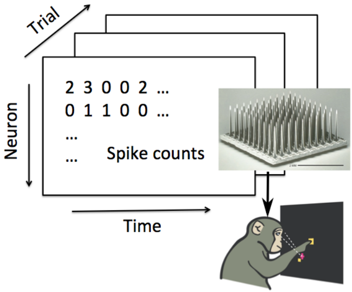


NLL reduction

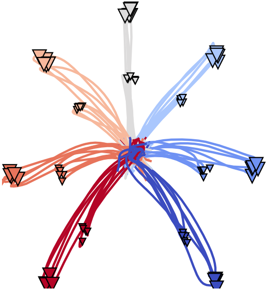


Real data analysis: Primate motor cortex

Data

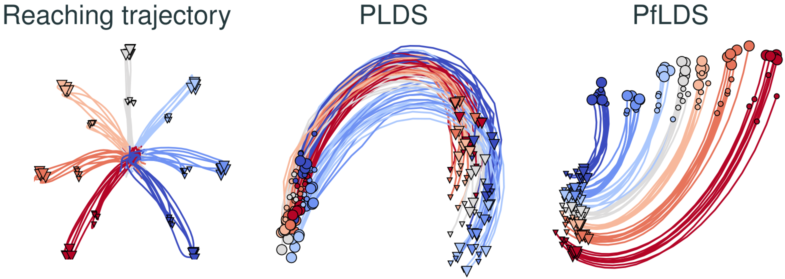


Reaching trajectory



Real data analysis: Primate motor cortex

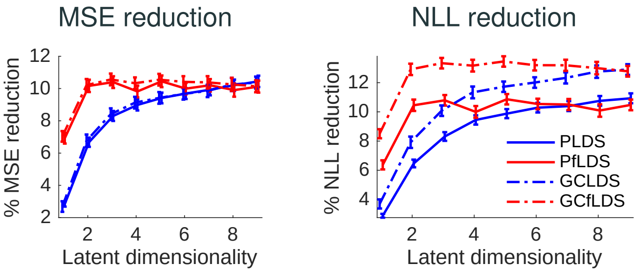
- Latent projection with 2 latent dimensions



- Recovering the behavior structure from neural data in a unsupervised fashion.
- Concise and informative latent dimensions.

Real data analysis: Primate motor cortex

- One-step-ahead predictive performance



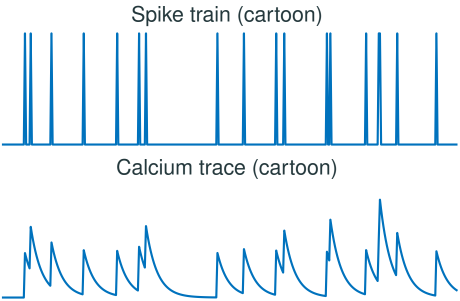
Conclusion and discussion

- Summary
  - Incorporated nonlinear observation into state space models.
  - Developed AEVB algorithm (flexible and scalable).
  - Obtained concise latent representations.
- Future work
  - Better stochastic optimization scheme.
  - Interpretable nonlinearity.
  - Application on more complex datasets.
- Gao Y*, Archer E*, Paninski L, Cunningham JP (2016) Linear dynamical neural population models through nonlinear embeddings. NIPS 2016. (* = equal contribution)

Table of Contents

Introduction: calcium imaging data

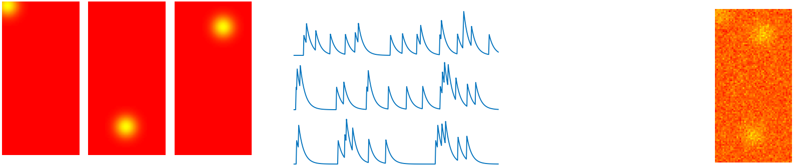
- Basic principle: the spiking activity of a neuron induces a transient increase in calcium concentration, which can be indirectly observed by recording the fluorescent properties of certain calcium indicators.



Introduction: calcium imaging data

- Calcium imaging enables simultaneous recording of many neurons.

Neuron shape $\times$ Calcium trace $+$ Noise $=$ Obseravtion



- Goal: recover the neuron shape and calcium trace given the observation.

