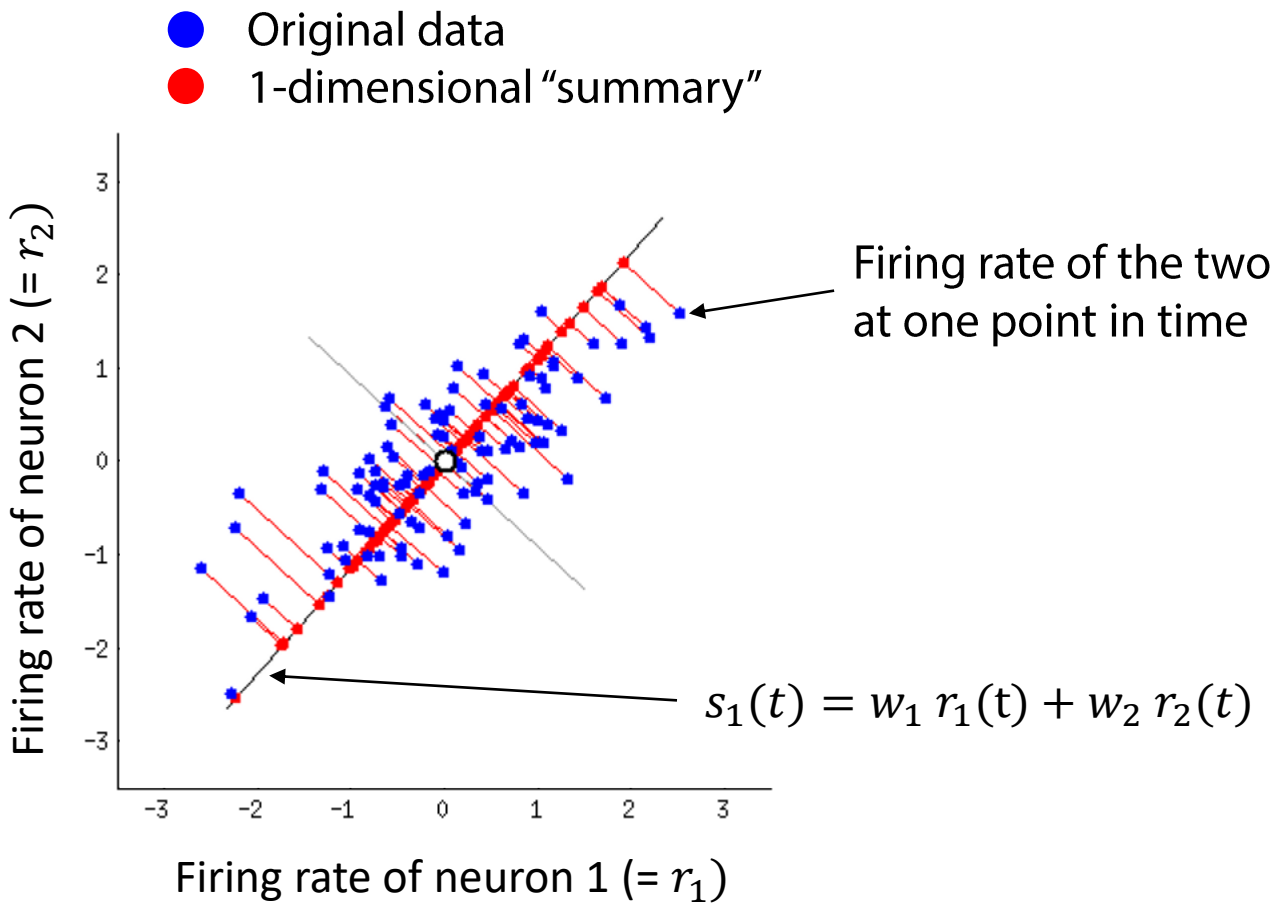
Dimensionality Reduction Part 2: Generative Models

