Latent variable methods for neural population analysis

Maneesh Sahani

Professor of Theoretical Neuroscience and Machine Learning
Gatsby Computational Neuroscience Unit
University College London

March 12, 2019

Latent variable methods

Most neural codes are distributed

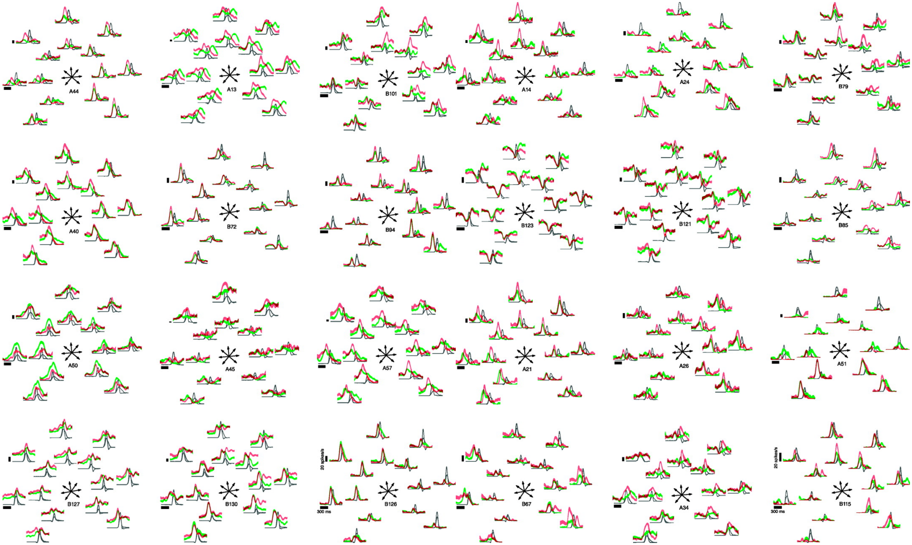
- Each neuron fires for a range of stimulus values and computations.
- Population activity must be taken together to identify stimulus.

Neurons are noisy

- Synaptic release failures.
- Branch-point spike propagation failures.
- Channel noise.
- Network chaos may amplify such noise.

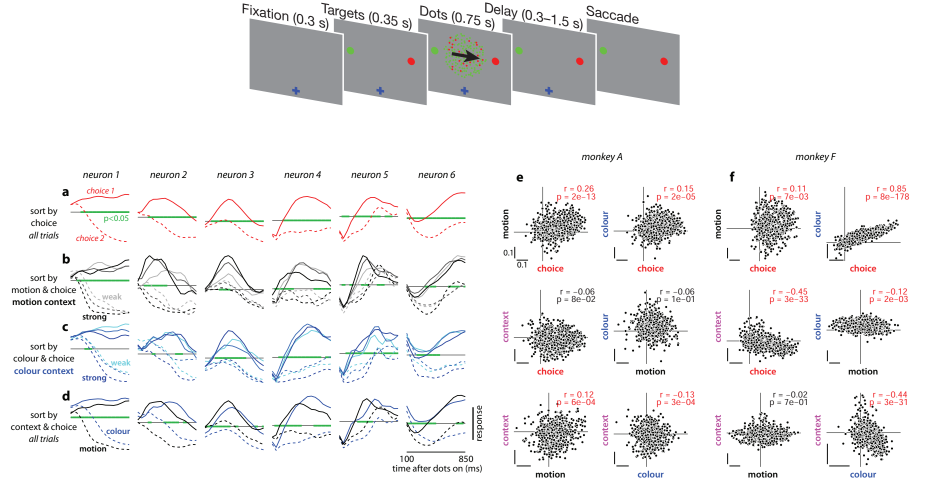
$\Rightarrow$ Network computation is carried in the coordinated activity of many neurons.

Heterogeneous dynamics



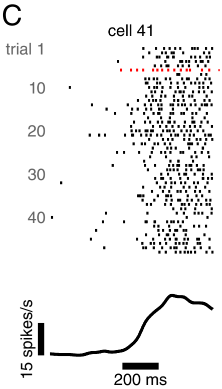
Churchland & Shenoy 2007

Mixed selectivity

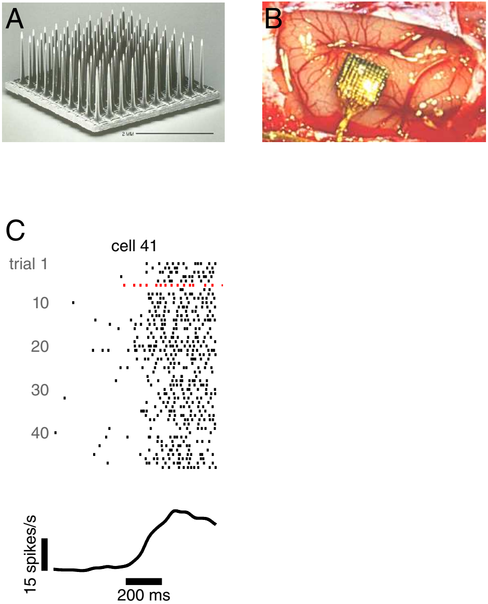


Mante et al. 2013

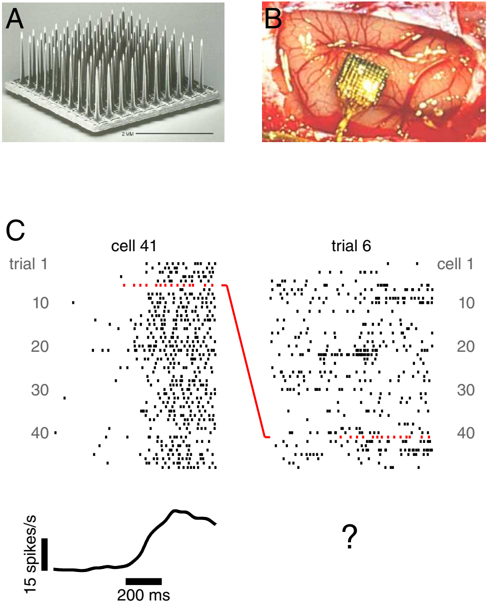
Population recording



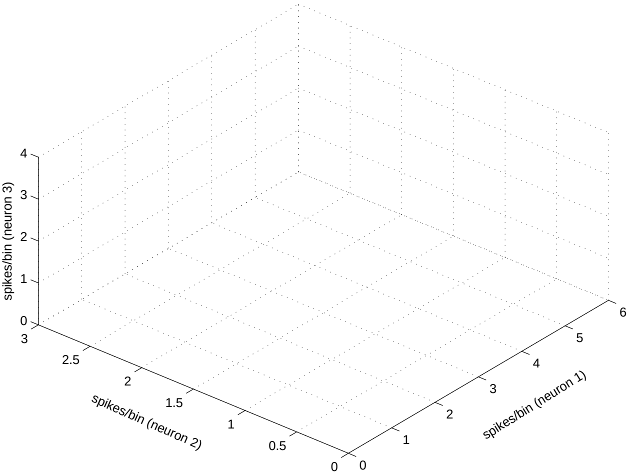
Population recording



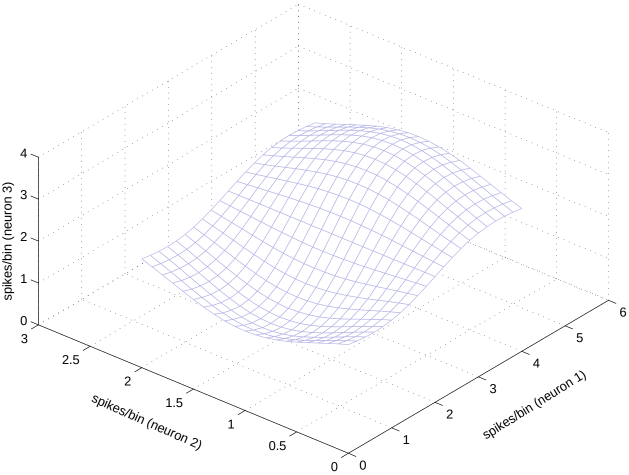
Population recording



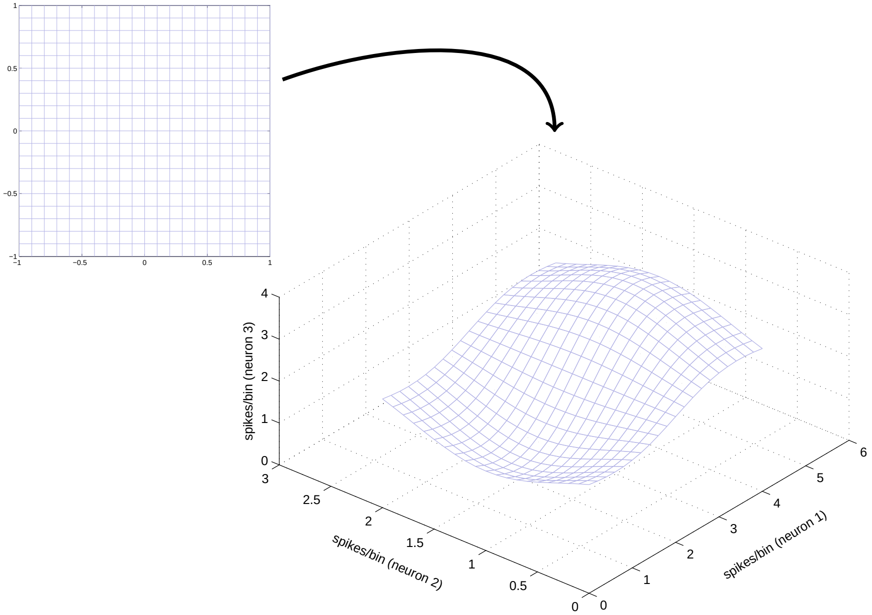
Latent variable methods



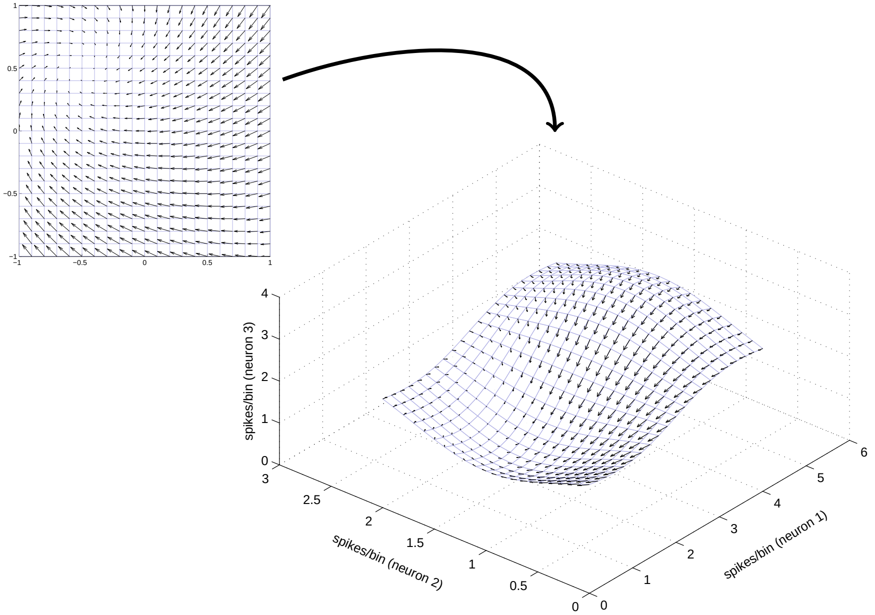
Latent variable methods



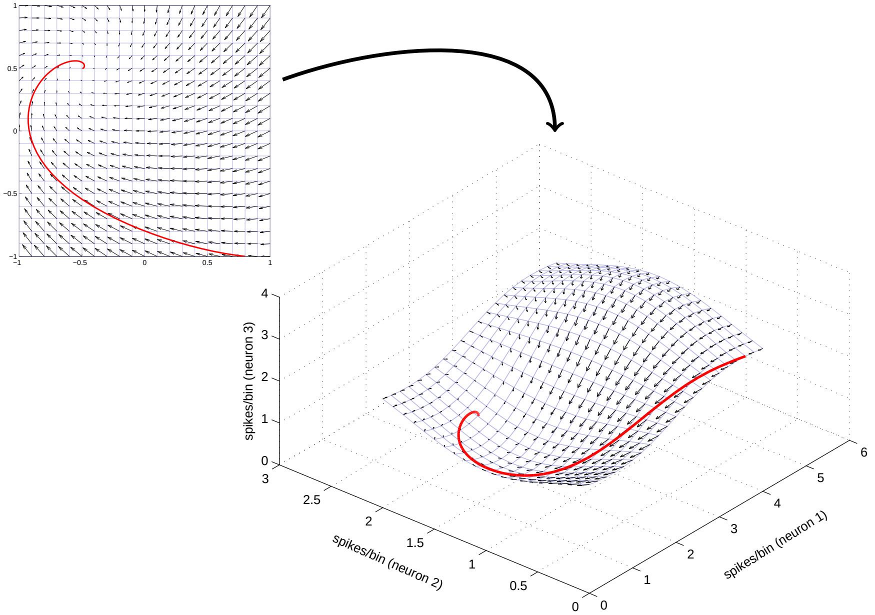
Latent variable methods



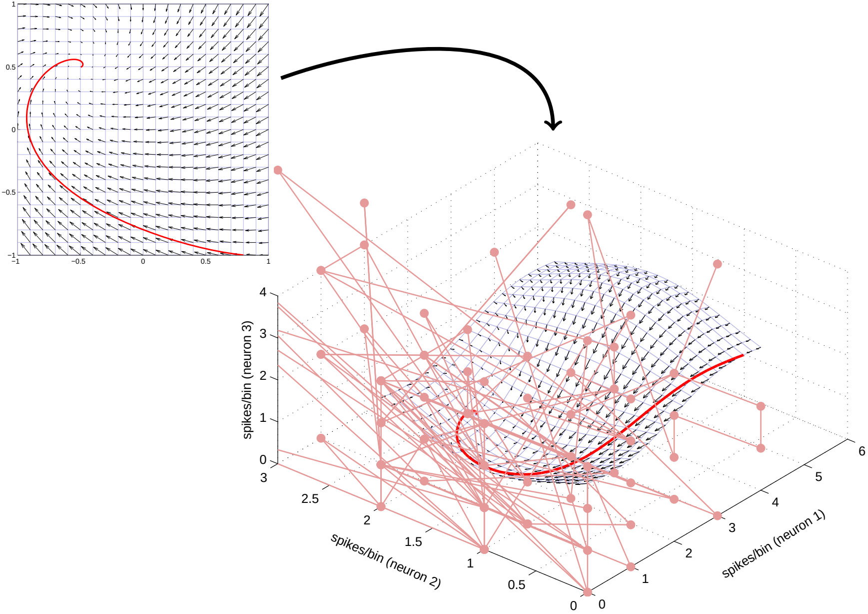
Latent variable methods



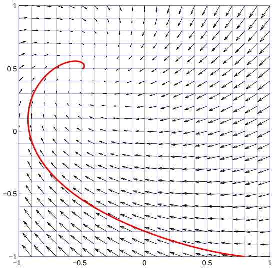
Latent variable methods

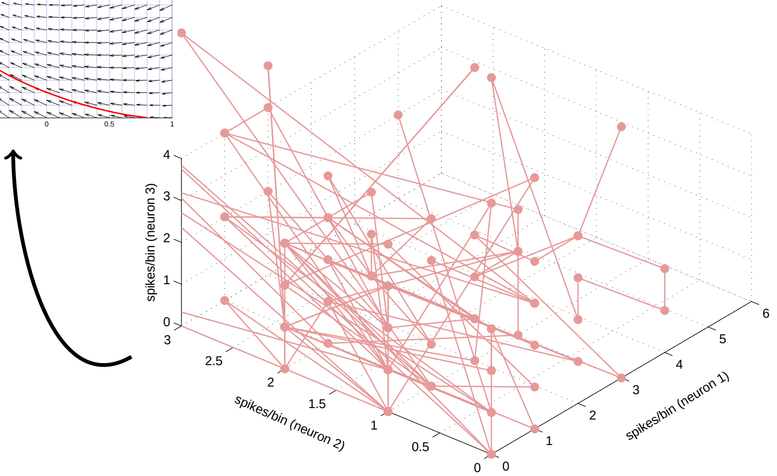


Latent variable methods



Latent variable methods





Two ideas

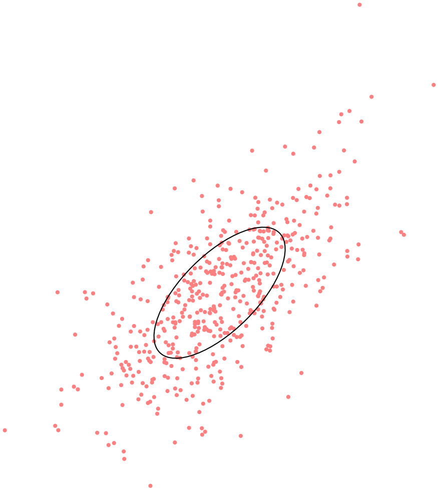
- Static dimensionality reduction
  - Requires data (population states) to be confined to low-dimensional manifold, with relatively small off-manifold noise.
  - In fact measured single-trial noise seems substantial, so single-trial analysis would require that the dominant modes of variability are not noise, but computational variablity *within* the manifold.
  - Conversely, if computational variability is small, then trial-averaging (PSTHs) may reduce off-manifold variation and allow dimensionality reduction.

- Low-dimensional latent dynamics
  - Noise may lift data off manifold, but only “manifold projection” influences future evolution.
  - Conceptually familiar from population coding – independent (or otherwise non-code-shaped) noise is easy to average away.

Linear Gaussian methods

Latent variables and Gaussians

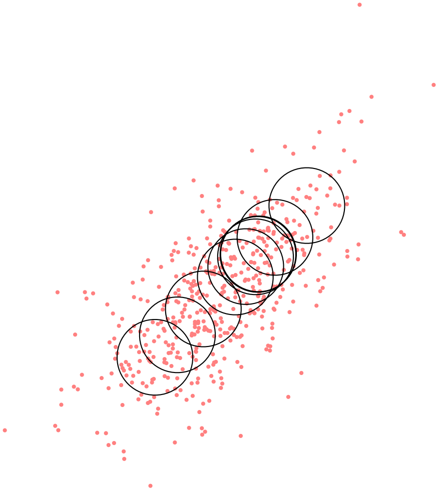
Gaussian correlation can be composed from latent components and uncorrelated noise.



$$\mathbf{x} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} 3 & 2 \\ 2 & 3 \end{bmatrix}\right)$$

Latent variables and Gaussians

Gaussian correlation can be composed from latent components and uncorrelated noise.



$$\mathbf{x} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} 3 & 2 \\ 2 & 3 \end{bmatrix}\right) \quad \Leftrightarrow \quad y \sim \mathcal{N}(0, 1) \quad \mathbf{x} \sim \mathcal{N}\left(\sqrt{2}\begin{bmatrix} 1 \\ 1 \end{bmatrix} y, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)$$

Probabilistic Principal Components Analysis (PPCA)

If the uncorrelated noise is assumed to be isotropic, this model is called PPCA.

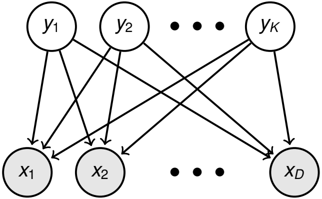
Probabilistic Principal Components Analysis (PPCA)

If the uncorrelated noise is assumed to be isotropic, this model is called PPCA.