Model formulation

- $X \in \mathbb{R}^{N \times T}$ 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

$$X = DA^T + \text{noise}$$

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

Model formulation: objective

- Structured matrix factorization

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

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

Greedy algorithm

- Identify ROIs one at a time, using the residual un-explained by existing signals.
- At iteration k, given the current residue (unexplained by existing ROIs)
  - 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 signal

Shape fine tuning

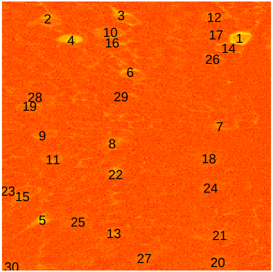
- Given current residue R, an identified center pixel p_k, denote S_k as a $w \times w$ window centered at p_k

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

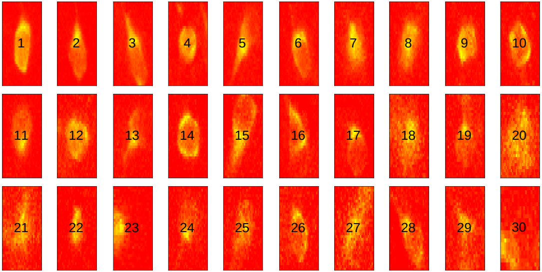
- $f(D_k) = \sum_{i=1}^{3} \lambda_i f_i(D_k)$
  - $f_1(D_k) = \sum_p \tau_{(p,p_k)} |D_{kp}|$ encourages sparsity
  - $f_2(D_k) = \sum_p (D_{kp} - G_{p_k})^2$ encourages Gaussian shape
  - $f_3(D_k) = \sum_{p_1 \text{ and } p_2 \text{ are neighbors}} (D_{kp_1} - D_{kp_2})^2$ encourages smoothness
-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

Real data analysis: sample patch, no shape regularization

Mean image with fitted ROI locations

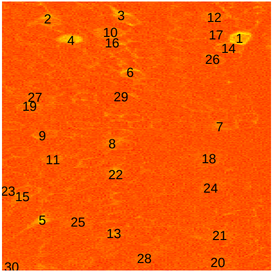


Fitted ROI shape

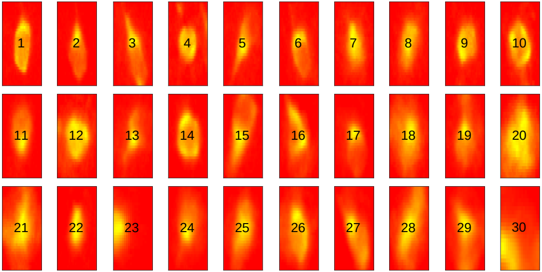


Real data analysis: sample patch, shape regularization

Mean image with fitted ROI locations



Fitted ROI shape



Conclusion and discussion

- Summary
    - Formulated calcium imaging ROI detection as a structured matrix factorization problem.
    - Developed a fast greedy algorithm.
- Future work
    - More spatial and temporal structure.
    - Overlapping neuron.
    - Online ROI detection.
    - Motion correction and background elimination.

- Pnevmatikakis EA, Soudry D, Gao Y, Machado TA, Merel J, Pfau D, Reardon T, Mu Y, Lacefield C, Yang W, Ahrens M, Bruno R, Jessell TM, Peterka DS, Yuste R, Paninski L (2016) Simultaneous denoising, deconvolution, and demixing of calcium imaging data. Neuron, 89(2), 285-299.

Table of Contents

Maximum entropy principle

- Entropy: for a continuous distribution with density $p(\mathbf{z})$ where $\mathbf{z} \in \mathbb{R}^d$, the entropy is defined as

$$H(p) = -\int p(\mathbf{z}) \log p(\mathbf{z}) d\mathbf{z} = E_{\mathbf{Z} \sim p} \left[-\log p(\mathbf{Z}) \right].$$

- A popular measure of diversity or information content.
- Maximum entropy (ME) principle: Subject to some given prior knowledge, the distribution that makes minimal additional assumptions is that which has the largest entropy of any distribution obeying those constraints

Applications of maximum entropy

- Neuroscience: generate a distribution of neural activity by specifying a set of features (pairwise correlation etc.) for hypothesis testing.
- Texture modeling: generate an image with a certain texture by specifying expected value of features relevant to texture.
- Ecology, natural language processing, finance...

