FACTOR ANALYSIS

Roweis & Gharamani 1999 Neural Comp

Takes axes as special
N neurons N-D activity
neuron axes special

FA assumes variance is
(1) components private to each axis
(2) shared/correlated component

Data X $\quad N \times T$

N: # neurons $\quad T$: # time points

$\underline{x}$ N-D vector at a given time

"standardize" $\quad \underline{x} \leftarrow \dfrac{\underline{x} - \underline{\mu}}{\sigma} \qquad \underline{\mu} = \langle \underline{x} \rangle \quad \sigma^2 = \langle (x-\mu)^2 \rangle$

Assume k shared factors

$L \quad N \times k$ "loadings"

$F \quad k \times T$ "factors" $\quad FF^T = \mathbb{1}_k$

Model: $\boxed{X = LF + \underline{\epsilon}} \qquad \mathrm{cov}(\epsilon) = \begin{pmatrix} \times & & 0 \\ & \times & \\ 0 & & \times \end{pmatrix}$

$$= \sum_k \underset{\uparrow}{\underline{l}_k}\, \underline{f}_k^T + \underline{\epsilon}$$

$\underline{l}_k$: columns of L, N-D $\qquad \underline{f}_k$: rows of F, T-D

$(\underline{l}_k \underline{f}_k^T)_{ij}$

$(l_k)_i (f_k)_j$

model as Gaussian

w/covariance

$$XX^T = (LF+\epsilon)(LF+\epsilon)^T$$

$$= L\underbrace{FF^T}_{I}L^T + \epsilon\epsilon^T$$

$$= \overset{d\times k}{L}\,\overset{k\times N}{L^T} + \text{cov}(\epsilon)$$

need not be orthog — Diagonal

Compare to PCA PPCA

$$X = USV^T$$

$$XX^T = USV^TVSU^T = US^2U$$

$$\text{cov} = \underset{N\times k}{U}S^2\underset{k\times N}{U^T} + \eta \qquad \text{cov}(\eta) = \epsilon I$$

FA: $X = LF + \epsilon$

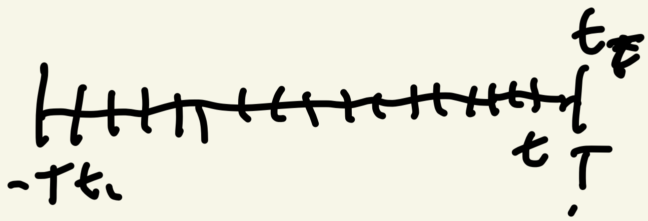
$L \to LQ$ $\quad F \to Q^T F$

$$\times\begin{pmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix}$$

PCA: rotate X

Gaussian Process Factor Analysis (GPFA)

GP:



Stochastic function $f(t) \to f(t_i) \equiv f_i$

$\underline{f}$ $\left(\frac{2T}{\Delta t}+1\right)$-D vector

$$\underline{f} \sim \frac{1}{\sqrt{2\pi \, \mathrm{Det}\, C}} e^{-\frac{1}{2}\underline{f}^T C^{-1} \underline{f}}$$

$\cancel{\sigma_1^2 \sigma_2^2 \cdots \sigma_N^2}$

$$(\underline{f}^T C^{-1} \underline{f}) = \sum_{ij} f_i C^{-1}_{ij} f_j$$

Let $\Delta t \to 0$ (let $T \to \infty$)

$$\sum_{ij} f_i C^{-1}_{ij} f_j \to \int dt\, dt'\, f(t)\, \underline{\underline{C^{-1}(t,t')}}\, f(t')$$

GP Big idea $\sim$ ∞-D Gaussian

Precise idea: Set of points (might be continuum)

s.t. the distribution of any

subset is a high-D Gaussian

$f(t)$ is G.P. if

$P(f(t_1), f(t_2), \ldots, f(t_N))$ is Gaussian for any t_i & any N

PCA, FA: typical approach

spike train: smooth (arbitrary timescale)

