

Probabilistic models for neural data: From single neurons to population dynamics

NEUROBIO 316QC

Jan Drugowitsch

jan_drugowitsch@hms.harvard.edu

Session 4: Generalized linear models #2, graphical models & the EM algorithm

Today

Q&A about previous session

Paper discussion (~1h)

Generalized linear models, graphical models & the EM algorithm (~40min)

Overview

Generalized linear models recap

General linear models (GLMs) vs. generalized linear models (GLMs)

Decoding in GLMs

Conditional independence and graphical models

Parameter reduction through conditional independence

d-separation

Mixture models and the EM algorithm

Mixture models, with (equivalent) latent variable representation

Parameter inference in mixture models

The EM algorithm in general

Overview

Generalized linear models recap

General linear models (GLMs) vs. generalized linear models (GLMs)

Decoding in GLMs

Conditional independence and graphical models

Parameter reduction through conditional independence

d-separation

Mixture models and the EM algorithm

Mixture models, with (equivalent) latent variable representation

Parameter inference in mixture models

The EM algorithm in general

General vs. generalized linear models (stats/neuroscience)

General linear models (mostly fMRI)

Standard linear model, but for multiple “observations” $\mathbf{y} = y_1, \dots, y_{N_y}$

$$\mathbf{y} = \mathbf{W}\mathbf{x} + \boldsymbol{\eta}$$

Generalized linear models (statistics)

$$E[y|\mathbf{x}] = f(\mathbf{w}^T \mathbf{x})$$

Nonlinear due to activation function f , or link function f^{-1}

Canonical link function determined by likelihood $p(y|\mathbf{x}, \mathbf{w})$, allows use of IRLS algorithm

Generalized linear models (neuroscience)

Special case of GLMs (statistics) for inhomogeneous Poisson process likelihood

Doesn't necessarily (but often) use canonical activation function $f(x) = \exp(x)$

Includes past neural activity in \mathbf{x} (otherwise standard LNP model)

Using GLMs (neuroscience) for decoding

Assume $y_{1:N}$ (spike count in small time bins δt), and $\mathbf{x}_{1:N}$ (any past info, including spikes)

Spike train log-likelihood

$$\log p(y_{1:N} | \mathbf{x}_{1:N}, \mathbf{w}) = \sum_n \log \text{Pois}(y_n | f(\mathbf{w}^T \mathbf{x}_n) \delta t) = \sum_n y_n \log f(\mathbf{w}^T \mathbf{x}_n) - \delta t \sum_n f(\mathbf{w}^T \mathbf{x}_n) + \text{const.}$$

Concave in \mathbf{w} (and \mathbf{x}_n) if $f(\cdot)$ convex and log-concave, e.g., ReLU, exp, softReLU

Using GLMs for decoding \mathbf{x} for given y (for estimated $\hat{\mathbf{w}}$):

$$\hat{\mathbf{x}}_{MAP} = \text{argmax}_{\mathbf{x}} (\log p(y | \mathbf{x}, \hat{\mathbf{w}}) + \log p(\mathbf{x}))$$

Concave in \mathbf{x} for above $f(\cdot)$'s and if $\log p(\mathbf{x})$ concave (e.g., Gaussian)

Note: works better for decoding from populations, \mathbf{y}_n is spike count vector

See Park, Meister, Huk & Pillow (2014) for details / example

Decoding tutorial at https://github.com/pillowlab/GLMspiketraintutorial_python

Overview

Generalized linear models recap

General linear models (GLMs) vs. generalized linear models (GLMs)

Decoding in GLMs

Conditional independence and graphical models

Parameter reduction through conditional independence

d-separation

Mixture models and the EM algorithm

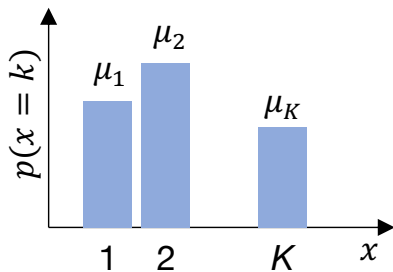
Mixture models, with (equivalent) latent variable representation

Parameter inference in mixture models

The EM algorithm in general




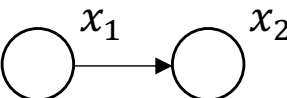
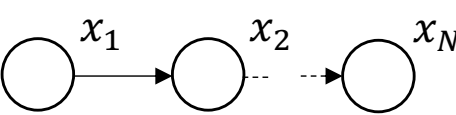
Conditional independence reduces parameters #1