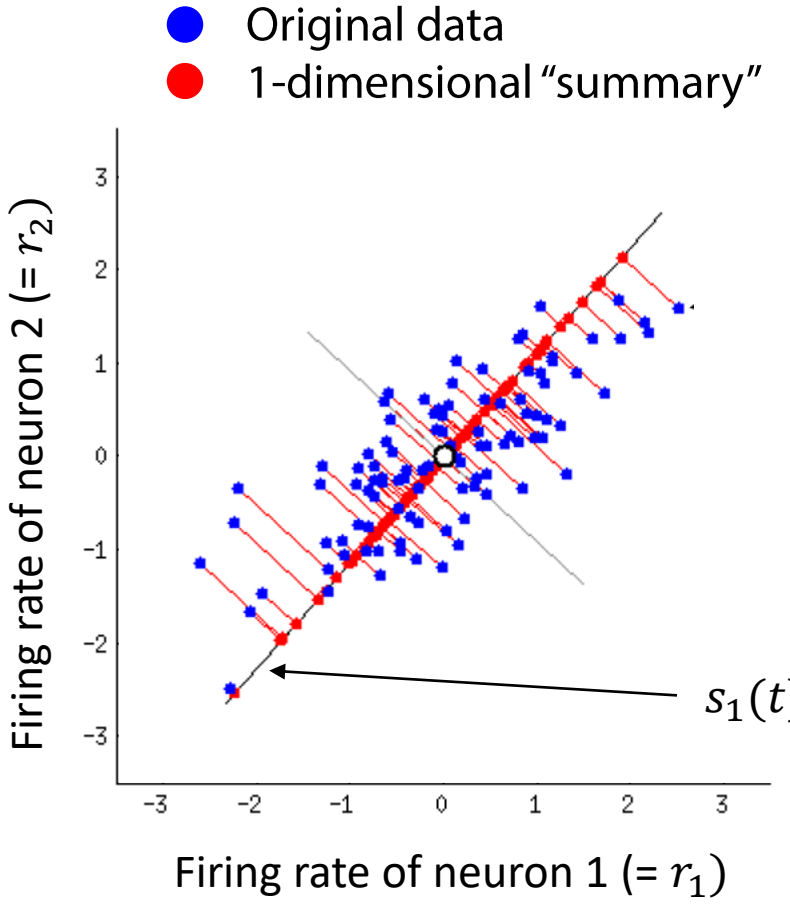
Bi23: Methods in Neural Data Analysis

3/8/2019

Last time: if we pick the right basis, we can reduce data dimensionality without losing much information



Last time: PCA finds the dimensions that capture the most variance in neural activity (ie the dimensions that lose the least information)



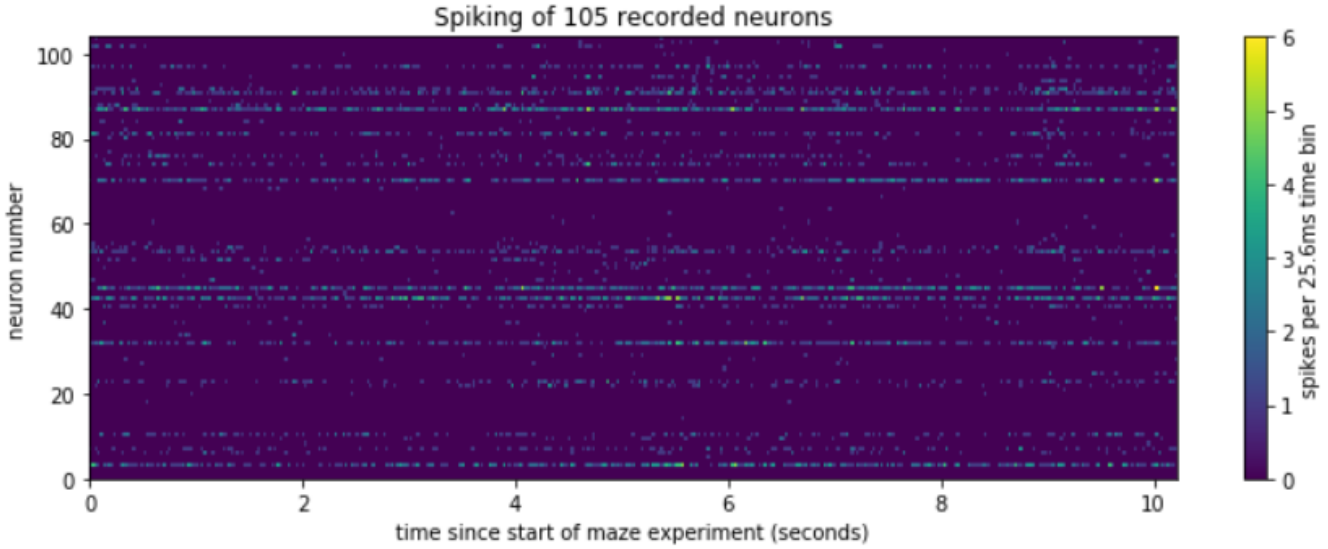
$$s_1(t) = w_1\, r_1(\mathrm{t}) + w_2\, r_2(t)$$

