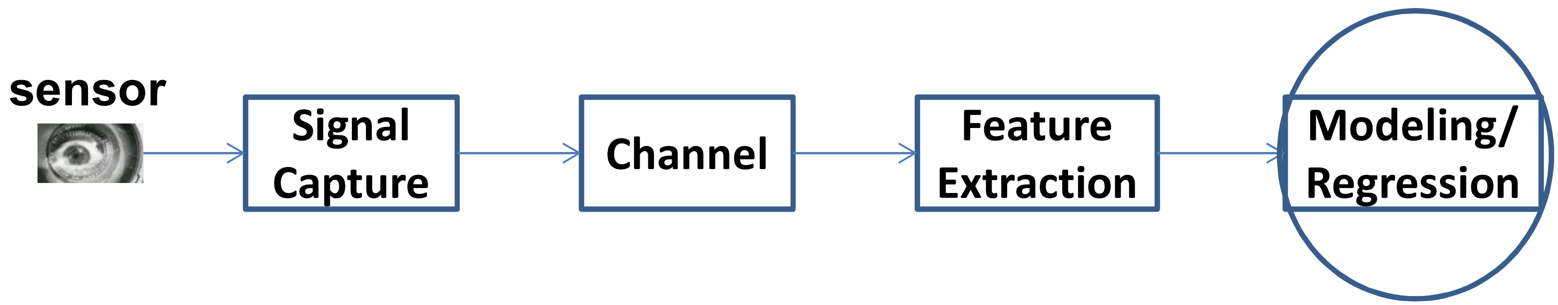
Machine Learning for Signal Processing

Linear Gaussian Models

MLSP

- Application of Machine Learning techniques to the analysis of signals



- Modeling
  - *Representation*
  - *Classification*

Linear Gaussian Models

- MAP and MMSE prediction with Gaussian models
  - Estimation
  - Regularization
- Representation
  - PCA
  - Probabilistic PCA
- Gaussian Classifier

Recap: MAP Estimators

- MAP (*Maximum A Posteriori*): Find most probable value of $\mathbf{y}$ given $\mathbf{x}$

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

- We have used this for classification earlier. But we can also use it for *regression*
  – Estimating *continuous* valued variables

- Lets do this for a Gaussian RV..