Assuming discrete x_n 's



Discrete random variable $p(x = k) = \mu_k, k = 1, \dots, K$

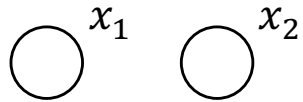
Requires $K - 1$ parameters, as $\sum_k \mu_k = 1$

	factorization	#parameters	
	$p(x_1)$	$K - 1$	
	$p(x_1, x_2)$	$K^2 - 1$	
	$p(x_{1:N})$	$K^N - 1$	$O(K^N)$ in N
	$p(x_2 x_1)p(x_1)$	$(K - 1)K + K - 1 = K^2 - 1$	
	$p(x_N x_{N-1}) \dots p(x_2 x_1)p(x_1)$ for $p(x_n x_{n-1})$ indep. of n :	$(N - 1)(K - 1)K + K - 1$ $(K - 1)K + K - 1$	$O(K^2N)$ in N $O(K^2)$

Conditional independence reduces parameters #2

Assuming Gaussian x_n 's

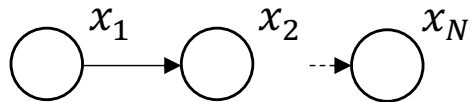
Independence



$$p \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = N \left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \middle| \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \Sigma_{11} & 0 \\ 0 & \Sigma_{22} \end{pmatrix} \right)$$

independence
↙

Markov chain



$$p \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = N \left(\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \middle| \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_N \end{pmatrix}, \Sigma \right) \propto \exp \left(-\frac{1}{2} \Delta^T \Sigma^{-1} \Delta \right)$$

full covariance matrix
↙

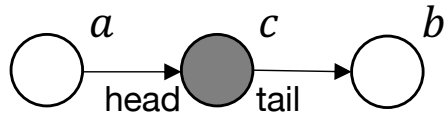
$$\Delta = \begin{pmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \\ \vdots \\ x_N - \mu_N \end{pmatrix}$$

Precision matrix Λ

$$\Sigma^{-1} = \Lambda = \begin{pmatrix} \Lambda_{11} & \Lambda_{12} & 0 & & 0 \\ \Lambda_{12} & \Lambda_{22} & \Lambda_{23} & & 0 \\ 0 & \Lambda_{23} & \Lambda_{33} & \ddots & 0 \\ & & \ddots & \ddots & \Lambda_{N-1,N} \\ 0 & 0 & 0 & \Lambda_{N-1,N} & \Lambda_N \end{pmatrix}$$

reveals Markov structure (coupling consecutive x_n 's)

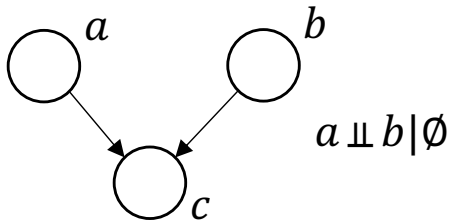
Conditional independence & d-separation



independent

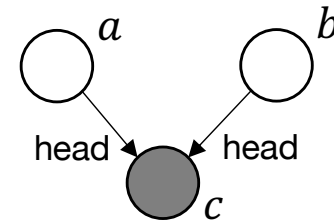
$$a \perp\!\!\!\perp b|c \quad a \not\perp\!\!\!\perp b|\emptyset$$

a and b are ...
“head-to-tail”
of c

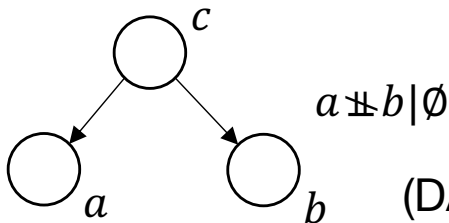


e.g., a image orientation
 b image contrast
 c neural response

$$a \not\perp\!\!\!\perp b|c$$



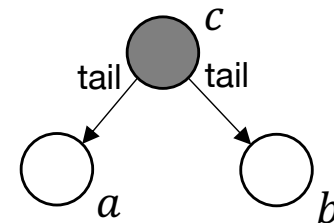
“head-to-head”



e.g., a activity neuron 1
 b activity neuron 2
 c image orientation

(DAG assumes no noise correlations)

$$a \perp\!\!\!\perp b|c$$



“tail-to-tail”

d-separation (Pearl, 1988)

$A \perp\!\!\!\perp B|C$ $A \rightarrow B$ blocked if h-t or t-t of node in set C ,
or if h-h of node, and neither nodes nor descendants in C