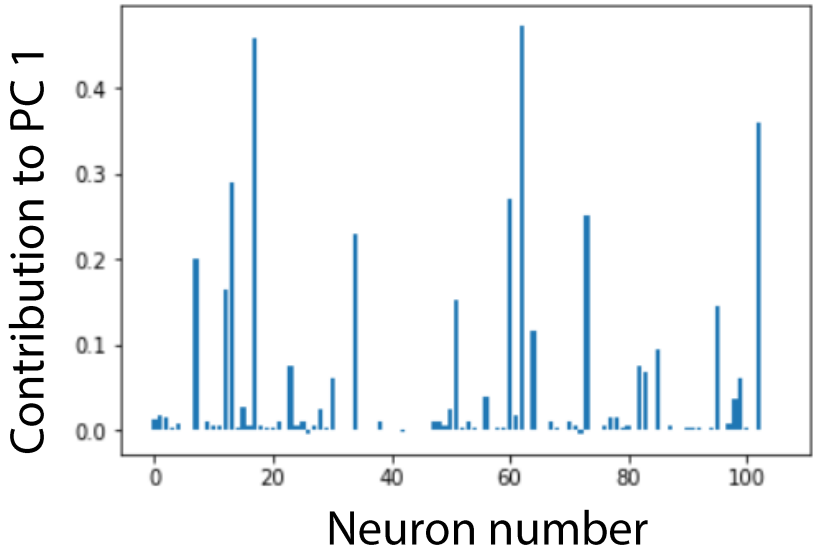
Our solution for w is an eigenvector of the **covariance matrix** Σ ,

$$\Sigma\, \boldsymbol{w}^T = \boldsymbol{\lambda}\, \boldsymbol{w}^T$$

Then the eigenvector $\boldsymbol{w_k}$ the maximizes $\boldsymbol{w_k} \Sigma \boldsymbol{w_k}^T$ is the one with the **LARGEST** eigenvalue $\boldsymbol{\lambda_k}$

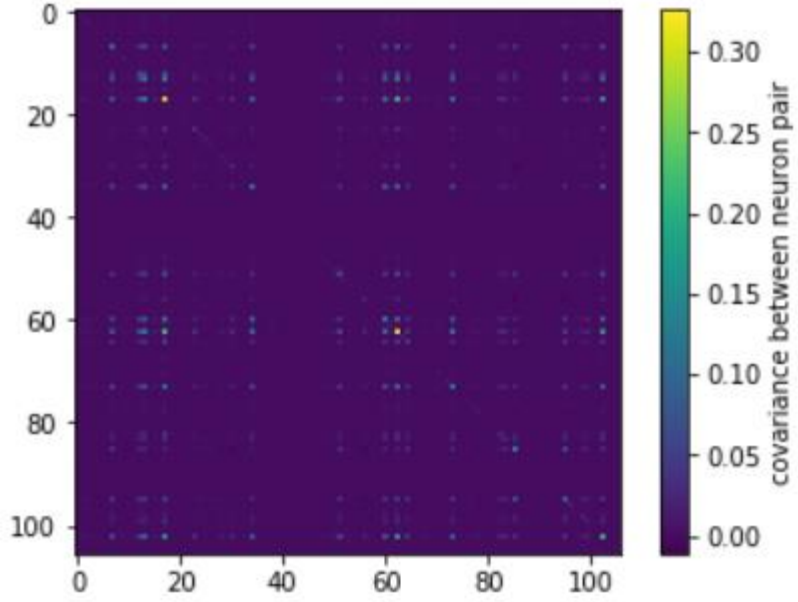
So: PCA is computing the *eigenvectors* of the *covariance matrix* (Σ)

