Machine Learning for Signal Processing
Applications of Linear Gaussian Models

Class 15. 12 Nov 2015

Instructor: Bhiksha Raj

Recap: MAP Estimators

- MAP (*Maximum A Posteriori*): Find a "best guess" for $\mathbf{y}$ (statistically), given known $\mathbf{x}$

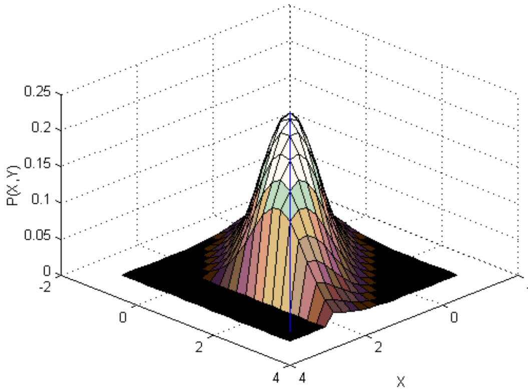
$$\mathbf{y} = argmax_{Y} P(\mathrm{Y}|\mathbf{x})$$

Conditional Probability of y|x

$$P(y \mid x) = N(\mu_y + C_{yx}C_{xx}^{-1}(x - \mu_x), C_{yy} - C_{yx}^{T}C_{xx}^{-1}C_{xy})$$

$$E_{y|x}[y] = \mu_{y|x} = \mu_y + C_{yx}C_{xx}^{-1}(x - \mu_x)$$

$$Var(y \mid x) = C_{yy} - C_{xy}^{T}C_{xx}^{-1}C_{xy}$$



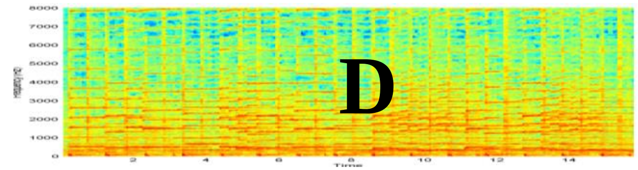
- The conditional probability of y given x is also Gaussian
  - The slice in the figure is Gaussian
- The mean of this Gaussian is a function of x
- The variance of y reduces if x is known
  - Uncertainty is reduced

Gaussians and more Gaussians..

- Linear Gaussian Models..

- PCA to develop the idea of LGM

A Brief Recap



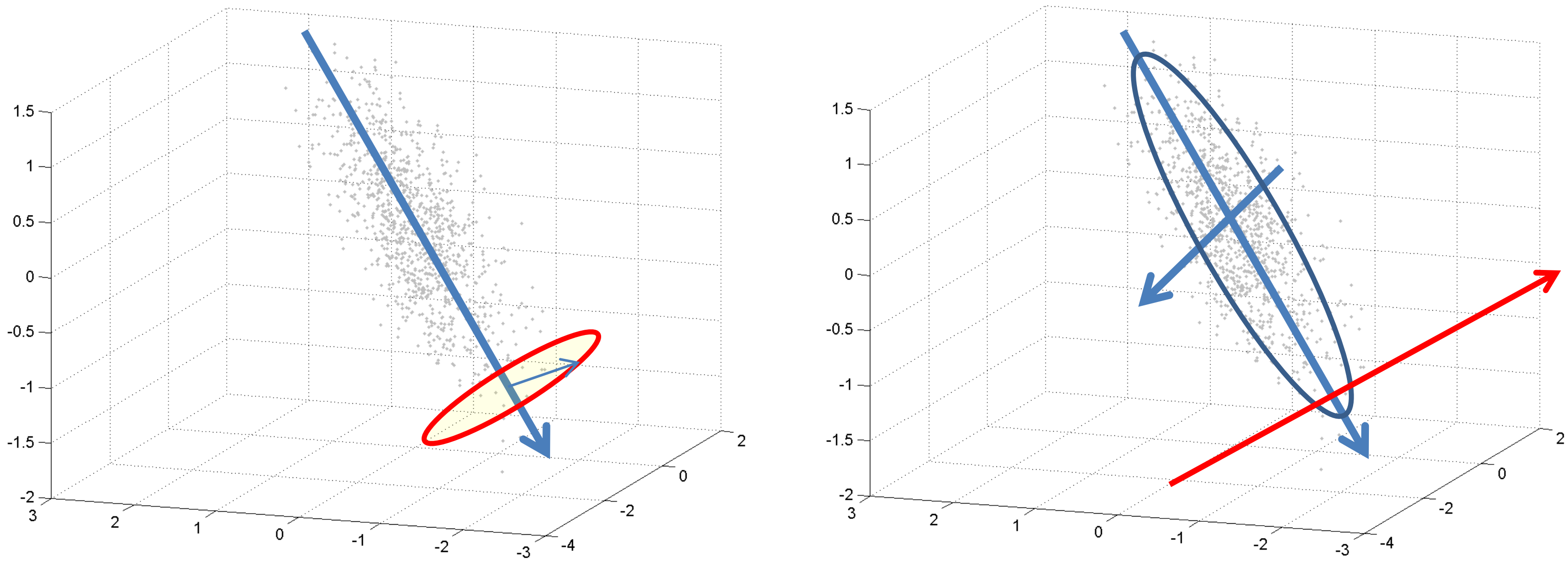
C

B

$$\mathbf{D} \approx \mathbf{BC}$$

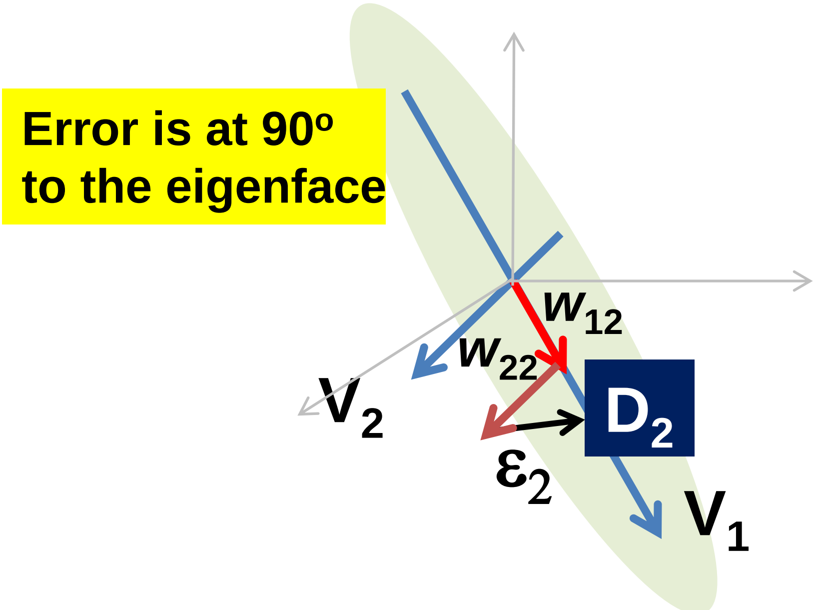
- Principal component analysis: Find the *K* bases that best explain the given data
- Find **B** and **C** such that the difference between **D** and **BC** is minimum
  - While constraining that the columns of **B** are orthonormal

Learning PCA



- For the given data: find the K-dimensional subspace such that it captures most of the variance in the data
  - Variance in remaining subspace is minimal

A Statistical Formulation of PCA



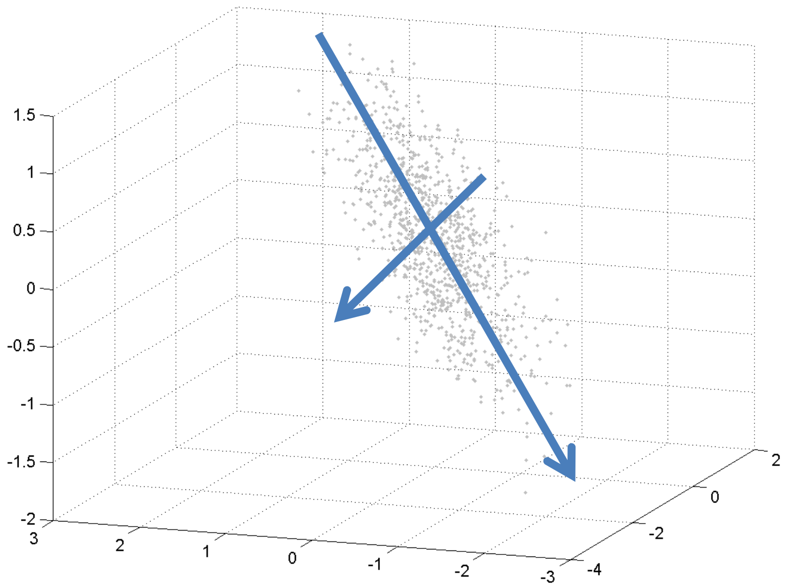
$$\mathbf{x} = \mathbf{V}\mathbf{w} + \mathbf{e}$$

$$\mathbf{w} \sim N(0, B)$$

$$\mathbf{e} \sim N(0, E)$$

- $\mathbf{x}$ is a random variable generated according to a linear relation
- $\mathbf{w}$ is drawn from an K-dimensional Gaussian with diagonal covariance
- $\mathbf{e}$ is drawn from a 0-mean (D-K)-rank D-dimensional Gaussian
- Estimate $\mathbf{V}$ (and B) given examples of $\mathbf{x}$

Linear Gaussian Models!!