$\Rightarrow$ rates

then apply PCA, FA

GPFA: infer latent factors

& (multiple) timescales on which those change

Yu, Cunningham, ..., Shenoy, Sahani 2009 J Neurophys

Record N neurons T time pts

$y_{i,t}$: i^{th} neuron, t^{th} time pt

$x_{j,t}$: j^{th} latent factor, t^{th} " "

$$Y = \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ \underline{y}_{:,1} & \underline{y}_{:,2} & \cdots & \underline{y}_{:,T} \\ \downarrow & \downarrow & & \downarrow \end{pmatrix} \quad N \times T$$

$$X = \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ \underline{x}_{:,1} & \underline{x}_{:,2} & \cdots & \underline{x}_{:,T} \\ \downarrow & \downarrow & & \downarrow \end{pmatrix} \quad p \times T$$

Find $x_{:,i}$ by FA at each time pt:

$$P(y_{:,t} \mid x_{:,t}) = \mathcal{N}(L\,\underline{x}_{:,t} + \underline{d},\ \underset{\substack{\uparrow \\ \text{diagonal}}}{R})$$

Neural states at diff times
smoothed as GP

$$x_{i,:} \sim \mathcal{N}(\underline{0}, \underset{T \times T}{K_i})$$

T-D
$i: 1 \to p$

Cov: "squared exponential"

$$K_i(t_1, t_2) = \underset{\substack{\uparrow \\ \text{signal variance}}}{\sigma_{s,i}^2} \exp\left(-\frac{(t_1-t_2)^2}{\underset{\substack{\uparrow \\ \text{chsr. timescale}}}{2\tau_i^2}}\right)$$

$FF^T = \mathbb{1}$

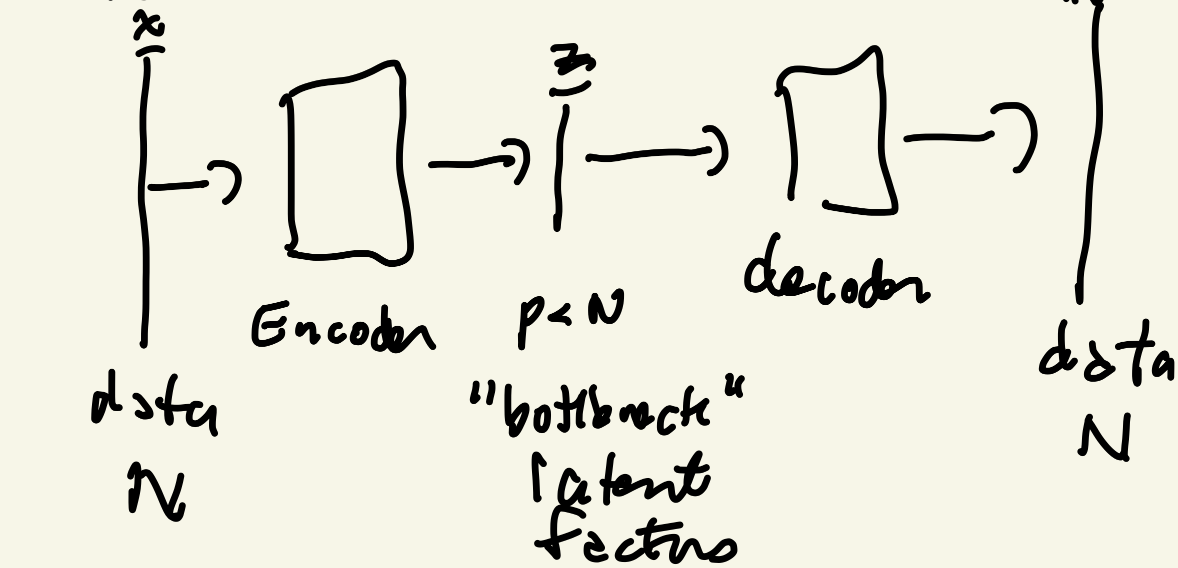
$$+ \underset{\substack{\uparrow \\ \text{noise variance}}}{\sigma_{ni}^2} \delta_{t_1 t_2}$$

$x_{:,t} \sim \mathcal{N}(0, \mathbb{1})$ $\quad K_i(t,t) \equiv 1$

$\sigma_{si}^2 + \sigma_{ni}^2 = 1$ $\quad \sigma_n \sim 10^{-3}$

enter τ_i

Autoencoder



PCA: linear autoencoder

$$\underset{\sim}{x} \Rightarrow \sum_{i=1}^{p} \underset{\sim}{e}_i \left(\underset{\sim}{e}_i^T \underset{\sim}{x}\right) \qquad \underset{\sim}{x} \text{ N-D}, \quad p \ll N$$