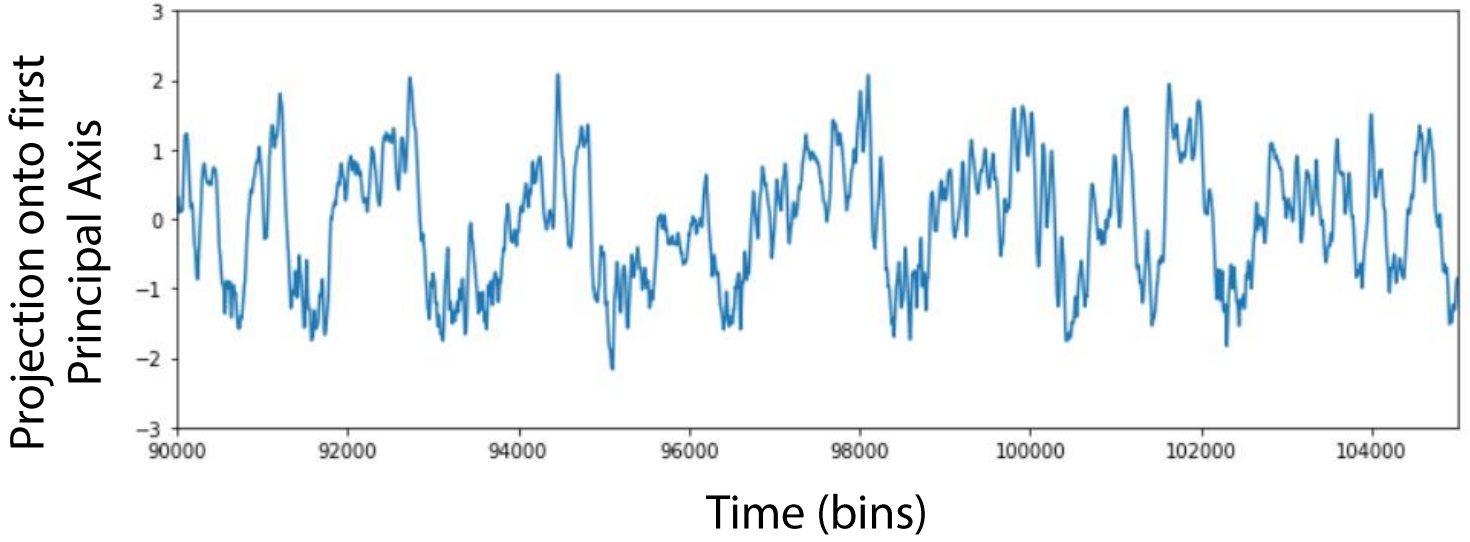




(aka **loadings**)

$$\Sigma = (\boldsymbol{r} - \bar{\boldsymbol{r}})(\boldsymbol{r} - \bar{\boldsymbol{r}})^T$$





A different perspective

Rather than projecting observed data into a low-dimensional space (eg with PCA), we want to construct a *hypothesis* about how the observed data was *generated*.

- Eg: the observed data (spikes) are noisy observations of a small number of underlying *latent variables*.

Generative models

A generative model describes the statistical process from which data arises.

- For example (GLM):

Underlying Spike rate

$$\mu = \theta x$$

unknown parameter

stimulus

Predicted spike distribution

$$y \sim N(\mu, \sigma^2)$$

Generative modeling: drawing a picture of a car
Discriminative modeling: distinguishing a car from a motorcycle

Some other types of generative models:

- Hidden Markov Model
- Gaussian Mixture Model
- Variational Autoencoder
- Generative Adversarial Network (GAN)

AI-generated faces created with GAN 2.0

A Generative Modeling Take on Dimensionality Reduction

We observe the spiking of a population of *N* neurons, $\boldsymbol{y(t)} = \{y_1(t), y_2(t), \ldots y_N(t)\}$.