$$\mathbf{x} = \mathbf{V}\mathbf{w} + \mathbf{e}$$

$$\mathbf{w} \sim N(0, B)$$

$$\mathbf{e} \sim N(0, E)$$

- $\mathbf{x}$ is a random variable generated according to a linear relation
- $\mathbf{w}$ is drawn from a Gaussian
- $\mathbf{e}$ is drawn from a 0-mean Gaussian
- Estimate $\mathbf{V}$ given examples of $\mathbf{x}$
  - In the process also estimate $\mathbf{B}$ and $\mathbf{E}$

Estimating the variables of the model

$$\mathbf{x} = \boldsymbol{\mu} + \mathbf{V}\mathbf{w} + \mathbf{e}$$

$$\mathbf{w} \sim N(0, I)$$
$$\mathbf{e} \sim N(0, E)$$

$$\boxed{\mathbf{x} \sim N(\boldsymbol{\mu}, \mathbf{V}\mathbf{V}^T + E)}$$

- Estimating the variables of the LGM is equivalent to estimating P($\mathbf{x}$)
  - The variables are $\boldsymbol{\mu}$, $\mathbf{V}$, and E

The Maximum Likelihood Estimate

$$\mathbf{x} \sim N(\boldsymbol{\mu}, \mathbf{V}\mathbf{V}^T + E)$$

- **Given training set** $\mathbf{x}_1, \mathbf{x}_2, .. \mathbf{x}_N$, **find** $\boldsymbol{\mu}, \mathbf{V}, E$

- The ML estimate of $\boldsymbol{\mu}$ does not depend on the covariance of the Gaussian

$$\boldsymbol{\mu} = \frac{1}{N}\sum_i \mathbf{x}_i$$

Simplified Model

$$\mathbf{x} = \mathbf{V}\mathbf{w} + \mathbf{e}$$

$$\mathbf{w} \sim N(0, I)$$
$$\mathbf{e} \sim N(0, E)$$

$$\boxed{\mathbf{x} \sim N(0, \mathbf{V}\mathbf{V}^T + E)}$$

- Estimating the variables of the LGM is equivalent to estimating P($\mathbf{x}$)
  - The variables are $\mathbf{V}$, and E

LGM: The complete EM algorithm

- Initialize $\mathbf{V}$ and E
- E step:

$$E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}] = \mathbf{V}^T(\mathbf{V}\mathbf{V}^T + E)^{-1}\mathbf{x}_i$$

$$E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}\mathbf{w}^T] = I - \mathbf{V}^T(\mathbf{V}\mathbf{V}^T + E)^{-1}\mathbf{V} + E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]^T$$

- M step:

$$\mathbf{V} = \left(\sum_i \mathbf{x}_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}^T]\right)\left(\sum_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}\mathbf{w}^T]\right)^{-1}$$

- $$E = \frac{1}{N}\sum_i \mathbf{x}_i\mathbf{x}_i^T - \frac{1}{N}\mathbf{V}\sum_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]\mathbf{x}_i^T$$

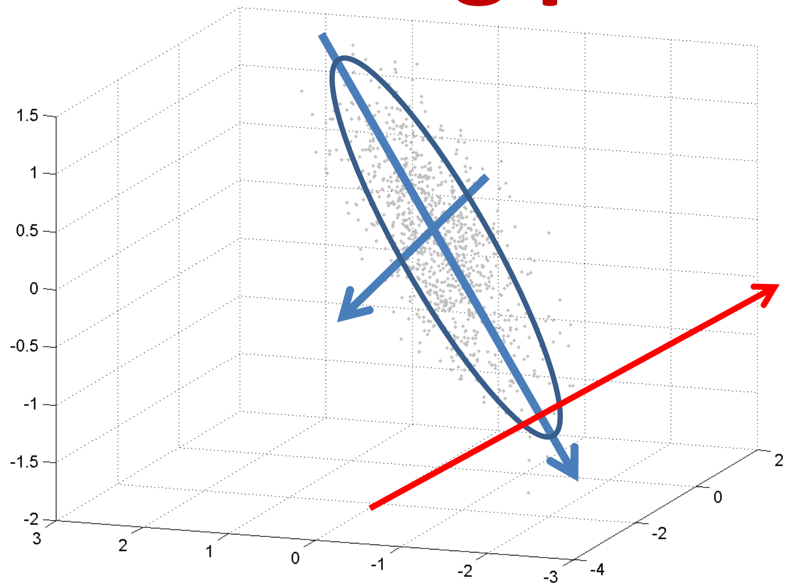
So what have we achieved

- Employed a complicated EM algorithm to learn a *Gaussian* PDF for a variable x
- What have we gained???

- Example uses:
  - PCA
    - Sensible PCA
    - EM algorithms for PCA
  - Factor Analysis
    - FA for feature extraction

LGMs : Application 1

Learning principal components



$$\mathbf{x} = \mathbf{V}\mathbf{w} + \mathbf{e}$$

$$\mathbf{w} \sim N(0, I)$$

$$\mathbf{e} \sim N(0, E)$$

- Find directions that capture most of the variation in the data
- **Error is orthogonal to principal directions**
  - $\mathbf{V}^T\mathbf{e} = 0$; $\mathbf{e}^T\mathbf{V} = 0$

Some Observations: 1

$$\mathbf{x} = \mathbf{V}\mathbf{w} + \mathbf{e} \qquad \mathbf{e} \sim N(0, E)$$

$$E = \mathrm{E}[\mathbf{e}\mathbf{e}^T]$$

$$\mathbf{V}^T E = \mathrm{E}[\mathbf{V}^T\mathbf{e}\mathbf{e}^T] = \mathrm{E}[0\mathbf{e}^T] = 0$$

- The covariance $\mathbf{E}$ of $\mathbf{e}$ is orthogonal to $\mathbf{V}$

Observation 2

$$\mathbf{V}^T E = 0$$

$$\mathbf{V}^T(\mathbf{V}\mathbf{V}^T + E)^{-1} = (\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T$$

- Proof

$$\mathbf{V}^T(\mathbf{V}\mathbf{V}^T + E)^{-1}(\mathbf{V}\mathbf{V}^T + E) = (\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T(\mathbf{V}\mathbf{V}^T + E)$$

$$\mathbf{V}^T = (\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T\mathbf{V}\mathbf{V}^T + (\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T E$$

$$\mathbf{V}^T = \mathbf{I}\mathbf{V}^T + (\mathbf{V}^T\mathbf{V})^{-1}0$$

$$\mathbf{V}^T = \mathbf{V}^T$$

Observation 3

$$\boxed{\mathbf{V}^T E = 0}$$

$$\boxed{\mathbf{V}^T(\mathbf{V}\mathbf{V}^T + E)^{-1} = (\mathbf{V}^T\mathbf{V})^{-1}\mathbf{V}^T}$$

$$\boxed{= pinv(\mathbf{V})}$$

LGM: The complete EM algorithm

$$\mathbf{x} = \mathbf{V}\mathbf{w} + \mathbf{e} \qquad \mathbf{X} \approx \mathbf{V}\mathbf{W}$$

- Initialize $\mathbf{V}$ and E
- E step: $\boxed{E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}] = \mathbf{V}^T(\mathbf{V}\mathbf{V}^T + E)^{-1}\mathbf{x}_i}$

$$\boxed{E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}\mathbf{w}^T] = I - \mathbf{V}^T(\mathbf{V}\mathbf{V}^T + E)^{-1}\mathbf{V} + E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]^T}$$

- M step:

$$\boxed{\mathbf{V} = \left(\sum_i \mathbf{x}_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}^T]\right)\left(\sum_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}\mathbf{w}^T]\right)^{-1}}$$

$$\boxed{E = \frac{1}{N}\sum_i \mathbf{x}_i\mathbf{x}_i^T - \frac{1}{N}\mathbf{V}\sum_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]\mathbf{x}_i^T}$$

LGM: The complete EM algorithm

$$\mathbf{x} = \mathbf{V}\mathbf{w} + \mathbf{e} \qquad \mathbf{X} \approx \mathbf{V}\mathbf{W}$$

- Initialize $\mathbf{V}$ and E
- E step:

$$\boxed{E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}] = \mathbf{V}^T(\mathbf{V}\mathbf{V}^T + E)^{-1}\mathbf{x}_i}$$

$$\boxed{E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}\mathbf{w}^T] = I - \mathbf{V}^T(\mathbf{V}\mathbf{V}^T + E)^{-1}\mathbf{V} + E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]^T}$$

- M step:

$$\boxed{\mathbf{V} = \left(\sum_i \mathbf{x}_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}^T]\right)\left(\sum_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}\mathbf{w}^T]\right)^{-1}}$$

$$\boxed{E = \frac{1}{N}\sum_i \mathbf{x}_i\mathbf{x}_i^T - \frac{1}{N}\mathbf{V}\sum_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]\mathbf{x}_i^T}$$

LGM: The complete EM algorithm

$$\mathbf{x} = \mathbf{V}\mathbf{w} + \mathbf{e} \qquad \mathbf{X} \approx \mathbf{V}\mathbf{W}$$

- Initialize $\mathbf{V}$ and E
- E step: $$\boxed{\mathbf{w}_i = \mathbf{V}^T(\mathbf{V}\mathbf{V}^T + E)^{-1}\mathbf{x}_i = pinv(\mathbf{V})\mathbf{x}_i}$$

$$\boxed{E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}\mathbf{w}^T] = I - \mathbf{V}^T(\mathbf{V}\mathbf{V}^T + E)^{-1}\mathbf{V} + E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]^T}$$

