Machine Learning for Signal Processing
Linear Gaussian Models

Class 21. 12 Nov 2013

Instructor: Bhiksha Raj

Administrivia

- HW3 is up
  - .
- Projects – please send us an update

Recap: MAP Estimators

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

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

Recap: MAP estimation

- x and y are jointly Gaussian

$$z = \begin{bmatrix} x \\ y \end{bmatrix}$$

$$E[z] = \mu_z = \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}$$

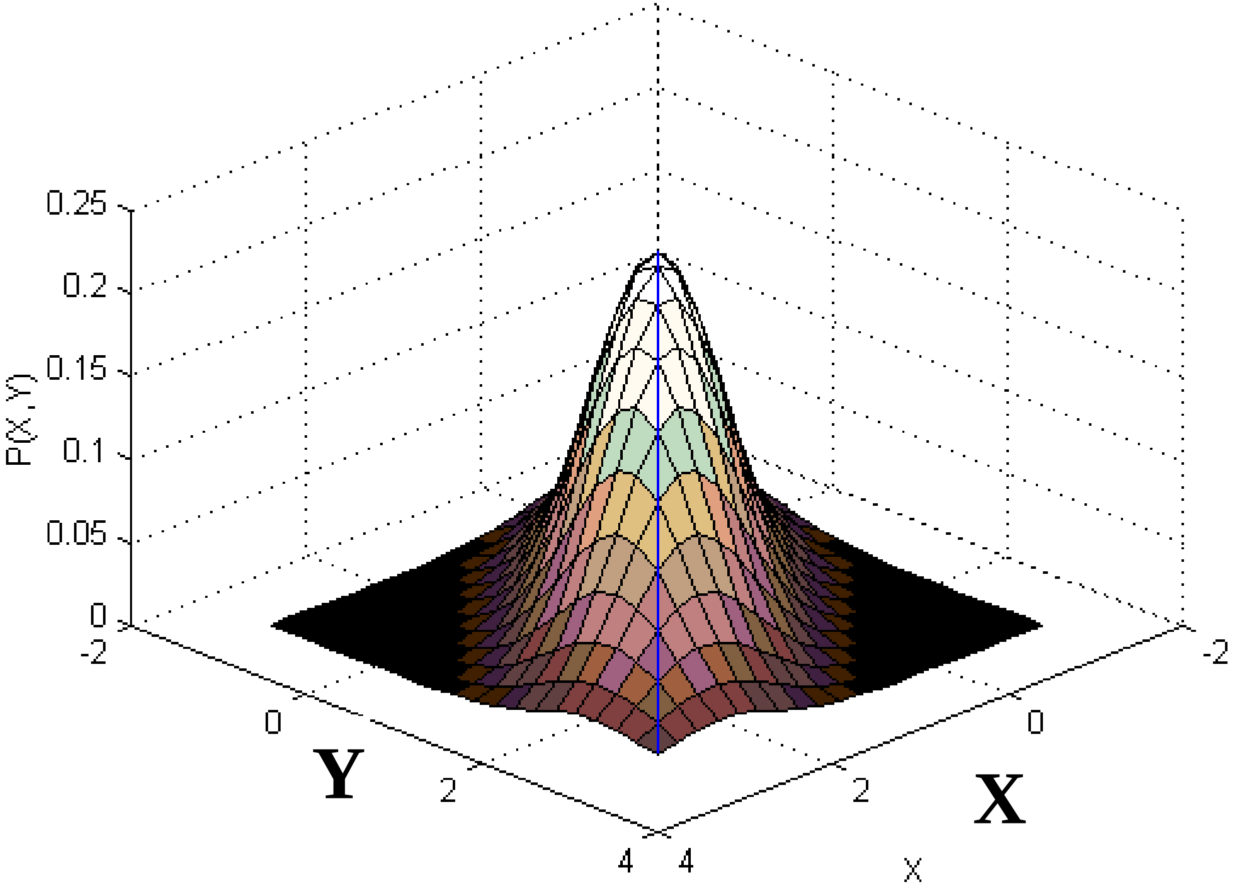
$$Var(z) = C_{zz} = \begin{bmatrix} C_{xx} & C_{xy} \\ C_{yx} & C_{yy} \end{bmatrix}$$

$$C_{xy} = E[(x - \mu_x)(y - \mu_y)^T]$$

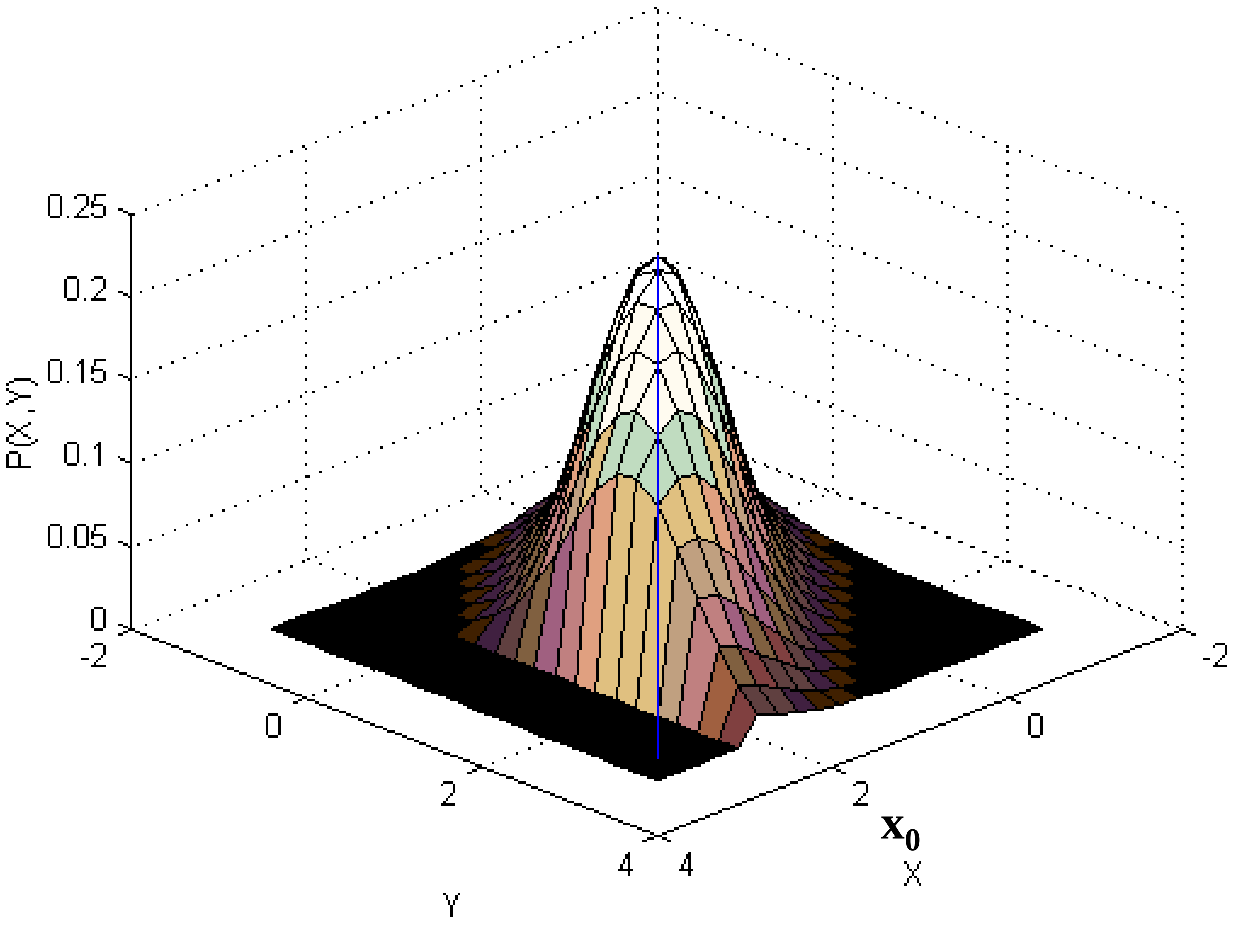
$$P(z) = N(\mu_z, C_{zz}) = \frac{1}{\sqrt{2\pi |C_{zz}|}} \exp\left(-0.5(z - \mu_z)(z - \mu_z)^T\right)$$

- ***z* is Gaussian**

MAP estimation: Gaussian PDF



MAP estimation: The Gaussian at a particular value of X

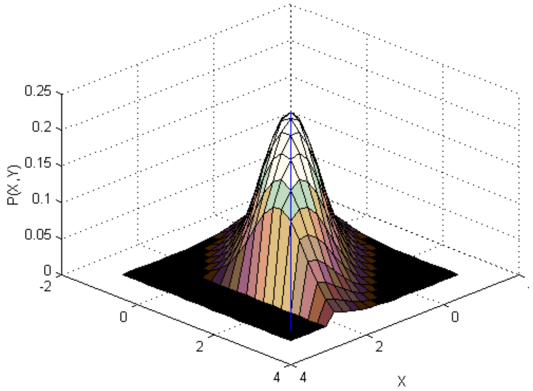


Conditional Probability of y|x

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

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

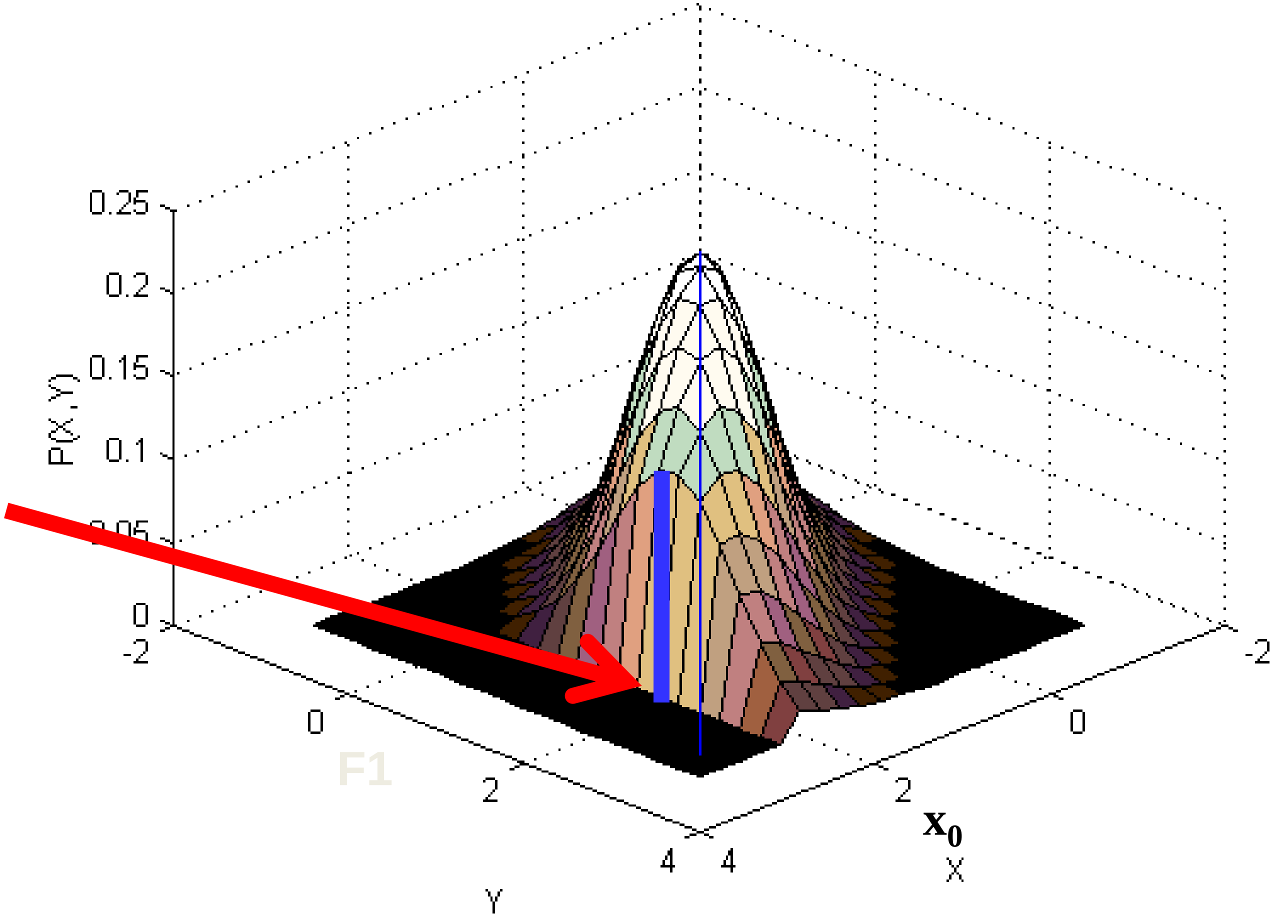
$$Var(y\,|\,x) = C_{yy} - C_{xy}^T C_{xx}^{-1} C_{xy}$$