Data: $\mathcal{D} = \mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}; \mathbf{x}_i \in \mathbb{R}^D$
Latents: $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}; \mathbf{y}_i \in \mathbb{R}^K$

Linear generative model: $x_d = \sum_{k=1}^{K} \Lambda_{dk} \, y_k + \epsilon_d$

- y_k are independent $\mathcal{N}(0, 1)$ Gaussian factors
- ϵ_d are independent $\mathcal{N}(0, \psi)$ Gaussian noise
- $K < D$



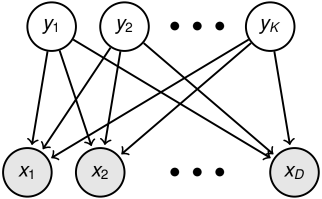
Probabilistic Principal Components Analysis (PPCA)

If the uncorrelated noise is assumed to be isotropic, this model is called PPCA.

Data: $\mathcal{D} = \mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}; \mathbf{x}_i \in \mathbb{R}^D$
Latents: $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}; \mathbf{y}_i \in \mathbb{R}^K$

Linear generative model: $x_d = \sum_{k=1}^{K} \Lambda_{dk}\, y_k + \epsilon_d$

- y_k are independent $\mathcal{N}(0, 1)$ Gaussian factors
- ϵ_d are independent $\mathcal{N}(0, \psi)$ Gaussian noise
- $K < D$



Model for observations $\mathbf{x}$ is a correlated Gaussian:

$$p(\mathbf{y}) = \mathcal{N}(0, I)$$

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\Lambda\mathbf{y}, \psi I)$$

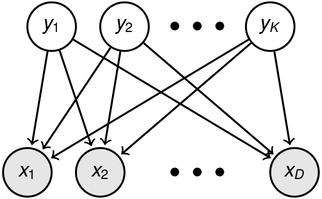
where Λ is a $D \times K$ matrix.

Probabilistic Principal Components Analysis (PPCA)

If the uncorrelated noise is assumed to be isotropic, this model is called PPCA.

Data: $\mathcal{D} = \mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}; \mathbf{x}_i \in \mathbb{R}^D$
Latents: $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}; \mathbf{y}_i \in \mathbb{R}^K$

Linear generative model: $x_d = \sum_{k=1}^{K} \Lambda_{dk}\, y_k + \epsilon_d$



- y_k are independent $\mathcal{N}(0, 1)$ Gaussian factors
- ϵ_d are independent $\mathcal{N}(0, \psi)$ Gaussian noise
- $K < D$

Model for observations $\mathbf{x}$ is a correlated Gaussian:

$$p(\mathbf{y}) = \mathcal{N}(0, I)$$

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\Lambda\mathbf{y}, \psi I)$$

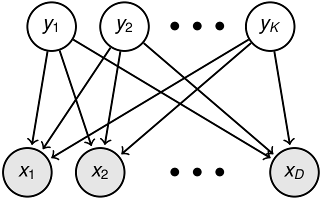
$$p(\mathbf{x}) = \int p(\mathbf{y})p(\mathbf{x}|\mathbf{y})d\mathbf{y}$$

where Λ is a $D \times K$ matrix.

Probabilistic Principal Components Analysis (PPCA)

If the uncorrelated noise is assumed to be isotropic, this model is called PPCA.

Data: $\mathcal{D} = \mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}; \mathbf{x}_i \in \mathbb{R}^D$
Latents: $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}; \mathbf{y}_i \in \mathbb{R}^K$



Linear generative model: $x_d = \sum_{k=1}^{K} \Lambda_{dk}\, y_k + \epsilon_d$

- y_k are independent $\mathcal{N}(0, 1)$ Gaussian factors
- ϵ_d are independent $\mathcal{N}(0, \psi)$ Gaussian noise
- $K < D$

Model for observations **x** is a correlated Gaussian:

$$p(\mathbf{y}) = \mathcal{N}(0, I)$$

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\Lambda\mathbf{y}, \psi I)$$

$$p(\mathbf{x}) = \int p(\mathbf{y})p(\mathbf{x}|\mathbf{y})d\mathbf{y} = \mathcal{N}\left(\mathbb{E}_{\mathbf{y}}[\Lambda\mathbf{y}], \mathbb{E}_{\mathbf{y}}\left[\Lambda\mathbf{y}\mathbf{y}^{\mathsf{T}}\Lambda^{\mathsf{T}}\right] + \psi I\right)$$

Note: $\mathbb{E}_{\mathbf{x}}[f(x)] = \mathbb{E}_{\mathbf{y}}\left[\mathbb{E}_{\mathbf{x}|\mathbf{y}}[f(x)]\right]$

$\mathbb{V}_{\mathbf{x}}[x] = \mathbb{E}_{\mathbf{y}}[\mathbb{V}[\mathbf{x}|\mathbf{y}]] + \mathbb{V}_{\mathbf{y}}[\mathbb{E}[\mathbf{x}|\mathbf{y}]]$

where Λ is a $D \times K$ matrix.

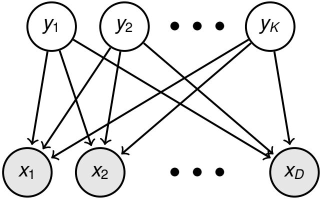
Probabilistic Principal Components Analysis (PPCA)

If the uncorrelated noise is assumed to be isotropic, this model is called PPCA.

Data: $\mathcal{D} = \mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}; \mathbf{x}_i \in \mathbb{R}^D$
Latents: $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}; \mathbf{y}_i \in \mathbb{R}^K$

Linear generative model: $x_d = \sum_{k=1}^{K} \Lambda_{dk}\, y_k + \epsilon_d$

- y_k are independent $\mathcal{N}(0, 1)$ Gaussian factors
- ϵ_d are independent $\mathcal{N}(0, \psi)$ Gaussian noise
- $K < D$



Model for observations **x** is a correlated Gaussian:

$$p(\mathbf{y}) = \mathcal{N}(0, I)$$

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\Lambda\mathbf{y}, \psi I)$$

Note: $\mathbb{E}_{\mathbf{x}}[f(x)] = \mathbb{E}_{\mathbf{y}}\left[\mathbb{E}_{\mathbf{x}|\mathbf{y}}[f(x)]\right]$

$\mathbb{V}_{\mathbf{x}}[x] = \mathbb{E}_{\mathbf{y}}[\mathbb{V}[\mathbf{x}|\mathbf{y}]] + \mathbb{V}_{\mathbf{y}}[\mathbb{E}[\mathbf{x}|\mathbf{y}]]$

$$p(\mathbf{x}) = \int p(\mathbf{y})p(\mathbf{x}|\mathbf{y})d\mathbf{y} = \mathcal{N}\left(\mathbb{E}_{\mathbf{y}}[\Lambda\mathbf{y}], \mathbb{E}_{\mathbf{y}}\left[\Lambda\mathbf{y}\mathbf{y}^{\mathsf{T}}\Lambda^{\mathsf{T}}\right] + \psi I\right) = \mathcal{N}\left(0, \Lambda\Lambda^{\mathsf{T}} + \psi I\right)$$

where Λ is a $D \times K$ matrix.

PPCA likelihood

The marginal distribution on $\mathbf{x}$ gives us the PPCA likelihood:

$$\log p(\mathcal{X}|\Lambda, \psi) = -\frac{N}{2}\log\left|2\pi(\Lambda\Lambda^{\mathsf{T}} + \psi I)\right| - \frac{1}{2}\mathrm{Tr}\left[(\Lambda\Lambda^{\mathsf{T}} + \psi I)^{-1} \underbrace{\sum_n \mathbf{x}\mathbf{x}^{\mathsf{T}}}_{NS}\right]$$

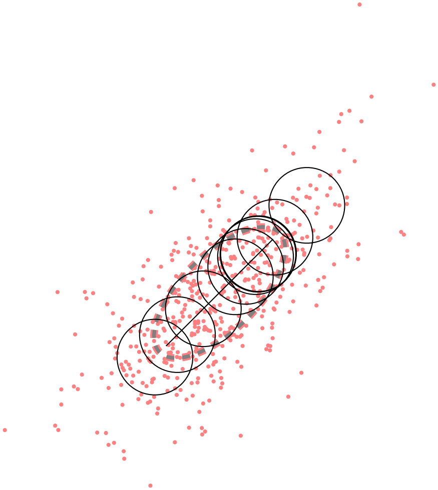
To find the ML values of (Λ, ψ) we could optimise numerically (gradient ascent / Newton's method), or we could use a different iterative algorithm called EM which we'll introduce soon.

In fact, however, ML for PPCA is more straightforward in principle, as we will see by first considering the limit $\psi \to 0$.

[Note: We may also add a constant mean $\boldsymbol{\mu}$ to the output, so as to model data that are not distributed around 0. In this case, the ML estimate $\widehat{\boldsymbol{\mu}} = \frac{1}{N}\sum_n \mathbf{x}_n$ and we can define $S = \frac{1}{N}\sum_n(\mathbf{x} - \widehat{\boldsymbol{\mu}})(\mathbf{x} - \widehat{\boldsymbol{\mu}})^{\mathsf{T}}$ in the likelihood above.]

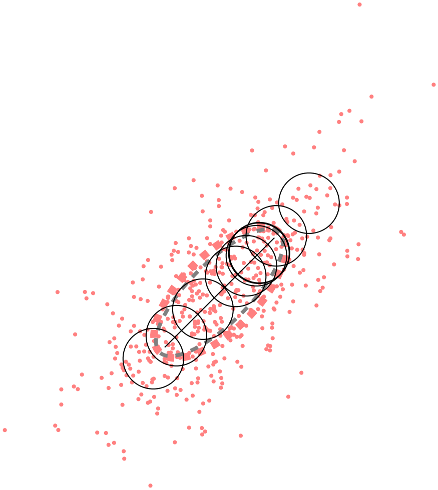
The $\psi \to 0$ limit

As $\psi \to 0$, the latent model can only capture *K* dimensions of variance.



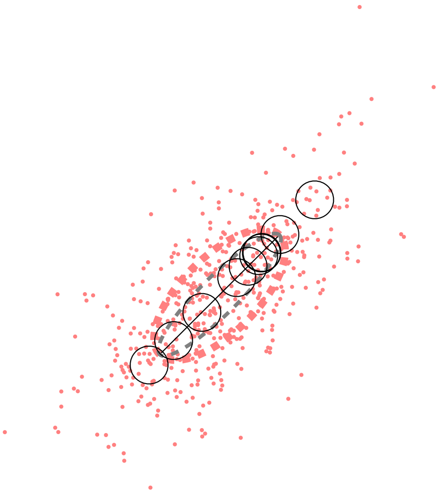
The $\psi \to 0$ limit

As $\psi \to 0$, the latent model can only capture *K* dimensions of variance.



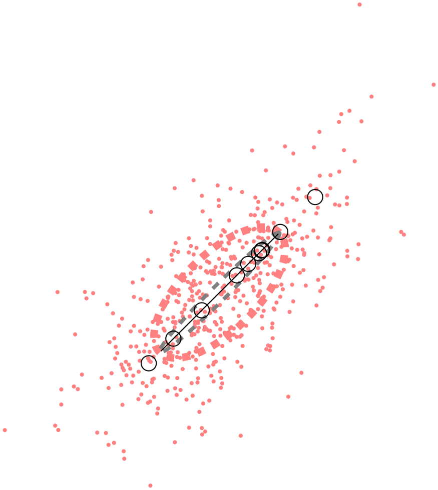
The $\psi \to 0$ limit

As $\psi \to 0$, the latent model can only capture K dimensions of variance.



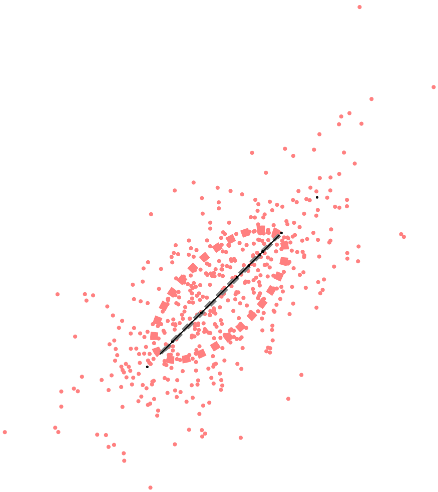
The $\psi \to 0$ limit

As $\psi \to 0$, the latent model can only capture K dimensions of variance.



The $\psi \to 0$ limit

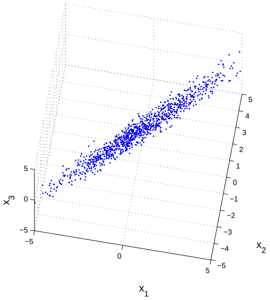
As $\psi \to 0$, the latent model can only capture *K* dimensions of variance.



In a Gaussian model, the ML parameters will find the *K*-dimensional space of most variance.

Principal Components Analysis

This leads us to an (old) algorithm called Principal Components Analysis (PCA).



Assume data $\mathcal{D} = \{\mathbf{x}_i\}$ have zero mean (if not, subtract it).

- Find direction of greatest variance – $\boldsymbol{\lambda}_{(1)}$.

$$\boldsymbol{\lambda}_{(1)} = \underset{\|\mathbf{v}\|=1}{\operatorname{argmax}} \sum_n (\mathbf{x}_n^T \mathbf{v})^2$$

- Find direction orthogonal to $\boldsymbol{\lambda}_{(1)}$ with greatest variance – $\boldsymbol{\lambda}_{(2)}$

$\vdots$

- Find direction orthogonal to $\{\boldsymbol{\lambda}_{(1)}, \boldsymbol{\lambda}_{(2)}, \ldots, \boldsymbol{\lambda}_{(n-1)}\}$ with greatest variance – $\boldsymbol{\lambda}_{(n)}$.
- Terminate when remaining variance drops below a threshold.

Eigendecomposition of a covariance matrix

The eigendecomposition of a covariance matrix makes finding the PCs easy.

Eigendecomposition of a covariance matrix

The eigendecomposition of a covariance matrix makes finding the PCs easy.

Recall that **u** is an eigenvector, with scalar eigenvalue ω, of a matrix S if

$$S\mathbf{u} = \omega\mathbf{u}$$

u can have any norm, but we will define it to be unity (i.e., $\mathbf{u}^{\mathsf{T}}\mathbf{u} = 1$).

Eigendecomposition of a covariance matrix

The eigendecomposition of a covariance matrix makes finding the PCs easy.
Recall that $\mathbf{u}$ is an eigenvector, with scalar eigenvalue ω, of a matrix S if

$$S\mathbf{u} = \omega\mathbf{u}$$

$\mathbf{u}$ can have any norm, but we will define it to be unity (i.e., $\mathbf{u}^\mathsf{T}\mathbf{u} = 1$).
For a covariance matrix $S = \langle \mathbf{x}\mathbf{x}^\mathsf{T} \rangle$ (which is $D \times D$, symmetric, positive semi-definite):

- In general there are D eigenvector-eigenvalue pairs $(\mathbf{u}_{(i)}, \omega_{(i)})$, except if two or more eigenvectors share the same eigenvalue (in which case the eigenvectors are degenerate — any linear combination is also an eigenvector).

Eigendecomposition of a covariance matrix

The eigendecomposition of a covariance matrix makes finding the PCs easy.

Recall that $\mathbf{u}$ is an eigenvector, with scalar eigenvalue ω, of a matrix S if

$$S\mathbf{u} = \omega\mathbf{u}$$

$\mathbf{u}$ can have any norm, but we will define it to be unity (i.e., $\mathbf{u}^{\mathsf{T}}\mathbf{u} = 1$).

For a covariance matrix $S = \left\langle \mathbf{x}\mathbf{x}^{\mathsf{T}} \right\rangle$ (which is $D \times D$, symmetric, positive semi-definite):

- In general there are D eigenvector-eigenvalue pairs $(\mathbf{u}_{(i)}, \omega_{(i)})$, except if two or more eigenvectors share the same eigenvalue (in which case the eigenvectors are degenerate — any linear combination is also an eigenvector).
- The D eigenvectors are orthogonal (or orthogonalisable, if $\omega_{(i)} = \omega_{(j)}$). Thus, they form an orthonormal basis. $\sum_i \mathbf{u}_{(i)}\mathbf{u}_{(i)}^{\mathsf{T}} = I$.

Eigendecomposition of a covariance matrix

The eigendecomposition of a covariance matrix makes finding the PCs easy.
Recall that $\mathbf{u}$ is an eigenvector, with scalar eigenvalue ω, of a matrix S if

$$S\mathbf{u} = \omega\mathbf{u}$$

$\mathbf{u}$ can have any norm, but we will define it to be unity (i.e., $\mathbf{u}^\mathsf{T}\mathbf{u} = 1$).
For a covariance matrix $S = \langle \mathbf{x}\mathbf{x}^\mathsf{T} \rangle$ (which is $D \times D$, symmetric, positive semi-definite):

- In general there are D eigenvector-eigenvalue pairs $(\mathbf{u}_{(i)}, \omega_{(i)})$, except if two or more eigenvectors share the same eigenvalue (in which case the eigenvectors are degenerate — any linear combination is also an eigenvector).
- The D eigenvectors are orthogonal (or orthogonalisable, if $\omega_{(i)} = \omega_{(j)}$). Thus, they form an orthonormal basis. $\sum_i \mathbf{u}_{(i)}\mathbf{u}_{(i)}{}^\mathsf{T} = I$.
- Any vector $\mathbf{v}$ can be written as

$$\mathbf{v} = \Big(\sum_i \mathbf{u}_{(i)}\mathbf{u}_{(i)}{}^\mathsf{T}\Big)\mathbf{v} = \sum_i (\mathbf{u}_{(i)}{}^\mathsf{T}\mathbf{v})\mathbf{u}_{(i)} = \sum_i v_{(i)}\mathbf{u}_{(i)}$$

Eigendecomposition of a covariance matrix

The eigendecomposition of a covariance matrix makes finding the PCs easy.
Recall that $\mathbf{u}$ is an eigenvector, with scalar eigenvalue ω, of a matrix S if

$$S\mathbf{u} = \omega\mathbf{u}$$

$\mathbf{u}$ can have any norm, but we will define it to be unity (i.e., $\mathbf{u}^{\mathsf{T}}\mathbf{u} = 1$).
For a covariance matrix $S = \langle \mathbf{x}\mathbf{x}^{\mathsf{T}} \rangle$ (which is $D \times D$, symmetric, positive semi-definite):

- In general there are D eigenvector-eigenvalue pairs $(\mathbf{u}_{(i)}, \omega_{(i)})$, except if two or more eigenvectors share the same eigenvalue (in which case the eigenvectors are degenerate — any linear combination is also an eigenvector).
- The D eigenvectors are orthogonal (or orthogonalisable, if $\omega_{(i)} = \omega_{(j)}$). Thus, they form an orthonormal basis. $\sum_i \mathbf{u}_{(i)}\mathbf{u}_{(i)}^{\mathsf{T}} = I$.
- Any vector $\mathbf{v}$ can be written as

$$\mathbf{v} = \left(\sum_i \mathbf{u}_{(i)}\mathbf{u}_{(i)}^{\mathsf{T}}\right)\mathbf{v} = \sum_i (\mathbf{u}_{(i)}^{\mathsf{T}}\mathbf{v})\mathbf{u}_{(i)} = \sum_i v_{(i)}\mathbf{u}_{(i)}$$

- The original matrix S can be written:

$$S = \sum_i \omega_{(i)}\mathbf{u}_{(i)}\mathbf{u}_{(i)}^{\mathsf{T}} = UWU^{\mathsf{T}}$$

  where $U = [\mathbf{u}_{(1)}, \mathbf{u}_{(2)}, \ldots, \mathbf{u}_{(D)}]$ collects the eigenvectors and $W = \operatorname{diag}\left[(\omega_{(1)}, \omega_{(2)}, \ldots, \omega_{(D)})\right]$.