- M step:

$$\boxed{\mathbf{V} = \left(\sum_i \mathbf{x}_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}^T]\right)\left(\sum_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}\mathbf{w}^T]\right)^{-1}}$$

$$\boxed{E = \frac{1}{N}\sum_i \mathbf{x}_i\mathbf{x}_i^T - \frac{1}{N}\mathbf{V}\sum_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]\mathbf{x}_i^T}$$

LGM: The complete EM algorithm

$$\mathbf{x} = \mathbf{V}\mathbf{w} + \mathbf{e} \qquad \mathbf{X} \approx \mathbf{V}\mathbf{W}$$

- Initialize $\mathbf{V}$ and E
- E step:

$$\boxed{\mathbf{w}_i = pinv(\mathbf{V})\mathbf{x}_i}$$

$$E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}\mathbf{w}^T] = I - \mathbf{V}^T(\mathbf{V}\mathbf{V}^T + E)^{-1}\mathbf{V} + E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]^T$$

- M step:

$$\mathbf{V} = \left(\sum_i \mathbf{x}_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}^T]\right)\left(\sum_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}\mathbf{w}^T]\right)^{-1}$$

$$E = \frac{1}{N}\sum_i \mathbf{x}_i\mathbf{x}_i^T - \frac{1}{N}\mathbf{V}\sum_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]\mathbf{x}_i^T$$

LGM: The complete EM algorithm

$$\mathbf{X} \approx \mathbf{V}\mathbf{W}$$

- Initialize $\mathbf{V}$ and E
- E step:

$$\boxed{\mathbf{w}_i = pinv(\mathbf{V})\mathbf{x}_i} \qquad \boxed{\mathbf{W} = pinv(\mathbf{V})\mathbf{X}}$$

$$E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}\mathbf{w}^T] = I - \mathbf{V}^T(\mathbf{V}\mathbf{V}^T + E)^{-1}\mathbf{V} + E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]^T$$

- M step:

$$\mathbf{V} = \left(\sum_i \mathbf{x}_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}^T]\right)\left(\sum_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}\mathbf{w}^T]\right)^{-1}$$

$$E = \frac{1}{N}\sum_i \mathbf{x}_i\mathbf{x}_i^T - \frac{1}{N}\mathbf{V}\sum_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]\mathbf{x}_i^T$$

LGM: The complete EM algorithm

$$\mathbf{X} \approx \mathbf{V}\mathbf{W}$$

- Initialize $\mathbf{V}$ and E
- E step:

$$\boxed{\mathbf{w}_i = pinv(\mathbf{V})\mathbf{x}_i} \qquad \boxed{\mathbf{W} = pinv(\mathbf{V})\mathbf{X}}$$

$$E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}\mathbf{w}^T] = I - \mathbf{V}^T(\mathbf{V}\mathbf{V}^T + E)^{-1}\mathbf{V} + E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]^T$$

- M step:

$$\mathbf{V} = \left(\sum_i \mathbf{x}_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}^T]\right)\left(\sum_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}\mathbf{w}^T]\right)^{-1}$$

$$E = \frac{1}{N}\sum_i \mathbf{x}_i\mathbf{x}_i^T - \frac{1}{N}\mathbf{V}\sum_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]\mathbf{x}_i^T$$

EM for PCA

$$\mathbf{X} \approx \mathbf{V}\mathbf{W}$$

- Initialize $\mathbf{V}$ and E
- E step:

$$\boxed{\mathbf{w}_i = pinv(\mathbf{V})\mathbf{x}_i} \qquad \boxed{\mathbf{W} = pinv(\mathbf{V})\mathbf{X}}$$

$$\boxed{E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}\mathbf{w}^T] = I - \mathbf{V}^T(\mathbf{V}\mathbf{V}^T + E)^{-1}\mathbf{V} + E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]^T}$$

- M step:

$$\boxed{\mathbf{V} = \left(\sum_i \mathbf{x}_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}^T]\right)\left(\sum_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}\mathbf{w}^T]\right)^{-1}}$$

$$\boxed{E = \frac{1}{N}\sum_i \mathbf{x}_i\mathbf{x}_i^T - \frac{1}{N}\mathbf{V}\sum_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]\mathbf{x}_i^T}$$

EM for PCA

- Initialize $\mathbf{V}$ and E
- E step:

$$\boxed{\mathbf{w}_i = pinv(\mathbf{V})\mathbf{x}_i} \qquad \boxed{\mathbf{W} = pinv(\mathbf{V})\mathbf{X}}$$

$$E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}\mathbf{w}^T] = I - \mathbf{V}^T(\mathbf{V}\mathbf{V}^T + E)^{-1}\mathbf{V} + E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]^T$$

- M step:

$$\mathbf{V} = \left(\sum_i \mathbf{x}_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}^T]\right)\left(\sum_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}\mathbf{w}^T]\right)^{-1}$$

$$E = \frac{1}{N}\sum_i \mathbf{x}_i\mathbf{x}_i^T - \frac{1}{N}\mathbf{V}\sum_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]\mathbf{x}_i^T$$

EM for PCA

- Initialize $\mathbf{V}$ and E
- E step:

$$\boxed{\mathbf{w}_i = pinv(\mathbf{V})\mathbf{x}_i} \qquad \boxed{\mathbf{W} = pinv(\mathbf{V})\mathbf{X}}$$

$$\boxed{E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}\mathbf{w}^T] = I - \mathbf{V}^T(\mathbf{V}\mathbf{V}^T + E)^{-1}\mathbf{V} + E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]^T}$$

- M step:

$$\boxed{\mathbf{V} = \left(\sum_i \mathbf{x}_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}^T]\right)\left(\sum_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}\mathbf{w}^T]\right)^{-1} = \mathbf{X}\mathbf{W}^T(\mathbf{W}\mathbf{W}^T)^{-1}}$$

$$\boxed{E = \frac{1}{N}\sum_i \mathbf{x}_i\mathbf{x}_i^T - \frac{1}{N}\mathbf{V}\sum_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]\mathbf{x}_i^T}$$

EM for PCA

- Initialize $\mathbf{V}$ and E
- E step:

$$\boxed{\mathbf{w}_i = pinv(\mathbf{V})\mathbf{x}_i} \qquad \boxed{\mathbf{W} = pinv(\mathbf{V})\mathbf{X}}$$

$$E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}\mathbf{w}^T] = I - \mathbf{V}^T(\mathbf{V}\mathbf{V}^T + E)^{-1}\mathbf{V} + E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]^T$$

- M step:

$$\mathbf{V} = \left(\sum_i \mathbf{x}_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}^T]\right)\left(\sum_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}\mathbf{w}^T]\right)^{-1} = \mathbf{X}\mathbf{W}^T(\mathbf{W}\mathbf{W}^T)^{-1} = \mathbf{X}pinv(\mathbf{W})$$

$$E = \frac{1}{N}\sum_i \mathbf{x}_i\mathbf{x}_i^T - \frac{1}{N}\mathbf{V}\sum_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]\mathbf{x}_i^T$$

EM for PCA

- Initialize $\mathbf{V}$ and E
- E step:

$$\boxed{\mathbf{w}_i = pinv(\mathbf{V})\mathbf{x}_i} \qquad \boxed{\mathbf{W} = pinv(\mathbf{V})\mathbf{X}}$$

$$E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}\mathbf{w}^T] = I - \mathbf{V}^T(\mathbf{V}\mathbf{V}^T + E)^{-1}\mathbf{V} + E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]^T$$

- M step:

$$\boxed{\mathbf{V} = \mathbf{X}\, pinv(\mathbf{W})}$$

$$E = \frac{1}{N}\sum_i \mathbf{x}_i\mathbf{x}_i^T - \frac{1}{N}\mathbf{V}\sum_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]\mathbf{x}_i^T$$

EM for PCA

- Initialize $\mathbf{V}$ and E
- E step:

$$\boxed{\mathbf{W} = pinv(\mathbf{V})\mathbf{X}}$$

$$E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}\mathbf{w}^T] = I - \mathbf{V}^T(\mathbf{V}\mathbf{V}^T + E)^{-1}\mathbf{V} + E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]^T$$

- M step:

$$\boxed{\mathbf{V} = \mathbf{X}\, pinv(\mathbf{W})}$$

$$E = \frac{1}{N}\sum_i \mathbf{x}_i\mathbf{x}_i^T - \frac{1}{N}\mathbf{V}\sum_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]\mathbf{x}_i^T$$

EM for PCA

- Initialize $\mathbf{V}$ and E
- E step:

$$\boxed{\mathbf{W} = pinv(\mathbf{V})\mathbf{X}}$$

$$\cancel{E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}\mathbf{w}^T] = I - \mathbf{V}^T(\mathbf{V}\mathbf{V}^T + E)^{-1}\mathbf{V} + E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]^T}$$

- M step:

irrelevant

$$\boxed{\mathbf{V} = \mathbf{X}\, pinv(\mathbf{W})}$$

$$\cancel{E = \frac{1}{N}\sum_i \mathbf{x}_i\mathbf{x}_i^T - \frac{1}{N}\mathbf{V}\sum_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]\mathbf{x}_i^T}$$

EM for PCA

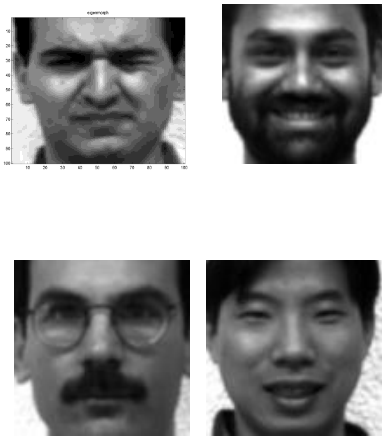
- Initialize $\mathbf{V}$
- Iterate

$$\boxed{\mathbf{W} = pinv(\mathbf{V})\mathbf{X}}$$

$$\boxed{\mathbf{V} = \mathbf{X}\, pinv(\mathbf{W})}$$

- Note: $\mathbf{V}$ will not be actual eigenvectors, but a set of bases in space spanned by principal eigenvectors
  - Additional decorrelation within PC space may be needed

Why EM PCA?