Maximum entropy problem

- Maximum entropy (ME) problem

$$\begin{aligned} p^* \quad = \quad & \underset{p}{\text{maximize}} && H(p) \\ & \text{subject to} && E_{\mathbf{Z}\sim p}[T(\mathbf{Z})] = 0 \\ & && \text{supp}(p) = \mathcal{Z}, \end{aligned}$$

where $T(\mathbf{z}) = (T_1(\mathbf{z}), ..., T_m(\mathbf{z})) : \mathcal{Z} \to \mathbb{R}^m$ is the vector of known statistics, and $\mathcal{Z}$ is the given support.

Gibbs distribution

- Lagrange multipliers argument $\Rightarrow$ an exponential family form for ME distribution (Gibbs distribution):

$$p^*(\mathbf{z}) \propto e^{<\eta, T(\mathbf{z})>} \mathbb{1}(\mathbf{z} \in \mathcal{Z})$$

- Hard to Identify $\eta \in \mathbb{R}^m$.
- Hard to sample from the distribution.
- Question: is there a better way to do this?

Maximum entropy flow network (MEFN)

- Identify a transformation $f_{\phi^*} \in \mathcal{F}$ that transforms a simple distribution $\mathbf{Z}_0 \sim p_0$ (Gaussian) to a complicated one $\mathbf{Z} = f_{\phi^*}(\mathbf{Z}_0) \sim p_{\phi^*}$ that approximates the ME distribution.

$$
\begin{aligned}
p^* \quad = \quad & \underset{p}{\text{maximize}} && H(p) \\
& \text{subject to} && E_{\mathbf{Z}\sim p}[T(\mathbf{Z})] = 0 \\
& && \text{supp}(p) = \mathcal{Z},
\end{aligned}
$$

$$\Downarrow$$

$$
\begin{aligned}
\phi^* \quad = \quad & \underset{\phi}{\text{maximize}} && H(p_\phi) \\
& \text{subject to} && R(\phi) = E_{\mathbf{Z}_0\sim p_0}[T(f_\phi(\mathbf{Z}_0))] = 0 \\
& && \text{supp}(p_\phi) = \mathcal{Z}.
\end{aligned}
$$

where p_ϕ is the distribution of $\mathbf{Z} = f_\phi(\mathbf{Z}_0)$ for $\mathbf{Z}_0 \sim p_0$.

Normalizing flow

- Normalizing flow: A parameterized family of smooth and invertible transformation

$$\mathcal{F} = \{f_\phi : \mathbb{R}^d \to \mathbb{R}^d, \phi \in \mathbb{R}^q\}.$$

- Explicit probability density form for $Z = f_\phi(Z_0) \sim p_\phi$ by change-of-variable theorem

$$p_\phi(\mathbf{z}) = p_0\left(f_\phi^{-1}(\mathbf{z})\right) |\det(J_\phi(\mathbf{z}))|^{-1}.$$

- Constructing a flexible family of normalizing flows by composing simple normalizing flows (deep learning)

$$\mathbf{Z} = f_k \circ f_{k-1} \circ \cdots \circ f_1(\mathbf{Z}_0).$$

Augmented Lagrangian method

- Augmented Lagrangian method minimizes the objective

$$L(\phi; \lambda, c) = -H(p_\phi) + \lambda^\top R(\phi) + \frac{c}{2}||R(\phi)||^2.$$

  for a sequence of $\lambda \in \mathbb{R}^m$ and $c \geq 0$.
- Update rule: at iteration k, given λ_k and c_k.
  - Find ϕ_k that optimizes $L(\phi; \lambda_k, c_k)$.
  - Update λ and c by

$$\begin{aligned} \lambda_{k+1} =& \lambda_k + c_k R(\phi_k) \\ c_{k+1} =& \begin{cases} \beta c_k & ||R(\phi_k)|| > \gamma ||R(\phi_{k-1})|| \\ c_k & \text{otherwise} \end{cases} \end{aligned}$$

    for some $\gamma \in (0, 1)$, $\beta > 1$

Augmented Lagrangian method in stochastic setting

- Both $H(p_\phi)$ and $R(\phi)$ are intractable, but we can approximate with an unbiased estimation,

$$
\begin{aligned}
H(p_\phi) &\approx \frac{1}{n}\sum_{i=1}^{n} - \log p_\phi(\mathbf{z}^{(i)}), \\
R(\phi) &\approx \frac{1}{n}\sum_{i=1}^{n} T(f_\phi(\mathbf{z}^{(i)})),
\end{aligned}
$$

where $\mathbf{z}^{(i)} \sim p_0$. We can then optimize the objective by stochastic gradient descent.