Overview

Generalized linear models recap

General linear models (GLMs) vs. generalized linear models (GLMs)

Decoding in GLMs

Conditional independence and graphical models

Parameter reduction through conditional independence

d-separation

Mixture models and the EM algorithm

Mixture models, with (equivalent) latent variable representation

Parameter inference in mixture models

The EM algorithm in general

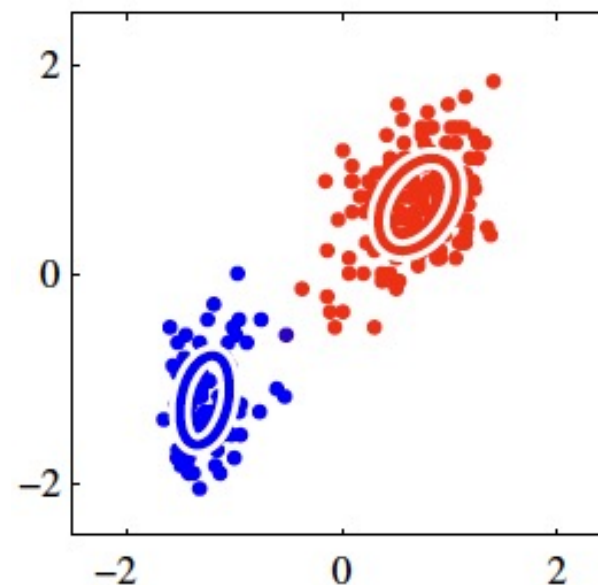
Mixture models

Weighted sum of probability distributions

$$p(x|\boldsymbol{\theta}) = \sum_k \pi_k p(x|\boldsymbol{\theta}_k) \quad \sum_k \pi_k = 1$$

mixture weights likelihood of k th mixture component

Most popular: Gaussian mixture, $p(x|\boldsymbol{\theta}_k) = \mathcal{N}(x|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$



PRLM, Fig. 9.8(f)

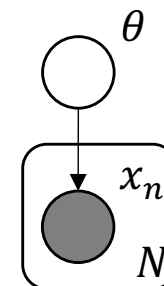
“Direct” ML parameter estimates

N data points, $x_{1:N}$: $p(x_{1:N}|\boldsymbol{\theta}) = \prod_n \sum_k \pi_k \mathcal{N}(x_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$

ML estimate:

$$\nabla_{\theta_k} \log p(x_{1:N}|\boldsymbol{\theta}) = \sum_n \frac{\pi_k \mathcal{N}(x_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(x_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \nabla_{\theta_k} \log \mathcal{N}(x_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = 0$$

no closed-form solution for $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$



(Equivalent) formulation with latent variables

For each datum x , define **“one-hot” latent binary vector** $\mathbf{z} = (z_1, \dots, z_K)^T$ with $z_k = 1$ and $z_{j \neq k} = 0$ otherwise

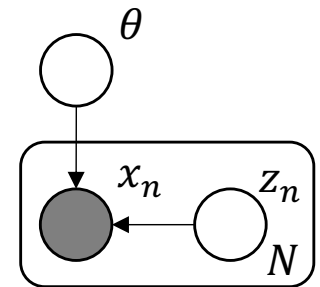
“ $z_k = 1$ ” means “data generated by mixture component k ”

$$p(z_k = 1 | \boldsymbol{\theta}) = \pi_k \quad \text{or (equivalently)} \quad p(\mathbf{z} | \boldsymbol{\theta}) = \prod_k \pi_k^{z_k}$$

Update data likelihood

$$p(x | \boldsymbol{\theta}, \mathbf{z}) = \underbrace{\prod_k p(x | \theta_k)^{z_k}}_{\text{data “generated” by mixture component for which } z_k = 1} \quad \text{and} \quad p(\mathbf{x}, \mathbf{z} | \boldsymbol{\theta}) = \prod_k \pi_k^{z_k} p(x | \theta_k)^{z_k}$$

data “generated” by mixture component
for which $z_k = 1$



Mixture component posterior

$$\gamma(z_k) \equiv p(z_k = 1 | \mathbf{x}, \boldsymbol{\theta}) = \frac{\pi_k p(\mathbf{x} | \boldsymbol{\theta}_k)}{\sum_j \pi_j p(\mathbf{x} | \boldsymbol{\theta}_j)}$$