MAP estimation

- x and y are jointly Gaussian

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

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

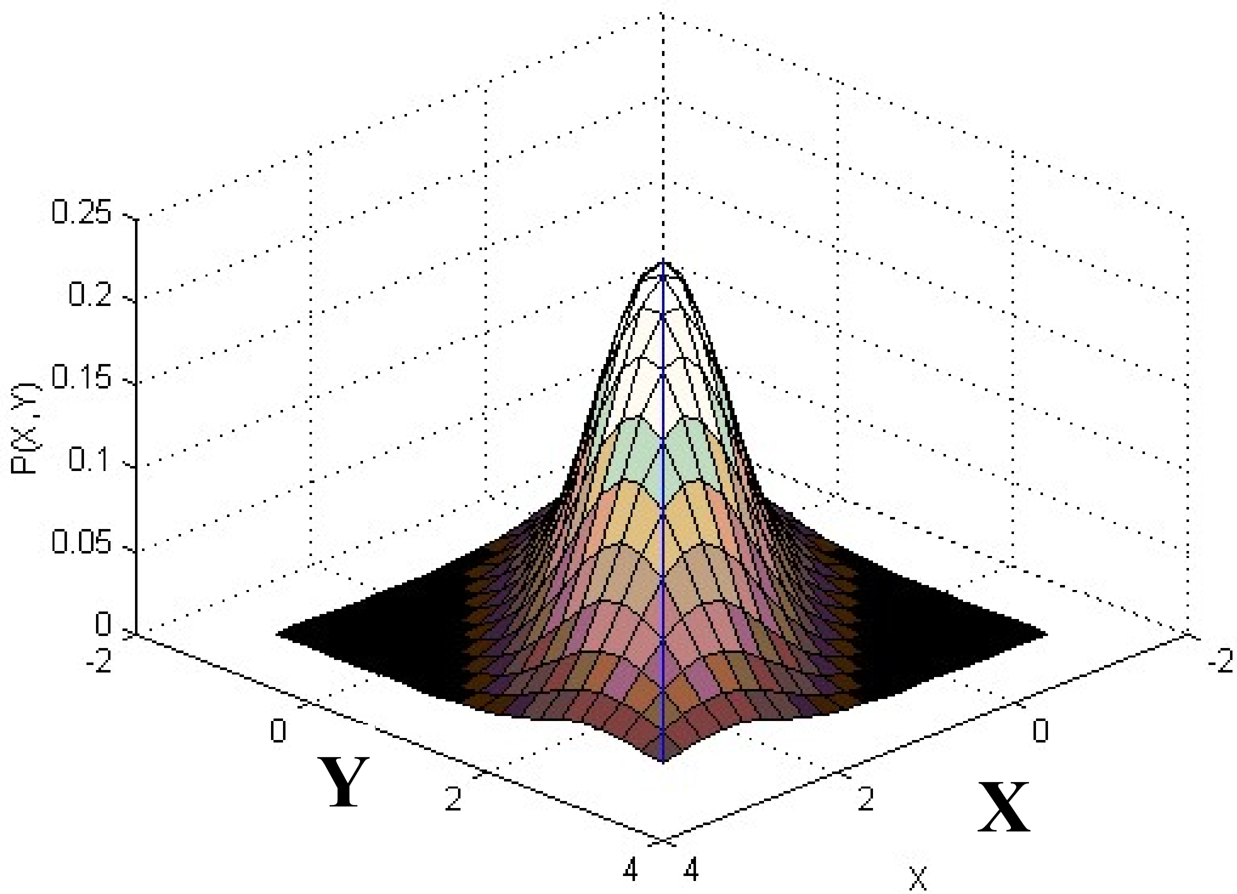
$$Var(z) = C_{zz} = \begin{bmatrix} C_{xx} & C_{xy} \\ C_{yx} & C_{yy} \end{bmatrix}$$