- 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

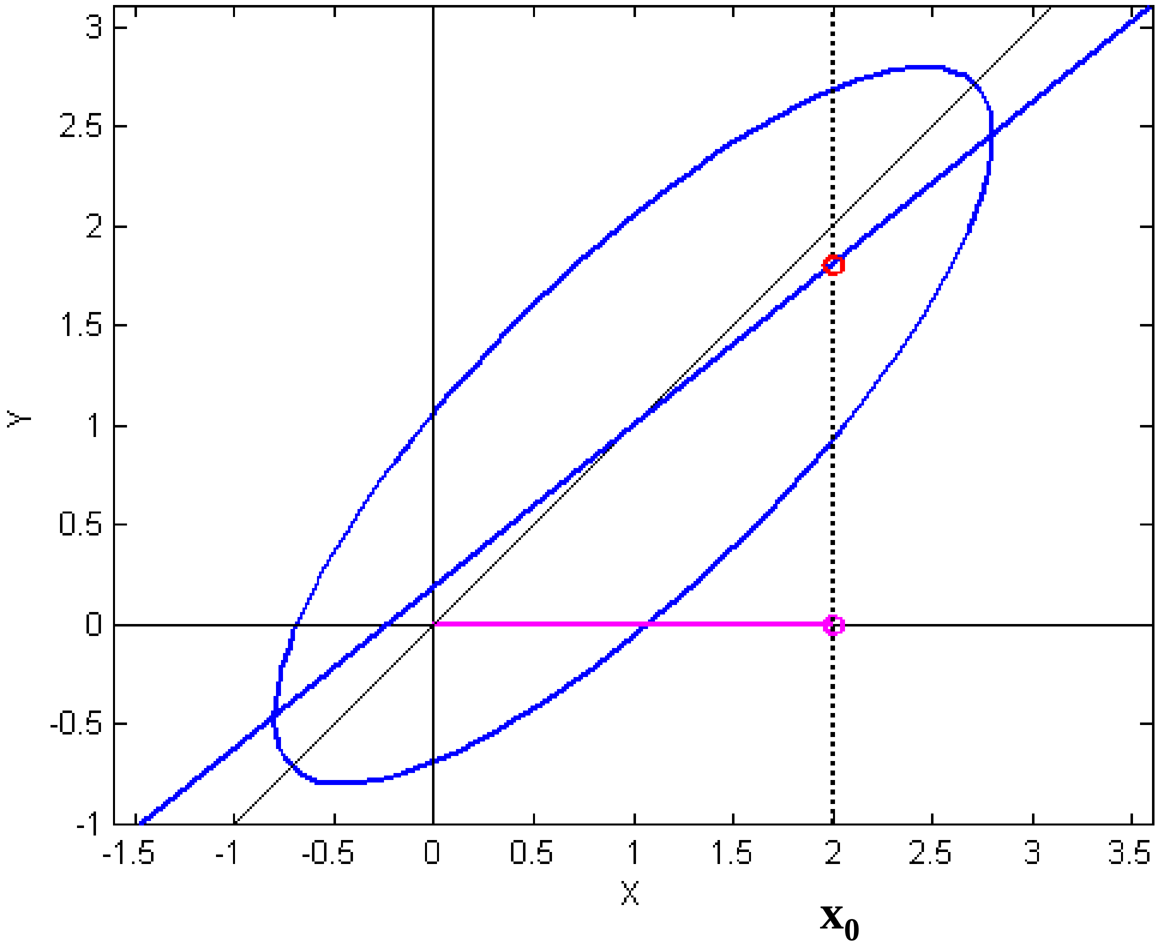
MAP estimation: The Gaussian at a particular value of X

Most likely value



MAP Estimation of a Gaussian RV

$$\hat{y} = \arg\max_y P(y \mid x) = E_{y|x}[y]$$



Its also a *minimum-mean-squared error* estimate

- Minimize error:

$$Err = E[\|\mathbf{y}-\hat{\mathbf{y}}\|^2 \mid \mathbf{x}] = E[(\mathbf{y}-\hat{\mathbf{y}})^T(\mathbf{y}-\hat{\mathbf{y}}) \mid \mathbf{x}]$$

$$Err = E[\mathbf{y}^T\mathbf{y} + \hat{\mathbf{y}}^T\hat{\mathbf{y}} - 2\hat{\mathbf{y}}^T\mathbf{y} \mid \mathbf{x}] = E[\mathbf{y}^T\mathbf{y} \mid \mathbf{x}] + \hat{\mathbf{y}}^T\hat{\mathbf{y}} - 2\hat{\mathbf{y}}^T E[\mathbf{y} \mid \mathbf{x}]$$

- Differentiating and equating to 0:

$$d.Err = 2\hat{\mathbf{y}}^T d\hat{\mathbf{y}} - 2E[\mathbf{y} \mid \mathbf{x}]^T d\hat{\mathbf{y}} = 0$$

$$\boxed{\hat{\mathbf{y}} = E[\mathbf{y} \mid \mathbf{x}]}$$

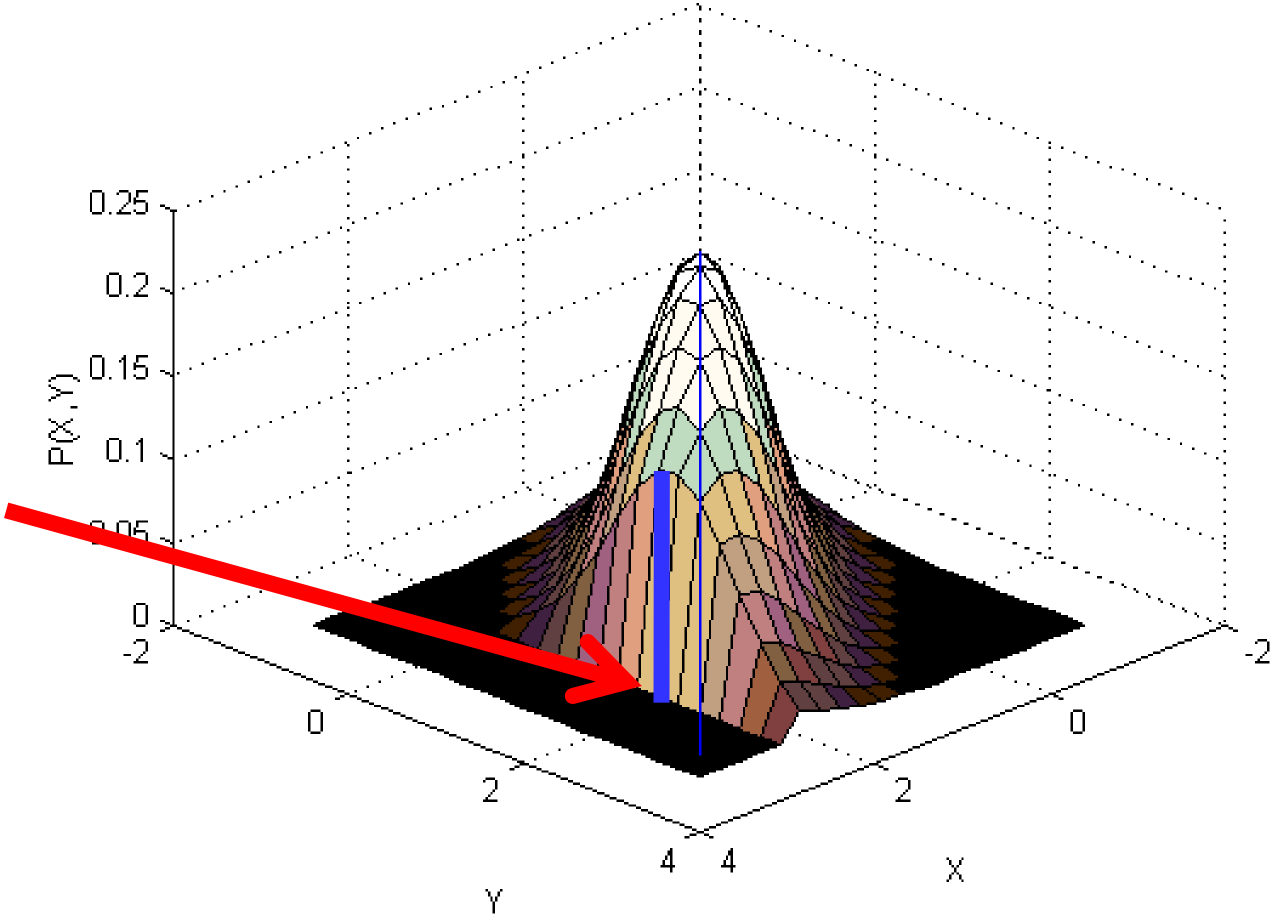
The MMSE estimate is the mean of the distribution

For the Gaussian: MAP = MMSE

Most likely value

is also

The MEAN value



- Would be true of any symmetric distribution

MMSE estimates for mixture distributions

$$P(\mathbf{y} \mid \mathbf{x}) = \sum_k P(k \mid \mathbf{x}) P(\mathbf{y} \mid k, \mathbf{x})$$

- Let $P(\mathbf{y}|\mathbf{x})$ be a mixture density
- The MMSE estimate of $\mathbf{y}$ is given by

$$E[\mathbf{y} \mid \mathbf{x}] = \int \mathbf{y} \sum_k P(k \mid \mathbf{x}) P(\mathbf{y} \mid k, \mathbf{x}) d\mathbf{y}$$

$$= \sum_k P(k \mid \mathbf{x}) \int \mathbf{y} P(\mathbf{y} \mid k, \mathbf{x}) d\mathbf{y}$$

$$= \sum_k P(k \mid \mathbf{x}) E[\mathbf{y} \mid k, \mathbf{x}]$$

- Just a weighted combination of the MMSE estimates from the component distributions

MMSE estimates from a Gaussian mixture

- Let P($\mathbf{x}$,$\mathbf{y}$) be a Gaussian Mixture

$$\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}$$

$$P(\mathbf{x},\mathbf{y}) = P(\mathbf{z}) = \sum_k P(k) N(\mathbf{z}; \mu_k, \Sigma_k)$$

- P($\mathbf{y}$|$\mathbf{x}$) is also a Gaussian mixture

$$P(\mathbf{y} \mid \mathbf{x}) = \frac{P(\mathbf{x},\mathbf{y})}{P(\mathbf{x})} = \frac{\sum_k P(k,\mathbf{x},\mathbf{y})}{P(\mathbf{x})} = \frac{\sum_k P(\mathbf{x}) P(k \mid \mathbf{x}) P(\mathbf{y} \mid \mathbf{x}, k)}{P(\mathbf{x})}$$

$$P(\mathbf{y} \mid \mathbf{x}) = \sum_k P(k \mid \mathbf{x}) P(\mathbf{y} \mid \mathbf{x}, k)$$

MMSE estimates from a Gaussian mixture

- Let $P(\mathbf{y}|\mathbf{x})$ is a Gaussian Mixture

$$P(\mathbf{y}\,|\,\mathbf{x}) = \sum_k P(k\,|\,\mathbf{x}) P(\mathbf{y}\,|\,\mathbf{x}, k)$$

$$P(\mathbf{y}, \mathbf{x}, k) = N(\begin{bmatrix} \mu_{k,\mathbf{x}} \\ \mu_{k,\mathbf{y}} \end{bmatrix}, \begin{bmatrix} C_{k,\mathbf{xx}} & C_{k,\mathbf{xy}} \\ C_{k,\mathbf{yx}} & C_{k,\mathbf{yy}} \end{bmatrix})$$

$$P(\mathbf{y}\,|\,\mathbf{x}, k) = N(\mu_{k,\mathbf{y}} + C_{k,\mathbf{yx}} C_{k,\mathbf{xx}}^{-1} (\mathbf{x} - \mu_{k,\mathbf{x}}), \Theta)$$

$$P(\mathbf{y}\,|\,\mathbf{x}) = \sum_k P(k\,|\,\mathbf{x}) N(\mu_{k,\mathbf{y}} + C_{k,\mathbf{yx}} C_{k,\mathbf{xx}}^{-1} (\mathbf{x} - \mu_{k,\mathbf{x}}), \Theta)$$