A simple model might be to assume that the firing rate of each neuron y_n is a function of a small number *D* (D<<N) of latent variables, $\boldsymbol{z}$, as follows:

$$z_d(t) \sim \mathrm{N}(0,1)$$

$$y_n(t) \sim \mathrm{N}\left(\sum_{d=1}^{D} W_{n,d}\, z_d(t), \sigma^2\right)$$

Or, in matrix notation,

$$\mathbf{z}(t) \sim \mathrm{N}(0, \mathrm{I})$$

Identity matrix

$$\mathbf{y}(t) \sim \mathrm{N}(\boldsymbol{W} \cdot \boldsymbol{z}(t), \sigma^2 \mathrm{I})$$

"factor loading" matrix (D x N)

Or written another way:

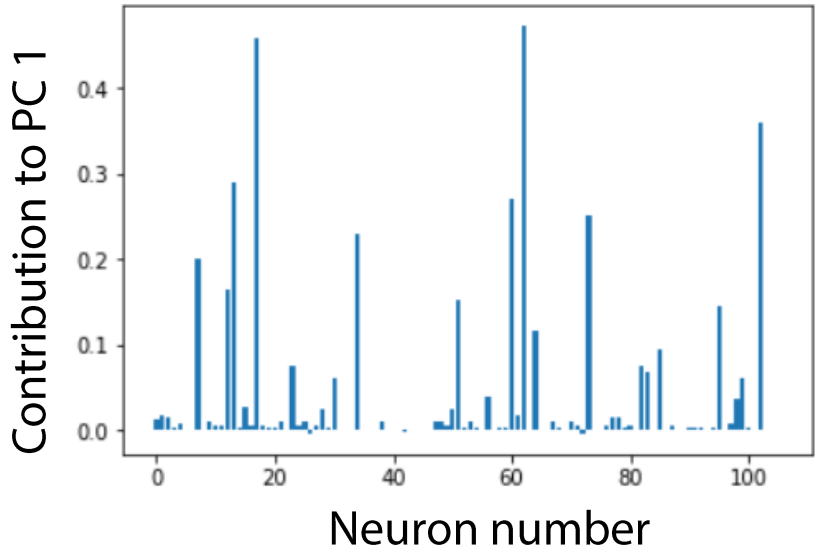
$$\mathbf{y}(t) = \boldsymbol{W} \cdot \boldsymbol{z}(t) + \epsilon, \qquad \epsilon \sim \mathrm{N}(0, \sigma^2 \mathrm{I})$$

This is "Probabilistic PCA"!

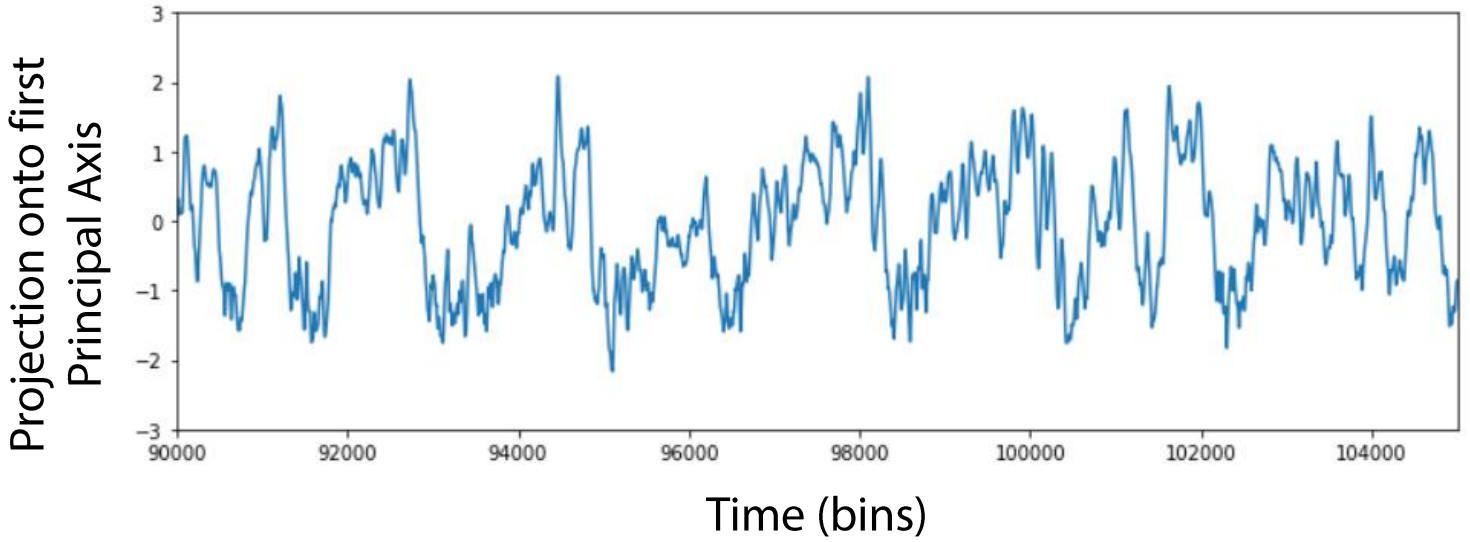
Reconciling Probabilistic with ordinary PCA