$\mathbf{X}$

10000 x 300

$\mathbf{X}\mathbf{X}^T$

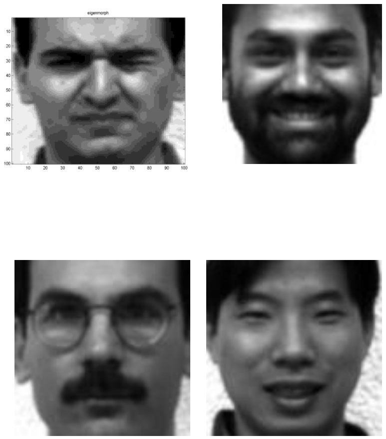
10000 x 10000

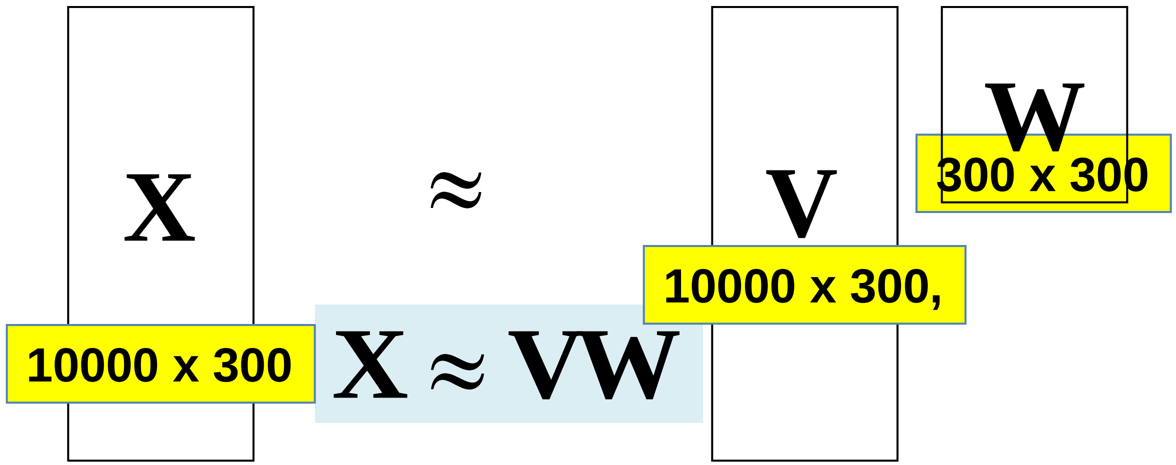
- Example: Computing eigenfaces
- Each face is 100x100 : 10000 dimensional
- But only 300 examples
  - $\mathbf{X}$ is 10000 x 300
- What is the size of the covariance matrix?
- What is its rank?

PCA on illconditioned data

- Few instances of high-dimensional data
  - No. instances < dimensionality
- Covariance matrix is very large
  - Eigen decomposition is expensive
  - E.g. 1000000-dimensional data: Covariance has 10^{12} elements
- But the rank of the covariance is low
  - Only the no. of instances of data

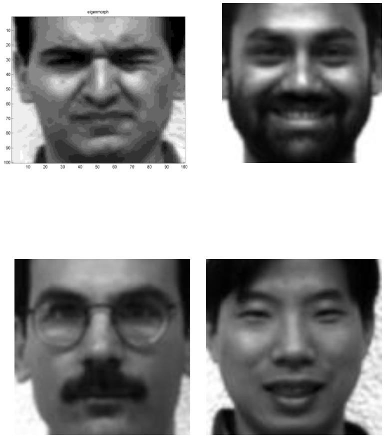
Why EM PCA?

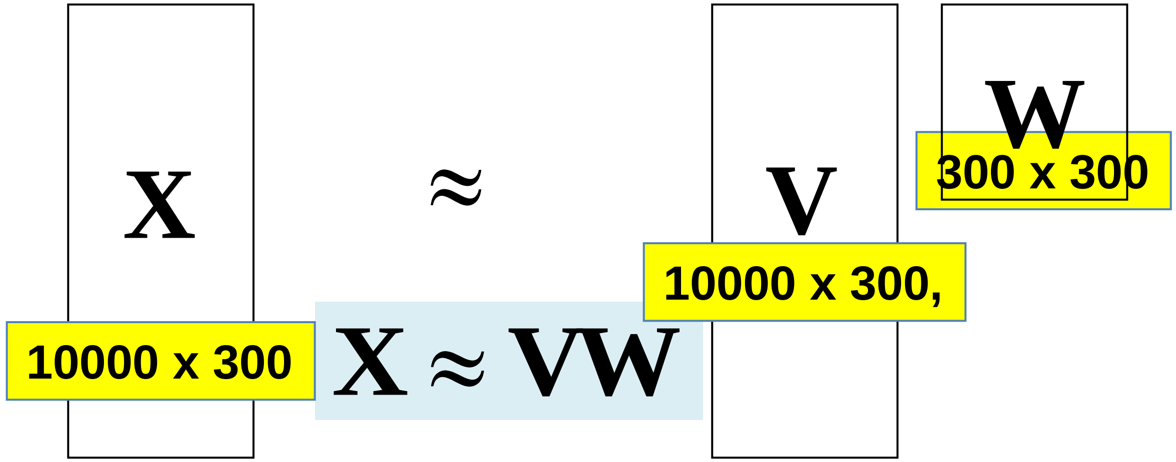




- Consequence of low rank $\mathbf{X}$
  - The actual number of bases is limited to the rank of $\mathbf{X}$
- Note actual size of $\mathbf{V}$
  - Max number of columns = min(dimension, no. data points)
  - No. of columns = rank of ($\mathbf{XX}^{\mathbf{T}}$)
- Note size of $\mathbf{W}$
  - Max number of rows = min(dimension, no. of data points)

Why EM PCA?





- If **X** is high dimensional
  - Particularly if the number of vectors in **X** is smaller than the dimensionality

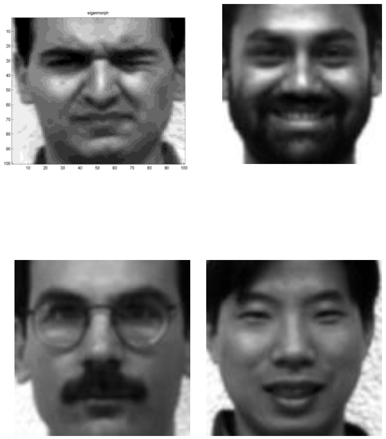
- Pinv(**V**) and pinv(**W**) are efficient to compute
  - **V** will have a max of 300 columns in the example
  - **W** will have a max of 300 rows

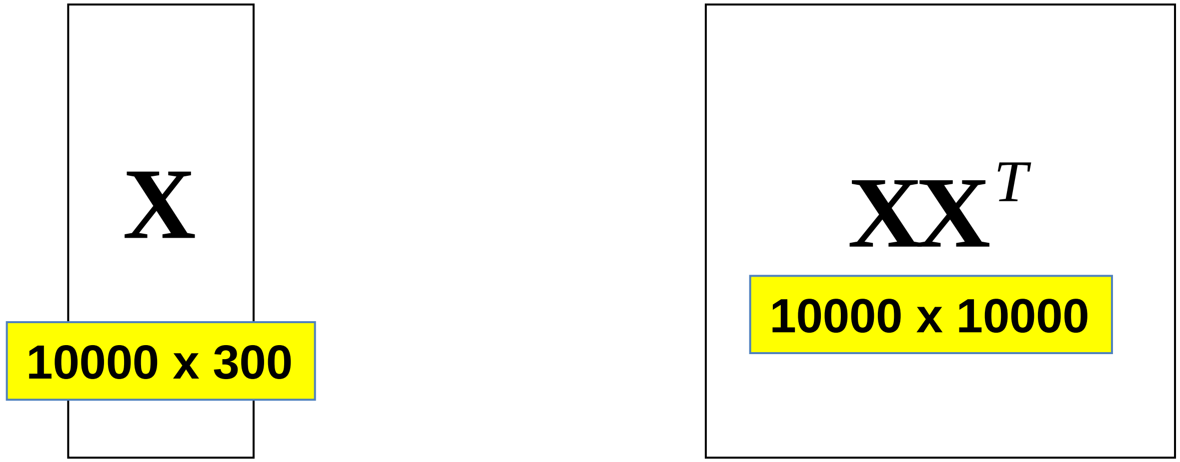
PCA as an instance of LGM

- Viewing PCA as an instance of linear Gaussian models leads to EM solution
- Very effective in dealing with high-dimensional and/or data poor situations

- An aside: Another simpler solution for the same situation..