MMSE estimates from a Gaussian mixture

$$P(\mathbf{y} \mid \mathbf{x}) = \sum_k P(k \mid \mathbf{x}) N(\mu_{k,\mathbf{y}} + C_{k,\mathbf{yx}} C_{k,\mathbf{xx}}^{-1} (\mathbf{x} - \mu_{k,\mathbf{x}}), \Theta)$$

- P($\mathbf{y}|\mathbf{x}$) is a mixture Gaussian density
- E[$\mathbf{y}|\mathbf{x}$] is also a mixture

$$E[\mathbf{y} \mid \mathbf{x}] = \sum_k P(k \mid \mathbf{x}) E[\mathbf{y} \mid k, \mathbf{x}]$$

$$E[\mathbf{y} \mid \mathbf{x}] = \sum_k P(k \mid \mathbf{x}) \left(\mu_{k,\mathbf{y}} + C_{k,\mathbf{yx}} C_{k,\mathbf{xx}}^{-1} (\mathbf{x} - \mu_{k,\mathbf{x}}) \right)$$

MMSE estimates from a Gaussian mixture

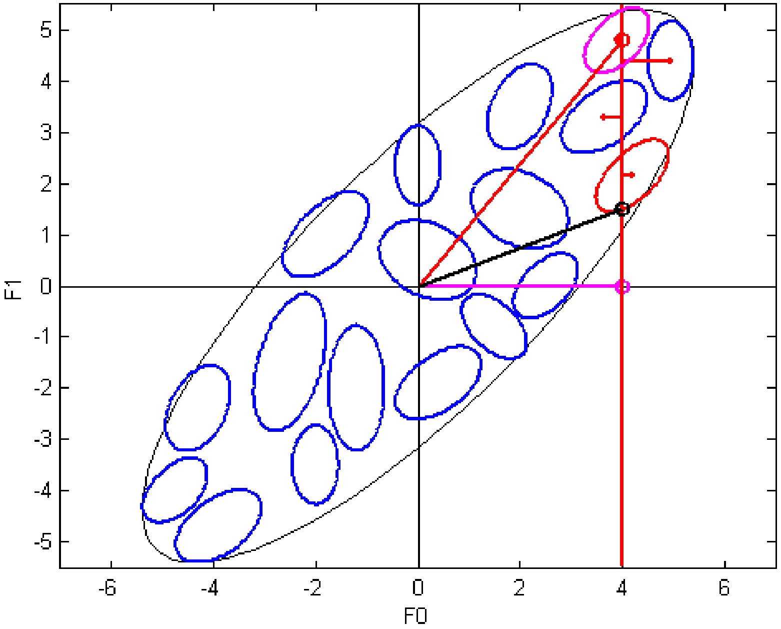
$$E[\mathbf{y} \mid \mathbf{x}] = \sum_k P(k \mid \mathbf{x})\left(\mu_{k,\mathbf{y}} + C_{k,\mathbf{yx}} C_{k,\mathbf{xx}}^{-1} (\mathbf{x} - \mu_{k,\mathbf{x}})\right)$$

- Weighted combination of MMSE estimates obtained from individual Gaussians!
- Weight $P(k|\mathbf{x})$ is easily computed too..

$$P(k \mid \mathbf{x}) = \frac{P(k,\mathbf{x})}{P(\mathbf{x})}$$

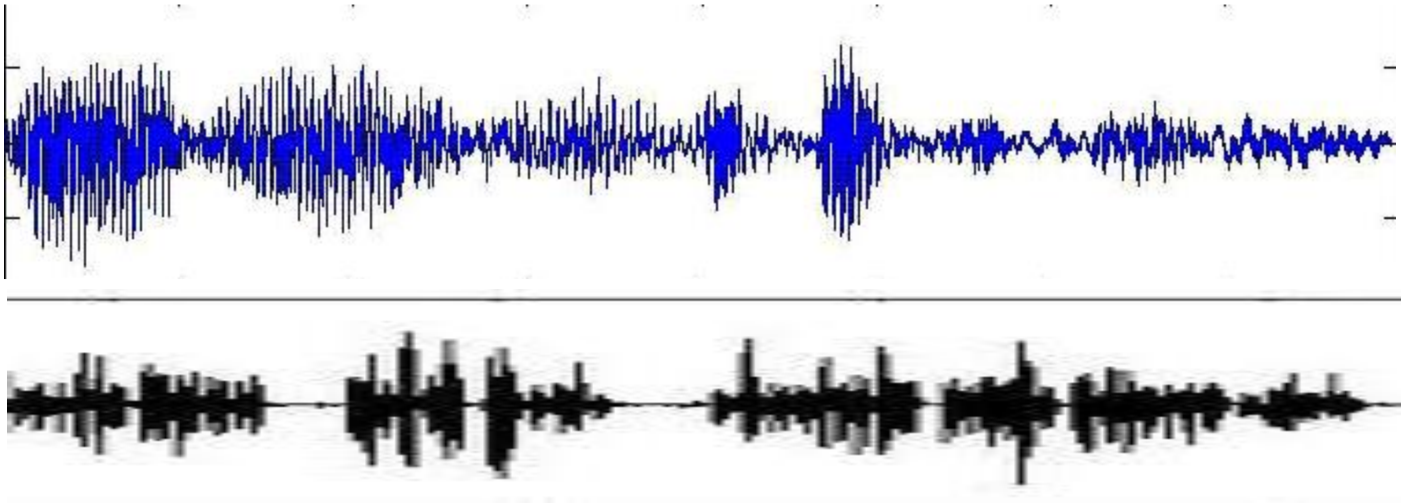
$$P(\mathbf{x}) = \sum_k P(k) N(\mu_{k,x}, C_{xx})$$

MMSE estimates from a Gaussian mixture



- A mixture of estimates from individual Gaussians

Voice Morphing



- *Align training recordings from both speakers*
  - Cepstral vector sequence
- Learn a GMM on joint vectors
- Given speech from one speaker, find MMSE estimate of the other
- *Synthesize from cepstra*

MMSE with GMM: Voice Transformation

- Festvox GMM transformation suite (Toda)

	awb	bdl	jmk	slt
awb	🔈	🔈	🔈	🔈
bdl	🔈	🔈	🔈	🔈
jmk	🔈	🔈	🔈	🔈
slt	🔈	🔈	🔈	🔈

MAP / ML / MMSE

- General statistical estimators
- All used to predict a variable, based on other parameters related to it..

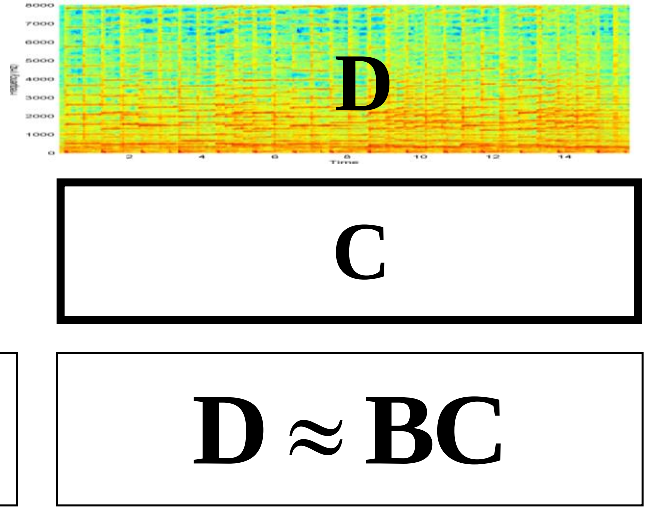
- Most common assumption: Data are Gaussian, all RVs are Gaussian
  - Other probability densities may also be used..

- For Gaussians relationships are linear as we saw..

Gaussians and more Gaussians..

- Linear Gaussian Models..
- But first a recap

A Brief Recap



D

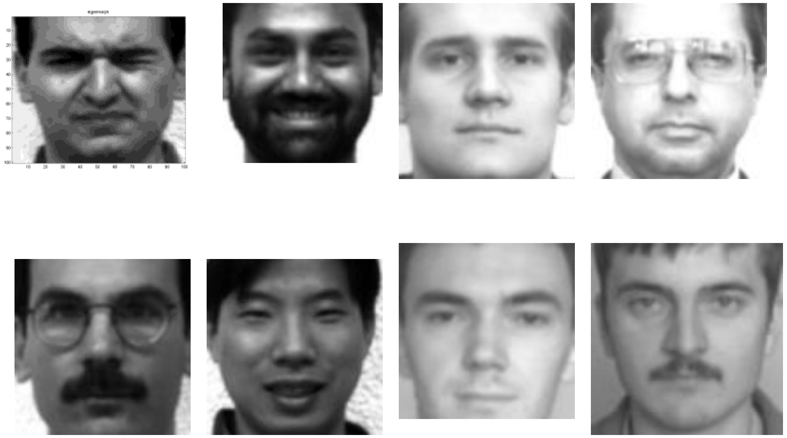
C

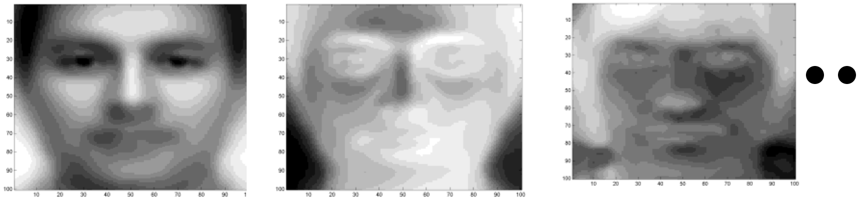
B

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

- 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

Remember Eigenfaces

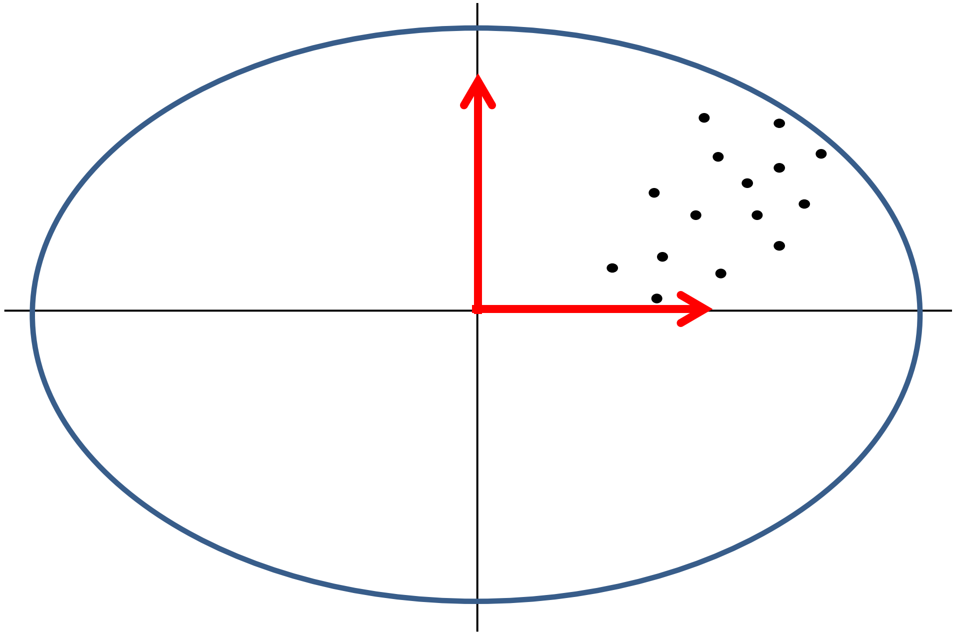




- Approximate every face f as
  $f = w_{f,1} V_1 + w_{f,2} V_2 + w_{f,3} V_3 +.. + w_{f,k} V_k$
- Estimate V to minimize the squared error