$$\mathbf{z}(t) \sim \mathrm{N}(0, \mathrm{I})$$

$$\mathbf{y}(t) = \boldsymbol{W} \cdot \boldsymbol{z}(t) + \epsilon, \qquad \epsilon \sim \mathrm{N}(0, \sigma^2 \mathrm{I})$$



(aka **loadings**)



We can make lots of models from this same general framework:

Probabilistic PCA:

$$\mathbf{z}(t) \sim \mathrm{N}(0, \mathrm{I})$$
$$\mathbf{y}(t) = \boldsymbol{W} \cdot \boldsymbol{z}(t) + \epsilon, \qquad \epsilon \sim \mathrm{N}(0, \sigma^2 \mathrm{I})$$

Factor Analysis:

$$\mathbf{z}(t) \sim \mathrm{N}(0, \mathrm{I})$$
$$\mathbf{y}(t) = \boldsymbol{W} \cdot \boldsymbol{z}(t) + \epsilon, \qquad \epsilon \sim \mathrm{N}(0, \Psi) \qquad \Psi = \begin{bmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_N^2 \end{bmatrix}$$

Linear Dynamical System:

$$\mathbf{z}(t) = \boldsymbol{A} \cdot \boldsymbol{z}(t-1) + v, \qquad v \sim \mathrm{N}(\mu, \sigma^2 \mathrm{I}_z)$$
$$\mathbf{y}(t) = \boldsymbol{W} \cdot \boldsymbol{z}(t) + b + \epsilon, \qquad \epsilon \sim \mathrm{N}(0, \sigma^2 \mathrm{I}_x)$$

Gaussian Process Factor Analysis:

$$\boldsymbol{z_i}(t) \sim \mathrm{N}(0, K_i) \qquad K_i(t_1, t_2) = \sigma_{f,i}^2 \exp\left(-\frac{(t_1 - t_2)^2}{2\tau_i^2}\right) + \sigma_{n,i}^2 \cdot \delta_{t_1, t_2}$$
$$\mathbf{y}(t) = \boldsymbol{W} \cdot \boldsymbol{z}(t) + b + \epsilon, \qquad \epsilon \sim \mathrm{N}(0, \Psi)$$

…LOTS of models:

Hidden Markov Model:

$$\mathbf{z}(t) \in \{1,2,\ldots D\} \qquad \text{(discrete states!)}$$

$$\mathbf{y}(t) \sim \mathrm{N}(\mu_z, \Sigma_z)$$

Independent Component Analysis:

$$\mathbf{z}(t) \sim \text{maximally non} - \text{Gaussian}$$

$$\mathbf{y}(t) \sim \mathrm{N}(z, \sigma_z^2\ \mathrm{I})$$

Switching Linear Dynamical System:

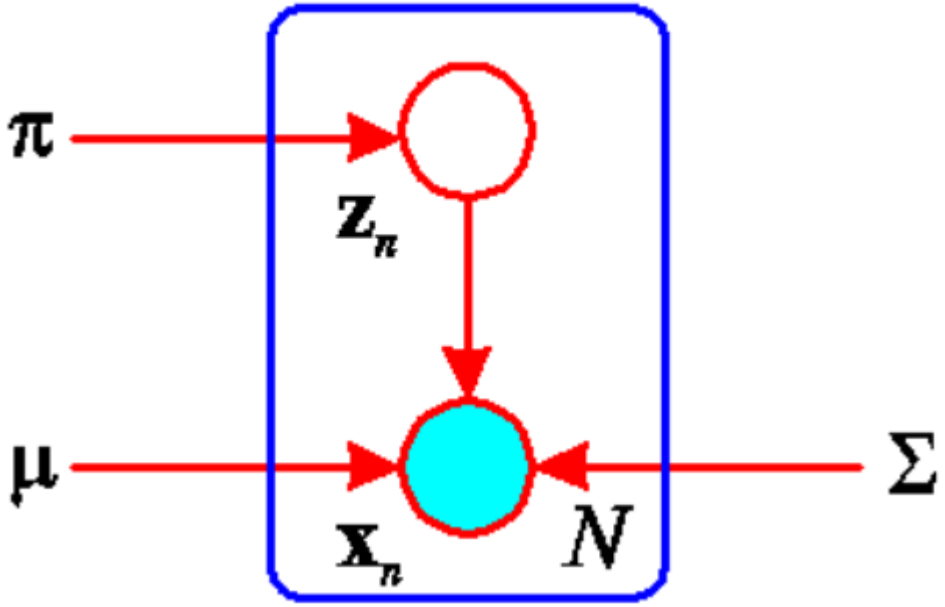
$$\pi(t) = \in \{1,2,\ldots D\}$$

$$\mathbf{z}(t) = \boldsymbol{A} \cdot \boldsymbol{z}(t-1) + v, \qquad v \sim \mathrm{N}(\mu_\pi, \sigma_\pi^2 \mathrm{I}_z)$$

$$\mathbf{y}(t) = \boldsymbol{W} \cdot \boldsymbol{z}(t) + b + \epsilon, \qquad \epsilon \sim \mathrm{N}(0, \sigma^2 \mathrm{I}_x)$$

Sometimes you'll see these models represented graphically:

Graphical representation of a Gaussian Mixture Model



Recall how we fit the GLM…

spikes (known)

Predicted spike distribution

$$y \sim N(z, \sigma^2)$$

model parameters (unknown)

Underlying Spike rate

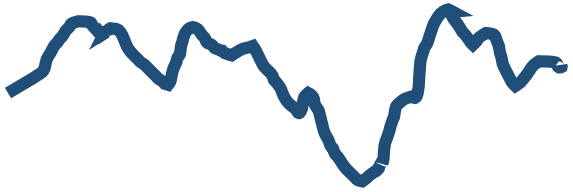
$$z = \theta x$$

stimulus feature (known)

$$P(y|x,\theta,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(y-\theta x)^2}{2\sigma^2}}$$

Now, find the "most likely" θ given observed y and x!

Stimulus feature **(x)**



time →

Encoding model

P (y | x)

Recorded spikes **(y)**

time →

Now we are instead trying to do the following:

spikes (known)

Predicted spike distribution

$$y_i \sim N(\boldsymbol{Az}, \sigma^2)$$

model parameters (unknown)

$$P(y_i|\boldsymbol{A}, \boldsymbol{z}, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(y_i - \boldsymbol{Az})^2}{2\sigma^2}}$$

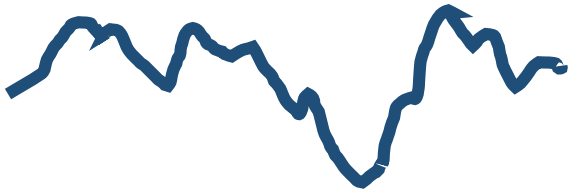
Latent Variable(s)

$$\boldsymbol{z} = ???$$

latent variable(s) (also unknown)

Now, find the "most likely" θ given observed y and x!

Latent variable **(z)**



time →

Encoding model

P (y | z)

Recorded spikes **(y_i)**

time →

The Expectation Maximization algorithm converges to a local solution

spikes (known)

Predicted spike distribution

$$y_i \sim N(\boldsymbol{Az}, \sigma^2) \longrightarrow P(y_i|\boldsymbol{A}, \boldsymbol{z}, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(y_i - \boldsymbol{Az})^2}{2\sigma^2}}$$

model parameters (unknown)