PCA and eigenvectors

- The variance in direction $\mathbf{u}_{(i)}$ is

$$\left\langle (\mathbf{x}^{\mathsf{T}}\mathbf{u}_{(i)})^2 \right\rangle = \left\langle \mathbf{u}_{(i)}{}^{\mathsf{T}}\mathbf{x}\mathbf{x}^{\mathsf{T}}\mathbf{u}_{(i)} \right\rangle = \mathbf{u}_{(i)}{}^{\mathsf{T}} S \mathbf{u}_{(i)} = \mathbf{u}_{(i)}{}^{\mathsf{T}} \omega_{(i)} \mathbf{u}_{(i)} = \omega_{(i)}$$

PCA and eigenvectors

- The variance in direction $\mathbf{u}_{(i)}$ is

$$\left\langle (\mathbf{x}^\mathsf{T}\mathbf{u}_{(i)})^2 \right\rangle = \left\langle \mathbf{u}_{(i)}{}^\mathsf{T}\mathbf{x}\mathbf{x}^\mathsf{T}\mathbf{u}_{(i)} \right\rangle = \mathbf{u}_{(i)}{}^\mathsf{T} S\mathbf{u}_{(i)} = \mathbf{u}_{(i)}{}^\mathsf{T}\omega_{(i)}\mathbf{u}_{(i)} = \omega_{(i)}$$

- The variance in an arbitrary direction $\mathbf{v}$ is

$$\left\langle (\mathbf{x}^\mathsf{T}\mathbf{v})^2 \right\rangle = \left\langle \left(\mathbf{x}^\mathsf{T}\left(\sum_i v_{(i)}\mathbf{u}_{(i)} \right) \right)^2 \right\rangle = \sum_{ij} v_{(i)}\mathbf{u}_{(i)}{}^\mathsf{T} S\mathbf{u}_{(j)} v_{(j)}$$
$$= \sum_{ij} v_{(i)}\omega_{(j)} v_{(j)}\mathbf{u}_{(i)}{}^\mathsf{T}\mathbf{u}_{(j)} = \sum_i v_{(i)}^2\omega_{(i)}$$

PCA and eigenvectors

- The variance in direction $\mathbf{u}_{(i)}$ is

$$\left\langle (\mathbf{x}^{\mathsf{T}}\mathbf{u}_{(i)})^2 \right\rangle = \left\langle \mathbf{u}_{(i)}{}^{\mathsf{T}}\mathbf{x}\mathbf{x}^{\mathsf{T}}\mathbf{u}_{(i)} \right\rangle = \mathbf{u}_{(i)}{}^{\mathsf{T}} S \mathbf{u}_{(i)} = \mathbf{u}_{(i)}{}^{\mathsf{T}} \omega_{(i)} \mathbf{u}_{(i)} = \omega_{(i)}$$

- The variance in an arbitrary direction $\mathbf{v}$ is

$$\left\langle (\mathbf{x}^{\mathsf{T}}\mathbf{v})^2 \right\rangle = \left\langle \left(\mathbf{x}^{\mathsf{T}} \left(\sum_i v_{(i)} \mathbf{u}_{(i)} \right) \right)^2 \right\rangle = \sum_{ij} v_{(i)} \mathbf{u}_{(i)}{}^{\mathsf{T}} S \mathbf{u}_{(j)} v_{(j)}$$

$$= \sum_{ij} v_{(i)} \omega_{(j)} v_{(j)} \mathbf{u}_{(i)}{}^{\mathsf{T}} \mathbf{u}_{(j)} = \sum_i v_{(i)}^2 \omega_{(i)}$$

- If $\mathbf{v}^{\mathsf{T}}\mathbf{v} = 1$, then $\sum_i v_{(i)}^2 = 1$ and so $\operatorname{argmax}_{\|\mathbf{v}\|=1} \left\langle (\mathbf{x}^{\mathsf{T}}\mathbf{v})^2 \right\rangle = \mathbf{u}_{(\max)}$
  The direction of greatest variance is the eigenvector the largest eigenvalue.

PCA and eigenvectors

- The variance in direction $\mathbf{u}_{(i)}$ is

$$\left\langle (\mathbf{x}^\mathsf{T}\mathbf{u}_{(i)})^2 \right\rangle = \left\langle \mathbf{u}_{(i)}{}^\mathsf{T}\mathbf{x}\mathbf{x}^\mathsf{T}\mathbf{u}_{(i)} \right\rangle = \mathbf{u}_{(i)}{}^\mathsf{T} S \mathbf{u}_{(i)} = \mathbf{u}_{(i)}{}^\mathsf{T} \omega_{(i)} \mathbf{u}_{(i)} = \omega_{(i)}$$

- The variance in an arbitrary direction $\mathbf{v}$ is

$$\left\langle (\mathbf{x}^\mathsf{T}\mathbf{v})^2 \right\rangle = \left\langle \left(\mathbf{x}^\mathsf{T} \left(\sum_i v_{(i)} \mathbf{u}_{(i)} \right) \right)^2 \right\rangle = \sum_{ij} v_{(i)} \mathbf{u}_{(i)}{}^\mathsf{T} S \mathbf{u}_{(j)} v_{(j)}$$

$$= \sum_{ij} v_{(i)} \omega_{(j)} v_{(j)} \mathbf{u}_{(i)}{}^\mathsf{T} \mathbf{u}_{(j)} = \sum_i v_{(i)}^2 \omega_{(i)}$$

- If $\mathbf{v}^\mathsf{T}\mathbf{v} = 1$, then $\sum_i v_{(i)}^2 = 1$ and so $\operatorname{argmax}_{\|\mathbf{v}\|=1} \left\langle (\mathbf{x}^\mathsf{T}\mathbf{v})^2 \right\rangle = \mathbf{u}_{(\max)}$
  The direction of greatest variance is the eigenvector the largest eigenvalue.
- In general, the PCs are exactly the eigenvectors of the empirical covariance matrix, ordered by decreasing eigenvalue.

PCA and eigenvectors

- The variance in direction $\mathbf{u}_{(i)}$ is

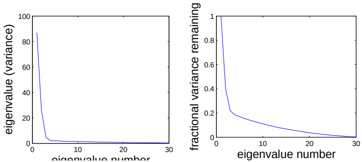
$$\left\langle (\mathbf{x}^{\mathsf{T}}\mathbf{u}_{(i)})^2 \right\rangle = \left\langle \mathbf{u}_{(i)}{}^{\mathsf{T}}\mathbf{x}\mathbf{x}^{\mathsf{T}}\mathbf{u}_{(i)} \right\rangle = \mathbf{u}_{(i)}{}^{\mathsf{T}} S \mathbf{u}_{(i)} = \mathbf{u}_{(i)}{}^{\mathsf{T}} \omega_{(i)} \mathbf{u}_{(i)} = \omega_{(i)}$$

- The variance in an arbitrary direction $\mathbf{v}$ is

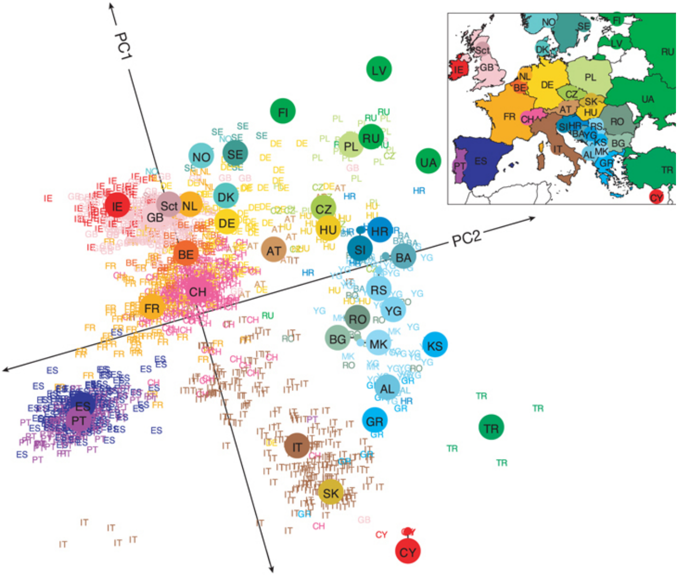
$$\left\langle (\mathbf{x}^{\mathsf{T}}\mathbf{v})^2 \right\rangle = \left\langle \left(\mathbf{x}^{\mathsf{T}} \left(\sum_i v_{(i)} \mathbf{u}_{(i)} \right) \right)^2 \right\rangle = \sum_{ij} v_{(i)} \mathbf{u}_{(i)}{}^{\mathsf{T}} S \mathbf{u}_{(j)} v_{(j)}$$

$$= \sum_{ij} v_{(i)} \omega_{(j)} v_{(j)} \mathbf{u}_{(i)}{}^{\mathsf{T}} \mathbf{u}_{(j)} = \sum_i v_{(i)}^2 \omega_{(i)}$$

- If $\mathbf{v}^{\mathsf{T}}\mathbf{v} = 1$, then $\sum_i v_{(i)}^2 = 1$ and so $\operatorname{argmax}_{\|\mathbf{v}\|=1} \left\langle (\mathbf{x}^{\mathsf{T}}\mathbf{v})^2 \right\rangle = \mathbf{u}_{(\max)}$
  The direction of greatest variance is the eigenvector the largest eigenvalue.
- In general, the PCs are exactly the eigenvectors of the empirical covariance matrix, ordered by decreasing eigenvalue.
- The eigenspectrum shows how the variance is distributed across dimensions; can identify transitions that might separate signal from noise, or the number of PCs that capture a pre-determined fraction of variance.

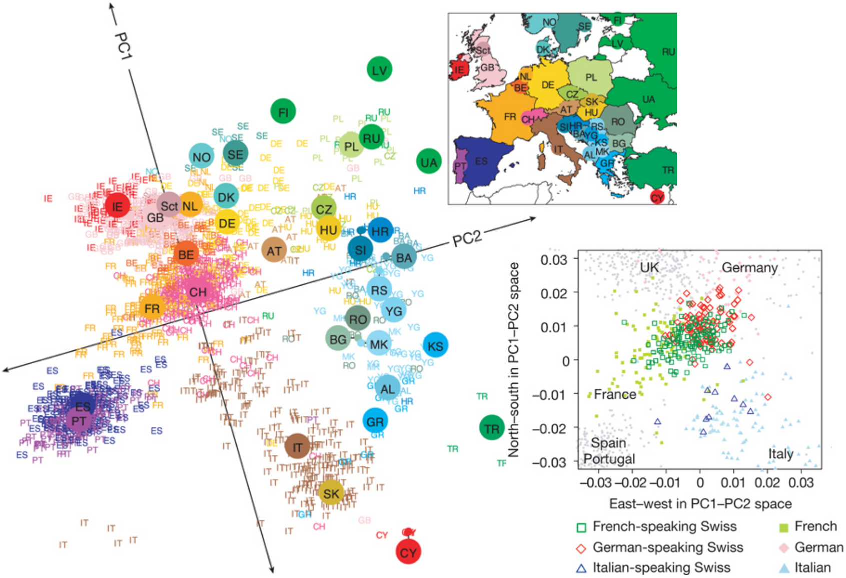


Example of PCA: Genetic variation within Europe



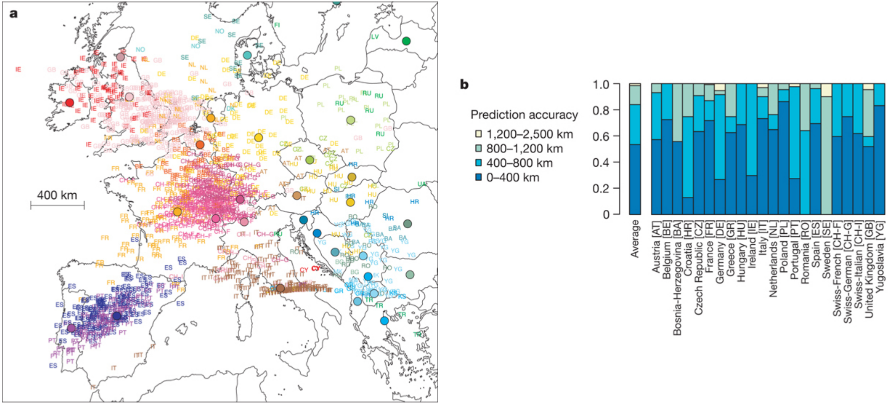
Novembre et al. (2008) Nature 456:98-101

Example of PCA: Genetic variation within Europe



Novembre et al. (2008) Nature 456:98-101

Example of PCA: Genetic variation within Europe

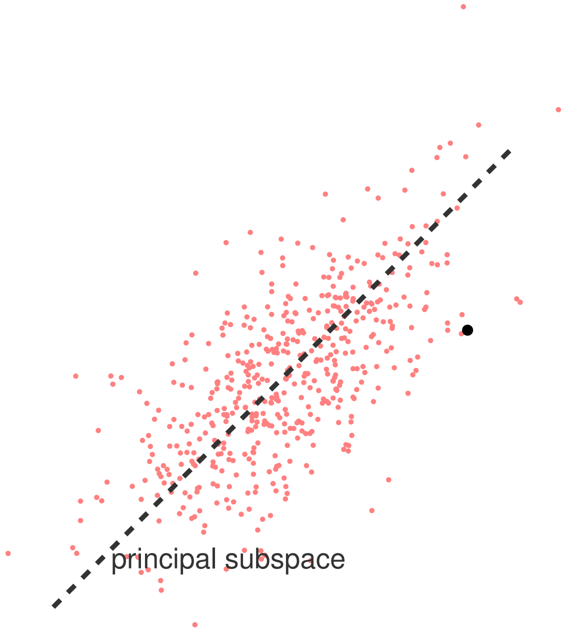


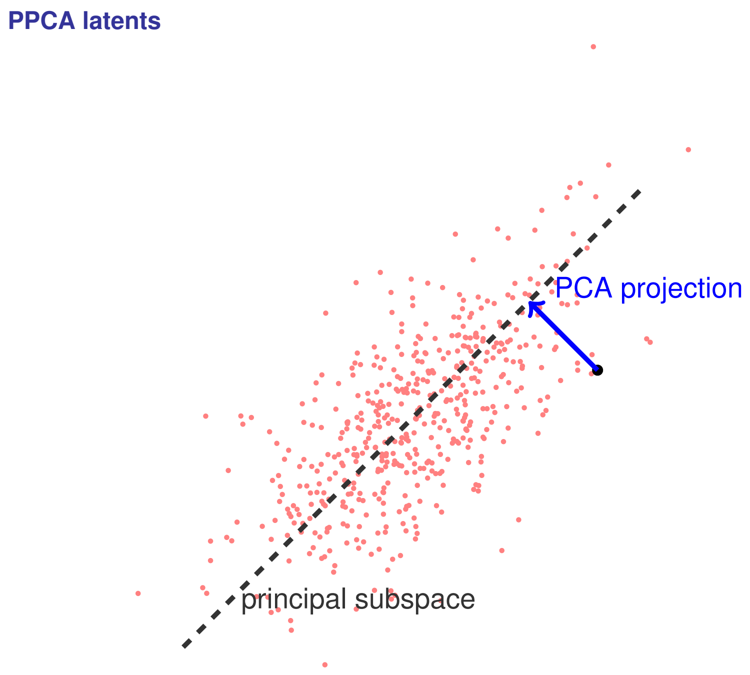
Novembre et al. (2008) Nature 456:98-101

Equivalent definitions of PCA

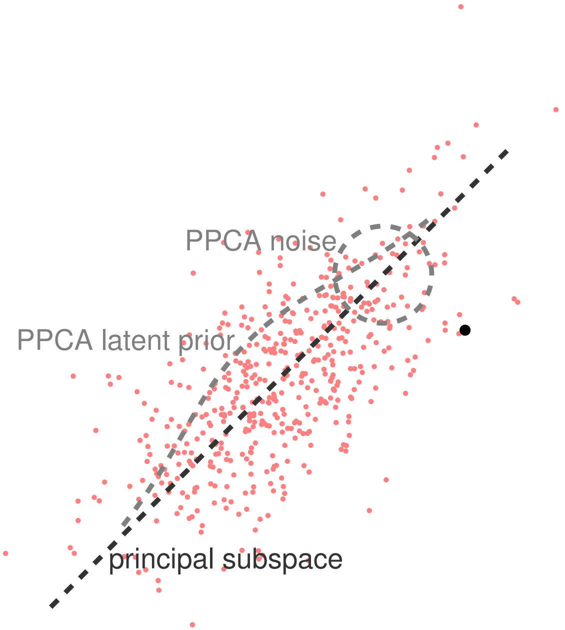
- Find K directions of greatest variance in data.
- Find K-dimensional orthogonal projection that *preserves* greatest variance.
- Find K-dimensional vectors $\mathbf{y}_i$ and matrix Λ so that $\hat{\mathbf{x}}_i = \Lambda \mathbf{y}_i$ is as close as possible (in squared distance) to $\mathbf{x}_i$.
- Find the approximate rank-K factorisation of the data matrix $X \approx \Lambda Y$ with smallest squared error (SVD!)
- . . . (many others)

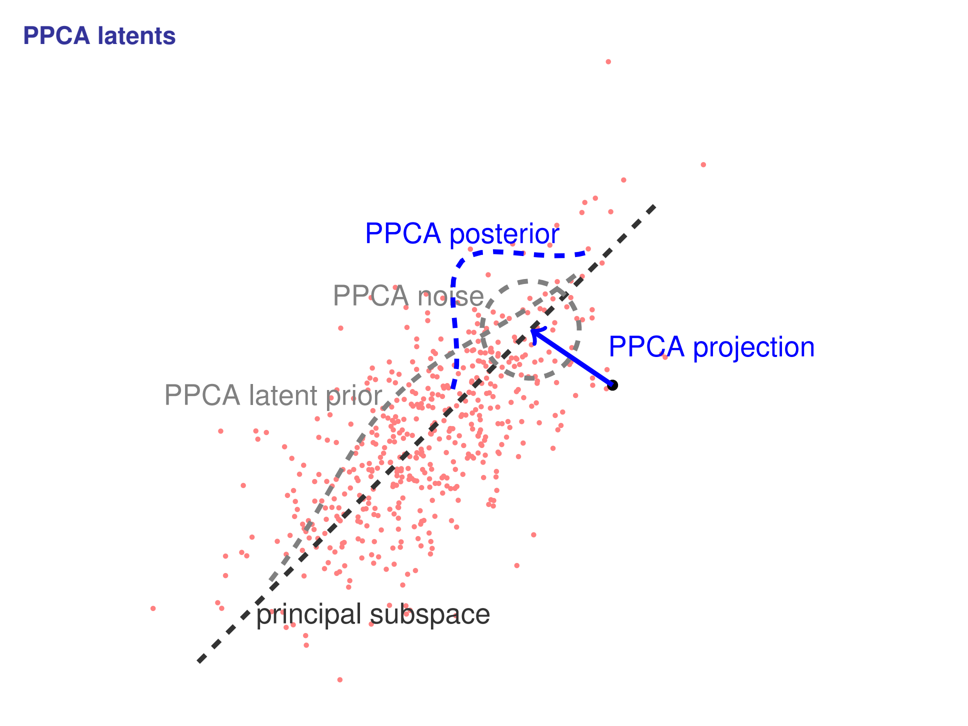
PPCA latents




PPCA latents
PCA projection
principal subspace

PPCA latents



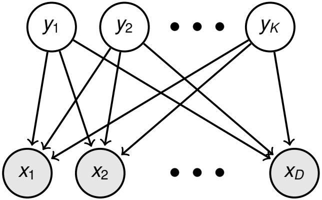

PPCA latents
PPCA posterior
PPCA noise
PPCA projection
PPCA latent prior
principal subspace

Factor Analysis

If dimensions are not equivalent, equal variance assumption is inappropriate.