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

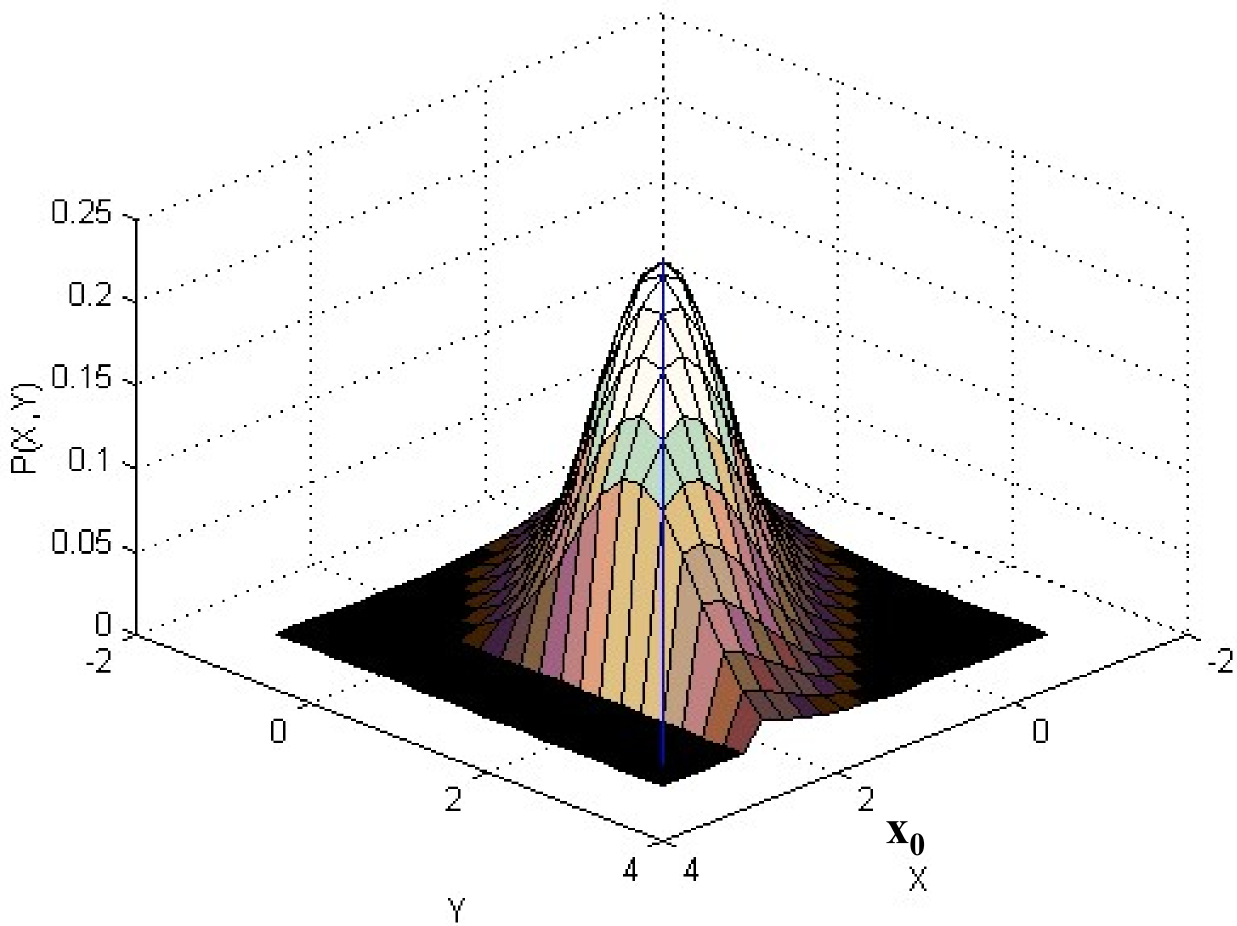
$$P(z) = N(\mu_z, C_{zz}) = \frac{1}{\sqrt{2\pi\,|C_{zz}|}} \exp\left(-0.5(z - \mu_z)^T C_{zz}^{-1}(z - \mu_z)\right)$$