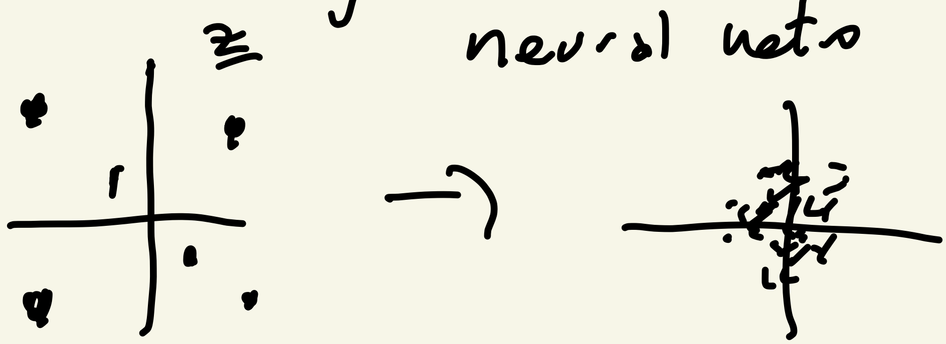
$$= E\, E^T \underset{\sim}{x} \qquad E = \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ e_1 & e_2 & \cdots & e_p \\ \downarrow & \downarrow & & \downarrow \end{pmatrix}$$

encoder $N \times p$ decoder $p \times N$

$$\underline{\min} \; |\underset{\sim}{x} - E E^T \underset{\sim}{x}|^2$$

More general (nonlinear)

neural nets



Variational Autoencoder: stochastic

$$\underline{z} \sim \mathcal{N}(0, \mathbb{I})$$

encoder outputs $\underline{\mu}$, $\underline{\sigma}^2$ of a Gaussian

$\underline{x}_i \to \underline{\mu}, \underline{\sigma}$ $\quad z_i \sim \mathcal{N}(\underline{\mu}, \underline{\sigma}_i^2)$

$q_\theta(\underline{z}_i | \underline{x}_i)$ encoder

θ: parameters

decoder $p_\phi(\underline{x} | \underline{z})$ $\quad \phi$: parameters

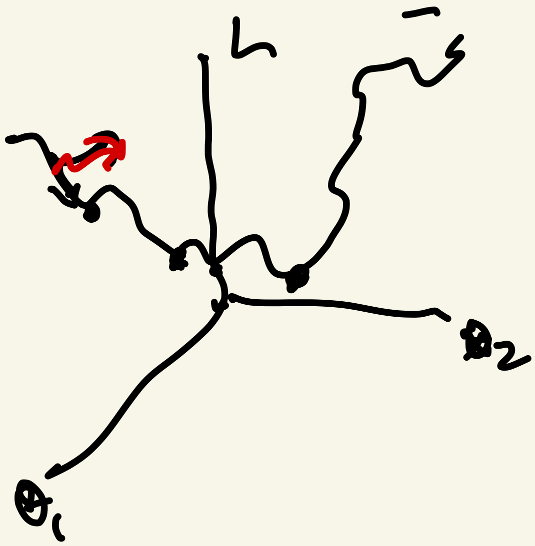
Loss Function:

$$l_i(\theta, \phi) = -E_{z \sim q_\theta(\underline{z}_i | \underline{x}_i)}\left[\log p_\phi(\underline{x}_i | \underline{z}_i)\right]$$

$$L = \sum_i l_i \qquad + KL\left[q_\theta(\underline{z} | \underline{x}_i) \,||\, \mathcal{N}(0, \mathbb{I})\right]$$

$\underline{\theta}$ $\quad \theta_i^{t+1} = \theta^t - \lambda \frac{\partial L}{\partial \theta_i}$

λ: learning rate



$$\nabla_\theta L = \left(\frac{\partial L}{\partial \theta_1}, \frac{\partial L}{\partial \theta_2}, \dots, \frac{\partial L}{\partial \theta_p}\right)^T$$