An Aside: The GRAM trick

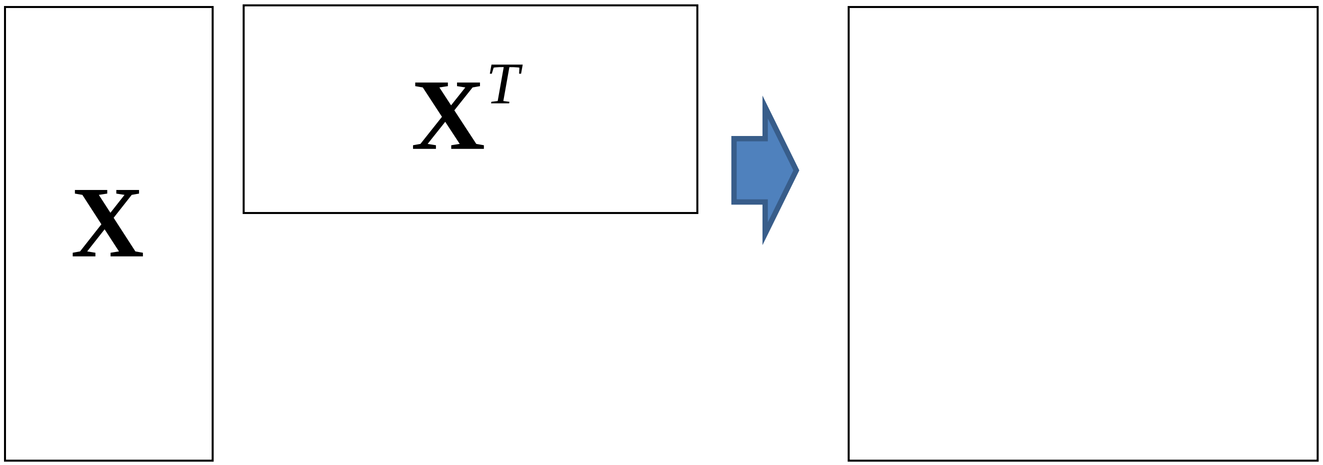




- The number of non-zero Eigen values is no more than the length of the smallest “edge” of **X**
  - 300 in this case
- This leads to the “gram” trick..

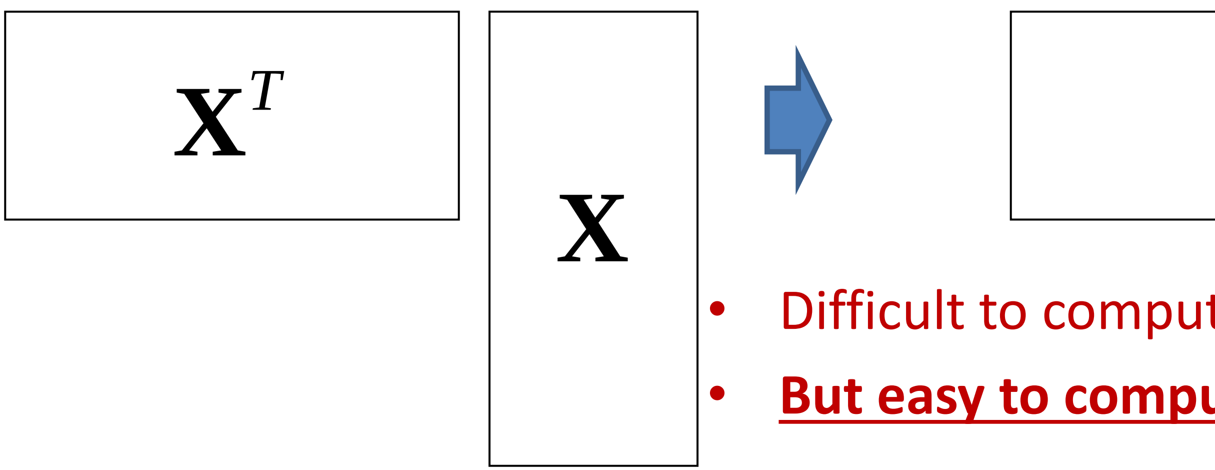
- Assumption $\mathbf{X}^T\mathbf{X}$ is invertible: the instances are linearly independent

An Aside: The GRAM trick



If X is 10000 x 300,
XX^T = 10000 x 10000

- $\mathbf{XX^T}$ is large but $\mathbf{X^TX}$ is not



If X is 10000 x 300,
X^TX = 300 x 300

- Difficult to compute Eigen vectors of $\mathbf{XX^T}$
- **But easy to compute Eigen vectors of $\mathbf{X^TX}$**

The Gram Trick

- To compute principal vectors we Eigendecompose $\mathbf{XX^T}$

$$\left(\mathbf{XX}^T\right)\mathbf{E} = \mathbf{E}\Lambda$$

- Let us find the Eigen vectors of $\mathbf{X^TX}$ instead

$$\left(\mathbf{X}^T\mathbf{X}\right)\hat{\mathbf{E}} = \hat{\mathbf{E}}\hat{\Lambda}$$

- Manipulating it slightly

Note that for a diagonal matrix:
$\Lambda\Lambda^{-0.5} = \Lambda^{-0.5}\Lambda$

$$\mathbf{X}^T\mathbf{X}\hat{\mathbf{E}}\hat{\Lambda}^{-0.5} = \hat{\mathbf{E}}\hat{\Lambda}^{-0.5}\hat{\Lambda}$$

The Gram Trick

- Eigendecompose $\mathbf{X}^T\mathbf{X}$ instead of $\mathbf{X}\mathbf{X}^T$

$$\left(\mathbf{X}^T\mathbf{X}\right)\hat{\mathbf{E}} = \hat{\mathbf{E}}\hat{\Lambda}$$

$$\mathbf{X}^T\mathbf{X}\hat{\mathbf{E}}\hat{\Lambda}^{-0.5} = \hat{\mathbf{E}}\hat{\Lambda}^{-0.5}\hat{\Lambda}$$

$$\left(\mathbf{X}\mathbf{X}^T\right)\left(\mathbf{X}\hat{\mathbf{E}}\hat{\Lambda}^{-0.5}\right) = \left(\mathbf{X}\hat{\mathbf{E}}\hat{\Lambda}^{-0.5}\right)\hat{\Lambda}$$

- Letting: $\mathbf{X}\hat{\mathbf{E}}\hat{\Lambda}^{-0.5} = \mathbf{E}$

$$\left(\mathbf{X}\mathbf{X}^T\right)\mathbf{E} = \mathbf{E}\hat{\Lambda}$$

- E is the matrix of Eigenvectors of $\mathbf{X}\mathbf{X}^T$!!!

The Gram Trick

- **When X is low rank or XX^T is too large:**

- Compute X^TX instead
  - Will be manageable size
- Perform Eigen Decomposition of X^TX

$$\left(\mathbf{X}^T\mathbf{X}\right)\hat{\mathbf{E}} = \hat{\mathbf{E}}\hat{\Lambda}$$

- **Compute Eigenvectors of XX^T as**

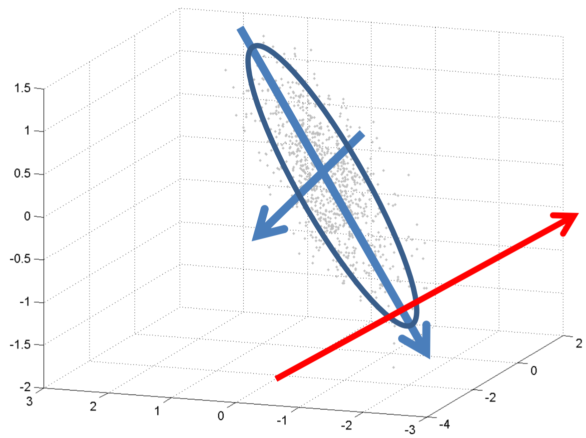
$$\mathbf{X}\hat{\mathbf{E}}\hat{\Lambda}^{-0.5} = \mathbf{E}$$

- **These are the principal components of X**

Why EM PCA

- Dimensionality / Rank has alternate potential solution
  - Gram Trick
- Other uses?
  - Noise
  - Incomplete data

PCA with noisy data



$$\mathbf{x} = \mathbf{V}\mathbf{w} + \mathbf{e} + \mathbf{n}$$

$$\mathbf{w} \sim N(0, I)$$

$$\mathbf{e} \sim N(0, E)$$

$$\mathbf{n} \sim N(0, B)$$

- Error is orthogonal to principal directions
  - $\mathbf{V}^T\mathbf{e} = \mathbf{0}; \ \mathbf{e}^T\mathbf{V} = \mathbf{0}$
- Noise is isotropic
  - B is diagonal
  - Noise is not orthogonal to either **V** or **e**

LGM: The complete EM algorithm

- Initialize $\mathbf{V}$ and E
- E step:

$$E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}] = \mathbf{V}^T(\mathbf{V}\mathbf{V}^T + E)^{-1}\mathbf{x}_i$$

$$E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}\mathbf{w}^T] = I - \mathbf{V}^T(\mathbf{V}\mathbf{V}^T + E)^{-1}\mathbf{V} + E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]^T$$

- M step:

$$\mathbf{V} = \left(\sum_i \mathbf{x}_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}^T]\right)\left(\sum_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}\mathbf{w}^T]\right)^{-1}$$

$$E = \frac{1}{N}\sum_i \mathbf{x}_i\mathbf{x}_i^T - \frac{1}{N}\mathbf{V}\sum_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]\mathbf{x}_i^T$$