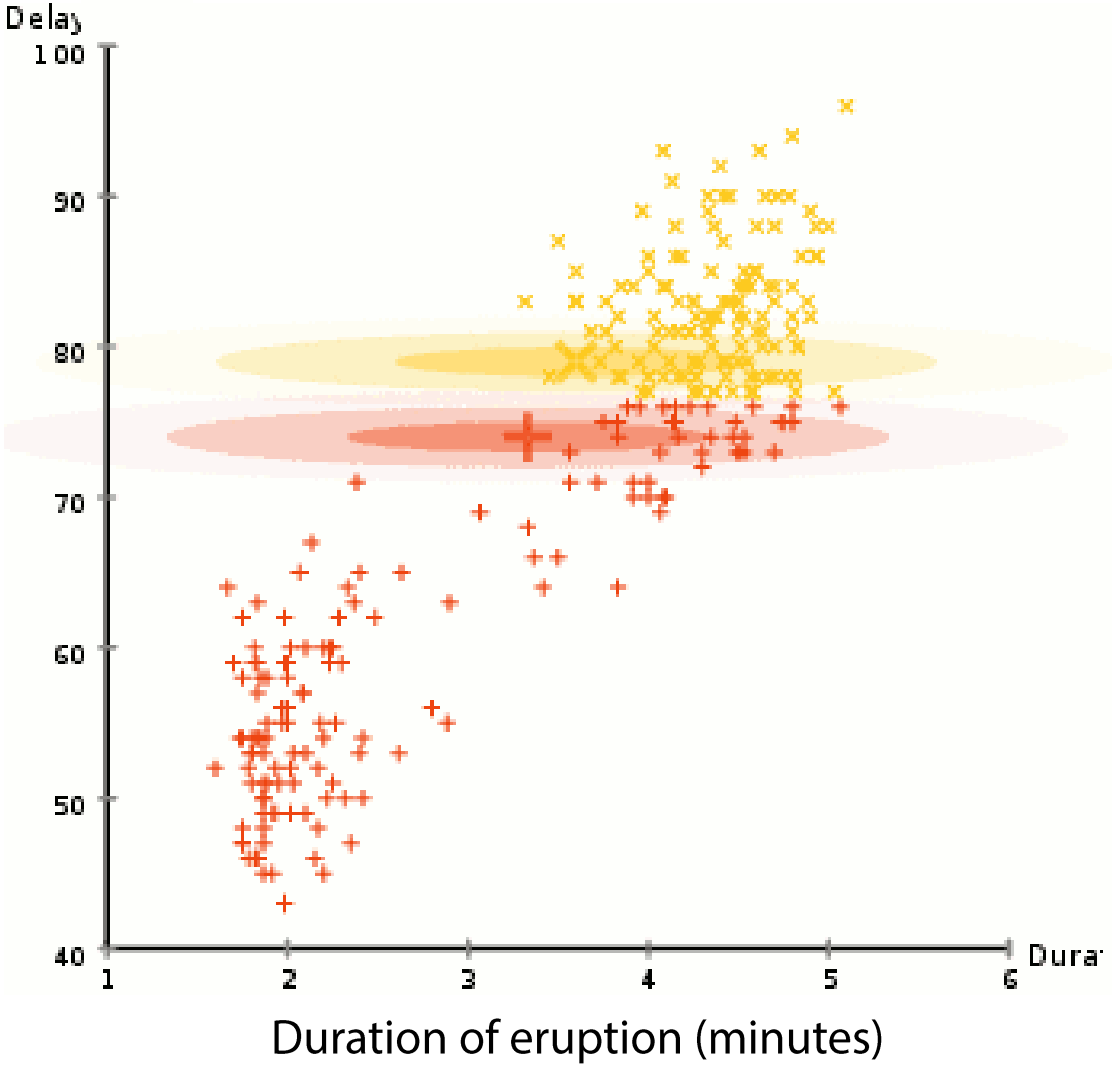
Latent Variable(s)

$$\boldsymbol{z} = ???$$

1) **Initialization:** make a random guess for model parameters (here **A** and σ).
2) **Expectation step:** compute the expected value of Z given y, A, and σ.
3) **Maximization step:** re-estimate **A** and σ given y and our new estimate of Z.
4) **Iterate** steps 2-3 until converged.

The Old Faithful Dataset

Time since last eruption (minutes)



Duration of eruption (minutes)