LFADS

Latent factor analysis via dynamical systems

Pandarinath, ..., Shenoy, Abbott, Sussillo
2018

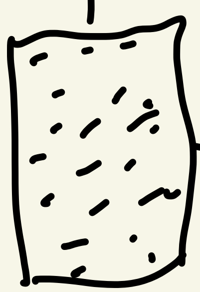
Basic idea: data $\underline{x}$ evolves by D.S.

$$\dot{\underline{x}}(t) = F\left(\underline{x}(t), \underline{u}(t)\right)$$

$\uparrow$ inputs

RNN learn dynamics

T

N 

spikes

$\rightarrow$ Encoder (RNN) $\rightarrow$ $\underline{\mu}$, $\underline{\sigma}^2$, $g_0 \sim N(\mu, \sigma^2)$ $\rightarrow$ Generator (Decoder) RNN $g(t)$

$g(t) = \text{RNN}(g(t-1), \underline{u}(t))$

Controller $\rightarrow$ Generator

Controller RNN $\rightarrow \underline{u}(t)$

factors $\underline{f}(t) = W^{fac}\, g(t)$

$\underline{f}$: F-D, W^{fac}: $F \times [\text{RNN}]$

Rate $\underline{r}(t) = \exp\left[W^{rate}\, \underline{f}(t)\right]$

W^{rate}: $N \times F$, $\underline{f}(t)$: F-D

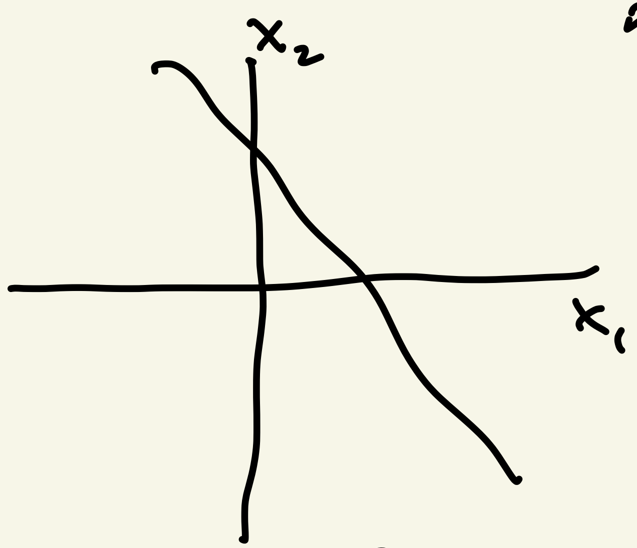
Spikes ~ Poisson(rates)

Low-D visualization (2 or 3 D) — t-SNE, umap

Constrained optimization
(Lagrange multipliers)

Opt: min $L(\underline{x})$ $\underline{x}$: N-D
subject to constraints
- could be inequalities
consider equalities
$f_i(\underline{x}) = k_i$, $i = 1, \ldots, C$
$\underline{f}(\underline{x}) = \underline{k}$