- **z is Gaussian**

MAP estimation: Gaussian PDF



MAP estimation: The Gaussian at a particular value of X

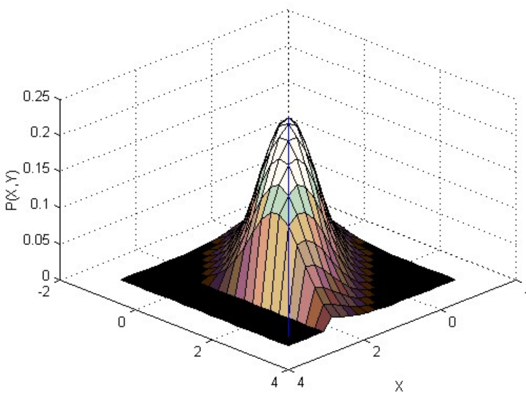


Conditional Probability of y|x

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

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

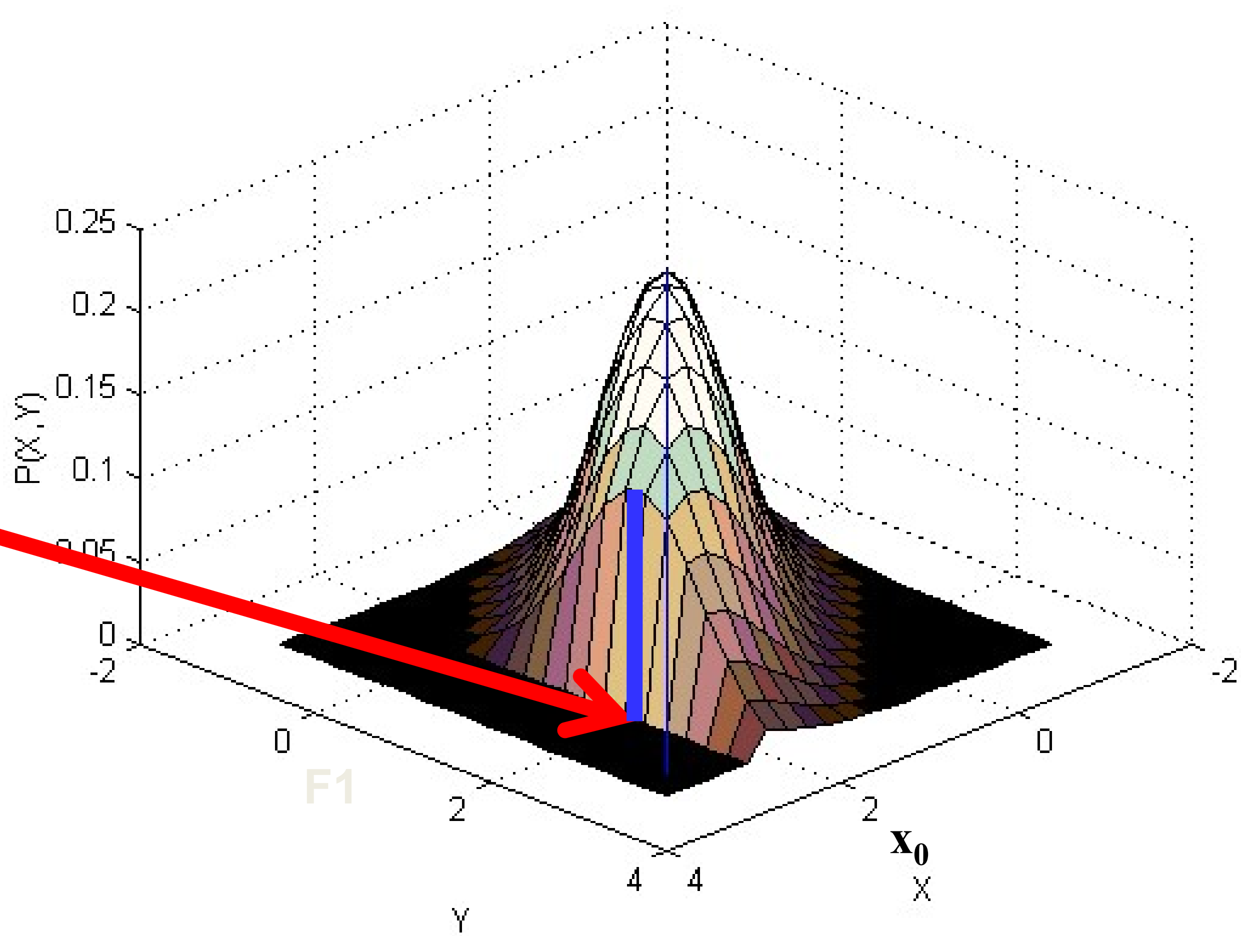
$$Var(y\,|\,x) = C_{yy} - C_{yx}C_{xx}^{-1}C_{xy}$$