Data: $\mathcal{D} = \mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}; \mathbf{x}_i \in \mathbb{R}^D$

Latents: $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}; \mathbf{y}_i \in \mathbb{R}^K$



Linear generative model: $x_d = \sum_{k=1}^{K} \Lambda_{dk}\ y_k + \epsilon_d$

- y_k are independent $\mathcal{N}(0, 1)$ Gaussian factors
- ϵ_d are independent $\mathcal{N}(0, \Psi_{dd})$ Gaussian noise
- $K < D$

Model for observations $\mathbf{x}$ is still a correlated Gaussian:

$$p(\mathbf{y}) = \mathcal{N}(0, I)$$

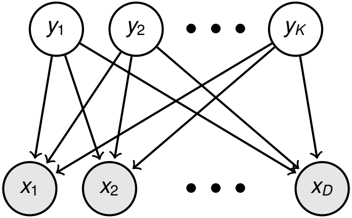
$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\Lambda\mathbf{y}, \Psi)$$

$$p(\mathbf{x}) = \int p(\mathbf{y})p(\mathbf{x}|\mathbf{y})d\mathbf{y} = \mathcal{N}\left(0, \Lambda\Lambda^{\mathsf{T}} + \Psi\right)$$

where Λ is a $D \times K$, and Ψ is $K \times K$ and diagonal.

Dimensionality Reduction: Finds a low-dimensional projection of high dimensional data that captures the correlation structure of the data.

Factor Analysis (cont.)



- ML learning finds Λ ("common factors") and Ψ ("unique factors" or "uniquenesses") given data
- parameters (corrected for symmetries): $DK + D - \frac{K(K-1)}{2}$
- If number of parameters $> \frac{D(D+1)}{2}$ model is not identifiable (even after accounting for rotational degeneracy discussed later)
- no closed form solution for ML params: $\mathcal{N}(0, \Lambda\Lambda^\mathsf{T} + \Psi)$

Factor Analysis projections

Our analysis for PPCA still applies:

$$\tilde{\mathbf{x}}_n = \Lambda(I + \Lambda^\mathsf{T}\Psi^{-1}\Lambda)^{-1}\Lambda^\mathsf{T}\Psi^{-1}\mathbf{x}_n = \mathbf{x}_n - \Psi(\Lambda\Lambda^\mathsf{T} + \Psi)^{-1}\mathbf{x}_n$$

but now Ψ is diagonal but not spherical.

Note, though, that Λ is generally different from that found by PPCA.

Factor Analysis projections

Our analysis for PPCA still applies:

$$\tilde{\mathbf{x}}_n = \Lambda(I + \Lambda^\top \Psi^{-1}\Lambda)^{-1}\Lambda^\top \Psi^{-1}\mathbf{x}_n = \mathbf{x}_n - \Psi(\Lambda\Lambda^\top + \Psi)^{-1}\mathbf{x}_n$$

but now Ψ is diagonal but not spherical.

Note, though, that Λ is generally different from that found by PPCA.

And Λ is not unique: the latent space may be transformed by an arbitrary orthogonal transform U ($U^\top U = UU^\top = I$) without changing the likelihood.

Factor Analysis projections

Our analysis for PPCA still applies:

$$\tilde{\mathbf{x}}_n = \Lambda(I + \Lambda^{\mathsf{T}}\Psi^{-1}\Lambda)^{-1}\Lambda^{\mathsf{T}}\Psi^{-1}\mathbf{x}_n = \mathbf{x}_n - \Psi(\Lambda\Lambda^{\mathsf{T}} + \Psi)^{-1}\mathbf{x}_n$$

but now Ψ is diagonal but not spherical.

Note, though, that Λ is generally different from that found by PPCA.

And Λ is not unique: the latent space may be transformed by an arbitrary orthogonal transform U ($U^{\mathsf{T}}U = UU^{\mathsf{T}} = I$) without changing the likelihood.

$$\tilde{\mathbf{y}} = U\mathbf{y} \quad \text{and} \quad \tilde{\Lambda} = \Lambda U^{\mathsf{T}} \quad \Rightarrow \quad \tilde{\Lambda}\tilde{\mathbf{y}} = \Lambda U^{\mathsf{T}}U\mathbf{y} = \Lambda\mathbf{y}$$

Factor Analysis projections

Our analysis for PPCA still applies:

$$\tilde{\mathbf{x}}_n = \Lambda(I + \Lambda^{\mathsf{T}}\Psi^{-1}\Lambda)^{-1}\Lambda^{\mathsf{T}}\Psi^{-1}\mathbf{x}_n = \mathbf{x}_n - \Psi(\Lambda\Lambda^{\mathsf{T}} + \Psi)^{-1}\mathbf{x}_n$$

but now Ψ is diagonal but not spherical.

Note, though, that Λ is generally different from that found by PPCA.

And Λ is not unique: the latent space may be transformed by an arbitrary orthogonal transform U ($U^{\mathsf{T}}U = UU^{\mathsf{T}} = I$) without changing the likelihood.

$$\tilde{\mathbf{y}} = U\mathbf{y} \quad \text{and} \quad \tilde{\Lambda} = \Lambda U^{\mathsf{T}} \quad \Rightarrow \quad \tilde{\Lambda}\tilde{\mathbf{y}} = \Lambda U^{\mathsf{T}}U\mathbf{y} = \Lambda\mathbf{y}$$

$$-\ell = \frac{1}{2}\log\left|2\pi(\Lambda\Lambda^{\mathsf{T}} + \Psi)\right| + \frac{1}{2}\mathbf{x}^{\mathsf{T}}(\Lambda\Lambda^{\mathsf{T}} + \Psi)^{-1}\mathbf{x}$$

Factor Analysis projections

Our analysis for PPCA still applies:

$$\tilde{\mathbf{x}}_n = \Lambda(I + \Lambda^{\mathsf{T}}\Psi^{-1}\Lambda)^{-1}\Lambda^{\mathsf{T}}\Psi^{-1}\mathbf{x}_n = \mathbf{x}_n - \Psi(\Lambda\Lambda^{\mathsf{T}} + \Psi)^{-1}\mathbf{x}_n$$

but now Ψ is diagonal but not spherical.

Note, though, that Λ is generally different from that found by PPCA.

And Λ is not unique: the latent space may be transformed by an arbitrary orthogonal transform U ($U^{\mathsf{T}}U = UU^{\mathsf{T}} = I$) without changing the likelihood.

$$\tilde{\mathbf{y}} = U\mathbf{y} \quad \text{and} \quad \tilde{\Lambda} = \Lambda U^{\mathsf{T}} \quad \Rightarrow \quad \tilde{\Lambda}\tilde{\mathbf{y}} = \Lambda U^{\mathsf{T}}U\mathbf{y} = \Lambda\mathbf{y}$$

$$-\ell = \frac{1}{2}\log\left|2\pi(\Lambda U^{\mathsf{T}}U\Lambda^{\mathsf{T}} + \Psi)\right| + \frac{1}{2}\mathbf{x}^{\mathsf{T}}(\Lambda U^{\mathsf{T}}U\Lambda^{\mathsf{T}} + \Psi)^{-1}\mathbf{x}$$

Factor Analysis projections

Our analysis for PPCA still applies:

$$\tilde{\mathbf{x}}_n = \Lambda(I + \Lambda^\top \Psi^{-1} \Lambda)^{-1} \Lambda^\top \Psi^{-1} \mathbf{x}_n = \mathbf{x}_n - \Psi(\Lambda\Lambda^\top + \Psi)^{-1} \mathbf{x}_n$$

but now Ψ is diagonal but not spherical.

Note, though, that Λ is generally different from that found by PPCA.

And Λ is not unique: the latent space may be transformed by an arbitrary orthogonal transform U ($U^\top U = UU^\top = I$) without changing the likelihood.

$$\tilde{\mathbf{y}} = U\mathbf{y} \quad \text{and} \quad \tilde{\Lambda} = \Lambda U^\top \quad \Rightarrow \quad \tilde{\Lambda}\tilde{\mathbf{y}} = \Lambda U^\top U \mathbf{y} = \Lambda \mathbf{y}$$

$$-\ell = \frac{1}{2} \log \left| 2\pi(\Lambda U^\top U \Lambda^\top + \Psi) \right| + \frac{1}{2} \mathbf{x}^\top (\Lambda U^\top U \Lambda^\top + \Psi)^{-1} \mathbf{x}$$

$$= \frac{1}{2} \log \left| 2\pi(\tilde{\Lambda}\tilde{\Lambda}^\top + \Psi) \right| + \frac{1}{2} \mathbf{x}^\top (\tilde{\Lambda}\tilde{\Lambda}^\top + \Psi)^{-1} \mathbf{x}$$

Factor analysis rotations

- FA (like many other latent methods) finds a subspace not a basis.
- Indeed, the columns of Λ need not be orthogonal.
- Many standard choices of basis:
  - Principal factors: orthogonalise columns in order of variance contribution to $\Lambda\Lambda^{\mathsf{T}}$ (analgous to PCA – achieved by eigendecomp of $\Lambda\Lambda^{\mathsf{T}}$ or equivalent SVD of Λ.
  - Varimax factors:

$$\underset{\Lambda:\Lambda^{\mathsf{T}}\Lambda=I}{\operatorname{argmax}} \left(\frac{1}{D} \sum_k \sum_d (\Lambda_{dk})^4 - \sum_k \left(\frac{1}{D} \sum_d \Lambda_{dk}^2 \right)^2 \right)$$

    sparse along columns, so each observation is explained by few factors.
  - Other rotations: Quartimax, Equimax, Oblimin, Promax . . . all consider loading pattern alone.
  - Independent components: usually formed from PCA sphered representation (assuming no noise), but noisy complete case could be seen as FA rotation.

FA vs PCA

- PCA and PPCA are rotationally invariant; FA is not

$$\text{If } \mathbf{x} \to U\mathbf{x} \text{ for unitary } U, \quad \text{then } \boldsymbol{\lambda}_{(i)}^{\text{PCA}} \to U\boldsymbol{\lambda}_{(i)}^{\text{PCA}}$$

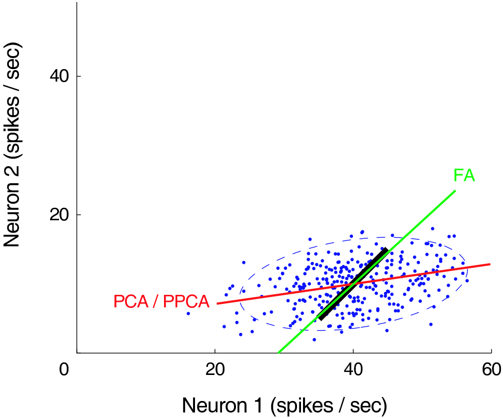
- FA is measurement scale invariant; PCA and PPCA are not

$$\text{If } \mathbf{x} \to S\mathbf{x} \text{ for diagonal } S, \quad \text{then } \boldsymbol{\lambda}_{(i)}^{\text{FA}} \to S\boldsymbol{\lambda}_{(i)}^{\text{FA}}$$

- FA and PPCA define a probabilistic model; PCA does not

[Note: it may be tempting to try to eliminate the scale-dependence of (P)PCA by pre-processing data to equalise total variance on each axis. But P(PCA) assume equal *noise* variance. Total variance has contributions from both $\Lambda\Lambda^{\mathsf{T}}$ and noise, so this approach does not exactly solve the problem.]

FA vs PCA for neural data



Non-Gaussian noise

Other noise models

- Both Gaussian noise, and mean-independent stationary variance, are unrealistic assumptions for spike counts, particularly in small bins
- Square-rooting improves matters, but is inaccurate for small counts and transforms the shape of the manifold.
- Instead: use a conditionally Poisson count distribution:
  - Poisson Factor Analysis
  - Exponential Family PCA
  - Covariance transformation

Likelihood-based approaches

One approach uses the following model:

$$p(\mathbf{y}) = \mathcal{N}(0, I)$$

$$p(\mathbf{x}|\mathbf{y}) = \prod_d \text{Poisson}[f([\Lambda\mathbf{y} + \mathbf{b}]_d)] = \prod_d \frac{f([\Lambda\mathbf{y} + \mathbf{b}]_d)^{x_d} e^{-f([\Lambda\mathbf{y}+\mathbf{b}]_d)}}{x_d!}$$

This is the Poisson noise equivalent of FA (note that we include an explicit "bias" **b** to control the mean of the generative distribution — it does not make sense to centre non-negative data).

Unfortunately, the E-step inference of $p(\mathbf{y}|\mathbf{x})$ has no simple closed form solution, and so true maximum likelihood learning is not tractable.

Instead, we can follow the steps of EM, but using an approximate estimate of the posterior. This is called a variational approximation.

Exponential Family PCA

$$p(\mathbf{x}|\mathbf{y}) = \prod_d \text{Poisson}[f([\Lambda \mathbf{y} + \mathbf{b}]_d)] = \prod_d \frac{f([\Lambda \mathbf{y} + \mathbf{b}]_d)^{x_d} e^{-f([\Lambda \mathbf{y}+\mathbf{b}]_d)}}{x_d!}$$

- Maximise likelihood over latents $\mathbf{y}$ and parameters $\Lambda, \mathbf{b}$ jointly.
- Convex if $f() \equiv \exp()$ (and other convex, log-concave functions).
- Noise model, but no uncertainty in latents — analagous to PCA.
- Can be seen as matrix factorisation (like SVD) with different cost function.
- Incorporating "nuclear norm" penalty (sum of singular values of ΛY finds low-rank log-rates while retaining convexity.

Covariance transformation

- Assume

$$\mathbf{x} \sim \text{Poisson}[\exp(\mathbf{z})]$$

and

$$\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$$

- Then we can compute the expected mean and covariance of $\mathbf{z}$ in terms of $\boldsymbol{\mu}$ and Σ in closed form.
- This relationship can be inverted to give Σ from the observed mean and covariance of the data.
- Can then perform PCA or factor analysis on Σ.

Dynamics

Dynamics

- Slow features analysis: SFA
- [Noise (and slowness)] Gaussian Process Factor Analysis : GPFA
- [Markov dynamics] Linear Gaussian State-Space Models: LGSSM
  also called (Hidden) Linear Dynamical Systems models: LDS.
  - related to the Kalman Filter
  - a particular 0 noise limit $\rightarrow$ SFA.
  - consistent spectral learning [Subspace Identification: SSID] possible, but inefficient.
- Poisson noise: PLDS
  - EM intractable – requires approximation.
  - SSID can be adapted exactly.

Gaussian process latents

$$\mathbf{x}(t) \sim \mathcal{GP}\left[\boldsymbol{\mu}(t); \mathsf{K}_{\boldsymbol{\theta}}(t, t')\right] \qquad \text{state model}$$

$$\mathbf{y}(t) \sim \mathit{Dist}\left[f(\mathbf{x}(t))\right] \qquad \text{observation model}$$

$\mathcal{GP}$ is a Gaussian process: this implies that any finite set of measurements at fixed times is jointly normal.

- Includes linear-Gaussian dynamical systems (LDS).

$$\mathbf{x}_t \sim \mathcal{N}\left(A\mathbf{x}_{t-1}, Q\right)$$

- Allows generalisation to non-(first-order-)Markov systems.

Gaussian process dynamics

$$\mathbf{x}(t) \sim \mathcal{GP}\left[\boldsymbol{\mu}(t); \mathsf{K}_{\boldsymbol{\theta}}(t, t')\right]$$
$$\mathbf{y}(t) \sim \mathit{Dist}\left[f(\mathbf{x}(t))\right]$$

- $K_{\boldsymbol{\theta}}(t, t')$ gives the covariance between values of $\mathbf{x}(t)$ and $\mathbf{x}(t')$.
- Parameterised by covariance. LDS (or auto-regressive models) are parameterised by precision (inverse covariance).
- Easier to specify priors wih interesting properties:

  – LDS: $\mathsf{K}(t, t') \propto a^{|t-t'|}$

  – Smooth: $\mathsf{K}(t, t') \propto \exp(-(t - t')^2/2\lambda)$

  – Oscillatory: $\mathsf{K}(t, t') \propto \sin(2\pi\omega(t - t'))$

  – Stationary “Brownian”: $\mathsf{K}(t, t') \propto [1 - |t - t'|/\lambda]^+$

- Inference naively $O(T^3)$ instead of $O(T)$.
  - Numerical methods based on regularities in matrices.
  - Sparsifying methods select (or create) subset of data with similar predictive power.

Link functions

$$\mathbf{x}(t) \sim \mathcal{GP}\left[\boldsymbol{\mu}(t); \mathsf{K}_{\boldsymbol{\theta}}(t, t')\right]$$
$$\mathbf{y}(t) \sim \mathit{Dist}\left[f(\mathbf{x}(t))\right]$$

f maps the latent GP values to (mean) intensity.

- Nonlinear
  - Exponential – danger: emphasises variability at high values.
  - Threshold-linear or soft-threshold.
- Linear
  - Requires observation model tolerant of negative values.
  - Alternatively, can use a truncated prior.
    - Requires approximation (but so does non-linearity).
    - Posterior often not far from Gaussian (multi-d truncation – draws are suprisingly smooth).
    - EP can be powerful approximation technique.

Observation models

$$\mathbf{x}(t) \sim \mathcal{GP}\left[\boldsymbol{\mu}(t); \mathsf{K}_{\boldsymbol{\theta}}(t, t')\right]$$
$$\mathbf{y}(t) \sim \textit{Dist}\left[f(\mathbf{x}(t))\right]$$

- Point process (continuous time)
  - Rescaled renewal process. (next)
  - Inhomogeneous Markov-interval.

$$\lambda(t) = f(\mathbf{x}(t), s_{last}) \qquad \big(\text{often } = f(\mathbf{x}(t)) \cdot h(s_{last})\big)$$

  - GM-like sum.

$$\lambda(t) = f\Big(\mathbf{x}(t) + \sum_i \alpha_i h(s_i)\Big)$$

- Spike count (discrete time)
  - Poisson counts.
  - (Square-rooted) Gaussian counts.

Examples

- Example 1: GP-based intensity estimates

  Cunningham, Yu, Shenoy, and Sahani. Inferring neural firing rates from spike trains using Gaussian processes. In *Adv. Neural Info. Proc. Sys. 20*, Cambridge, MA, 2008. MIT Press.
  Cunningham, Shenoy, and Sahani. Fast Gaussian process methods for point process intensity estimation. In *ICML '08*, pp. 192–199, Helsinki Finland, 2008. Omni Press.

- Example 2: Gaussian process factor analysis

  Yu, Cunningham, Santhanam, Ryu, Shenoy, and Sahani. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. *J. Neurophysiol.* 102: 614-635, 2009.