Texture modeling (overview)

- Goal: construct a distribution on images that mimics a given texture (from a training image).
  - Authenticity: the samples from the distribution should mimic the given texture (constraints).
  - Sample diversity: the distribution should generate a variety of images with certain texture (entropy).
- Natural application of maximum entropy modeling.

Texture modeling (formulation)

- Sample from the space of images $\mathbf{z} \in [0,1]^{d=w\times h\times c}$ where $w \times h$ is the image size, c is the number of channels ($c = 3$ for RGB representation).
- Texture loss: $T(\mathbf{z}) : [0,1]^{w\times h\times c} \to \mathbb{R}$, a complicated form proposed in Ulyanov et al. (2016).
- Ulyanov et al. (2016) proposes texture net, which solves the problem without considering entropy term.

$$\min E_{\mathbf{Z}\sim p_0}\left[T(f_\phi(\mathbf{Z}))\right]$$

- We build MEFN using $T(\mathbf{z})$ as the expectation constraint.
- We use real-nvp (Dinh, Sohl-Dickstein, and Bengio 2016) as the normalizing flow structure.

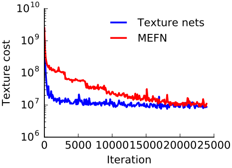
Experiment: Texture modeling (result)

Input

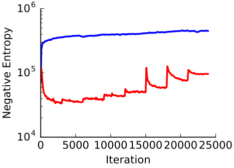
Texture net
(Ulyanov et al. 2016)

MEFN (ours)

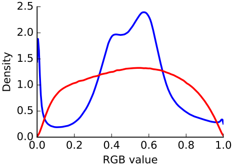
Texture loss



Negative entropy



RGB histogram



Experiment: Texture modeling (diversity measure)

Method	d_{L^2}	SST	SSW	SSB
Texture net	0.077	0.043	0.036	0.006
MEFN	0.113	0.058	0.054	0.005

- For $n = 20$ randomly sampled image $\mathbf{z}^{(1)}, ..., \mathbf{z}^{(20)}$
- $d_{L^2} = \text{mean}_{i \neq j} \frac{1}{d} \|\mathbf{z}^{(i)} - \mathbf{z}^{(j)}\|_2^2$: Mean Euclidean distance.
- ANOVA: SST $=$ SSW $+$ SSB, $\bar{\mathbf{z}} = \frac{1}{n} \sum_i \mathbf{z}^{(i)}$.
  - SST $= \frac{1}{nd} \sum_{i,k} (z_k^{(i)} - \bar{z})^2$: Total var.
  - SSW $= \frac{1}{nd} \sum_{i,k} (z_k^{(i)} - \bar{z}_k)^2$: residue var, larger $\Rightarrow$ better.
  - SSB $= \frac{1}{n} \sum_k (\bar{z}_k - \bar{z})^2$: mean image var, smaller $\Rightarrow$ better.

Conclusion and discussion

- Summary
  - Solved maximum entropy problem by optimizing a normalizing flow.
  - Combined augmented Lagrangian optimization with stochastic optimization.
  - Obtained promising texture modeling result.
- Future work
  - Better normalizing flow structure
  - Better constrained stochastic optimization algorithm
- Loaiza G*, Gao Y*, Cunningham JP (2017) Maximum entropy flow networks. ICLR 2017. (*=equal contribution)

Acknowledgement

Thank you!

Table of Contents

End of the talk.

Questions?