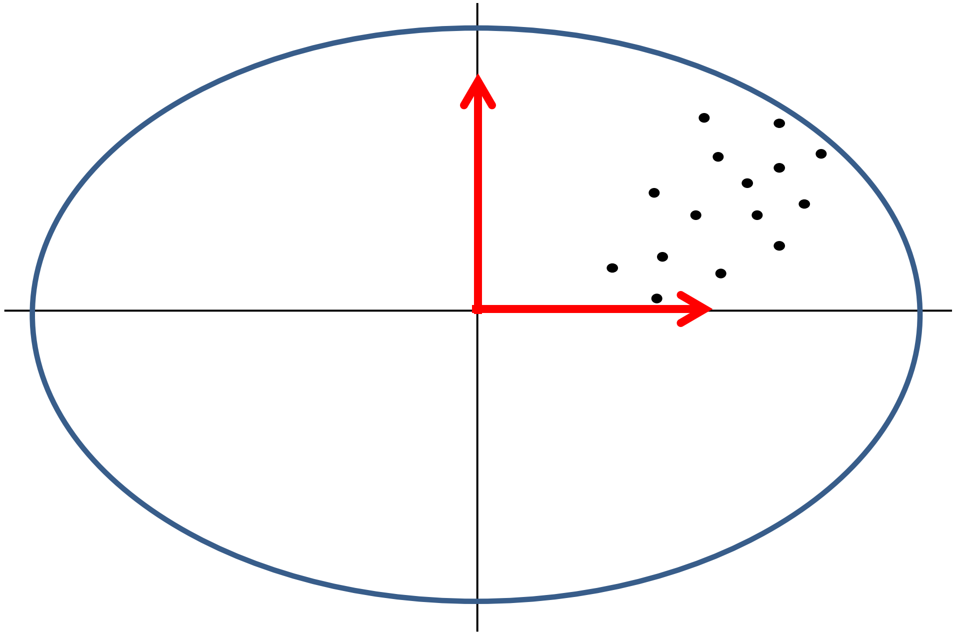
- *Error is unexplained by V_1.. V_k*
- ***Error is orthogonal to Eigenfaces***

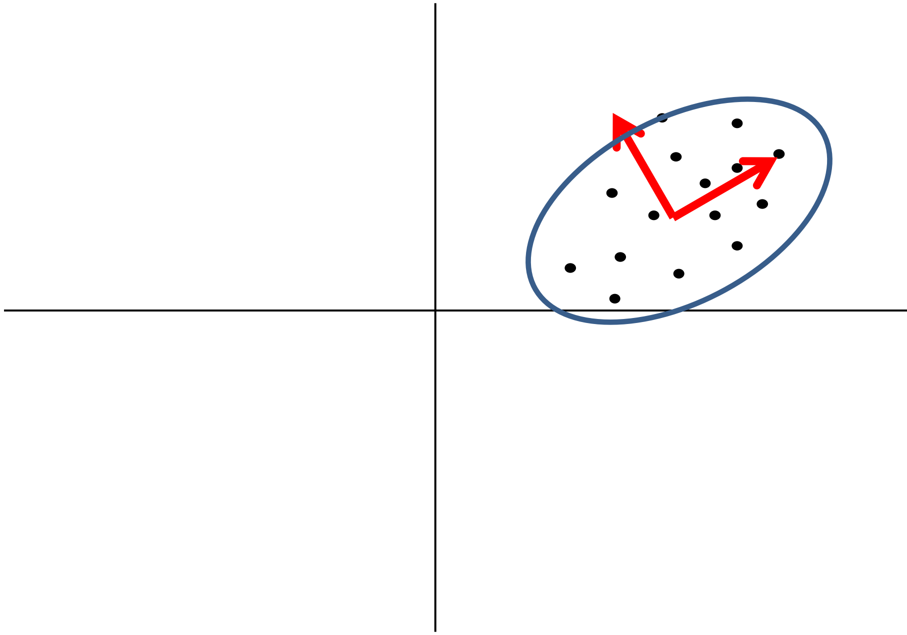
Karhunen Loeve vs. PCA



- Eigenvectors of the *Correlation* matrix:
  - Principal directions of tightest ellipse ***centered on origin***
  - Directions that retain maximum ***<u>energy</u>***

Karhunen Loeve vs. PCA

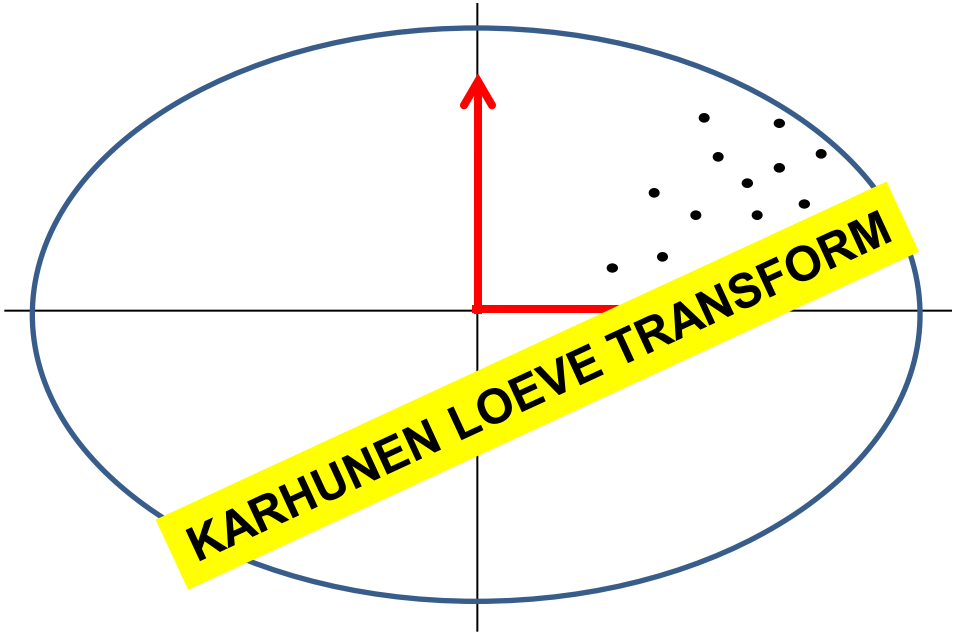


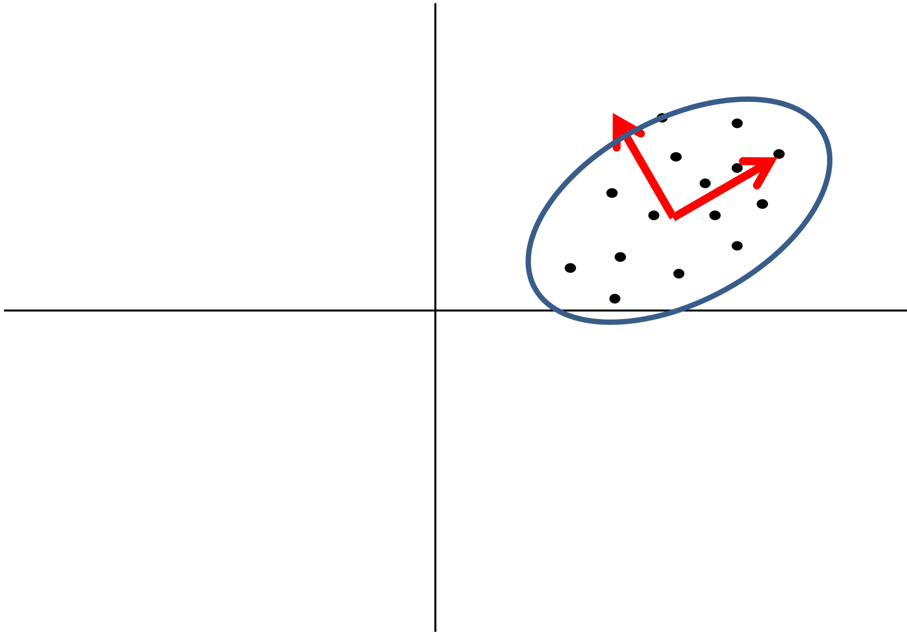


- Eigenvectors of the *Correlation* matrix:
  - Principal directions of tightest ellipse ***centered on origin***
  - Directions that retain maximum ***energy***

- **Eigenvectors of the *Covariance* matrix:**
  - **Principal directions of tightest ellipse *centered on data***
  - **Directions that retain maximum *variance***

Karhunen Loeve vs. PCA

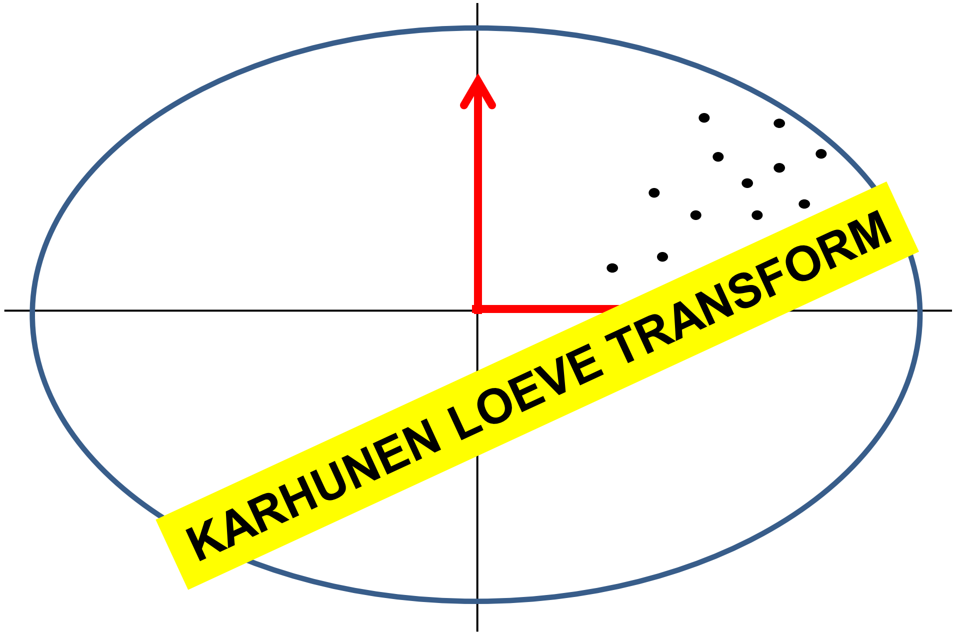


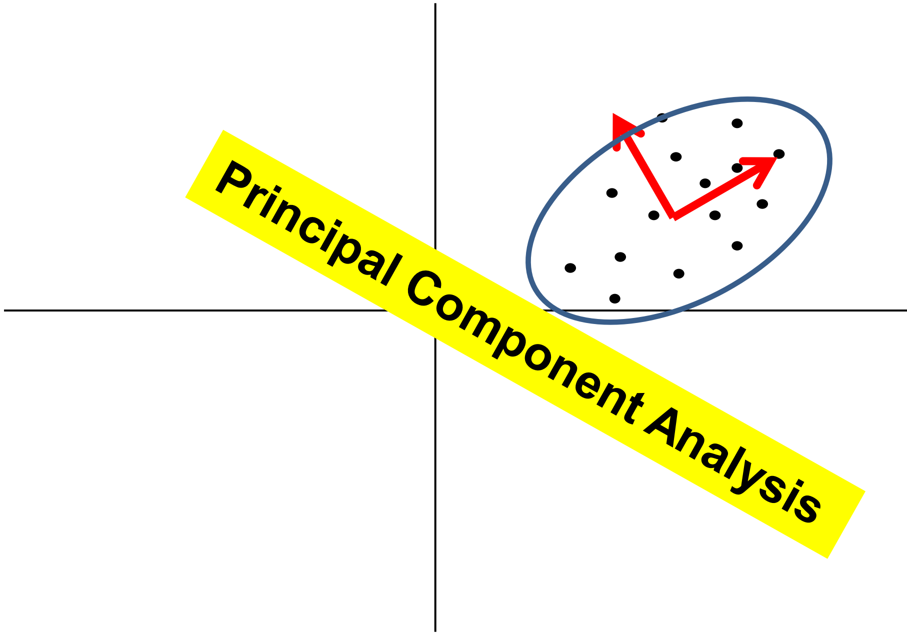


- Eigenvectors of the *Correlation* matrix:
  - Principal directions of tightest ellipse ***centered on origin***
  - Directions that retain maximum ***<u>energy</u>***

- **Eigenvectors of the *Covariance* matrix:**
  - **Principal directions of tightest ellipse *centered on data***
  - **Directions that retain maximum *<u>variance</u>***