PCA with Noisy Data

- Initialize $\mathbf{V}$ and B
- E step:

$$\beta = \mathbf{V}^T(\mathbf{V}\mathbf{V}^T + B)^{-1}$$

$$\mathbf{W} = \beta\mathbf{X}$$

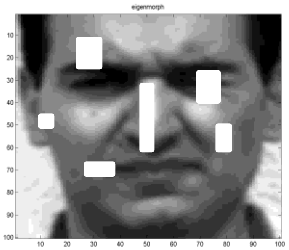
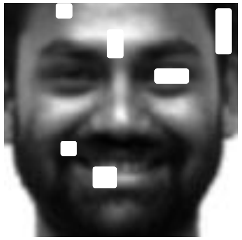
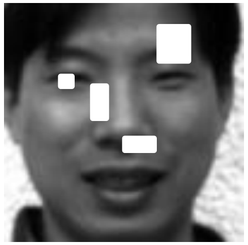
$$\mathbf{C} = NI - N\beta\mathbf{V} + \mathbf{W}\mathbf{W}^T$$

- M step:

$$\mathbf{V} = \mathbf{X}\mathbf{W}^T\mathbf{C}^{-1}$$

$$B = \frac{1}{N} diag\left(\mathbf{X}\mathbf{X}^T - \mathbf{V}\mathbf{W}\mathbf{X}^T\right)$$

PCA with *Incomplete* Data

- How to compute principal directions when some components in your training data are missing?

- Eigen decomposition is not possible
  – Cannot compute correlation matrix with missing data

PCA with missing data

- How it goes
- Given : $\mathbf{X} = \{\mathbf{X}_c, \mathbf{X}_m\}$
  - $\mathbf{X}_\mathbf{m}$ are missing components

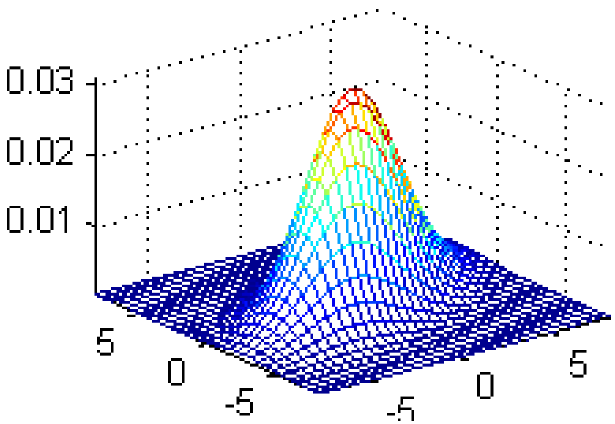
1. Initialize: Initialize $\mathbf{X}_m$
2. Build "complete" data $\mathbf{X} = \{\mathbf{X}_c, \mathbf{X}_m\}$
3. PCA ($\mathbf{X} = \mathbf{VW}$): Estimate $\mathbf{V}$
   - $\mathbf{V}$ must have fewer bases than dimensions of $\mathbf{X}$
4. $\mathbf{W} = \mathbf{V}^\mathbf{T}\mathbf{X}$
5. $\hat{\mathbf{X}} = \mathbf{VW}$
6. Select $\mathbf{X}_\mathbf{m}$ from $\hat{\mathbf{X}}$
7. Return to 2

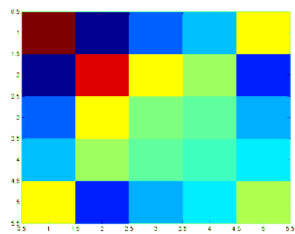
LGM for PCA

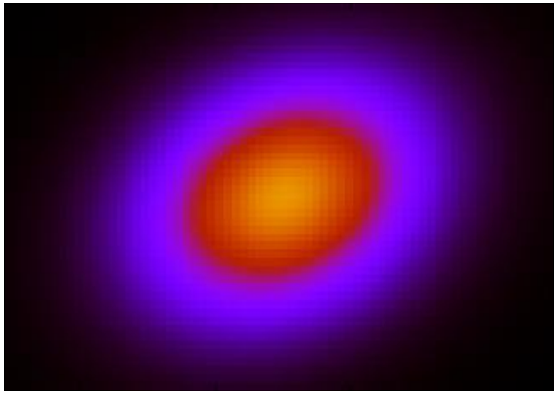
- Obviously many uses:
  - Ill-conditioned data
  - Noise
  - Missing data
  - Any combination of the above..

LGMs : Application 2

Learning with insufficient data

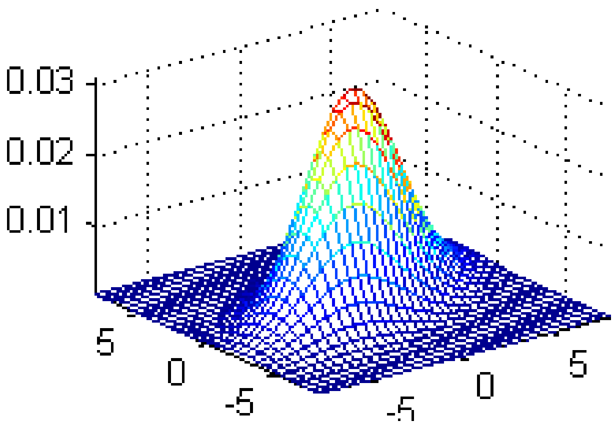


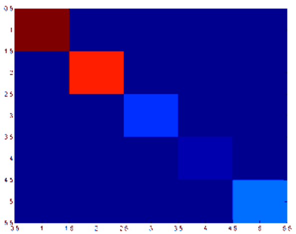


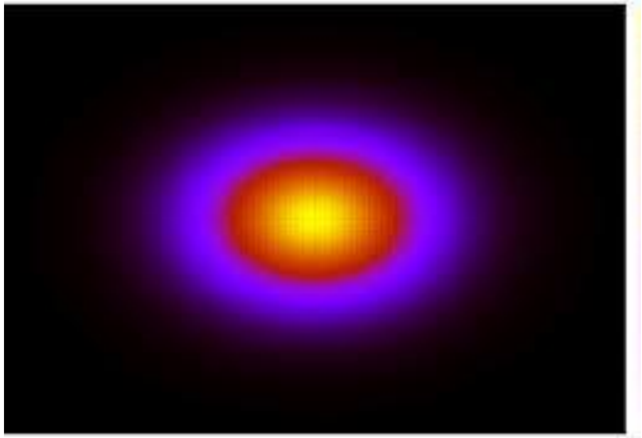


- The full covariance matrix of a Gaussian has D^2 terms
- Fully captures the relationships between variables
- Problem: **Needs a lot of data to estimate robustly**

An Approximation







- Assume the covariance is diagonal
  - Gaussian is aligned to axes : no correlation between dimensions
  - Covariance has only *D* terms
- **Needs less data**
- **Problem : Model loses all information about correlation between dimensions**

Is There an Intermediate

- Capture the most important correlations
- But require less data

- Solution: Find the key subspaces in the data
  – Capture the complete correlations in these subspaces
  – Assume data is otherwise uncorrelated

Factor Analysis

$$\mathbf{x} = \mathbf{V}\mathbf{w} + \mathbf{e}$$

$$\mathbf{w} \sim N(0, I)$$
$$\mathbf{e} \sim N(0, E)$$

$$\boxed{\mathbf{x} \sim N(0, \mathbf{V}\mathbf{V}^T + E)}$$

- *E* is a full rank diagonal matrix
- **V** has *K* columns: K-dimensional subspace
  - We will capture all the correlations in the subspace represented by **V**
- Estimated covariance: Diagonal covariance *E* plus the covariance between dimensions in **V**

Factor Analysis

- Initialize $\mathbf{V}$ and E
- E step:

$$E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}] = \mathbf{V}^T(\mathbf{V}\mathbf{V}^T + E)^{-1}\mathbf{x}_i$$

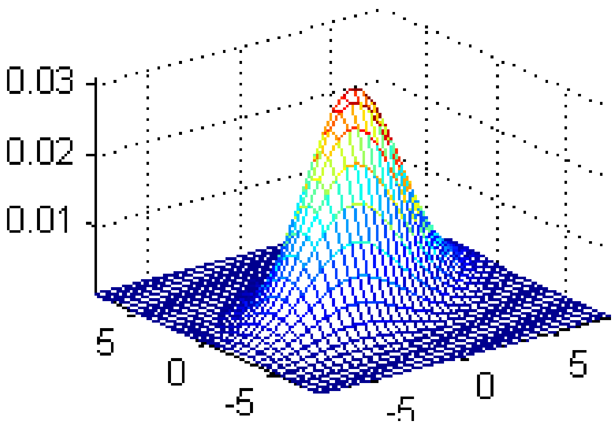
$$E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}\mathbf{w}^T] = I - \mathbf{V}^T(\mathbf{V}\mathbf{V}^T + E)^{-1}\mathbf{V} + E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]^T$$

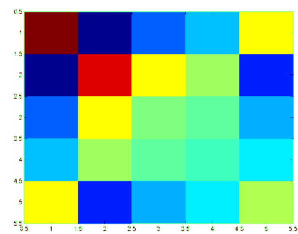
- M step:

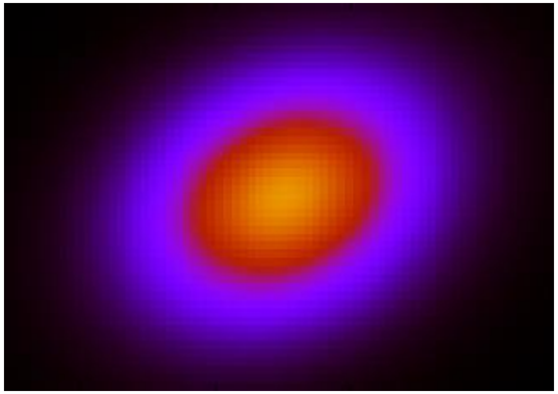
$$\mathbf{V} = \left(\sum_i \mathbf{x}_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}^T]\right)\left(\sum_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}\mathbf{w}^T]\right)^{-1}$$

$$E = \frac{1}{N} diag\left(\sum_i \mathbf{x}_i\mathbf{x}_i^T - \frac{1}{N}\mathbf{V}\sum_i E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]\mathbf{x}_i^T\right)$$

FA Gaussian







- Will get a full covariance matrix
- But only estimate DK terms
- Data insufficiency less of a problem

The Factor Analysis Model

$$\mathbf{x} = \underbrace{\mathbf{V}}_{\text{LOADINGS}}\underbrace{\mathbf{w}}_{\text{FACTORS}} + \mathbf{e}$$

$$\mathbf{w} \sim N(0, I)$$
$$\mathbf{e} \sim N(0, E)$$

- Often used to learn distribution of data when we have insufficient data
- Often used in psychometrics
  - Underlying model: The actual systematic variations in the data are totally explained by a small number of "factors"
  - FA uncovers these factors

FA: Example

- Hypothesis: there are two kinds of intelligence, "verbal" and "mathematical",
  - neither is directly observed.
  - Evidence sought from examination scores from each of 10 different academic fields of 1000 students.
- Solution: Find out if distribution is well explained by two factors
  - Hack: Attempt to relate factors to verbal and math IQ

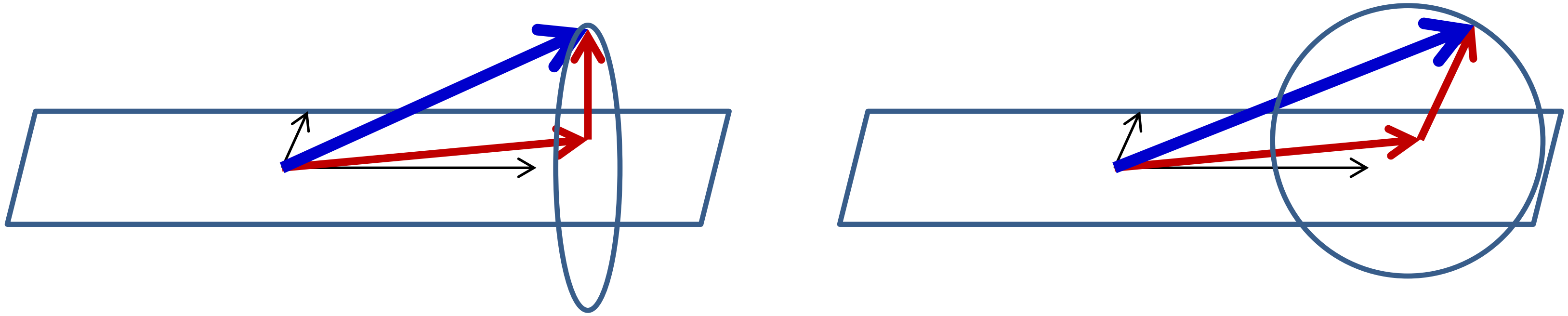
FA, PCA etc.

$$\mathbf{x} = \mathbf{V}\mathbf{w} + \mathbf{e}$$

$$\mathbf{w} \sim N(0, I)$$
$$\mathbf{e} \sim N(0, E)$$

- Note: distinction between PCA and FA is only in the assumptions about **e**
- FA looks a lot like PCA with noise
- FA can also be performed with incomplete data

FA, PCA etc.



- PCA: Error is always at 90 degrees to the bases in $\mathbf{V}$
- FA: Error may be at any angle
- PCA used mainly to find *principal* directions that capture most of the variance
  - Bases in V will be orthogonal to one another
- FA tries to capture most of the covariance

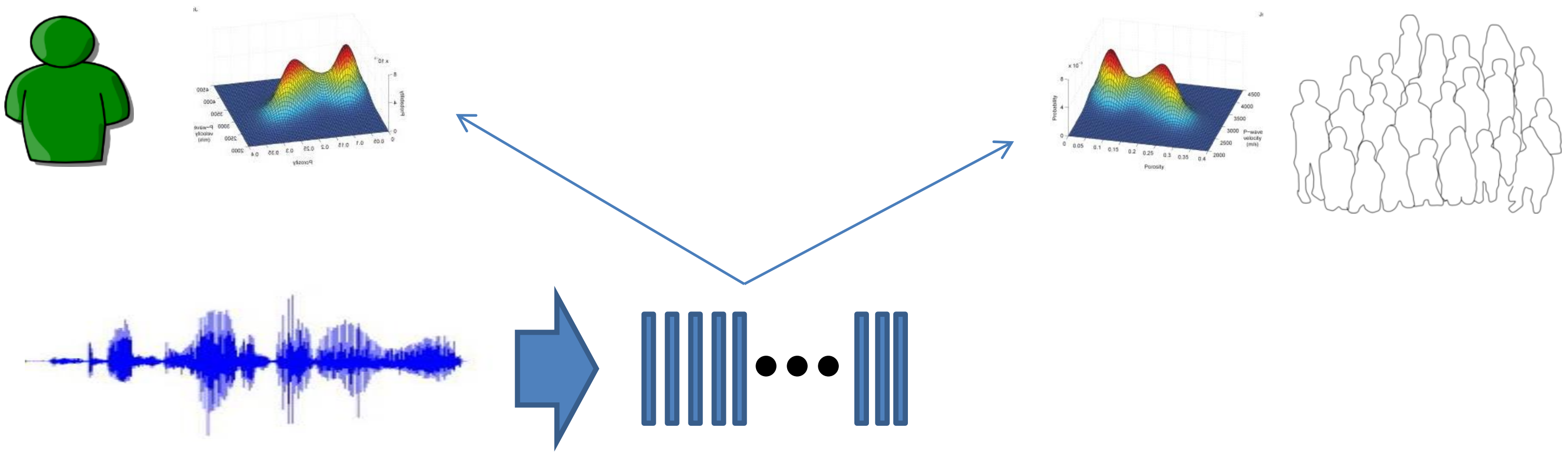
FA: A very successful use

- Voice biometrics: Speaker identification

- Given: Only a small amount of training data from a speaker, learn model for speaker
  - Use to verify speaker later

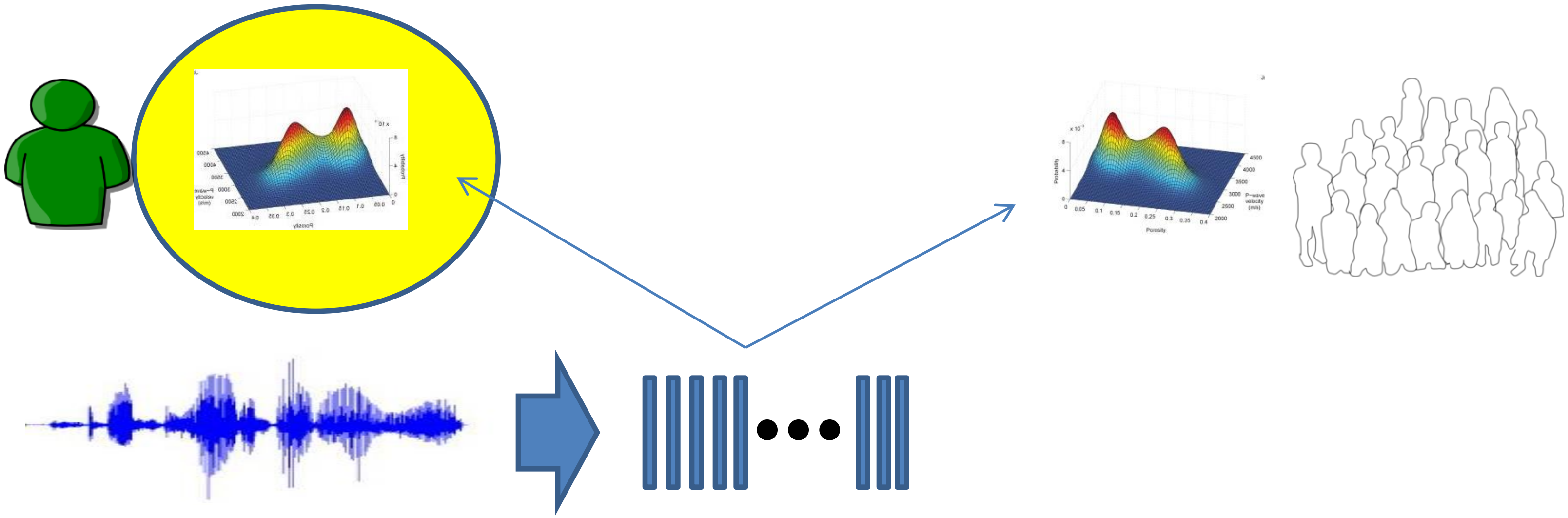
- Problem: Immense variation in ways people can speak
  - 15 minutes of training data totally insufficient!