Example 1: GP-based intensity estimates

Spike train discretised in (arbitrarily small) time-bins.

$$\mathbf{x} \sim \mathcal{N}(\mu\mathbf{1}, \mathsf{K}_{\boldsymbol{\theta}})$$

$$p(\mathbf{y} \mid \mathbf{x}) = \prod_{i=1}^{N} \left[\frac{\gamma x_{y_i}}{\Gamma(\gamma)} \left(\gamma \sum_{k=y_{i-1}}^{y_i - 1} x_k \Delta \right)^{\gamma - 1} \exp\left\{ -\gamma \sum_{k=y_{i-1}}^{y_i - 1} x_k \Delta \right\} \right]$$

- This is a Gamma-interval process

$$p(\tau) = \frac{\gamma^{\gamma}}{\Gamma(\gamma)} \tau^{\gamma - 1} e^{-\gamma\tau}$$

with order γ and mean 1, with time rescaled according to GP rate.

Example 1: GP-based intensity estimates

Modal Inference:

$$\mathbf{x}^* = \underset{\mathbf{x} \succeq \mathbf{0}}{\operatorname{argmax}}\, p(\mathbf{x} \mid \mathbf{y}) = \underset{\mathbf{x} \succeq \mathbf{0}}{\operatorname{argmax}}\, p(\mathbf{y} \mid \mathbf{x}) p(\mathbf{x}).$$

- Note that the nonnegativity constraint eliminates need for a space warping link function (equivalent to truncated prior).
- Convex. Solve using a log barrier Newton Method.
- Computational complexity is a major challenge. We exploit problem structure to minimize run-time and memory requirements.

Example 1: GP-based intensity estimates

Learning:

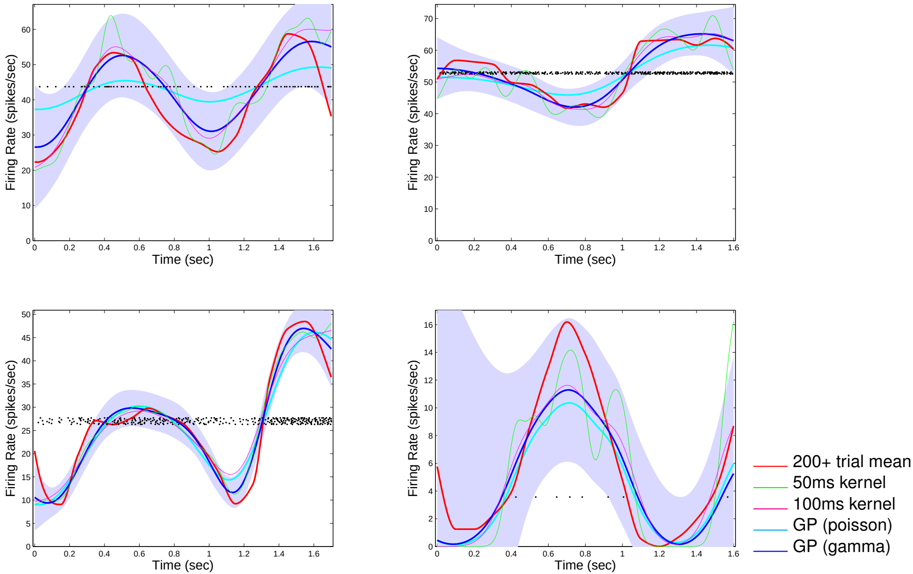
- The hyperparameters are $\theta = [\sigma_f^2, \kappa, \gamma, \mu]$ (where σ_f^2 and κ are the variance and lengthscale of the covariance kernel).
- Laplace approximation to approximate the intractable integral over **x**:

$$p(\mathbf{y} \mid \theta) = \int_{\mathbf{x}} p(\mathbf{y} \mid \mathbf{x}, \theta) p(\mathbf{x} \mid \theta) d\mathbf{x} \;\approx\; p(\mathbf{y} \mid \mathbf{x}^*, \theta) p(\mathbf{x}^* \mid \theta) \frac{(2\pi)^{\frac{n}{2}}}{|\Lambda^* + \mathsf{K}^{-1}|^{\frac{1}{2}}}$$

- This can be optimised to find “best” parameter values. Or can be used to weight different parameter values on a grid to integrate approximately over parameter settings.

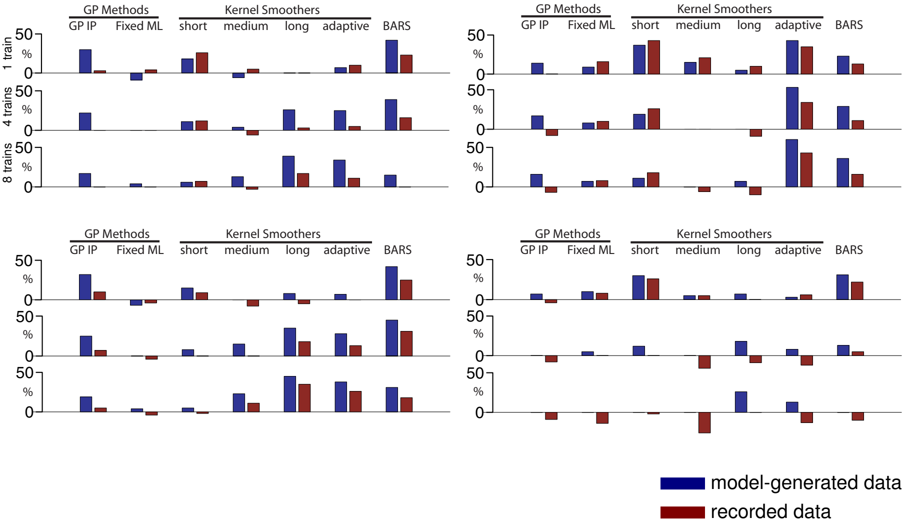
Example 1: GP-based intensity estimates

Results: (reconstructing simulated data)



Example 1: GP-based intensity estimates

Results: (percent improvement of full GP method over competitor)



Example 2: GPFA

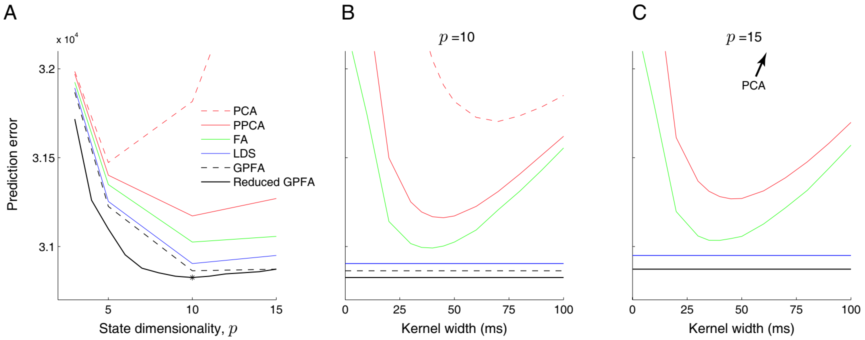
Spike train binned (10 – 20 ms) to yield spike counts.

$$x_i(t) \sim \mathcal{GP}[\mathbf{0}; K_i]$$

$$K_i(t_1, t_2) = (1 - \sigma_n^2) \exp\left(-\frac{(t_1 - t_2)^2}{2\tau_i^2}\right) + \sigma_n^2 \delta_{t_1, t_2}$$

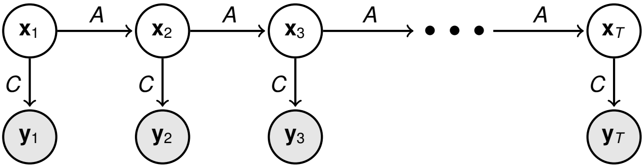
$$\mathbf{y}(t)|\mathbf{x}(t) \sim \mathcal{N}\left(C\mathbf{x}(t) + \mathbf{d}, R\right)$$

- Spike counts may be square-rooted to stabilise variance of (and Gaussianise) Poisson counts
- The model is jointly Gaussian! Exact inference and learning is possible using Factor-Analysis-like methods.


A
B
C
x 10^4
3.2
3.15
3.1
Prediction error
5
10
15
State dimensionality, p
PCA
PPCA
FA
LDS
GPFA
Reduced GPFA
p =10
0
25
50
75
100
Kernel width (ms)
p =15
PCA
0
25
50
75
100
Kernel width (ms)

Learning dynamics

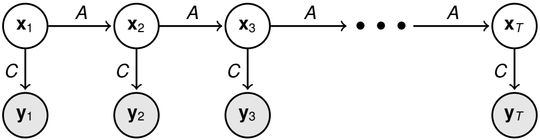
State space models.



$$\mathbf{x}_t|\mathbf{x}_{t-1} \sim \mathcal{N}\left(A\mathbf{x}_{t-1}, Q\right)$$
$$\mathbf{y}_t|\mathbf{x}_t \sim \mathcal{N}\left(C\mathbf{x}_t, R\right)$$

- Dynamics in latent space are self-contained.
- An **innovations** process introduces stochasticity, and allows inference and learning to compensate for model mismatch.
- Poisson, or other point-process observation models are not easy to handle. (But see Smith & Brown 2003, Yu et al. 2006, Macke et al. 2011, Buesing et al. 2012).

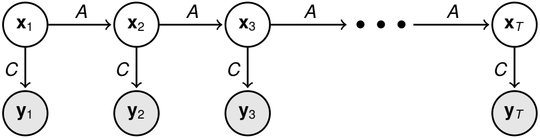
The Kalman Filter



$$
\begin{aligned}
{\color{blue}P(\mathbf{x}_t|\mathbf{y}_{1:t})} &= \int P(\mathbf{x}_t, \mathbf{x}_{t-1}|\mathbf{y}_t, \mathbf{y}_{1:t-1})\, d\mathbf{x}_{t-1} \\
&= \int \frac{P(\mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{y}_t|\mathbf{y}_{1:t-1})}{P(\mathbf{y}_t|\mathbf{y}_{1:t-1})}\, d\mathbf{x}_{t-1} \\
&\propto \int P(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) P(\mathbf{x}_t|\mathbf{x}_{t-1}, {\color{red}\mathbf{y}_{1:t-1}}) P(\mathbf{y}_t|\mathbf{x}_t, {\color{red}\mathbf{x}_{t-1}, \mathbf{y}_{1:t-1}})\, d\mathbf{x}_{t-1} \\
&\underset{\text{Markov property}}{=} \int {\color{blue}P(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})} P(\mathbf{x}_t|\mathbf{x}_{t-1}) P(\mathbf{y}_t|\mathbf{x}_t)\, d\mathbf{x}_{t-1}
\end{aligned}
$$

This is a **forward recursion** based on Bayes rule.

The Kalman Filter



Notation: $\hat{\mathbf{x}}_t^{\tau} \equiv E[\mathbf{x}_t|\mathbf{y}_1, \ldots, \mathbf{y}_\tau]$

Prediction: $\hat{\mathbf{x}}_t^{t-1} = A\hat{\mathbf{x}}_{t-1}^{t-1}$

Correction: $\hat{\mathbf{x}}_t^{t} = \hat{\mathbf{x}}_t^{t-1} + K_t(\mathbf{y}_t - C\hat{\mathbf{x}}_t^{t-1})$

Kalman gain: $K_t = \hat{V}_t^{t-1} C^\top (C\hat{V}_t^{t-1} C^\top + R)^{-1}$

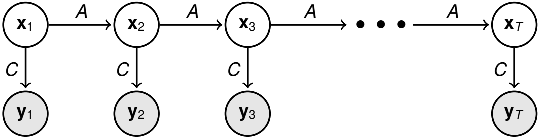
Prediction variance: $\hat{V}_t^{t-1} = A\hat{V}_{t-1}^{t-1} A^\top + Q$

Corrected variance: $\hat{V}_t^{t} = \hat{V}_t^{t-1} - K_t C \hat{V}_t^{t-1}$

To get these equations we need the Gaussian integral: $\int e^{-\frac{1}{2}(\mathbf{x}-\mu)^\top \Sigma^{-1}(\mathbf{x}-\mu)} d\mathbf{x} = |2\pi\Sigma|^{1/2}$

and the Matrix Inversion Lemma: $(\Phi + \Lambda\Psi\Lambda^\top)^{-1} = \Phi^{-1} - \Phi^{-1}\Lambda(\Psi^{-1} + \Lambda^\top\Phi^{-1}\Lambda)^{-1}\Lambda^\top\Phi^{-1}$ assuming Φ and Ψ are symmetric and invertible.

The Kalman Smoother



$$P(\mathbf{x}_t|\mathbf{y}_{1:\tau}) = \int P(\mathbf{x}_t, \mathbf{x}_{t+1}|\mathbf{y}_{1:\tau})\, d\mathbf{x}_{t+1}$$

$$= \int P(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{y}_{1:\tau})(\mathbf{x}_{t+1}|\mathbf{y}_{1:\tau})\, d\mathbf{x}_{t+1}$$

$$\underset{\text{Markov property}}{=} \int P(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{y}_{1:t})(\mathbf{x}_{t+1}|\mathbf{y}_{1:\tau})\, d\mathbf{x}_{t+1}$$

Additional **backward recursion**:

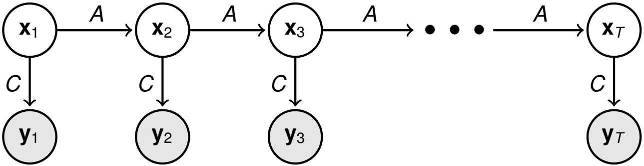
$$J_t = \hat{V}_t^t A^\top (\hat{V}_{t+1}^t)^{-1}$$

$$\hat{\mathbf{x}}_t^\tau = \hat{\mathbf{x}}_t^t + J_t(\hat{\mathbf{x}}_{t+1}^\tau - A\hat{\mathbf{x}}_t^t)$$

$$\hat{V}_t^\tau = \hat{V}_t^t + J_t(\hat{V}_{t+1}^\tau - \hat{V}_{t+1}^t)J_t^\top$$

The Kalman filter

For a Gaussian SSM, the Kalman filter finds the expected latent state.



- Model likelihood can be computed from filtered expected state and variance.

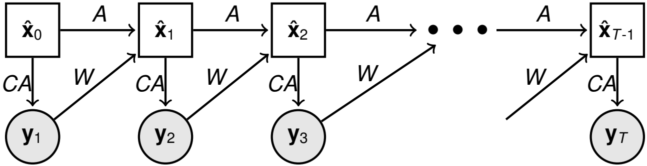
$$P(\mathbf{y}_1 \ldots \mathbf{y}_T) = P(\mathbf{y}_1) \prod_{t=2}^{T} P(\mathbf{y}_t | \mathbf{y}_1 \ldots \mathbf{y}_{t-1})$$

$$\begin{aligned} P(\mathbf{y}_{t+1} | \mathbf{y}_1 \ldots \mathbf{y}_t) &= \int d\mathbf{x}_{t+1} \; P(\mathbf{y}_{t+1} | \mathbf{x}_{t+1}) P(\mathbf{x}_{t+1} | \mathbf{y}_1 \ldots \mathbf{y}_t) \\ &= \int d\mathbf{x}_{t+1} \; \mathcal{N}(\mathbf{y}_{t+1} | C\mathbf{x}_{t+1}, R) \; \mathcal{N}(\mathbf{x}_{t+1} | A\hat{\mathbf{x}}_t, V_{t+1}) \\ &= \mathcal{N}(\mathbf{y}_{t+1} | CA\hat{\mathbf{x}}_t, CV_{t+1}C^{\mathsf{T}} + R) \,, \end{aligned}$$

- K_t and V_t converge to stationary values.

Recurrent Linear Models

The RLM parametrises the likelihood with a stationary feedback gain:



- Learning by direct gradient ascent: backpropagation through time.
- For Gaussian SSM data converges to equivalent model – learns the Kalman filter directly.
- Generalisation to Poisson (or other point process) output is remains tractable with stable learning.
- Not identical to Poisson-output SSM, but empirically close.

Supervised methods

Not so latent variables

- Controlled experiments use repeated trials
  - One or more experimental parameter or factor varied systematically.
  - Each unique configuration of factors is a condition.
- May also observe (generally continuous-valued) behavioural outputs or a random/natural stimulus: covariates.
- Ideally, unsupervised structure in data would reflect these values.
- Weak signals? Non-linearities?
- Unsupervised projections may not naturally separate the different factors: unmixing.

- We will look at supervised methods designed to relate multivariate data to known experimental factors or covariates.
- Methods we consider are also used to study structure in the condition averages: equivalent to having one trial per condition
  - averaging may make noise more Gaussian
  - **but still not equal variance**

Two cases

The tools needed in two different cases are slightly different:

- Categorical factors: discrete repeated values (almost always experimental control).
  - Stimulus (say, object) identity.
  - Behavioural instruction.
  - “Context” signal.
  - We sometimes ignore the metricity of factors: time bin, gabor orientation, . . .
- Continuous or ordinal covariates: experimental factors or covariates themselves lie in a metric space.
  - time in trial
  - orientation
  - reaching movement kinematics

Categorical factors: decomposition of variance

Suppose on *i*th trial we have:

- factor value $k^{(i)} \in 1 \ldots K$
- recorded (binned) data $\mathbf{x}_t^{(i)} \in \mathbb{R}^N$, $t = [1 \ldots T]$, $N = \#$ neurons; remove global mean.

Consider time t.

- For each condition κ we have the condition mean (PSTH): $\bar{\mathbf{x}}_t^{(\kappa)} = \left\langle \mathbf{x}_t^{(i)} \right\rangle_{i:k^{(i)}=\kappa}$
- Let us write $\mathbf{x}_t^{(i)} = \bar{\mathbf{x}}_t^{(k^{(i)})} + \Delta\mathbf{x}_t^{(i)}$.
- Then total scatter or variance:

$$S_t = \left\langle \mathbf{x}_t^{(i)} \mathbf{x}_t^{(i)\mathsf{T}} \right\rangle = \left\langle (\bar{\mathbf{x}}_t^{(k^{(i)})} + \Delta\mathbf{x}_t^{(i)})(\bar{\mathbf{x}}_t^{(k^{(i)})} + \Delta\mathbf{x}_t^{(i)})^{\mathsf{T}} \right\rangle$$

Categorical factors: decomposition of variance

Suppose on ith trial we have:

- factor value $k^{(i)} \in 1 \dots K$
- recorded (binned) data $\mathbf{x}_t^{(i)} \in \mathbb{R}^N$, $t = [1 \dots T]$, $N =$ # neurons; remove global mean.

Consider time t.

- For each condition κ we have the condition mean (PSTH): $\bar{\mathbf{x}}_t^{(\kappa)} = \left\langle \mathbf{x}_t^{(i)} \right\rangle_{i:k^{(i)}=\kappa}$
- Let us write $\mathbf{x}_t^{(i)} = \bar{\mathbf{x}}_t^{(k^{(i)})} + \Delta\mathbf{x}_t^{(i)}$.
- Then total scatter or variance:

$$S_t = \left\langle \mathbf{x}_t^{(i)} \mathbf{x}_t^{(i)\mathsf{T}} \right\rangle = \left\langle (\bar{\mathbf{x}}_t^{(k^{(i)})} + \Delta\mathbf{x}_t^{(i)})(\bar{\mathbf{x}}_t^{(k^{(i)})} + \Delta\mathbf{x}_t^{(i)})^\mathsf{T} \right\rangle$$

$$= \left\langle \left\langle (\bar{\mathbf{x}}_t^{(\kappa)} + \Delta\mathbf{x}_t^{(i)})(\bar{\mathbf{x}}_t^{(\kappa)} + \Delta\mathbf{x}_t^{(i)})^\mathsf{T} \right\rangle_{i:k^{(i)}=\kappa} \right\rangle_\kappa$$

Categorical factors: decomposition of variance

Suppose on ith trial we have:

- factor value $k^{(i)} \in 1 \ldots K$
- recorded (binned) data $\mathbf{x}_t^{(i)} \in \mathbb{R}^N$, $t = [1 \ldots T]$, $N =$ # neurons; remove global mean.