Karhunen Loeve vs. PCA

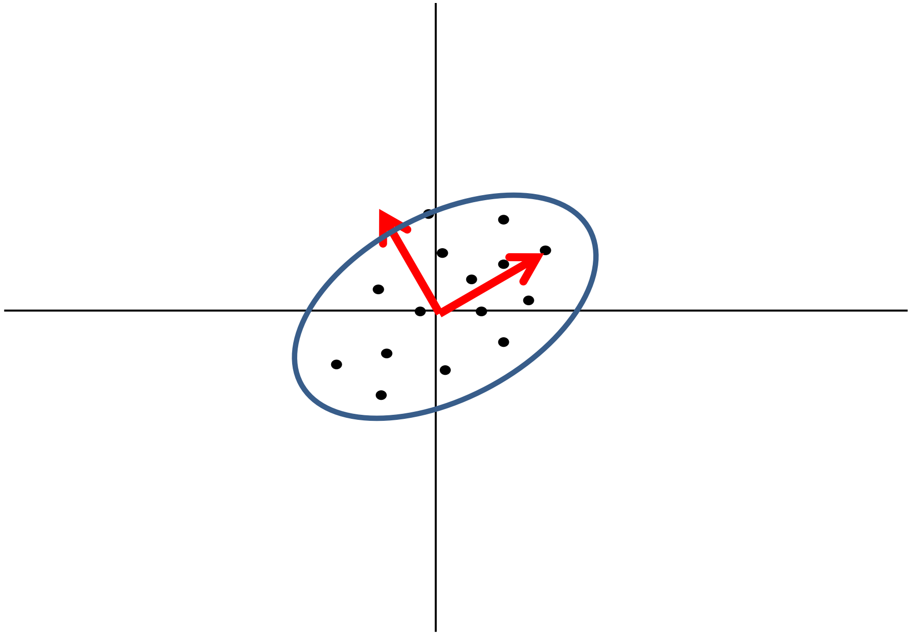




- Eigenvectors of the *Correlation* matrix:
  - Principal directions of tightest ellipse ***centered on origin***
  - Directions that retain maximum ***energy***

- **Eigenvectors of the *Covariance* matrix:**
  - **Principal directions of tightest ellipse *centered on data***
  - **Directions that retain maximum *variance***

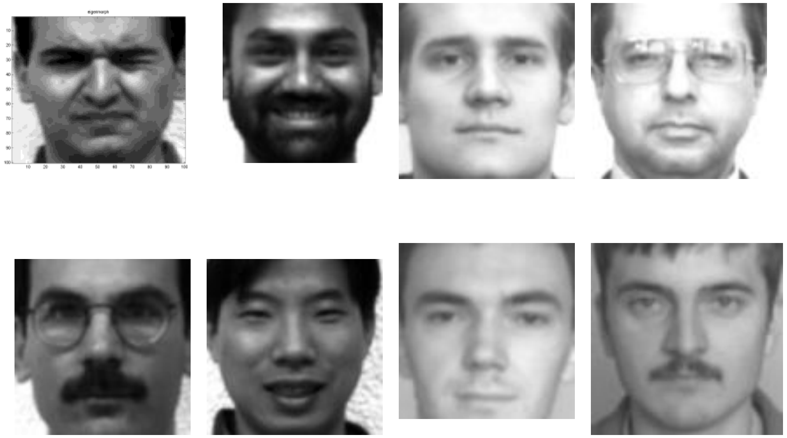
Karhunen Loeve vs. PCA

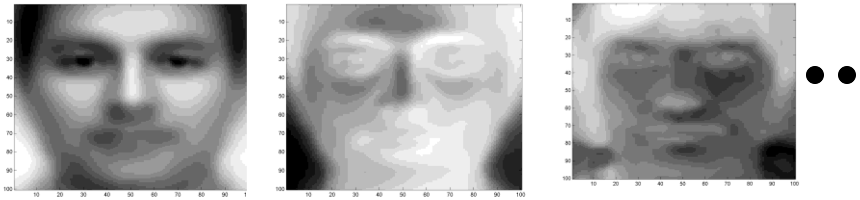


- If the data are naturally centered at origin, KLT == PCA

- Following slides refer to PCA!
  - Assume data centered at origin for simplicity
    - Not essential, as we will see..

Remember Eigenfaces





- Approximate every face f as
  $f = w_{f,1} V_1 + w_{f,2} V_2 + w_{f,3} V_3 + .. + w_{f,k} V_k$
- Estimate V to minimize the squared error

- *Error is unexplained by $V_1 .. V_k$*
- ***Error is orthogonal to Eigenfaces***

Eigen Representation

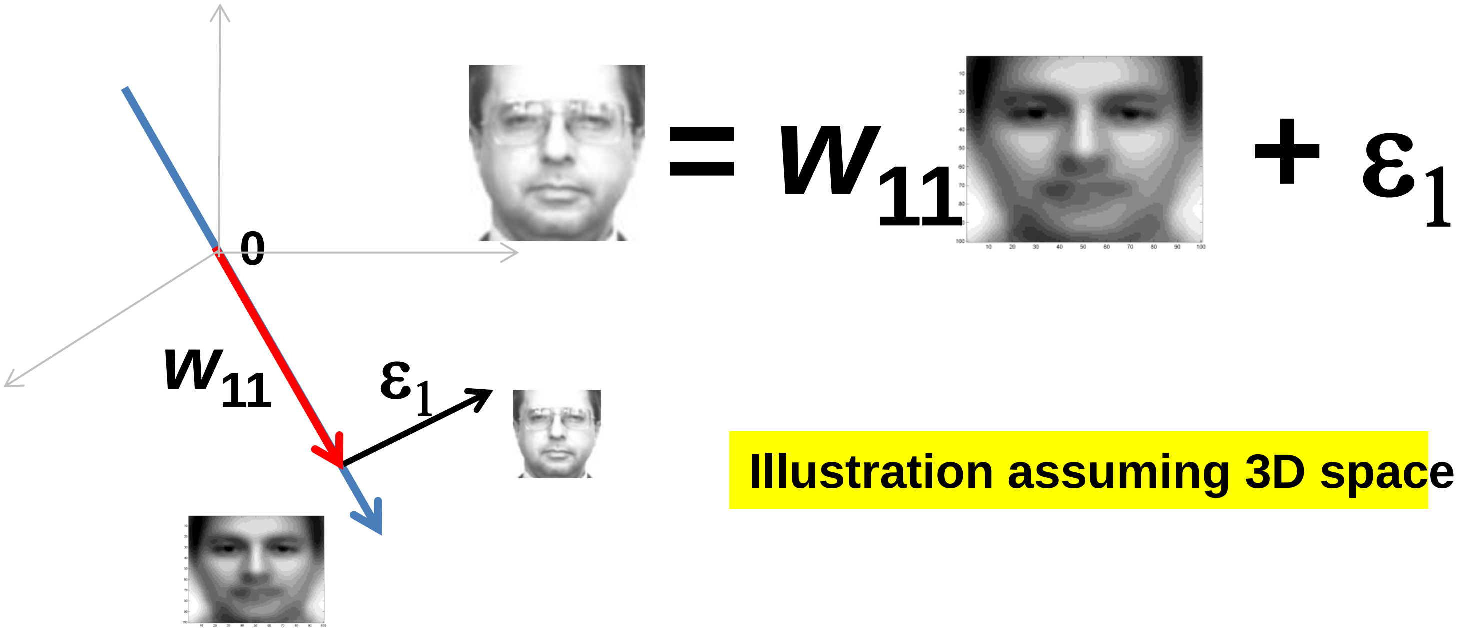


Illustration assuming 3D space

- K-dimensional representation
  - Error is orthogonal to representation
  - Weight and error are specific to data instance

Representation

Error is at 90º to the eigenface

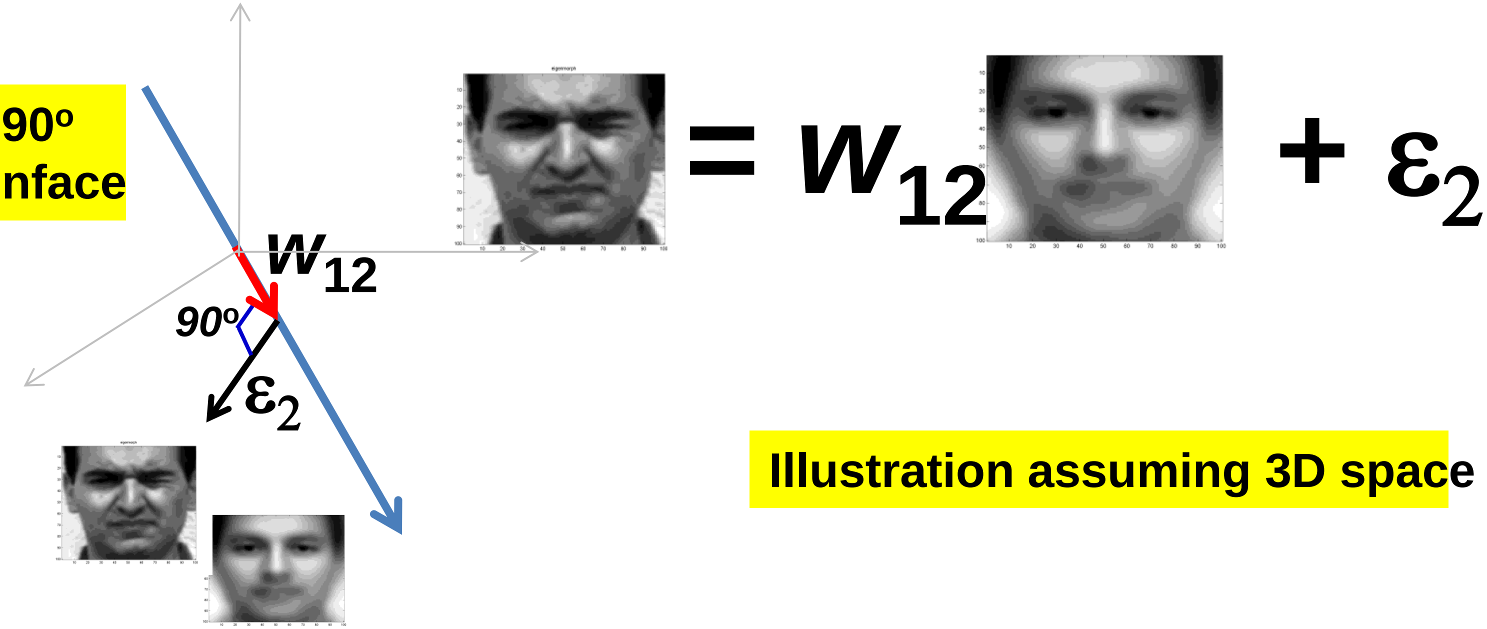
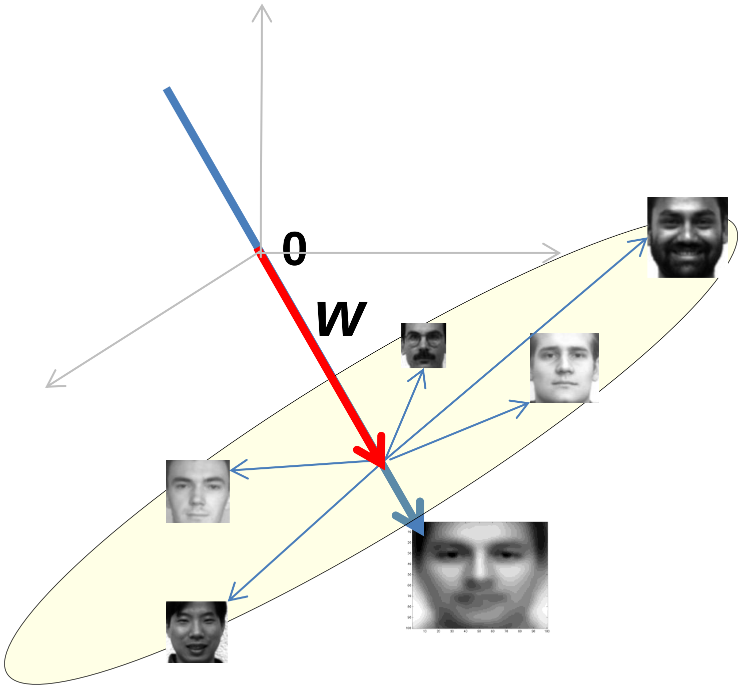