Speaker Verification



- A model represents distribution of cepstral vectors for the speaker
- A second model represents everyone else (potential imposters)
- The cepstra computed from a test recording are “scored” against both models
  - Accept the speaker if the speaker model scores higher

Speaker Verification

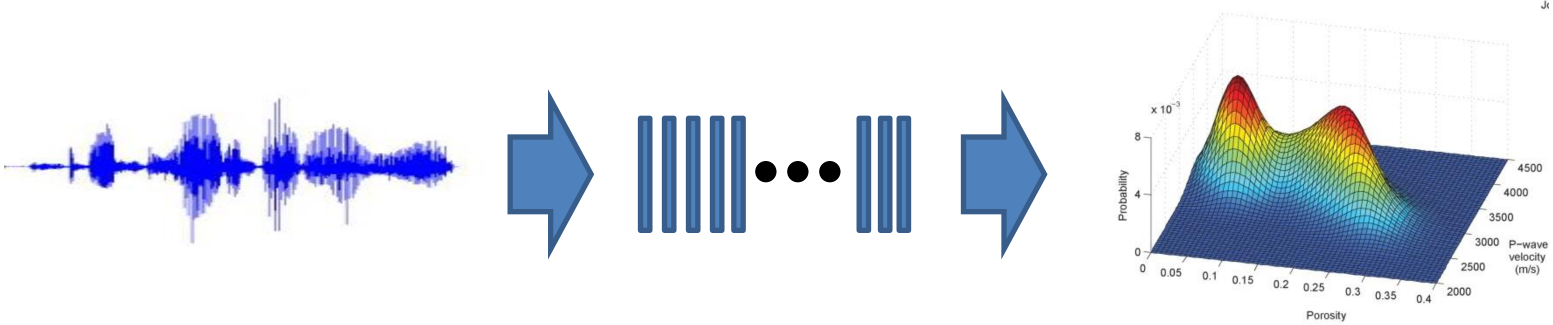


- Problem: One typically has only a few seconds or minutes of training data from the speaker
- Hard to estimate speaker model
- Test data may be spoken differently, or come over a different channel, or in noise
  - Wont really match

Hypothesis

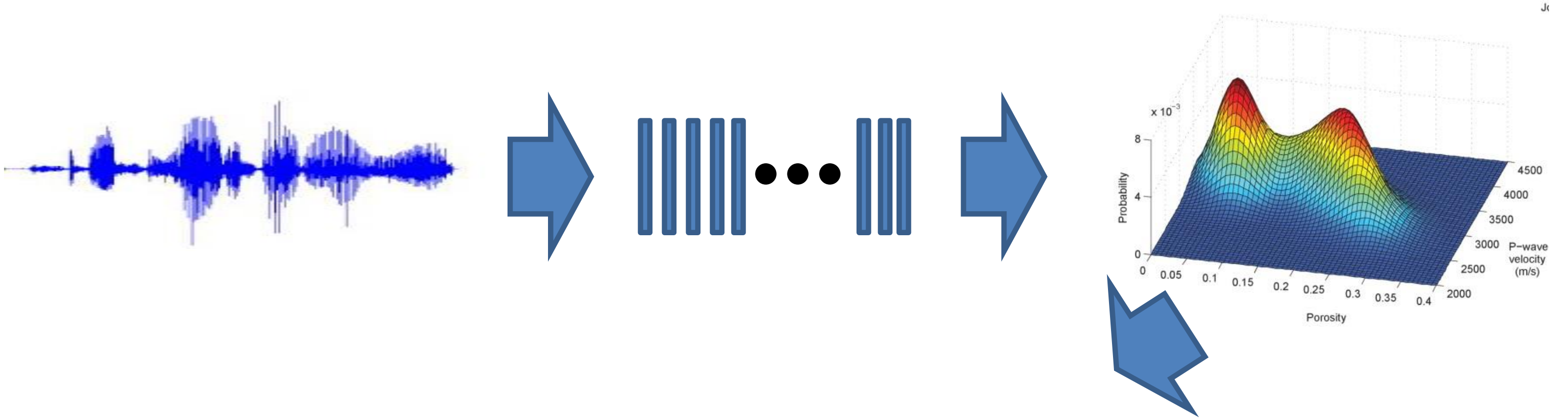
- Variations between different instances of the utterance spoken by the same speaker related to only a few factors
- Factors are common to all speakers
- Solution: Learn factors by analyzing many speakers
  – Use learned factors to predict variations for a given speaker
  – Can provide robust models for a speaker from very little data

Representing the Data: "super vectors"



- Convert recordings to a sequence of feature vectors
  - Cepstra
- Compute the probability distribution for the data
  - Modeled as a Gaussian mixture
- The data are represented by the parameters of the distribution

Representing the Data: "super vectors"

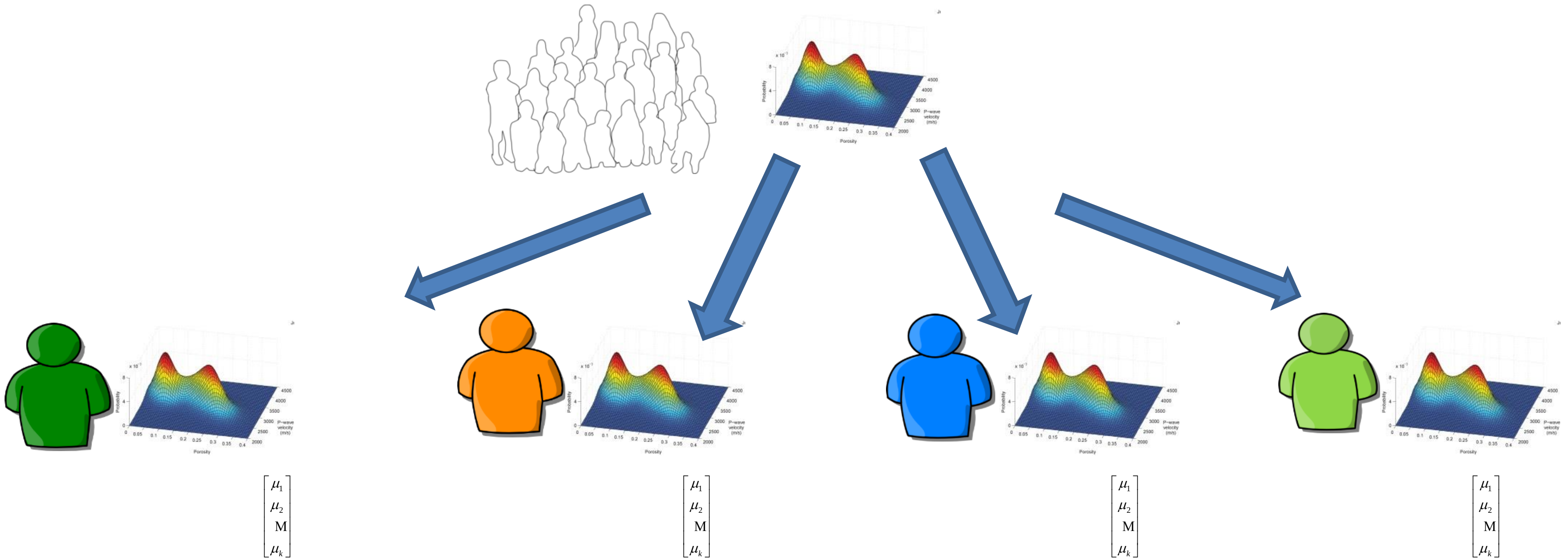


$$P(X) = \sum_k w_k N(X; \mu_k, \Theta_k)$$

This "supervector" is the feature that represents the recording

$$\begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_k \end{bmatrix}$$

Training



- Supervectors are obtained for each training speaker by adapting a "Universal background model" trained from large amounts of data

Training the Factor Analyzer

$\begin{bmatrix} \mu_1 \\ \mu_2 \\ \mathrm{M} \\ \mu_k \end{bmatrix}$

$\begin{bmatrix} \mu_1 \\ \mu_2 \\ \mathrm{M} \\ \mu_k \end{bmatrix}$

$\begin{bmatrix} \mu_1 \\ \mu_2 \\ \mathrm{M} \\ \mu_k \end{bmatrix}$

$\begin{bmatrix} \mu_1 \\ \mu_2 \\ \mathrm{M} \\ \mu_k \end{bmatrix}$

$$\mathbf{x} = \mathbf{V}\mathbf{w} + \mathbf{e} \qquad \mathbf{w} \sim N(0, I) \;\; \mathbf{e} \sim N(0, E)$$

- The supervectors are assumed to be the output of a linear Gaussian process
- Use FA to estimate $\mathbf{V}$
  - $\mathbf{V}$ are the factors that cause variations
  - The *real* information is in the factor $\mathbf{w}$

Training models for *a speaker*

$$\begin{bmatrix} \mu_1 \\ \mu_2 \\ \mathrm{M} \\ \mu_k \end{bmatrix}$$

$$\mathbf{x} = \mathbf{V}\mathbf{w}_S + \mathbf{e} \qquad \mathbf{w} \sim N(0, I) \quad \mathbf{e} \sim N(0, E)$$

- From training data: estimate the means for the speaker to conform to the factor analysis
  - Constrained estimation: requires much less data
- Use the estimated means as the distribution for the speaker
  - Solves data insufficiency problem
  - Also solves the problem of variations

Many other applications..

- Exploratory FA
- Confirmatory FA..