Consider time t.

- For each condition κ we have the condition mean (PSTH): $\bar{\mathbf{x}}_t^{(\kappa)} = \left\langle \mathbf{x}_t^{(i)} \right\rangle_{i:k^{(i)}=\kappa}$
- Let us write $\mathbf{x}_t^{(i)} = \bar{\mathbf{x}}_t^{(k^{(i)})} + \Delta\mathbf{x}_t^{(i)}$.
- Then total scatter or variance:

$$\begin{aligned}
S_t &= \left\langle \mathbf{x}_t^{(i)} \mathbf{x}_t^{(i)\mathsf{T}} \right\rangle = \left\langle (\bar{\mathbf{x}}_t^{(k^{(i)})} + \Delta\mathbf{x}_t^{(i)})(\bar{\mathbf{x}}_t^{(k^{(i)})} + \Delta\mathbf{x}_t^{(i)})^\mathsf{T} \right\rangle \\
&= \left\langle \left\langle (\bar{\mathbf{x}}_t^{(\kappa)} + \Delta\mathbf{x}_t^{(i)})(\bar{\mathbf{x}}_t^{(\kappa)} + \Delta\mathbf{x}_t^{(i)})^\mathsf{T} \right\rangle_{i:k^{(i)}=\kappa} \right\rangle_\kappa \\
&= \left\langle \left\langle \bar{\mathbf{x}}_t^{(\kappa)} \bar{\mathbf{x}}_t^{(\kappa)\mathsf{T}} - \bar{\mathbf{x}}_t^{(\kappa)} \Delta\mathbf{x}_t^{(i)\mathsf{T}} - \Delta\mathbf{x}_t^{(i)} \bar{\mathbf{x}}_t^{(\kappa)\mathsf{T}} + \Delta\mathbf{x}_t^{(i)} \Delta\mathbf{x}_t^{(i)\mathsf{T}} \right\rangle_{i:k^{(i)}=\kappa} \right\rangle_\kappa
\end{aligned}$$

Categorical factors: decomposition of variance

Suppose on ith trial we have:

- factor value $k^{(i)} \in 1 \dots K$
- recorded (binned) data $\mathbf{x}_t^{(i)} \in \mathbb{R}^N$, $t = [1 \dots T]$, $N =$ # neurons; remove global mean.

Consider time t.

- For each condition κ we have the condition mean (PSTH): $\bar{\mathbf{x}}_t^{(\kappa)} = \left\langle \mathbf{x}_t^{(i)} \right\rangle_{i:k^{(i)}=\kappa}$
- Let us write $\mathbf{x}_t^{(i)} = \bar{\mathbf{x}}_t^{(k^{(i)})} + \Delta\mathbf{x}_t^{(i)}$.
- Then total scatter or variance:

$$
\begin{aligned}
S_t &= \left\langle \mathbf{x}_t^{(i)} \mathbf{x}_t^{(i)\mathsf{T}} \right\rangle = \left\langle (\bar{\mathbf{x}}_t^{(k^{(i)})} + \Delta\mathbf{x}_t^{(i)})(\bar{\mathbf{x}}_t^{(k^{(i)})} + \Delta\mathbf{x}_t^{(i)})^\mathsf{T} \right\rangle \\
&= \left\langle \left\langle (\bar{\mathbf{x}}_t^{(\kappa)} + \Delta\mathbf{x}_t^{(i)})(\bar{\mathbf{x}}_t^{(\kappa)} + \Delta\mathbf{x}_t^{(i)})^\mathsf{T} \right\rangle_{i:k^{(i)}=\kappa} \right\rangle_\kappa \\
&= \left\langle \left\langle \bar{\mathbf{x}}_t^{(\kappa)} \bar{\mathbf{x}}_t^{(\kappa)\mathsf{T}} - \bar{\mathbf{x}}_t^{(\kappa)} \Delta\mathbf{x}_t^{(i)\mathsf{T}} - \Delta\mathbf{x}_t^{(i)} \bar{\mathbf{x}}_t^{(\kappa)\mathsf{T}} + \Delta\mathbf{x}_t^{(i)} \Delta\mathbf{x}_t^{(i)\mathsf{T}} \right\rangle_{i:k^{(i)}=\kappa} \right\rangle_\kappa \\
&= \left\langle \left\langle \bar{\mathbf{x}}_t^{(\kappa)} \bar{\mathbf{x}}_t^{(\kappa)\mathsf{T}} \right\rangle - \left\langle \bar{\mathbf{x}}_t^{(\kappa)} \Delta\mathbf{x}_t^{(i)\mathsf{T}} \right\rangle - \left\langle \Delta\mathbf{x}_t^{(i)} \bar{\mathbf{x}}_t^{(\kappa)\mathsf{T}} \right\rangle + \left\langle \Delta\mathbf{x}_t^{(i)} \Delta\mathbf{x}_t^{(i)\mathsf{T}} \right\rangle \right\rangle_\kappa
\end{aligned}
$$

Categorical factors: decomposition of variance

Suppose on ith trial we have:

- factor value $k^{(i)} \in 1 \ldots K$
- recorded (binned) data $\mathbf{x}_t^{(i)} \in \mathbb{R}^N$, $t = [1 \ldots T]$, $N =$ # neurons; remove global mean.

Consider time t.

- For each condition κ we have the condition mean (PSTH): $\bar{\mathbf{x}}_t^{(\kappa)} = \left\langle \mathbf{x}_t^{(i)} \right\rangle_{i:k^{(i)}=\kappa}$
- Let us write $\mathbf{x}_t^{(i)} = \bar{\mathbf{x}}_t^{(k^{(i)})} + \Delta\mathbf{x}_t^{(i)}$.
- Then total scatter or variance:

$$
\begin{aligned}
S_t &= \left\langle \mathbf{x}_t^{(i)} \mathbf{x}_t^{(i)\mathsf{T}} \right\rangle = \left\langle (\bar{\mathbf{x}}_t^{(k^{(i)})} + \Delta\mathbf{x}_t^{(i)})(\bar{\mathbf{x}}_t^{(k^{(i)})} + \Delta\mathbf{x}_t^{(i)})^{\mathsf{T}} \right\rangle \\
&= \left\langle \left\langle (\bar{\mathbf{x}}_t^{(\kappa)} + \Delta\mathbf{x}_t^{(i)})(\bar{\mathbf{x}}_t^{(\kappa)} + \Delta\mathbf{x}_t^{(i)})^{\mathsf{T}} \right\rangle_{i:k^{(i)}=\kappa} \right\rangle_{\kappa} \\
&= \left\langle \left\langle \bar{\mathbf{x}}_t^{(\kappa)} \bar{\mathbf{x}}_t^{(\kappa)\mathsf{T}} - \bar{\mathbf{x}}_t^{(\kappa)} \Delta\mathbf{x}_t^{(i)\mathsf{T}} - \Delta\mathbf{x}_t^{(i)} \bar{\mathbf{x}}_t^{(\kappa)\mathsf{T}} + \Delta\mathbf{x}_t^{(i)} \Delta\mathbf{x}_t^{(i)\mathsf{T}} \right\rangle_{i:k^{(i)}=\kappa} \right\rangle_{\kappa} \\
&= \left\langle \left\langle \bar{\mathbf{x}}_t^{(\kappa)} \bar{\mathbf{x}}_t^{(\kappa)\mathsf{T}} \right\rangle - \left\langle \bar{\mathbf{x}}_t^{(\kappa)} \Delta\mathbf{x}_t^{(i)\mathsf{T}} \right\rangle - \left\langle \Delta\mathbf{x}_t^{(i)} \bar{\mathbf{x}}_t^{(\kappa)\mathsf{T}} \right\rangle + \left\langle \Delta\mathbf{x}_t^{(i)} \Delta\mathbf{x}_t^{(i)\mathsf{T}} \right\rangle \right\rangle_{\kappa} \\
&= \left\langle \bar{\mathbf{x}}_t^{(\kappa)} \bar{\mathbf{x}}_t^{(\kappa)\mathsf{T}} - \bar{\mathbf{x}}_t^{(\kappa)} \left\langle \Delta\mathbf{x}_t^{(i)} \right\rangle^{\mathsf{T}} - \left\langle \Delta\mathbf{x}_t^{(i)} \right\rangle \bar{\mathbf{x}}_t^{(\kappa)\mathsf{T}} + \left\langle \Delta\mathbf{x}_t^{(i)} \Delta\mathbf{x}_t^{(i)\mathsf{T}} \right\rangle \right\rangle_{\kappa}
\end{aligned}
$$

Categorical factors: decomposition of variance

Suppose on ith trial we have:

- factor value $k^{(i)} \in 1 \dots K$
- recorded (binned) data $\mathbf{x}_t^{(i)} \in \mathbb{R}^N$, $t = [1 \dots T]$, $N = \#$ neurons; remove global mean.

Consider time t.

- For each condition κ we have the condition mean (PSTH): $\bar{\mathbf{x}}_t^{(\kappa)} = \left\langle \mathbf{x}_t^{(i)} \right\rangle_{i:k^{(i)}=\kappa}$
- Let us write $\mathbf{x}_t^{(i)} = \bar{\mathbf{x}}_t^{(k^{(i)})} + \Delta\mathbf{x}_t^{(i)}$.
- Then total scatter or variance:

$$\begin{aligned}
S_t &= \left\langle \mathbf{x}_t^{(i)} \mathbf{x}_t^{(i)\mathsf{T}} \right\rangle = \left\langle (\bar{\mathbf{x}}_t^{(k^{(i)})} + \Delta\mathbf{x}_t^{(i)})(\bar{\mathbf{x}}_t^{(k^{(i)})} + \Delta\mathbf{x}_t^{(i)})^{\mathsf{T}} \right\rangle \\
&= \left\langle \left\langle (\bar{\mathbf{x}}_t^{(\kappa)} + \Delta\mathbf{x}_t^{(i)})(\bar{\mathbf{x}}_t^{(\kappa)} + \Delta\mathbf{x}_t^{(i)})^{\mathsf{T}} \right\rangle_{i:k^{(i)}=\kappa} \right\rangle_{\kappa} \\
&= \left\langle \left\langle \bar{\mathbf{x}}_t^{(\kappa)} \bar{\mathbf{x}}_t^{(\kappa)\mathsf{T}} - \bar{\mathbf{x}}_t^{(\kappa)} \Delta\mathbf{x}_t^{(i)\mathsf{T}} - \Delta\mathbf{x}_t^{(i)} \bar{\mathbf{x}}_t^{(\kappa)\mathsf{T}} + \Delta\mathbf{x}_t^{(i)} \Delta\mathbf{x}_t^{(i)\mathsf{T}} \right\rangle_{i:k^{(i)}=\kappa} \right\rangle_{\kappa} \\
&= \left\langle \left\langle \bar{\mathbf{x}}_t^{(\kappa)} \bar{\mathbf{x}}_t^{(\kappa)\mathsf{T}} \right\rangle - \left\langle \bar{\mathbf{x}}_t^{(\kappa)} \Delta\mathbf{x}_t^{(i)\mathsf{T}} \right\rangle - \left\langle \Delta\mathbf{x}_t^{(i)} \bar{\mathbf{x}}_t^{(\kappa)\mathsf{T}} \right\rangle + \left\langle \Delta\mathbf{x}_t^{(i)} \Delta\mathbf{x}_t^{(i)\mathsf{T}} \right\rangle \right\rangle_{\kappa} \\
&= \left\langle \bar{\mathbf{x}}_t^{(\kappa)} \bar{\mathbf{x}}_t^{(\kappa)\mathsf{T}} - \bar{\mathbf{x}}_t^{(\kappa)} \left\langle \Delta\mathbf{x}_t^{(i)} \right\rangle^{\mathsf{T}} - \left\langle \Delta\mathbf{x}_t^{(i)} \right\rangle \bar{\mathbf{x}}_t^{(\kappa)\mathsf{T}} + \left\langle \Delta\mathbf{x}_t^{(i)} \Delta\mathbf{x}_t^{(i)\mathsf{T}} \right\rangle \right\rangle_{\kappa} \\
&= \left\langle \bar{\mathbf{x}}_t^{(\kappa)} \bar{\mathbf{x}}_t^{(\kappa)\mathsf{T}} \right\rangle_{\kappa} + \left\langle \left\langle \Delta\mathbf{x}_t^{(i)} \Delta\mathbf{x}_t^{(i)\mathsf{T}} \right\rangle_{i:k^{(i)}=\kappa} \right\rangle_{\kappa}
\end{aligned}$$

Categorical factors: decomposition of variance

Suppose on ith trial we have:

- factor value $k^{(i)} \in 1 \ldots K$
- recorded (binned) data $\mathbf{x}_t^{(i)} \in \mathbb{R}^N$, $t = [1 \ldots T]$, $N =$ # neurons; remove global mean.

Consider time t.

- For each condition κ we have the condition mean (PSTH): $\bar{\mathbf{x}}_t^{(\kappa)} = \left\langle \mathbf{x}_t^{(i)} \right\rangle_{i:k^{(i)}=\kappa}$
- Let us write $\mathbf{x}_t^{(i)} = \bar{\mathbf{x}}_t^{(k^{(i)})} + \Delta\mathbf{x}_t^{(i)}$.
- Then total scatter or variance:

$$
\begin{aligned}
S_t &= \left\langle \mathbf{x}_t^{(i)} \mathbf{x}_t^{(i)\mathsf{T}} \right\rangle = \left\langle (\bar{\mathbf{x}}_t^{(k^{(i)})} + \Delta\mathbf{x}_t^{(i)})(\bar{\mathbf{x}}_t^{(k^{(i)})} + \Delta\mathbf{x}_t^{(i)})^{\mathsf{T}} \right\rangle \\
&= \left\langle \left\langle (\bar{\mathbf{x}}_t^{(\kappa)} + \Delta\mathbf{x}_t^{(i)})(\bar{\mathbf{x}}_t^{(\kappa)} + \Delta\mathbf{x}_t^{(i)})^{\mathsf{T}} \right\rangle_{i:k^{(i)}=\kappa} \right\rangle_{\kappa} \\
&= \left\langle \left\langle \bar{\mathbf{x}}_t^{(\kappa)} \bar{\mathbf{x}}_t^{(\kappa)\mathsf{T}} - \bar{\mathbf{x}}_t^{(\kappa)} \Delta\mathbf{x}_t^{(i)\mathsf{T}} - \Delta\mathbf{x}_t^{(i)} \bar{\mathbf{x}}_t^{(\kappa)\mathsf{T}} + \Delta\mathbf{x}_t^{(i)} \Delta\mathbf{x}_t^{(i)\mathsf{T}} \right\rangle_{i:k^{(i)}=\kappa} \right\rangle_{\kappa} \\
&= \left\langle \left\langle \bar{\mathbf{x}}_t^{(\kappa)} \bar{\mathbf{x}}_t^{(\kappa)\mathsf{T}} \right\rangle - \left\langle \bar{\mathbf{x}}_t^{(\kappa)} \Delta\mathbf{x}_t^{(i)\mathsf{T}} \right\rangle - \left\langle \Delta\mathbf{x}_t^{(i)} \bar{\mathbf{x}}_t^{(\kappa)\mathsf{T}} \right\rangle + \left\langle \Delta\mathbf{x}_t^{(i)} \Delta\mathbf{x}_t^{(i)\mathsf{T}} \right\rangle \right\rangle_{\kappa} \\
&= \left\langle \bar{\mathbf{x}}_t^{(\kappa)} \bar{\mathbf{x}}_t^{(\kappa)\mathsf{T}} - \bar{\mathbf{x}}_t^{(\kappa)} \left\langle \Delta\mathbf{x}_t^{(i)} \right\rangle^{\mathsf{T}} - \left\langle \Delta\mathbf{x}_t^{(i)} \right\rangle \bar{\mathbf{x}}_t^{(\kappa)\mathsf{T}} + \left\langle \Delta\mathbf{x}_t^{(i)} \Delta\mathbf{x}_t^{(i)\mathsf{T}} \right\rangle \right\rangle_{\kappa} \\
&= \underbrace{\left\langle \bar{\mathbf{x}}_t^{(\kappa)} \bar{\mathbf{x}}_t^{(\kappa)\mathsf{T}} \right\rangle_{\kappa}}_{\text{Var(cond. mean)}} + \underbrace{\left\langle \left\langle \Delta\mathbf{x}_t^{(i)} \Delta\mathbf{x}_t^{(i)\mathsf{T}} \right\rangle_{i:k^{(i)}=\kappa} \right\rangle_{\kappa}}_{\text{Mean(cond. var)}} = S_t^{(\text{signal})} + S_t^{(\text{noise})}
\end{aligned}
$$

Multifactor decomposition of variance

We can consider time bin t to be another factor (and may have may experimental factors). Write

- $\bar{\mathbf{x}}_t = \left\langle \mathbf{x}_t^{(i)} \right\rangle_i$
- $\bar{\mathbf{x}}^{(\kappa)} = \left\langle \mathbf{x}_t^{(i)} \right\rangle_{t,i:k^{(i)}=\kappa}$
- $\Delta\bar{\mathbf{x}}_t^{(\kappa)} = \bar{\mathbf{x}}_t^{(\kappa)} - \bar{\mathbf{x}}_t - \bar{\mathbf{x}}^{(\kappa)}$

Then

$$\begin{aligned} S^{(\text{total})} &= \left\langle \mathbf{x}_t^{(i)} \mathbf{x}_t^{(i)\mathsf{T}} \right\rangle_{t,i} = \left\langle (\bar{\mathbf{x}}_t + \bar{\mathbf{x}}^{(k^{(i)})} + \Delta\bar{\mathbf{x}}_t^{(k^{(i)})} + \Delta\mathbf{x}_t^{(i)})(\bar{\mathbf{x}}_t + \bar{\mathbf{x}}^{(k^{(i)})} + \Delta\bar{\mathbf{x}}_t^{(k^{(i)})} + \Delta\mathbf{x}_t^{(i)})^{\mathsf{T}} \right\rangle \\ &= \left\langle \bar{\mathbf{x}}_t \bar{\mathbf{x}}_t^{\mathsf{T}} \right\rangle_t + \left\langle \bar{\mathbf{x}}^{(\kappa)} \bar{\mathbf{x}}^{(\kappa)\mathsf{T}} \right\rangle_\kappa + \left\langle \Delta\bar{\mathbf{x}}_t^{(\kappa)} \Delta\bar{\mathbf{x}}_t^{(\kappa)\mathsf{T}} \right\rangle_{t,\kappa} + \left\langle \Delta\mathbf{x}_t^{(i)} \Delta\mathbf{x}_t^{(i)\mathsf{T}} \right\rangle_{t,i} \\ &= S^{(\text{time})} + S^{(\text{factor})} + S^{(\text{interact})} + S^{(\text{noise})} \end{aligned}$$

In general, for multiple factors:

$$\begin{aligned} S^{(\text{total})} = S^{(t)} &+ S^{(f_1)} + S^{(f_2)} + \dots \\ &+ S^{(t \times f_1)} + S^{(t \times f_2)} + S^{(f_1 \times f_2)} + \dots \\ &+ S^{(t \times f_1 \times f_2)} + \dots + S^{(t \times f_1 \times f_2 \times \dots)} + \dots \\ &+ S^{(\text{noise})} \end{aligned}$$

This decomposition is fundamental to the Multivariate Analysis of Variance (MANOVA).