Illustration assuming 3D space

- K-dimensional representation
  - Error is orthogonal to representation
  - Weight and error are specific to data instance

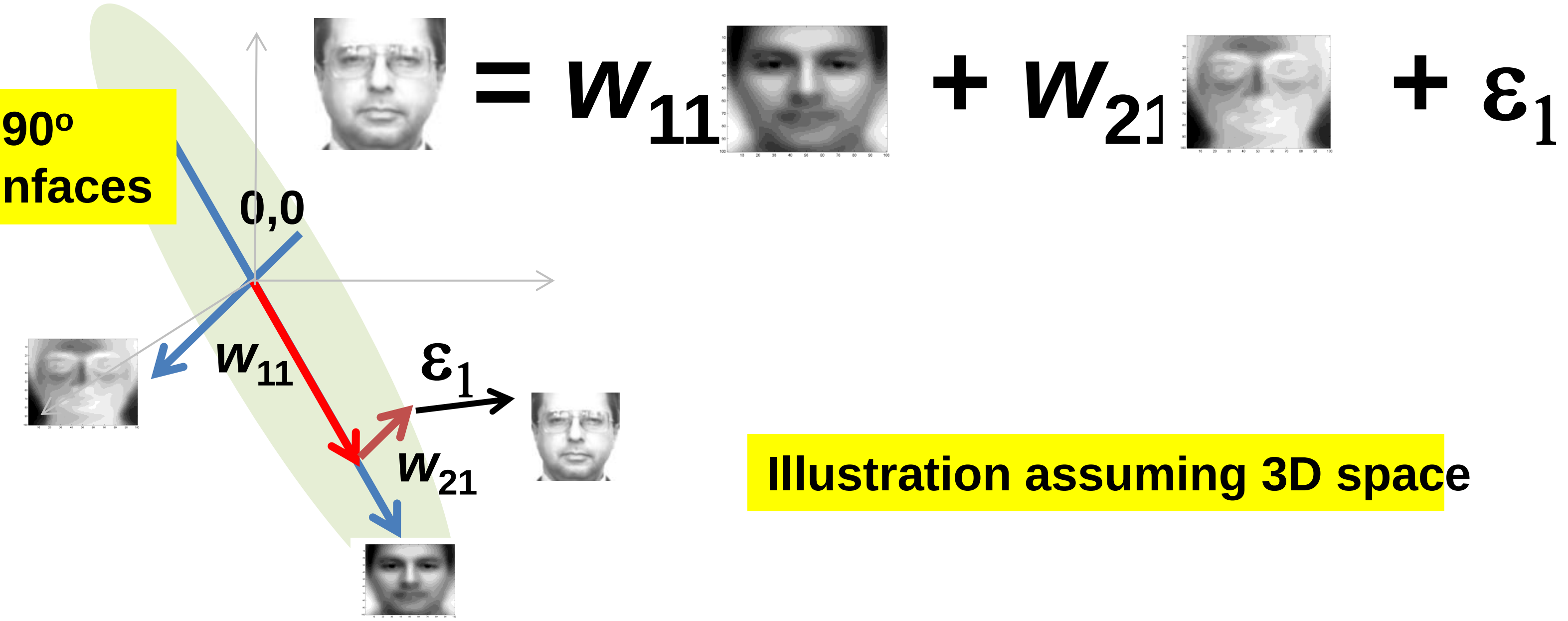
Representation



All data with the same representation wV_1 lie a plane orthogonal to wV_1

- K-dimensional representation
  - Error is orthogonal to representation

With 2 bases

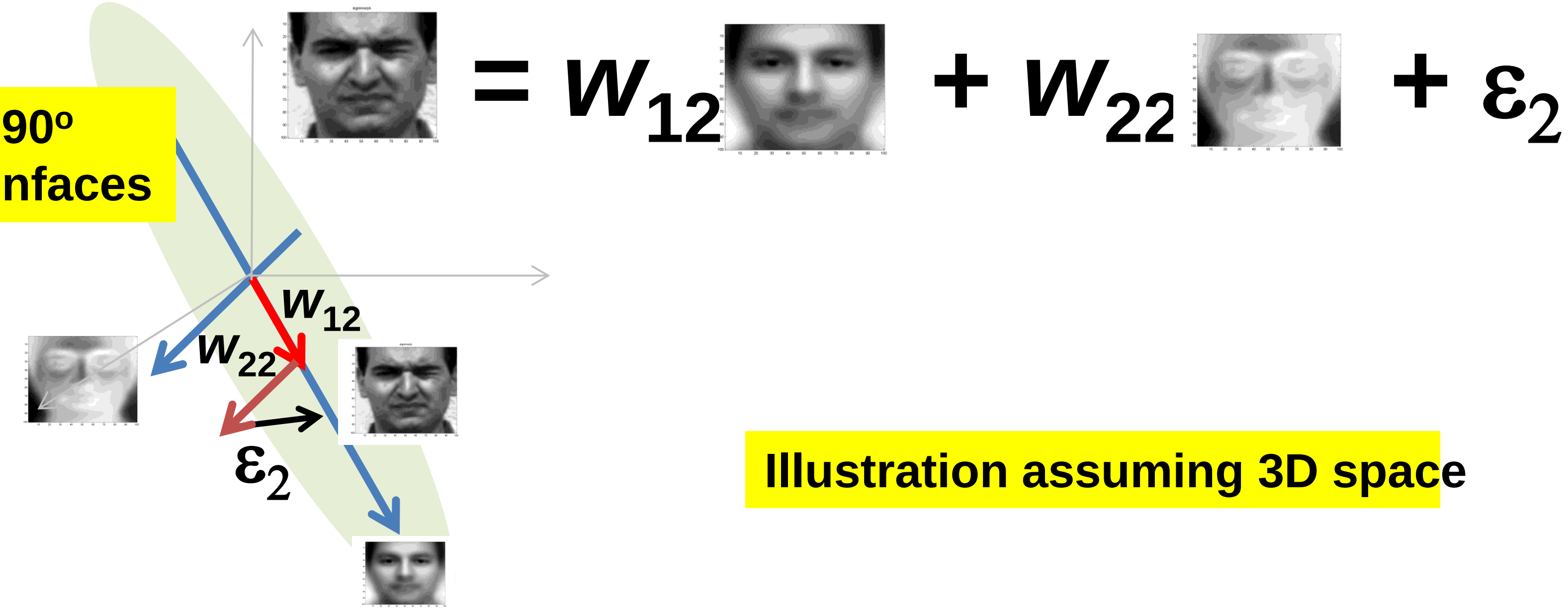


Error is at 90º to the eigenfaces

Illustration assuming 3D space

- K-dimensional representation
  - Error is orthogonal to representation
  - Weight and error are specific to data instance

With 2 bases

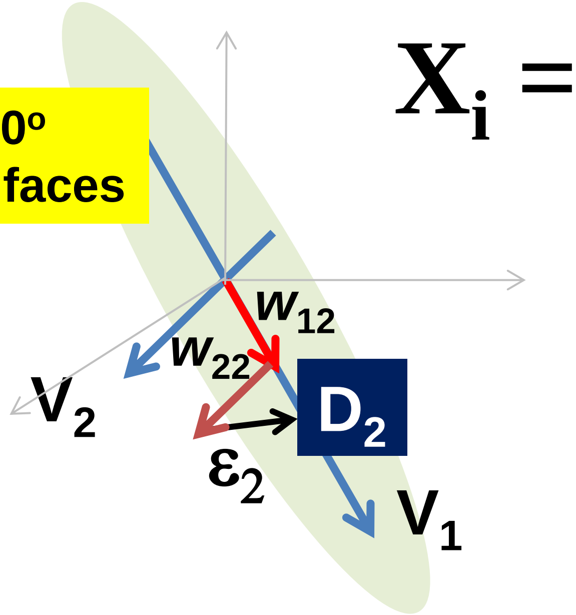


Error is at 90º to the eigenfaces

Illustration assuming 3D space

- K-dimensional representation
  - Error is orthogonal to representation
  - Weight and error are specific to data instance

In Vector Form



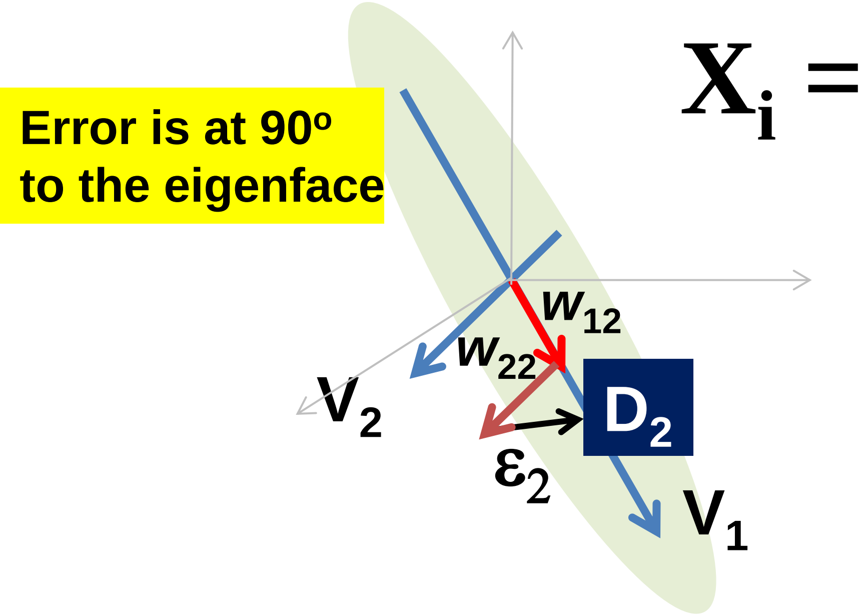
Error is at 90º to the eigenfaces

$$X_i = w_{1i}V_1 + w_{2i}V_2 + \varepsilon_i$$

$$X_i = \begin{bmatrix} V_1 & V_2 \end{bmatrix} \begin{bmatrix} w_{1i} \\ w_{2i} \end{bmatrix} + \varepsilon_i$$

- K-dimensional representation
  - Error is orthogonal to representation
  - Weight and error are specific to data instance

In Vector Form

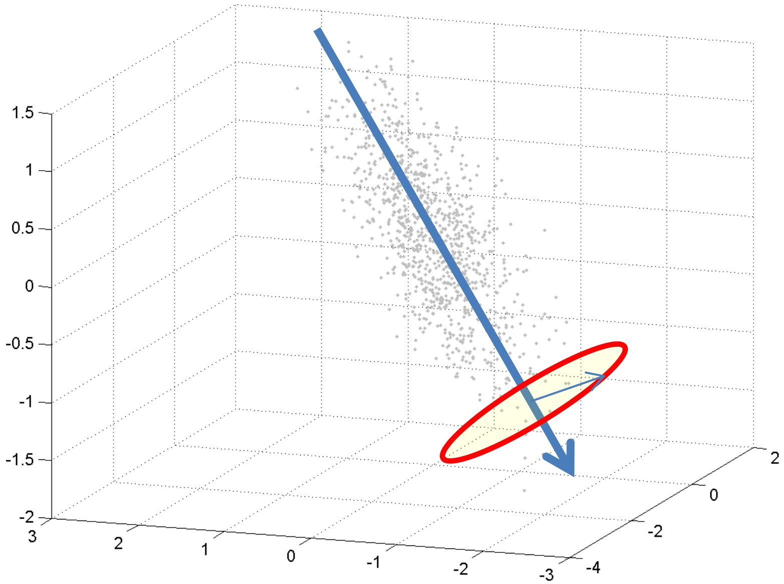


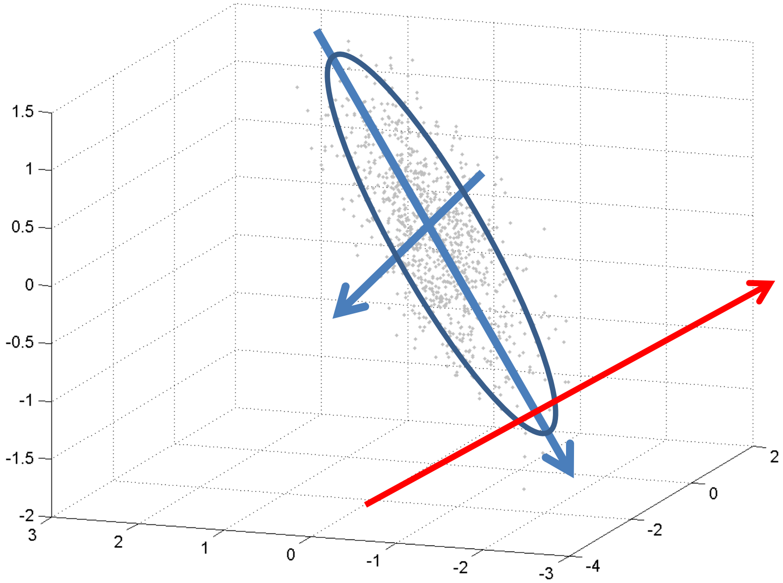
$$X_i = w_{1i}V_1 + w_{2i}V_2 + \varepsilon_i$$