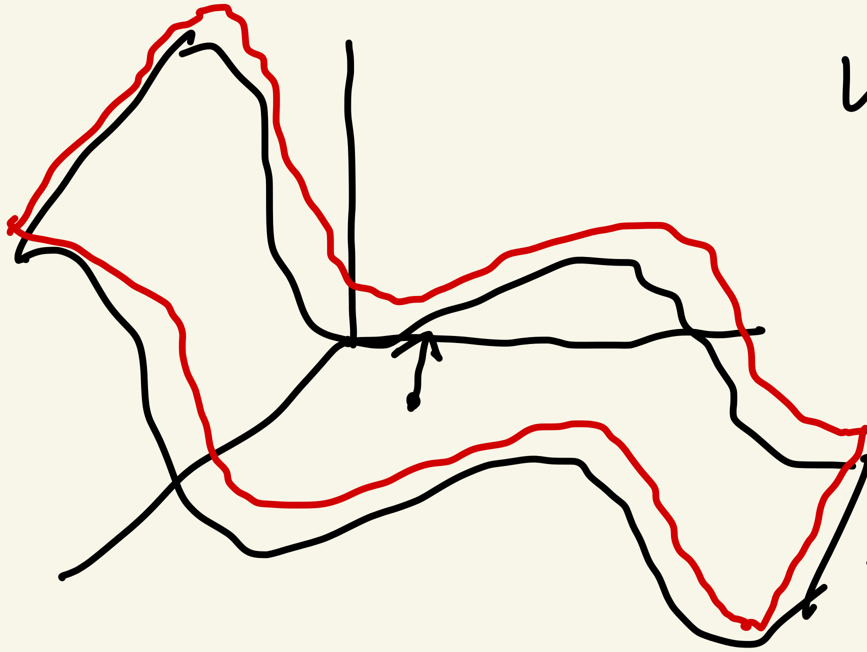
$ax_1 + bx_2 = c$



Magic formula (LaGrange mult)
form $\mathcal{L}(\underline{x}) = L(\underline{x}) + \sum_i \underbrace{\lambda_i}_{\text{L.M.'s}} f_i(\underline{x})$

set $\nabla_{\underline{x}} \mathcal{L}(\underline{x}) = 0$
$= \left(\frac{\partial \mathcal{L}}{\partial x_1}, \frac{\partial \mathcal{L}}{\partial x_2}, \ldots \frac{\partial \mathcal{L}}{\partial x_n}\right)$

Why?



$\to \underline{f} = \underline{k}_1 + \underline{e}$

$\underline{f} = \underline{k}_1$

Want min $L(\underline{x})$ on constraint surface

$$\nabla_x L = \left(\frac{\partial L}{\partial x_1}, \ldots \frac{\partial L}{\partial x_n}\right)^T \perp \text{C.S.}$$

Dir $\perp$ to C.S. = Dir of maximal change in $\underline{f}(x)$

$$= \nabla_x \underline{f}(x)$$

$$\cancel{\nabla_x L = 0} \qquad \nabla_x L \propto \nabla_x \underline{f}(\underline{x})$$

$$\triangleq \lambda_1 \nabla_x f_1(x) + \lambda_2 \nabla_x f_2(x) + \ldots + \lambda_N \nabla_x f_N(\underline{x})$$

Form $\mathcal{L}(\underline{x}) = L(\underline{x}) + \lambda_1 f_1(\underline{x}) + \lambda_2 f_2(\underline{x}) + \ldots$

$$\nabla_x \mathcal{L}(\underline{x}) = \nabla L(\underline{x}) + \lambda_1 \nabla f_1(\underline{x}) + \lambda_2 \nabla f_2(\underline{x}) + \ldots = 0$$

$$\nabla L(\underline{x}) = -\lambda_1 \nabla f_1(\underline{x}) - \lambda_2 \nabla f_2(\underline{x}) + \ldots$$

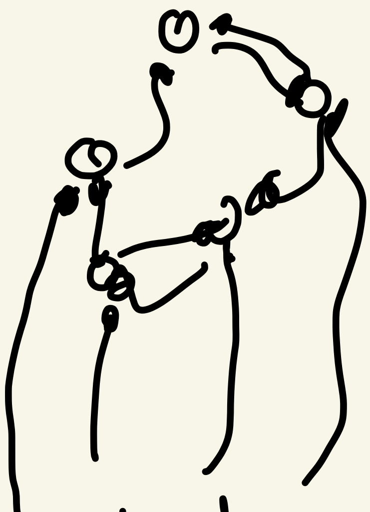
after minimization, set λ's to satisfy constraints

Rate networks

$$\underline{r}_{ss} = f\left(W\underline{r} + \underline{h}\right)$$

N-D
(N neurons)

$$r^i_{ss} = f_i\left(\sum_j W_{ij} r_j + h_i\right)$$



$$\tau_i \frac{dr_i}{dt} = -r_i + f_i\left(\sum_j W_{ij} r_j + h_i\right)$$

$$T \frac{d\underline{r}}{dt} = -\underline{r} + \underline{f}\left(\underbrace{W\underline{r} + \underline{h}}\right)$$

$$\begin{bmatrix}\tau_1 & & 0\\ & \ddots & \\ 0 & & \tau_N\end{bmatrix}$$

outside

Voltage picture $$T \frac{d\underline{V}}{dt} = -\underline{V} + \overbrace{W \underbrace{\underline{f}(V)}_{\underline{r}} + \underline{h}_r}$$

$$V = Wr + h_r$$

$$\tau \frac{\partial h_r}{\partial t} = -h_r + h_v$$

Miller &
Fumarola
2012