Studying factor-related variance

The idea behind our first group of methods is to look for a projection of the data that captures the structure related to one factor at a time.

Studying factor-related variance

The idea behind our first group of methods is to look for a projection of the data that captures the structure related to one factor at a time.

- A first thought: Use PCA / FA / etc. on the condition means.

Studying factor-related variance

The idea behind our first group of methods is to look for a projection of the data that captures the structure related to one factor at a time.

- A first thought: Use PCA / FA / etc. on the condition means.
  - Maximises projected signal variance, but does not reject variance from trial-to-trial noise, or from other factors (unmixing).

Studying factor-related variance

The idea behind our first group of methods is to look for a projection of the data that captures the structure related to one factor at a time.

- A first thought: Use PCA / FA / etc. on the condition means.
  - Maximises projected signal variance, but does not reject variance from trial-to-trial noise, or from other factors (unmixing).
- Rotate the projection vectors so as to find a good compromise between retaining variance related to signal and avoiding other sources.

Studying factor-related variance

The idea behind our first group of methods is to look for a projection of the data that captures the structure related to one factor at a time.

- A first thought: Use PCA / FA / etc. on the condition means.
  - Maximises projected signal variance, but does not reject variance from trial-to-trial noise, or from other factors (unmixing).

- Rotate the projection vectors so as to find a good compromise between retaining variance related to signal and avoiding other sources.
  Two ideas:

Studying factor-related variance

The idea behind our first group of methods is to look for a projection of the data that captures the structure related to one factor at a time.

- A first thought: Use PCA / FA / etc. on the condition means.
  - Maximises projected signal variance, but does not reject variance from trial-to-trial noise, or from other factors (unmixing).

- Rotate the projection vectors so as to find a good compromise between retaining variance related to signal and avoiding other sources.
  Two ideas:
  - Maximise projected signal to noise ratio.

Studying factor-related variance

The idea behind our first group of methods is to look for a projection of the data that captures the structure related to one factor at a time.

- A first thought: Use PCA / FA / etc. on the condition means.
  - Maximises projected signal variance, but does not reject variance from trial-to-trial noise, or from other factors (unmixing).

- Rotate the projection vectors so as to find a good compromise between retaining variance related to signal and avoiding other sources.
  Two ideas:
  - Maximise projected signal to noise ratio.
  - Minimise error between reconstructed trial and signal.

Studying factor-related variance

The idea behind our first group of methods is to look for a projection of the data that captures the structure related to one factor at a time.

- A first thought: Use PCA / FA / etc. on the condition means.
  - Maximises projected signal variance, but does not reject variance from trial-to-trial noise, or from other factors (unmixing).

- Rotate the projection vectors so as to find a good compromise between retaining variance related to signal and avoiding other sources.
  Two ideas:
  - Maximise projected signal to noise ratio.
  - Minimise error between reconstructed trial and signal.

We consider one factor at a time: $S^{(\text{total})} = S^{(\text{factor})} + S^{(\text{other})} = S_F + S_\Delta$.

Linear Discriminant Analysis (LDA)

Originally due to Fisher (1936), widely discussed in text books.

$$\text{Find } \mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmax}} \frac{\mathbf{w}^\mathsf{T} S_F \mathbf{w}}{\mathbf{w}^\mathsf{T} S_\Delta \mathbf{w}}$$

In this context, S_F is usually called between-class scatter – scatter between condition means. S_Δ is the average within-class scatter.

The projection is (heuristically) designed to maximise separation of the classes.

[The same idea, slightly generalised, has been discussed in neuroscience as "Denoising Source Separation" (Simon and de Cheveigné) and "Joint Decorrelation" (de Cheveigné and Parra).]

Linear Discriminant Analysis (LDA)

First note that $\frac{\mathbf{w}^{\mathsf{T}} S_F \mathbf{w}}{\mathbf{w}^{\mathsf{T}} S_\Delta \mathbf{w}} = \frac{\mathbf{w}^{\mathsf{T}} S_\Delta^{1/2} S_\Delta^{-1/2} S_F S_\Delta^{-1/2} S_\Delta^{1/2} \mathbf{w}}{\mathbf{w}^{\mathsf{T}} S_\Delta^{1/2} S_\Delta^{1/2} \mathbf{w}}$ so that we can define $\tilde{\mathbf{w}} = S_\Delta^{1/2} \mathbf{w}$ and find

$$\tilde{\mathbf{w}}^* = \underset{\tilde{\mathbf{w}}}{\operatorname{argmax}} \frac{\tilde{\mathbf{w}}^{\mathsf{T}} S_\Delta^{-1/2} S_F S_\Delta^{-1/2} \tilde{\mathbf{w}}}{\tilde{\mathbf{w}}^{\mathsf{T}} \tilde{\mathbf{w}}} = \underset{\|\tilde{\mathbf{w}}\|=1}{\operatorname{argmax}} \tilde{\mathbf{w}}^{\mathsf{T}} S_\Delta^{-1/2} S_F S_\Delta^{-1/2} \tilde{\mathbf{w}}$$

finally mapping back to obtain $\mathbf{w}^* = S_\Delta^{-1/2} \tilde{\mathbf{w}}^*$.

It may be easiest to think of this as a two-stage process:

- Whiten the non-factor scatter (transform data to $\tilde{\mathbf{x}}_t^{(i)} = S_\Delta^{-1/2} \mathbf{x}_t^{(i)}$), so that $\widetilde{S}_\Delta = I$.
- Run PCA on the means $\bar{\mathbf{x}}^{(\kappa)}$ in the whitened space; diagonalising $\widetilde{S}_F = S_\Delta^{-1/2} S_F S_\Delta^{-1/2}$.

  $\Rightarrow \widetilde{S}_F \tilde{\mathbf{w}}^* = \lambda \tilde{\mathbf{w}}^*$

  $\Rightarrow S_\Delta^{-1/2} S_F S_\Delta^{-1/2} S_\Delta^{1/2} \mathbf{w}^* = \lambda S_\Delta^{1/2} \mathbf{w}^*$

  $\Rightarrow S_\Delta^{-1} S_F \mathbf{w}^* = \lambda \mathbf{w}^*$

So solutions are eigenvectors of $S_\Delta^{-1} S_F$ (or generalised eigenvectors of S_Δ and S_F).

We can use more than one eigenvector of $\widetilde{S}_F$ to capture subspace with maximal whitened signal variance, although these will not be orthogonal when transformed back to the original space.

Demixed Principal Component Analysis (DPCA)

Two slightly different recent proposals from Machens and collaborators [NIPS and eLife]. We will describe the eLife version.

$$\text{Find } \underset{\mathbf{w}, \|\mathbf{u}\|=1}{\operatorname{argmin}} \sum_{i,t} \|\bar{\mathbf{x}}^{(k^{(i)})} - \mathbf{u}\mathbf{w}^{\mathsf{T}}\mathbf{x}_t^{(i)}\|^2$$

Reduced rank regression. Compress data to optimally preserve information about factor means: compare to bottleneck view of PCA.

Similar intuition to LDA, but slightly different cost function.

DPCA

Reduced rank regression has a well-known solution: The output direction ($\mathbf{u}^*$) will align with maximum output-variance mode of MSE regression.

That is:

$$\text{let } Q = \left\langle \mathbf{x}_t^{(i)} \mathbf{x}_t^{(i)} \right\rangle^{-1} \left\langle \mathbf{x}_t^{(i)} \bar{\mathbf{x}}^{(k^{(i)})} \right\rangle = (S_{Tot})^{-1} S_F$$

$$\text{then } \mathbf{u}^* = \text{eig}(Q^\mathsf{T} S_{Tot} Q) = \text{eig}(S_F {S_{Tot}}^{-1} S_{Tot} {S_{Tot}}^{-1} S_F) = \text{eig}(S_F {S_{Tot}}^{-1} S_F)$$

$$\text{and } \mathbf{w}^* = Q\mathbf{u}^*$$

Now,

$$S_F {S_{Tot}}^{-1} S_F \mathbf{u}^* = \mathbf{u}^* \lambda$$

$$\Rightarrow S_{Tot}^{-1} S_F S_F S_{Tot}^{-1} S_F \mathbf{u}^* = S_{Tot}^{-1} S_F \mathbf{u}^* \lambda$$

$$\Rightarrow S_{Tot}^{-1} S_F^2 \mathbf{w}^* = \mathbf{w}^* \lambda$$

So solutions are eigenvectors of $S_{Tot}^{-1} S_F^2$.

DPCA – alternative derivation

We can write the objective as:

$$\mathcal{C}(U,W) = \sum_{i,t} \|\bar{\mathbf{x}}^{(k^{(i)})} - UW^{\mathsf{T}}\mathbf{x}_t^{(i)}\|^2 \propto \mathrm{Tr}\left[\left\langle (\bar{\mathbf{x}}^{(k^{(i)})} - UW^{\mathsf{T}}\mathbf{x}_t^{(i)})(\bar{\mathbf{x}}^{(k^{(i)})} - UW^{\mathsf{T}}\mathbf{x}_t^{(i)})^{\mathsf{T}} \right\rangle\right]$$

$$= \mathrm{Tr}\left[\left\langle ((I - UW^{\mathsf{T}})\bar{\mathbf{x}}^{(k^{(i)})} - UW^{\mathsf{T}}\Delta\mathbf{x}_t^{(i)})((I - UW^{\mathsf{T}})\bar{\mathbf{x}}^{(k^{(i)})} - UW^{\mathsf{T}}\Delta\mathbf{x}_t^{(i)})^{\mathsf{T}} \right\rangle\right]$$

$$= \mathrm{Tr}\left[(I - UW^{\mathsf{T}})(I - UW^{\mathsf{T}})^{\mathsf{T}} S_F + WU^{\mathsf{T}}UW^{\mathsf{T}} S_\Delta\right]$$

$$= \mathrm{Tr}\left[(I - UW^{\mathsf{T}})(I - UW^{\mathsf{T}})^{\mathsf{T}} S_F + WW^{\mathsf{T}} S_\Delta\right]$$

$$= \mathrm{Tr}\left[S_F + WW^{\mathsf{T}} S_{Tot} - 2UW^{\mathsf{T}} S_F\right]$$

Differentiate wrt W to find maximum:

$$\frac{\partial \mathcal{C}}{\partial W} = 2S_{Tot}W - 2S_F U = 0 \qquad \Rightarrow W^* = S_{Tot}^{-1} S_F U$$

So

$$\mathcal{C}(U) = \mathrm{Tr}\left[S_f\right] + \mathrm{Tr}\left[U^{\mathsf{T}} S_F S_{Tot}^{-1} S_{Tot} S_{Tot}^{-1} S_F U - 2U^{\mathsf{T}} S_F S_{Tot}^{-1} S_F U\right]$$

$$= \mathrm{Tr}\left[S_f\right] - \mathrm{Tr}\left[U^{\mathsf{T}} S_F S_{Tot}^{-1} S_F U\right]$$

and U^* is given by the dominant eigenvectors of $S_F S_{Tot}^{-1} S_F$, giving us the same result.

DPCA – an aside

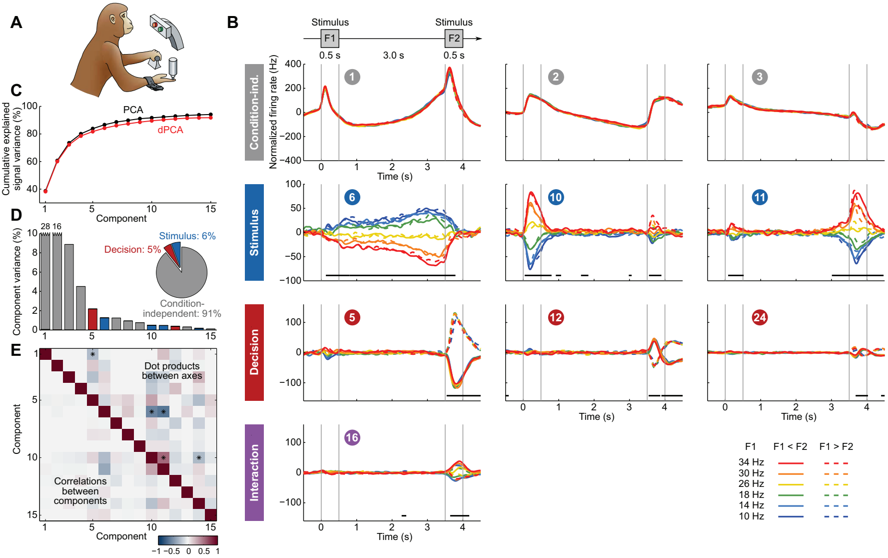
What if we require $W = U$ (i.e. projection and reconstruction are complementary orthogonal projections)?

Then, we can re-write the cost function again:

$$\begin{aligned}\mathcal{C}(U, W) &= \text{Tr}\left[S_F + WW^\mathsf{T} S_{Tot} - 2UW^\mathsf{T} S_F\right] \\ &= \text{Tr}\left[S_F + WW^\mathsf{T}(S_F + S_\Delta) - 2WW^\mathsf{T} S_F\right] \\ &= \text{const} + \text{Tr}\left[W^\mathsf{T}(S_\Delta - S_F)W\right]\end{aligned}$$

So with this constraint DPCA will find a projection which maximises the *difference* between S_F and S_Δ. Recall that LDA maximises the corresponding *ratio*.

DPCA – Romo data set



Relating LDA and DPCA

Relating LDA and DPCA

1. LDA projections are given by generalised eigenvectors:

$$S_{\Delta}\mathbf{w} = \lambda S_F \mathbf{w}$$

Relating LDA and DPCA

1. LDA projections are given by generalised eigenvectors:

$$
\begin{aligned}
& S_\Delta \mathbf{w} = \lambda S_F \mathbf{w} \\
\Rightarrow & S_\Delta \mathbf{w} + S_F \mathbf{w} = (\lambda + 1) S_F \mathbf{w}
\end{aligned}
$$

Relating LDA and DPCA

1. LDA projections are given by generalised eigenvectors:

$$\begin{aligned} & S_\Delta \mathbf{w} = \lambda S_F \mathbf{w} \\ \Rightarrow & S_\Delta \mathbf{w} + S_F \mathbf{w} = (\lambda + 1) S_F \mathbf{w} \\ \Rightarrow & S_{Tot} \mathbf{w} = (\lambda + 1) S_F \mathbf{w} \end{aligned}$$

   so LDA projections are also generalised eigenvectors of (S_{Tot}, S_F)

Relating LDA and DPCA

1. LDA projections are given by generalised eigenvectors:

$$
\begin{aligned}
& S_{\Delta}\mathbf{w} = \lambda S_F \mathbf{w} \\
\Rightarrow & S_{\Delta}\mathbf{w} + S_F\mathbf{w} = (\lambda + 1) S_F \mathbf{w} \\
\Rightarrow & S_{Tot}\mathbf{w} = (\lambda + 1) S_F \mathbf{w}
\end{aligned}
$$

   so LDA projections are also generalised eigenvectors of (S_{Tot}, S_F)
   $\Rightarrow$ eigenvectors of $S_{Tot}^{-1} S_F$ if inverse exists.

Relating LDA and DPCA

1. LDA projections are given by generalised eigenvectors:

$$\begin{aligned} & S_\Delta \mathbf{w} = \lambda S_F \mathbf{w} \\ \Rightarrow & S_\Delta \mathbf{w} + S_F \mathbf{w} = (\lambda + 1) S_F \mathbf{w} \\ \Rightarrow & S_{Tot} \mathbf{w} = (\lambda + 1) S_F \mathbf{w} \end{aligned}$$

   so LDA projections are also generalised eigenvectors of (S_{Tot}, S_F)
   $\Rightarrow$ eigenvectors of $S_{Tot}^{-1} S_F$ if inverse exists.

2. Define $\mathbf{k}^{(i)}$ to be the K-dimensional indicator vector (1 for coordinate $k^{(i)}$, 0 else). Then $\mathbf{w} = \underset{\|\mathbf{u}\|=1}{\text{argmin}} \sum_{i,t} \|\mathbf{k}^{(i)} - \mathbf{u}\mathbf{w}^\mathsf{T} \mathbf{x}_t^{(i)}\|^2$ yields LDA:

Relating LDA and DPCA

1. LDA projections are given by generalised eigenvectors:

$$\begin{aligned} & S_\Delta \mathbf{w} = \lambda S_F \mathbf{w} \\ \Rightarrow & S_\Delta \mathbf{w} + S_F \mathbf{w} = (\lambda + 1) S_F \mathbf{w} \\ \Rightarrow & S_{Tot} \mathbf{w} = (\lambda + 1) S_F \mathbf{w} \end{aligned}$$

   so LDA projections are also generalised eigenvectors of (S_{Tot}, S_F)
   $\Rightarrow$ eigenvectors of $S_{Tot}^{-1} S_F$ if inverse exists.

2. Define $\mathbf{k}^{(i)}$ to be the K-dimensional indicator vector (1 for coordinate $k^{(i)}$, 0 else). Then $\mathbf{w} = \underset{\|\mathbf{u}\|=1}{\text{argmin}} \sum_{i,t} \|\mathbf{k}^{(i)} - \mathbf{u}\mathbf{w}^\mathsf{T}\mathbf{x}_t^{(i)}\|^2$ yields LDA:
   Let $M = \left\langle \mathbf{x}_t^{(i)} \mathbf{k}^{(i)\mathsf{T}} \right\rangle = [\bar{\mathbf{x}}^{(1)} \bar{\mathbf{x}}^{(2)} \ldots]$.

Relating LDA and DPCA

1. LDA projections are given by generalised eigenvectors:

$$\begin{aligned} & S_\Delta \mathbf{w} = \lambda S_F \mathbf{w} \\ \Rightarrow & S_\Delta \mathbf{w} + S_F \mathbf{w} = (\lambda + 1) S_F \mathbf{w} \\ \Rightarrow & S_{Tot} \mathbf{w} = (\lambda + 1) S_F \mathbf{w} \end{aligned}$$

   so LDA projections are also generalised eigenvectors of (S_{Tot}, S_F)
   $\Rightarrow$ eigenvectors of $S_{Tot}^{-1} S_F$ if inverse exists.

2. Define $\mathbf{k}^{(i)}$ to be the K-dimensional indicator vector (1 for coordinate $k^{(i)}$, 0 else). Then $\mathbf{w} = \operatorname{argmin}_{\|\mathbf{u}\|=1} \sum_{i,t} \|\mathbf{k}^{(i)} - \mathbf{u}\mathbf{w}^\mathsf{T} \mathbf{x}_t^{(i)}\|^2$ yields LDA:
   Let $M = \left\langle \mathbf{x}_t^{(i)} \mathbf{k}^{(i)\mathsf{T}} \right\rangle = [\bar{\mathbf{x}}^{(1)} \bar{\mathbf{x}}^{(2)} \dots]$.
   Then $Q = S_{Tot}^{-1} M$, $\mathbf{u}^* = \operatorname{eig}(Q^\mathsf{T} S_{Tot} Q) = \operatorname{eig}(M^\mathsf{T} S_{Tot}^{-1} M)$ and $\mathbf{w}^* = Q\mathbf{u}^*$.

Relating LDA and DPCA

1. LDA projections are given by generalised eigenvectors:

$$\begin{aligned} & S_\Delta \mathbf{w} = \lambda S_F \mathbf{w} \\ \Rightarrow & S_\Delta \mathbf{w} + S_F \mathbf{w} = (\lambda + 1) S_F \mathbf{w} \\ \Rightarrow & S_{Tot} \mathbf{w} = (\lambda + 1) S_F \mathbf{w} \end{aligned}$$

so LDA projections are also generalised eigenvectors of (S_{Tot}, S_F)
$\Rightarrow$ eigenvectors of $S_{Tot}^{-1} S_F$ if inverse exists.