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

- *K*-dimensional representation
- $\mathbf{x}$ is a *D* dimensional vector
- $\mathbf{V}$ is a *D* x *K* matrix
- $\mathbf{w}$ is a *K* dimensional vector
- $\mathbf{e}$ is a *D* dimensional vector

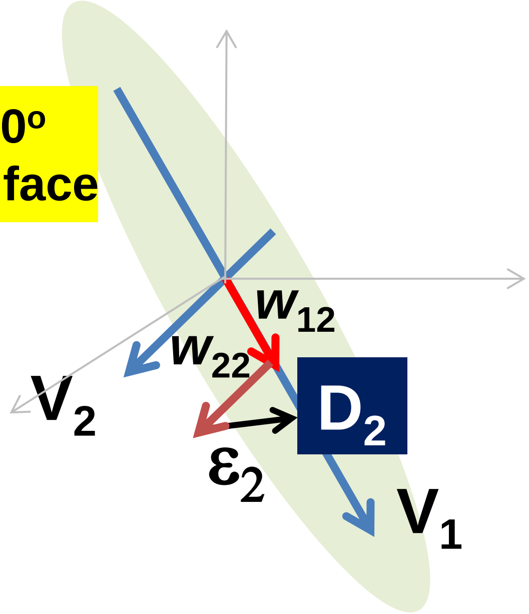
Learning PCA





- 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

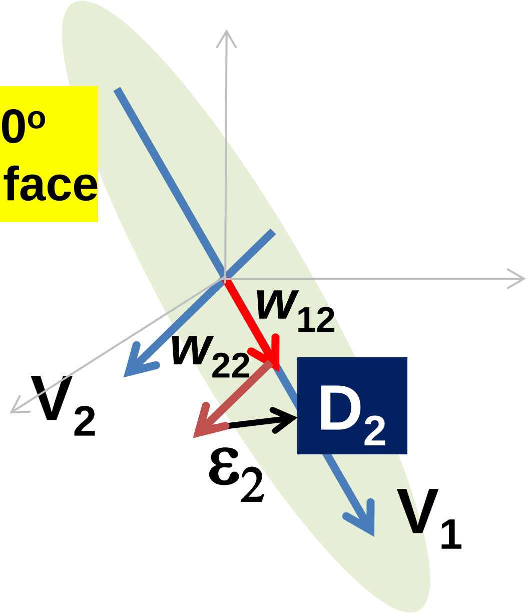
Constraints



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

- $\mathbf{V}^T\mathbf{V} = \mathbf{I}$: Eigen vectors are orthogonal to each other
- For every vector, error is orthogonal to Eigen vectors
  - $\mathbf{e}^T\mathbf{V} = 0$
- Over the *collection* of data
  - Average $\mathbf{w}^T\mathbf{w} = \mathbf{Diagonal}$: Eigen representations are uncorrelated
  - Determinant $\mathbf{e}^T\mathbf{e}$ = minimum: Error variance is minimum
    - Mean of error is 0

A Statistical Formulation of PCA



Error is at 90º to the eigenface

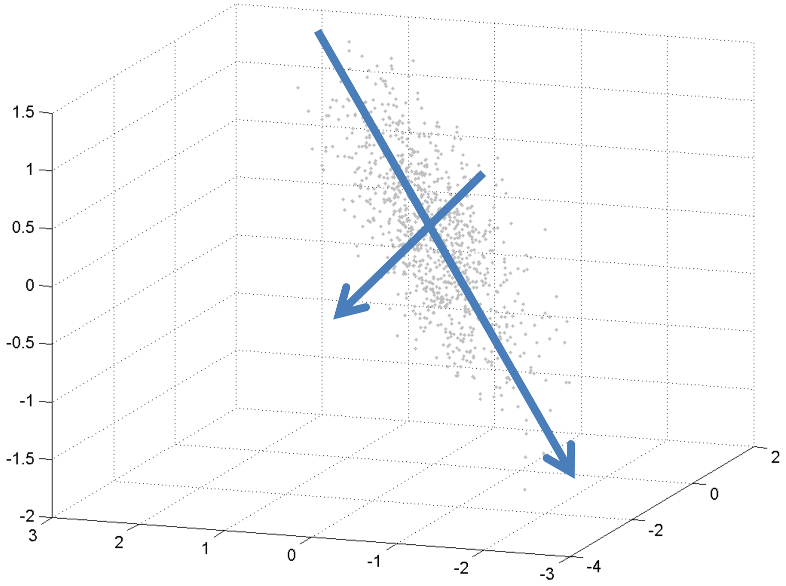
$$\mathbf{x} = \mathbf{V}\mathbf{w} + \mathbf{e}$$

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

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

- $\mathbf{x}$ 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!!



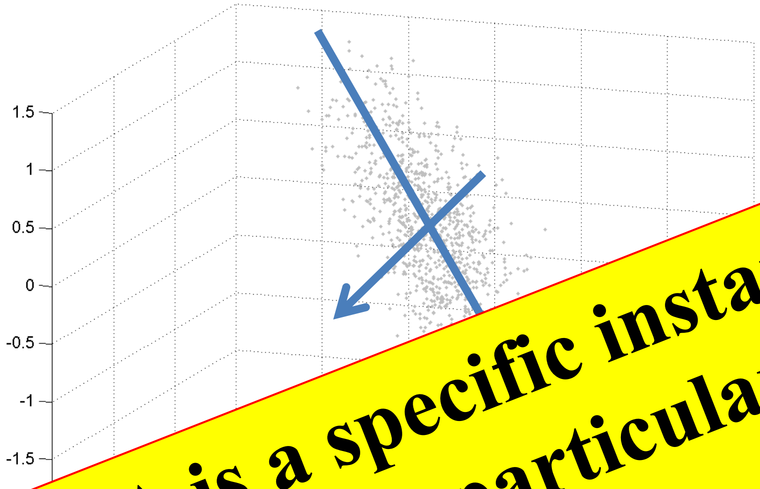
$$\mathbf{x} = \mathbf{V}\mathbf{w} + \mathbf{e}$$

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

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

- $\mathbf{x}$ 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 $\mathbf{E}$

Linear Gaussian Models!!



$\mathbf{X}$

…g to a linear relation

-

- …mean Gaussian

- Es… $\mathbf{V}$ given examples of $\mathbf{x}$
  - In the process also estimate $\mathbf{B}$ and $\mathbf{E}$

PCA is a specific instance of a linear Gaussian model with particular constraints

- ***B = Diagonal***
  - $\mathrm{V}^T\mathrm{V} = \mathrm{I}$
- ***E is low rank***

Linear Gaussian Models

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

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

- Observations are linear functions of two *uncorrelated* Gaussian random variables
  - A “weight” variable $\mathbf{w}$
  - An “error” variable $\mathbf{e}$
  - Error not correlated to weight: $\mathrm{E}[\mathbf{e}^{\mathrm{T}}\mathbf{w}] = 0$
- Learning LGMs: Estimate parameters of the model given instances of $\mathbf{x}$
  - The problem of learning the distribution of a Gaussian RV

LGMs: Probability Density

$$\mathbf{x} = \boldsymbol{\mu} + \mathbf{V}\mathbf{w} + \mathbf{e} \qquad \begin{aligned} \mathbf{w} &\sim N(0, B) \\ \mathbf{e} &\sim N(0, E) \end{aligned}$$

- The mean of $\mathbf{x}$:

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

- **The Covariance of $\mathbf{x}$:**

$$E[(\mathbf{x} - E[\mathbf{x}])(\mathbf{x} - E[\mathbf{x}])^T] = \mathbf{V}B\mathbf{V}^T + E$$

The probability of $\mathbf{x}$

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

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

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

$$\boxed{P(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^D \, |\mathbf{V}B\mathbf{V}^T + E|}} \exp\left(-0.5(\mathbf{x}-\boldsymbol{\mu})^T \left(\mathbf{V}B\mathbf{V}^T + E\right)^{-1} (\mathbf{x}-\boldsymbol{\mu})\right)}$$

- $\mathbf{x}$ is a linear function of Gaussians: $\mathbf{x}$ is also Gaussian
- Its mean and variance are as given

Estimating the variables of the model

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

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

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

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

Estimating the model

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

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

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

- The model is indeterminate:
  - $\mathbf{Vw} = \mathbf{VCC}^{-1}\mathbf{w} = (\mathbf{VC})(\mathbf{C}^{-1}\mathbf{w})$
  - We need extra constraints to make the solution unique
- Usual constraint : $B = \mathbf{I}$
  - Variance of $\mathbf{w}$ is an identity matrix

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

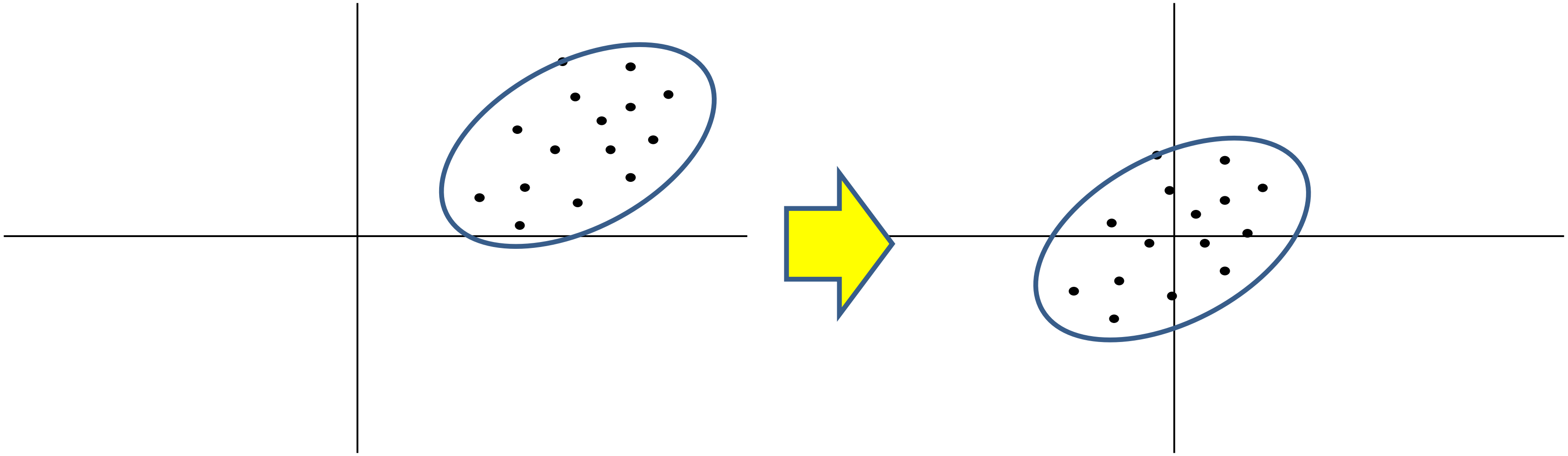
$$\boxed{\mathbf{x} \sim N(\boldsymbol{\mu}, \mathbf{V}\mathbf{V}^T + E)}$$

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

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

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

Centered Data



- We can safely assume "centered" data
  - $\mu = 0$
- If the data are not centered, "center" it
  - Estimate mean of data
    - Which is the maximum likelihood estimate
  - Subtract it from the data

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

Estimating the model

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

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

- Given a collection of xi terms
  - $\mathbf{x}_1, \mathbf{x}_2,..\mathbf{x}_N$
- Estimate $\mathbf{V}$ and E
- $\mathbf{w}$ is unknown for each $\mathbf{x}$
- But if assume we know $\mathbf{w}$ for each $\mathbf{x}$, then what do we get:

Estimating the Parameters

$\mathbf{x}_i = \mathbf{V}\mathbf{w}_i + \mathbf{e}$ $\quad P(\mathbf{e}) = N(0, E)$ $\quad P(\mathbf{x} \mid \mathbf{w}) = N(\mathbf{V}\mathbf{w}, E)$

$$P(\mathbf{x} \mid \mathbf{w}) = \frac{1}{\sqrt{(2\pi)^D \mid E \mid}} \exp\left(-0.5(\mathbf{x} - \mathbf{V}\mathbf{w})^T E^{-1}(\mathbf{x} - \mathbf{V}\mathbf{w})\right)$$

- We will use a *maximum-likelihood estimate*
- The log-likelihood of $\mathbf{x}_1..\mathbf{x}_N$ *knowing* their $\mathbf{w}_i$s

$$\log P(\mathbf{x}_1..\mathbf{x}_N \mid \mathbf{w}_1..\mathbf{w}_N) =$$

$$-0.5N \log \mid E^{-1} \mid -0.5\sum_i (\mathbf{x}_i - \mathbf{V}\mathbf{w}_i)^T E^{-1}(\mathbf{x}_i - \mathbf{V}\mathbf{w}_i)$$

Maximizing the log-likelihood

$$\boxed{LL = -0.5N\log|E^{-1}| - 0.5\sum_i (\mathbf{x}_i - \mathbf{V}\mathbf{w}_i)^T E^{-1}(\mathbf{x}_i - \mathbf{V}\mathbf{w}_i)}$$

- Differentiating w.r.t. $\mathbf{V}$ and setting to 0

$$\boxed{2\sum_i E^{-1}(\mathbf{x}_i - \mathbf{V}\mathbf{w}_i)\mathbf{w}_i^T = 0}$$

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

- **Differentiating w.r.t. E^{-1} and setting to 0**

$$\boxed{E = \frac{1}{N}\left(\sum_i \mathbf{x}_i\mathbf{x}_i^T - \mathbf{V}\sum_i \mathbf{w}_i\mathbf{x}_i^T\right)}$$

Estimating LGMs: If we know *w*

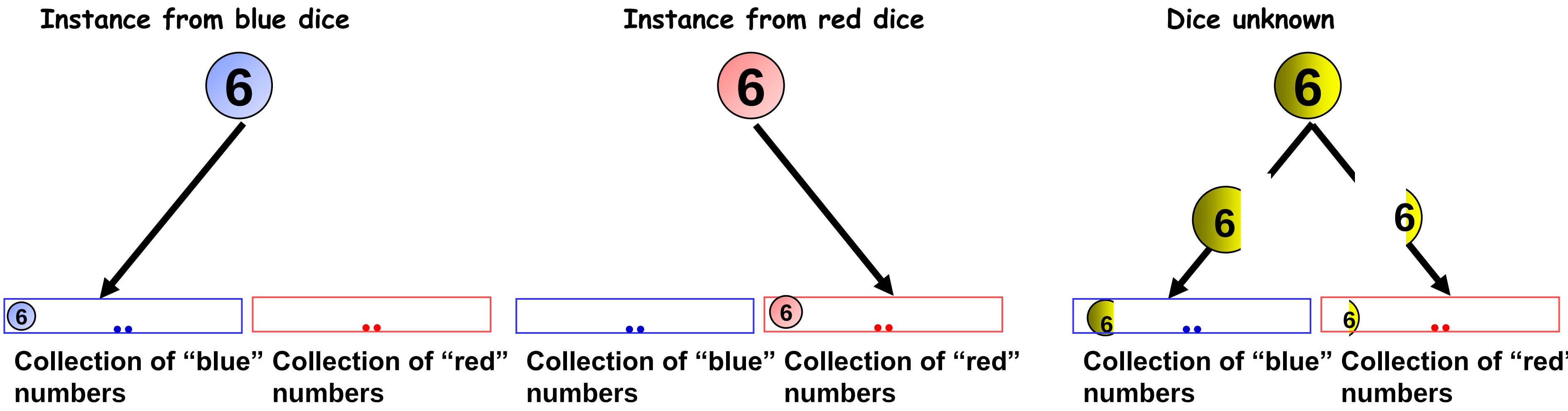
$$\mathbf{x}_i = \mathbf{V}\mathbf{w}_i + \mathbf{e} \qquad P(\mathbf{e}) = N(0, E)$$

$$\boxed{\mathbf{V} = \left(\sum_i \mathbf{x}_i \mathbf{w}_i^T\right)\left(\sum_i \mathbf{w}_i \mathbf{w}_i^T\right)^{-1}} \qquad \boxed{E = \frac{1}{N}\left(\sum_i \mathbf{x}_i \mathbf{x}_i^T - \mathbf{V}\sum_i \mathbf{w}_i \mathbf{x}_i^T\right)}$$

- But in reality we *don't* know the **w** for each **x**
  - So how to deal with this?

- EM..

Recall EM



Instance from blue dice
6
6
Collection of "blue" numbers
Collection of "red" numbers
Instance from red dice
6
6
Collection of "blue" numbers
Collection of "red" numbers
Dice unknown
6
6
6
6
6
Collection of "blue" numbers
Collection of "red" numbers

- We figured out how to compute parameters if we *knew* the missing information
- Then we "fragmented" the observations according to the posterior probability P(z|x) and counted as usual
- In effect we took the expectation with respect to the a posteriori probability of the missing data: P(z|x)

EM for LGMs

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

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

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

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

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

- Replace unseen data terms with expectations taken w.r.t. $\mathrm{P}(\mathbf{w}|\mathbf{x}_i)$

EM for LGMs

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

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

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

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

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

- Replace unseen data terms with expectations taken w.r.t. $P(\mathbf{w}|\mathbf{x}_i)$

Expected Value of w given x

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

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

- $\mathbf{x}$ and $\mathbf{w}$ are jointly Gaussian!
  - $\mathbf{x}$ is Gaussian
  - $\mathbf{w}$ is Gaussian
  - They are linearly related

$$\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \mathbf{w} \end{bmatrix} \qquad P(\mathbf{z}) = N(\mu_{\mathbf{z}}, C_{\mathbf{zz}})$$

Expected Value of w given x

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

$$\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \mathbf{w} \end{bmatrix}$$

$$P(\mathbf{z}) = N(\mu_{\mathbf{z}}, C_{\mathbf{zz}})$$

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

$$\mu_{\mathbf{z}} = \begin{bmatrix} \mu_{\mathbf{x}} \\ \mu_{\mathbf{w}} \end{bmatrix} = 0$$

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

$$C_{\mathbf{zz}} = \begin{bmatrix} C_{\mathbf{xx}} & C_{\mathbf{xw}} \\ C_{\mathbf{wx}} & C_{\mathbf{ww}} \end{bmatrix}$$

$$C_{\mathbf{xw}} = E[(\mathbf{x} - \mu_{\mathbf{x}})(\mathbf{w} - \mu_{\mathbf{w}})^T] = \mathbf{V}$$

$$C_{\mathbf{zz}} = \begin{bmatrix} \mathbf{V}\mathbf{V}^T + E & \mathbf{V} \\ \mathbf{V}^T & I \end{bmatrix}$$

- **x** and **w** are jointly Gaussian!

The conditional expectation of w given z

- P(w|z) is a Gaussian

$$\mu_{\mathbf{z}} = \begin{bmatrix} \mu_{\mathbf{x}} \\ \mu_{\mathbf{w}} \end{bmatrix} = 0$$

$$P(\mathbf{w}\,|\,\mathbf{x}) = N(\mu_{\mathbf{w}} + C_{\mathbf{wx}}C_{\mathbf{xx}}^{-1}(x - \mu_{\mathbf{x}}), C_{\mathbf{ww}} - C_{\mathbf{wx}}^{T}C_{\mathbf{xx}}^{-1}C_{\mathbf{xw}})$$

$$C_{\mathbf{zz}} = \begin{bmatrix} C_{\mathbf{xx}} & C_{\mathbf{xw}} \\ C_{\mathbf{wx}} & C_{\mathbf{ww}} \end{bmatrix}$$

$$C_{\mathbf{zz}} = \begin{bmatrix} \mathbf{V}\mathbf{V}^{T} + E & \mathbf{V} \\ \mathbf{V}^{T} & I \end{bmatrix}$$

$$P(\mathbf{w}\,|\,\mathbf{x}) = N(\mathbf{V}^{T}(\mathbf{V}\mathbf{V}^{T} + E)^{-1}\mathbf{x}, I - \mathbf{V}^{T}(\mathbf{V}\mathbf{V}^{T} + E)^{-1}\mathbf{V})$$

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

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

LGM: The complete EM algorithm

- 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:

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

-

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

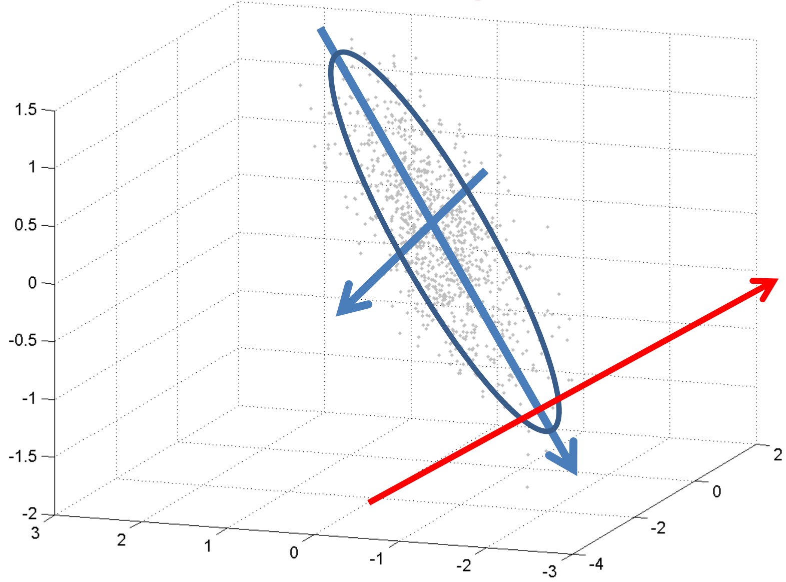
So what have we achieved

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

- Next class:
  - PCA
    - Sensible PCA
    - EM algorithms for PCA
  - Factor Analysis
    - FA for feature extraction

LGMs : Application 1

Learning principal components



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

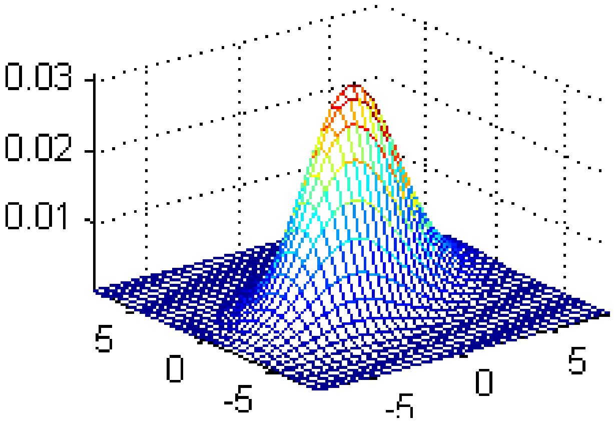
$$\mathbf{w} \sim N(0, I)$$

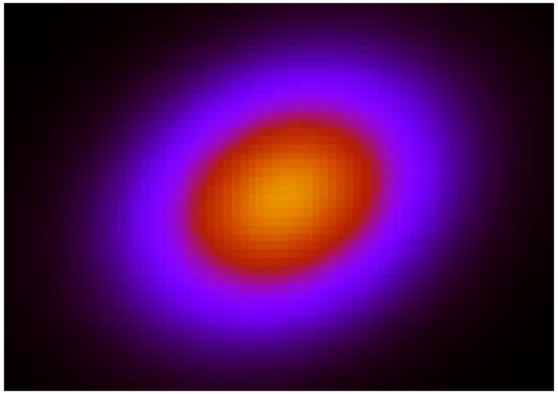
$$\mathbf{e} \sim N(0, E)$$

- Find directions that capture most of the variation in the data
- Error is orthogonal to these variations

LGMs : Application 2

Learning with insufficient data





FULL COV FIGURE

- 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**

To be continued..

- Other applications..
- Next class