Recover original formulation by $p(\mathbf{x} | \boldsymbol{\theta}) = \sum_{\mathbf{z}} p(\mathbf{x} | \mathbf{z}, \boldsymbol{\theta}) p(\mathbf{z} | \boldsymbol{\theta}) = \sum_k \pi_k p(\mathbf{x} | \boldsymbol{\theta}_k)$

Parameter inference in mixture models

Assume Gaussian mixture: $p(\mathbf{x}_{1:N}|\boldsymbol{\theta}) = \prod_n \sum_k \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$

no closed-form
solution for $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$

ML estimate:

$$\nabla_{\theta_k} \log p(\mathbf{x}_{1:N}|\boldsymbol{\theta}) = \sum_n \underbrace{\frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}}_{\text{"responsibility" } \gamma(z_{nk})} \nabla_{\theta_k} \log \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = 0$$

"responsibility" $\gamma(z_{nk})$
of component k for datum \mathbf{x}_n

$$\gamma(z_{nk}) = p(z_k = 1|\mathbf{x}_n, \boldsymbol{\theta})$$

1. Assume fixed responsibilities (ignore $\boldsymbol{\theta}_k$ -dependencies of $\gamma(\cdot)$)

reliability-weighted
empirical estimates

$$\hat{\boldsymbol{\mu}}_{k,ML} = \frac{1}{N_k} \sum_n \gamma(z_{nk}) \mathbf{x}_n \quad \text{with } N_k = \sum_n \gamma(z_{nk})$$

$$\hat{\boldsymbol{\Sigma}}_{k,ML} = \frac{1}{N_k} \sum_n \gamma(z_{nk}) (\mathbf{x}_n - \hat{\boldsymbol{\mu}}_{k,ML})(\mathbf{x}_n - \hat{\boldsymbol{\mu}}_{k,ML})^T$$

$$\pi_k = \frac{N_k}{N}$$

iterate
until
convergence

2. Update responsibilities (evaluating expression for $\gamma(z_{nk})$)

The expectation maximization algorithm

Assume “incomplete” data log-likelihood $\log p(\mathbf{X}|\boldsymbol{\theta})$ has complex form
e.g., log of sum, as in mixture model

“complete” data log-likelihood $\log p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$ has “nicer” form
e.g., $\log p(\mathbf{x}_n, \mathbf{z}_n|\boldsymbol{\theta}_k) = \sum_k z_{nk} (\log \pi_k + \log p(\mathbf{x}|\boldsymbol{\theta}_k))$ in mixture model

can compute conditional $p(\mathbf{z}_{1:N}|\mathbf{x}_{1:N}, \boldsymbol{\theta})$
e.g. responsibilities in mixture model

EM algorithm: iterate over

for mixture models

E-step evaluate $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{old})$

compute responsibilities

M-step $\hat{\boldsymbol{\theta}}^{new} = \operatorname{argmax}_{\boldsymbol{\theta}} E_{p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{old})} [\log p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})]$
set $\boldsymbol{\theta}^{old} \leftarrow \hat{\boldsymbol{\theta}}^{new}$

find $\hat{\boldsymbol{\mu}}_{ML,k}, \hat{\boldsymbol{\Sigma}}_{ML,k}, \pi_k$'s

Difference to standard ML

$$\hat{\boldsymbol{\theta}}_{ML} = \operatorname{argmax}_{\boldsymbol{\theta}} E_{p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta})} [p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})] = \operatorname{argmax}_{\boldsymbol{\theta}} p(\mathbf{X}|\boldsymbol{\theta})$$

↑
no log(·)!

Beware: EM algorithm prone to getting stuck in local minima! Run with multiple $\boldsymbol{\theta}^{ini}$

Overview

Generalized linear models recap

General linear models (GLMs) vs. generalized linear models (GLMs)

Decoding in GLMs

Conditional independence and graphical models

Parameter reduction through conditional independence

d-separation

Mixture models and the EM algorithm

Mixture models, with (equivalent) latent variable representation

Parameter inference in mixture models

The EM algorithm in general

Summary

3 types of GLMs: 1x General linear models, 2x Generalized linear models)

Conditional independence reduces #parameters (easier inference)

d-separation is formal way to determine conditional independence

Mixture models: weighted sum of mixture components

Inference becomes easier with latent variable assigning each datum to component

EM algorithm: general approach for ML estimation in latent variable models

Until next week

Complete GLM exercise (see notes for Session 4)

Read paper and prepare presentation (see notes for Session 5)

Read statistical methods sections (see notes for Session 5)

Next session

Discussing assignment (~15-25min)

Paper discussions (~1h)

Introducing dimensionality reduction (~30min)