2. Define $\mathbf{k}^{(i)}$ to be the K-dimensional indicator vector (1 for coordinate $k^{(i)}$, 0 else). Then $\mathbf{w} = \underset{\|\mathbf{u}\|=1}{\operatorname{argmin}} \sum_{i,t} \|\mathbf{k}^{(i)} - \mathbf{u}\mathbf{w}^\mathsf{T}\mathbf{x}_t^{(i)}\|^2$ yields LDA:
Let $M = \left\langle \mathbf{x}_t^{(i)} \mathbf{k}^{(i)\mathsf{T}} \right\rangle = [\bar{\mathbf{x}}^{(1)} \bar{\mathbf{x}}^{(2)} \ldots]$.
Then $Q = S_{Tot}^{-1} M$, $\mathbf{u}^* = \operatorname{eig}(Q^\mathsf{T} S_{Tot} Q) = \operatorname{eig}(M^\mathsf{T} S_{Tot}^{-1} M)$ and $\mathbf{w}^* = Q\mathbf{u}^*$.
So $\quad M^\mathsf{T} S_{Tot}^{-1} M \mathbf{u}^* = \mathbf{u}^* \lambda$

Relating LDA and DPCA

1. LDA projections are given by generalised eigenvectors:

$$
\begin{aligned}
& S_\Delta \mathbf{w} = \lambda S_F \mathbf{w} \\
\Rightarrow & S_\Delta \mathbf{w} + S_F \mathbf{w} = (\lambda + 1) S_F \mathbf{w} \\
\Rightarrow & S_{Tot} \mathbf{w} = (\lambda + 1) S_F \mathbf{w}
\end{aligned}
$$

   so LDA projections are also generalised eigenvectors of (S_{Tot}, S_F)
   $\Rightarrow$ eigenvectors of $S_{Tot}^{-1} S_F$ if inverse exists.

2. Define $\mathbf{k}^{(i)}$ to be the K-dimensional indicator vector (1 for coordinate $k^{(i)}$, 0 else). Then $\mathbf{w} = \underset{\|\mathbf{u}\|=1}{\operatorname{argmin}} \sum_{i,t} \|\mathbf{k}^{(i)} - \mathbf{u}\mathbf{w}^\mathsf{T}\mathbf{x}_t^{(i)}\|^2$ yields LDA:
   Let $M = \left\langle \mathbf{x}_t^{(i)} \mathbf{k}^{(i)\mathsf{T}} \right\rangle = [\bar{\mathbf{x}}^{(1)} \bar{\mathbf{x}}^{(2)} \ldots]$.
   Then $Q = S_{Tot}^{-1} M$, $\mathbf{u}^* = \operatorname{eig}(Q^\mathsf{T} S_{Tot} Q) = \operatorname{eig}(M^\mathsf{T} S_{Tot}^{-1} M)$ and $\mathbf{w}^* = Q\mathbf{u}^*$.
   So

$$
\begin{aligned}
& M^\mathsf{T} S_{Tot}^{-1} M \mathbf{u}^* = \mathbf{u}^* \lambda \\
\Rightarrow & S_{Tot}^{-1} M M^\mathsf{T} S_{Tot}^{-1} M \mathbf{u}^* = S_{Tot}^{-1} M \mathbf{u}^* \lambda
\end{aligned}
$$

Relating LDA and DPCA

1. LDA projections are given by generalised eigenvectors:

$$\begin{aligned} & S_\Delta \mathbf{w} = \lambda S_F \mathbf{w} \\ \Rightarrow & S_\Delta \mathbf{w} + S_F \mathbf{w} = (\lambda + 1) S_F \mathbf{w} \\ \Rightarrow & S_{Tot} \mathbf{w} = (\lambda + 1) S_F \mathbf{w} \end{aligned}$$

so LDA projections are also generalised eigenvectors of (S_{Tot}, S_F)
$\Rightarrow$ eigenvectors of $S_{Tot}^{-1} S_F$ if inverse exists.

2. Define $\mathbf{k}^{(i)}$ to be the K-dimensional indicator vector (1 for coordinate $k^{(i)}$, 0 else). Then $\mathbf{w} = \operatorname{argmin}_{\|\mathbf{u}\|=1} \sum_{i,t} \|\mathbf{k}^{(i)} - \mathbf{u}\mathbf{w}^\mathsf{T}\mathbf{x}_t^{(i)}\|^2$ yields LDA:
Let $M = \left\langle \mathbf{x}_t^{(i)} \mathbf{k}^{(i)\mathsf{T}} \right\rangle = [\bar{\mathbf{x}}^{(1)} \bar{\mathbf{x}}^{(2)} \ldots]$.
Then $Q = S_{Tot}^{-1} M$, $\mathbf{u}^* = \operatorname{eig}(Q^\mathsf{T} S_{Tot} Q) = \operatorname{eig}(M^\mathsf{T} S_{Tot}^{-1} M)$ and $\mathbf{w}^* = Q\mathbf{u}^*$.
So
$$\begin{aligned} & M^\mathsf{T} S_{Tot}^{-1} M \mathbf{u}^* = \mathbf{u}^* \lambda \\ \Rightarrow & S_{Tot}^{-1} M M^\mathsf{T} S_{Tot}^{-1} M \mathbf{u}^* = S_{Tot}^{-1} M \mathbf{u}^* \lambda \\ \Rightarrow & S_{Tot}^{-1} M M^\mathsf{T} \mathbf{w}^* = \mathbf{w}^* \lambda \end{aligned}$$

Relating LDA and DPCA

1. LDA projections are given by generalised eigenvectors:

$$\begin{aligned} & S_\Delta \mathbf{w} = \lambda S_F \mathbf{w} \\ \Rightarrow & S_\Delta \mathbf{w} + S_F \mathbf{w} = (\lambda + 1) S_F \mathbf{w} \\ \Rightarrow & S_{Tot} \mathbf{w} = (\lambda + 1) S_F \mathbf{w} \end{aligned}$$

so LDA projections are also generalised eigenvectors of (S_{Tot}, S_F)
$\Rightarrow$ eigenvectors of $S_{Tot}^{-1} S_F$ if inverse exists.

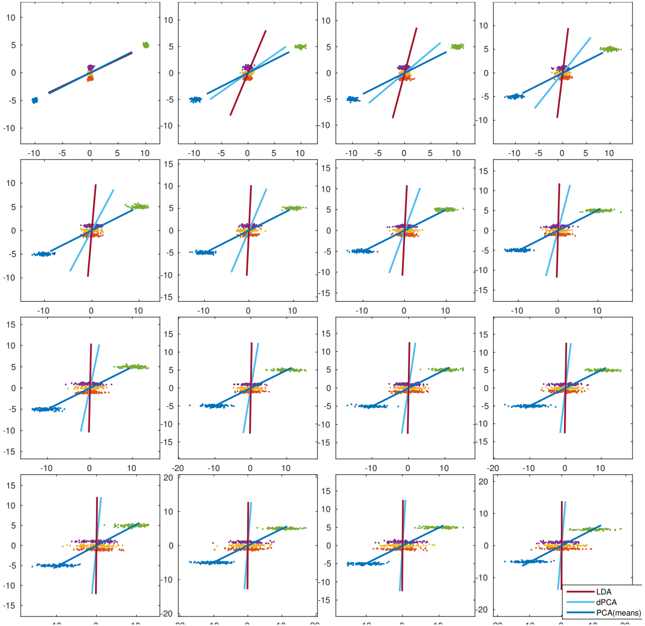
2. Define $\mathbf{k}^{(i)}$ to be the K-dimensional indicator vector (1 for coordinate $k^{(i)}$, 0 else). Then $\mathbf{w} = \underset{\|\mathbf{u}\|=1}{\operatorname{argmin}} \sum_{i,t} \|\mathbf{k}^{(i)} - \mathbf{u}\mathbf{w}^\mathsf{T}\mathbf{x}_t^{(i)}\|^2$ yields LDA:
Let $M = \left\langle \mathbf{x}_t^{(i)} \mathbf{k}^{(i)\mathsf{T}} \right\rangle = [\bar{\mathbf{x}}^{(1)} \bar{\mathbf{x}}^{(2)} \ldots]$.
Then $Q = S_{Tot}^{-1} M$, $\mathbf{u}^* = \operatorname{eig}(Q^\mathsf{T} S_{Tot} Q) = \operatorname{eig}(M^\mathsf{T} S_{Tot}^{-1} M)$ and $\mathbf{w}^* = Q\mathbf{u}^*$.
So

$$\begin{aligned} & M^\mathsf{T} S_{Tot}^{-1} M \mathbf{u}^* = \mathbf{u}^* \lambda \\ \Rightarrow & S_{Tot}^{-1} M M^\mathsf{T} S_{Tot}^{-1} M \mathbf{u}^* = S_{Tot}^{-1} M \mathbf{u}^* \lambda \\ \Rightarrow & S_{Tot}^{-1} \underbrace{M M^\mathsf{T}}_{K S_F} \mathbf{w}^* = \mathbf{w}^* \lambda \end{aligned}$$

Comparison

	LDA	DPCA
Cost	$\max \operatorname{Tr}\left[(W^{\mathsf{T}} S_{\Delta} W)^{-1}(W^{\mathsf{T}} S_F W)\right]$	$\min\limits_{U^{\mathsf{T}}U=I} \operatorname{Tr}\left[W^{\mathsf{T}} S_{Tot} W - 2W^{\mathsf{T}} S_F U\right]$
Eigenprob	$S_{\Delta}^{-1} S_F \equiv S_{Tot}^{-1} S_F$	$S_{Tot}^{-1} S_F^2$
RRR	$\mathbf{x}_t^{(i)} \to \mathbf{k}^{(i)}$	$\mathbf{x}_t^{(i)} \to \bar{\mathbf{x}}^{(k^{(i)})}$

Comparison



Continuous / ordinal covariates

- Regression
- Canonical correlation analysis: CCA
- Canonical covariance analysis: CVA / PLS

Canonical Correlations/Covariance Analysis

Data vector pairs: $\mathcal{D} = \{(\mathbf{u}_1, \mathbf{v}_1), (\mathbf{u}_2, \mathbf{v}_2) \dots\}$ in spaces $\mathcal{U}$ and $\mathcal{V}$.

Classic CCA

- Find unit vectors $\boldsymbol{\upsilon}_1 \in \mathcal{U}$, $\boldsymbol{\phi}_1 \in \mathcal{V}$ such that the (Pearson) correlation of $\mathbf{u}_i^\mathsf{T} \boldsymbol{\upsilon}_1$ and $\mathbf{v}_i^\mathsf{T} \boldsymbol{\phi}_1$ is maximised.
- As with PCA, repeat in orthogonal (wrt data covariance) subspaces.
- $\mathbf{svd}(\Sigma_u^{-1/2} \Sigma_{uv} \Sigma_v^{-1/2})$

CVA (or PLS – Partial Least Squares)

- $\mathbf{svd}(\Sigma_{uv})$

Probabilistic CCA

- Generative model with latent $\mathbf{x}_i \in \mathbb{R}^K$:

$$
\begin{aligned}
\mathbf{x} &\sim \mathcal{N}(0, I) \\
\mathbf{u} &\sim \mathcal{N}(\Upsilon \mathbf{x}, \Psi_u) \quad \Psi_u \succcurlyeq 0 \\
\mathbf{v} &\sim \mathcal{N}(\Phi \mathbf{x}, \Psi_v) \quad \Psi_v \succcurlyeq 0
\end{aligned}
$$

- Block diagonal noise.

Modes of covariation

What form does this population-movement covariation take?

Modes of covariation

What form does this population-movement covariation take?

"Canonical Covariance Analysis":

- For each reach target: find mean movement trajectory and mean firing profile (PSTH).

$$\bar{\mathbf{m}}_t^c = \frac{1}{N_{\text{trials}}^c} \sum_n \mathbf{m}_t^{n(c)} \qquad \bar{\mathbf{r}}_t^c = \frac{1}{N_{\text{trials}}^c} \sum_n \mathbf{r}_t^{n(c)}$$

$[\mathbf{m}_t \in \mathbb{R}^{\#\text{ move params}}; \mathbf{r}_t \in \mathbb{R}^{\#neurons}]$

Modes of covariation

What form does this population-movement covariation take?

“Canonical Covariance Analysis”:

- For each reach target: find mean movement trajectory and mean firing profile (PSTH).

$$\bar{\mathbf{m}}_t^c = \frac{1}{N_{\text{trials}}^c} \sum_n \mathbf{m}_t^{n(c)} \qquad \bar{\mathbf{r}}_t^c = \frac{1}{N_{\text{trials}}^c} \sum_n \mathbf{r}_t^{n(c)}$$

  [$\mathbf{m}_t \in \mathbb{R}^{\#\text{ move params}}$; $\mathbf{r}_t \in \mathbb{R}^{\# neurons}$]

- For each trial: find deviation from condition means.

$$\delta\mathbf{m}_t^{n(c)} = \mathbf{m}_t^{n(c)} - \bar{\mathbf{m}}_t^c \qquad \delta\mathbf{r}_t^{n(c)} = \mathbf{r}_t^{n(c)} - \bar{\mathbf{r}}_t^c$$

Modes of covariation

What form does this population-movement covariation take?

"Canonical Covariance Analysis":

- For each reach target: find mean movement trajectory and mean firing profile (PSTH).

$$\bar{\mathbf{m}}_t^c = \frac{1}{N^c_{\text{trials}}} \sum_n \mathbf{m}_t^{n(c)} \qquad \bar{\mathbf{r}}_t^c = \frac{1}{N^c_{\text{trials}}} \sum_n \mathbf{r}_t^{n(c)}$$

$[\mathbf{m}_t \in \mathbb{R}^{\#\text{ move params}}; \mathbf{r}_t \in \mathbb{R}^{\#neurons}]$

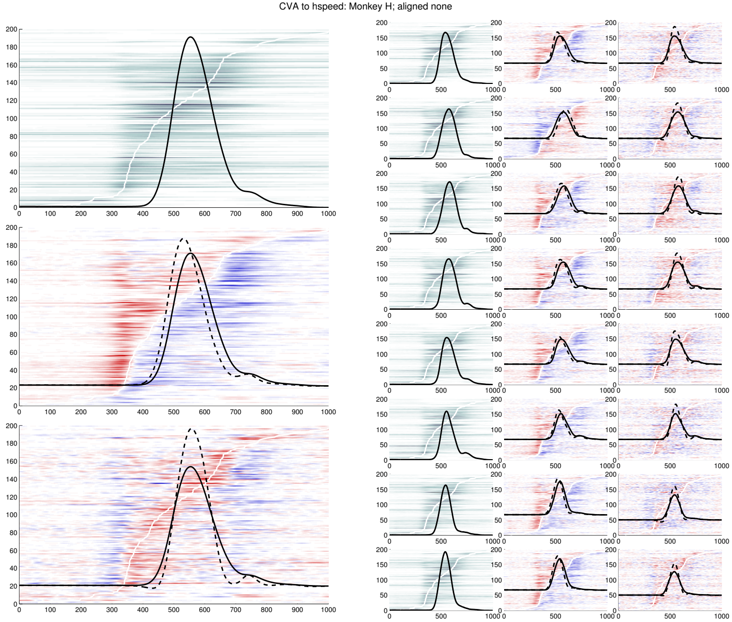
- For each trial: find deviation from condition means.

$$\delta\mathbf{m}_t^{n(c)} = \mathbf{m}_t^{n(c)} - \bar{\mathbf{m}}_t^c \qquad \delta\mathbf{r}_t^{n(c)} = \mathbf{r}_t^{n(c)} - \bar{\mathbf{r}}_t^c$$

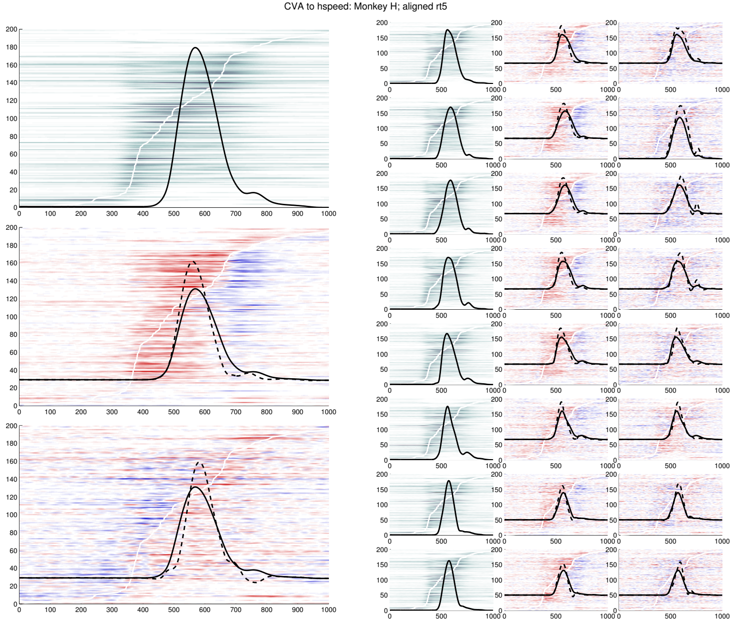
- For all trials: find simultaneous projection of deviations in movement and activity that have the highest covariance

$$(\mathbf{M}_t, \mathbf{R}_t) = \text{argmax} \sum_c \sum_n \underbrace{\Big(\sum_t \mathbf{M}_t^{\mathsf{T}}\, \delta\mathbf{m}_t^{n(c)}\Big)\ \Big(\sum_t \mathbf{R}_t^{\mathsf{T}}\, \delta\mathbf{r}_t^{n(c)}\Big)}_{\text{matrix dot products}}$$

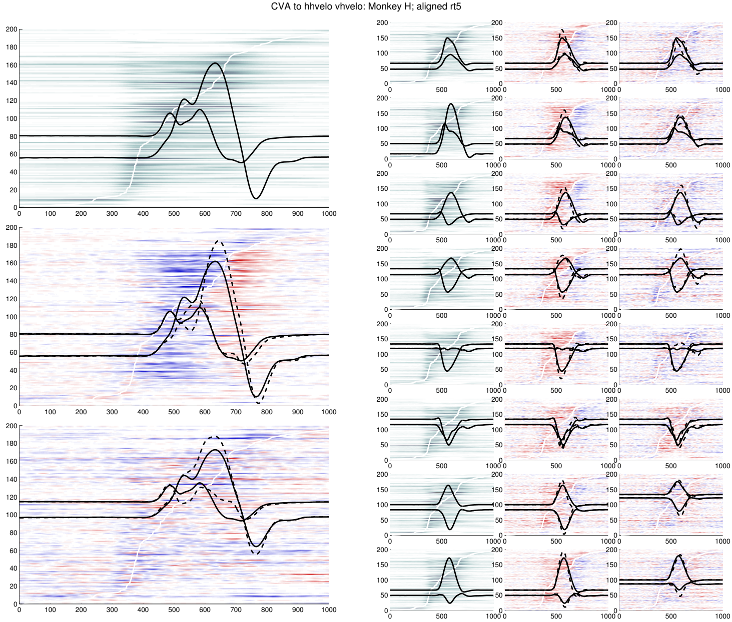
CVA: speed profile



CVA: speed profile aligned to movement start



CVA: velocity profile aligned to movement start



CVA: speed and velocity aligned to movement start