Backup slides

GCLDS: Supervised case, GCGLM

- Given count data $x_i \in \mathbb{N}$, and associated covariates $z_i \in \mathbb{R}^p$, build Generalized Count Generalized Linear Model (GCGLM).

$$x_i \sim \mathcal{GC}(\theta(z_i), g(\cdot)), \quad \text{where} \quad \theta(z_i) = z_i \beta.$$

- Reminder: generalized count distribution

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

GCLDS: special cases of GCGLM

Model Name	Typical Parameterization	GCGLM Parametrization
Logistic regression	$P(x=k) = \dfrac{\exp\left(k(\alpha+z\beta)\right)}{1+\exp(\alpha+x\beta)}$	$g(k) = \alpha k;\ k = 0, 1$
Poisson regression	$P(x=k) = \dfrac{\lambda^k}{k!}\exp(-\lambda);$ $\lambda = \exp(\alpha+z\beta)$	$g(k) = \alpha k$
Adjacent category regression	$\dfrac{P(x=k+1)}{P(x=k)} = \exp(\alpha_k + z\beta)$	$g(k) = \sum_{i=1}^{k}(\alpha_{i-1} + \log i);$ $k = 0, 1, ..., K$
Negative binomial regression	$P(x=k) = \dfrac{(k+r-1)!}{k!(r-1)!}(1-p)^r p^k$ $p = \exp(\alpha+z\beta)$	$g(k) = \alpha k + \log(k+r-1)!$
COM-Poisson regression	$P(x=k) = \dfrac{\lambda^k}{(k!)^\nu} / \sum_{j=1}^{+\infty}\dfrac{\lambda^j}{(j!)^\nu}$ $\lambda = \exp(\alpha+z\beta)$	$g(k) = \alpha k + (1-\nu)\log k!$

GCLDS: VBEM detail

MEFN: Normalizing flow structures

- Rezende and Mohamed (2015) Proposes two specific families of transformations for variational inference
- Planar flow

$$f_i(\mathbf{z}) = \mathbf{z} + \mathbf{u}_i h(\mathbf{w}_i^T \mathbf{z} + b_i),$$

where $b_i \in \mathbb{R}$, $\mathbf{u}_i, \mathbf{w}_i \in \mathbb{R}^d$ and h is an activation function.

- Circular flow

$$f_i(\mathbf{z}) = \mathbf{z} + \beta_i h(\alpha_i, r_i)(\mathbf{z} - \mathbf{z}_i'),$$

$\beta_i \in \mathbb{R}$, $\alpha_i > 0$, $\mathbf{z}_i' \in \mathbb{R}^d$, $h(\alpha, r) = 1/(\alpha + r)$ and $r_i = ||\mathbf{z} - \mathbf{z}_i'||$.

MEFN: real-nvp normalizing flow structure

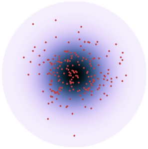
- Used in Dinh, Sohl-Dickstein, and Bengio (2016) for image denstiy estimation.
- Affine coupling layer: split variable $\mathbf{z} \in \mathbb{R}^D$ into $\mathbf{z}_1 \in \mathbb{R}^d$ and $\mathbf{z}_2 \in \mathbb{R}^{D-d}$. linearly transform $\mathbf{z}_2$ given $\mathbf{z}_1$.

$$f\begin{pmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \odot \exp\left(s(\mathbf{z}_1)\right) + t(\mathbf{z}_1) \end{pmatrix}$$

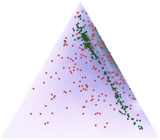
  where $\odot$ is element-wise product.
  $s(\mathbf{z}_1), t(\mathbf{z}_1) : \mathbb{R}^d \to \mathbb{R}^{D-d}$.
- The transformation family is flexible because
  - s and t can be complex while maintaining tractability (inversion and Jacobian computation).
  - Partitioning $\mathbf{z}$ into $(\mathbf{z}_1, \mathbf{z}_2)$ can be chosen arbitrarily.

Experiment: Dirichlet

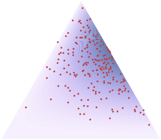
- Dirichlet distribution is the ME distribution on a simplex $\mathcal{S} = \{\mathbf{z} = (z_1, \ldots, z_{d-1}) : z_i \geq 0 \text{ and } \sum_{k=1}^{d-1} z_k \leq 1\}$ with expetation on the log of each coordinate $E[\log Z_k] = \kappa_k (k = 1, ..., d)$, where $Z_d = 1 - \sum_{k=1}^{d-1} Z_k$.
- 10 layers of planar flow (Rezende and Mohamed 2015).



Initial distribution p_0



MEFN result p_{ϕ^*} (Control case: moment matching)



Ground truth p^*