- 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

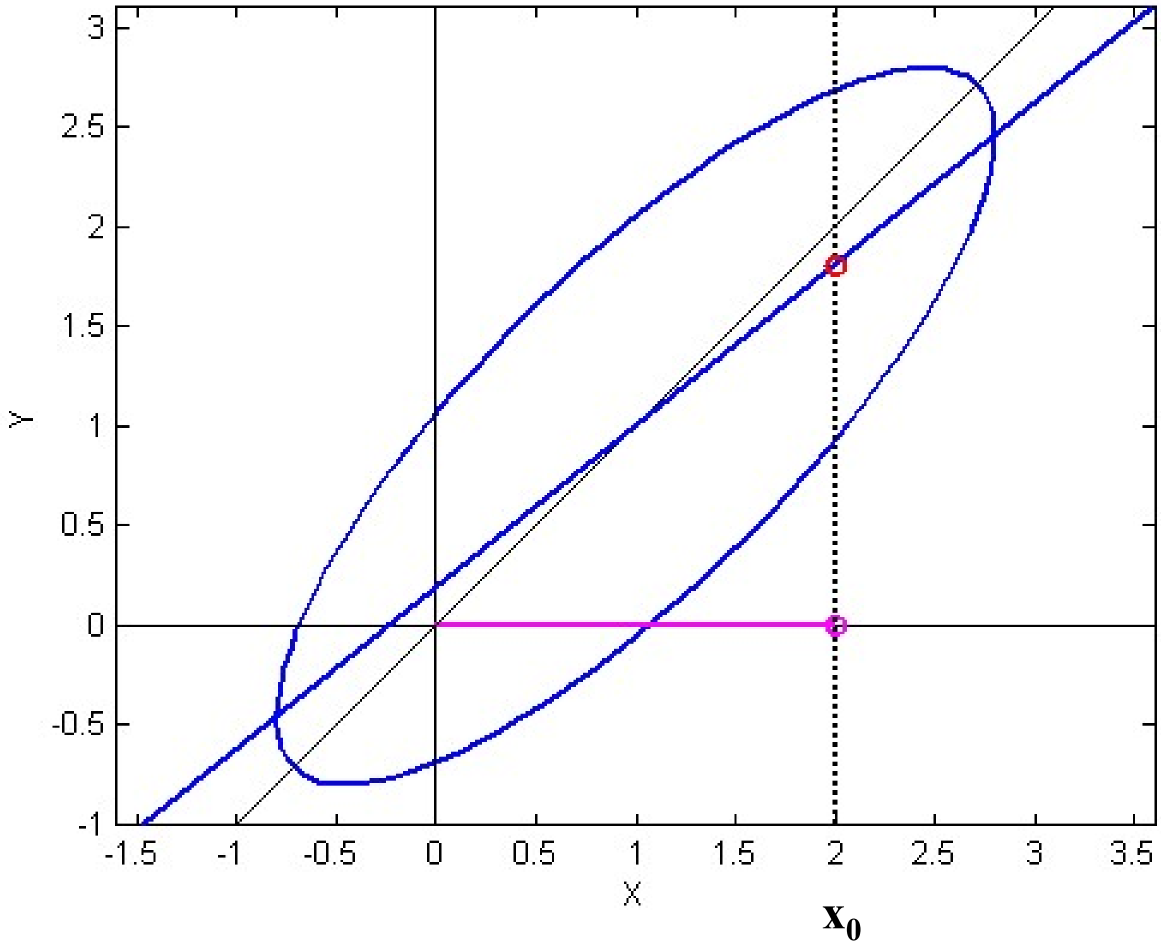
MAP estimation: The Gaussian at a particular value of X

Most likely value



MAP Estimation of a Gaussian RV

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

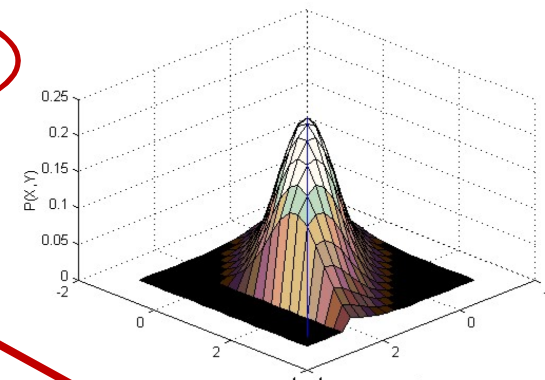


Conditional Probability of y|x

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

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

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



MAP Estimate.

Its actually a regression line

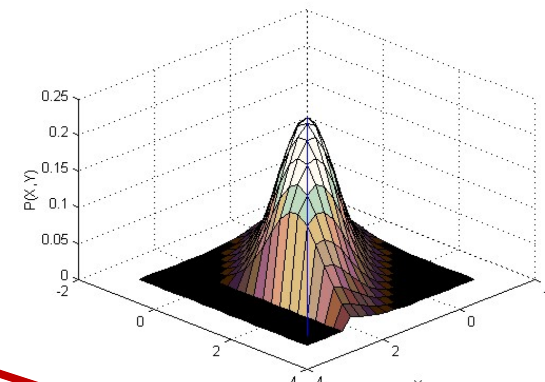
- The conditional pro ... ian
  - The slice in the fig ...
- The mean of this Gaussian is a function of x
- The variance of y reduces if x is known
  - Uncertainty is reduced

Conditional Probability of y|x

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

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

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



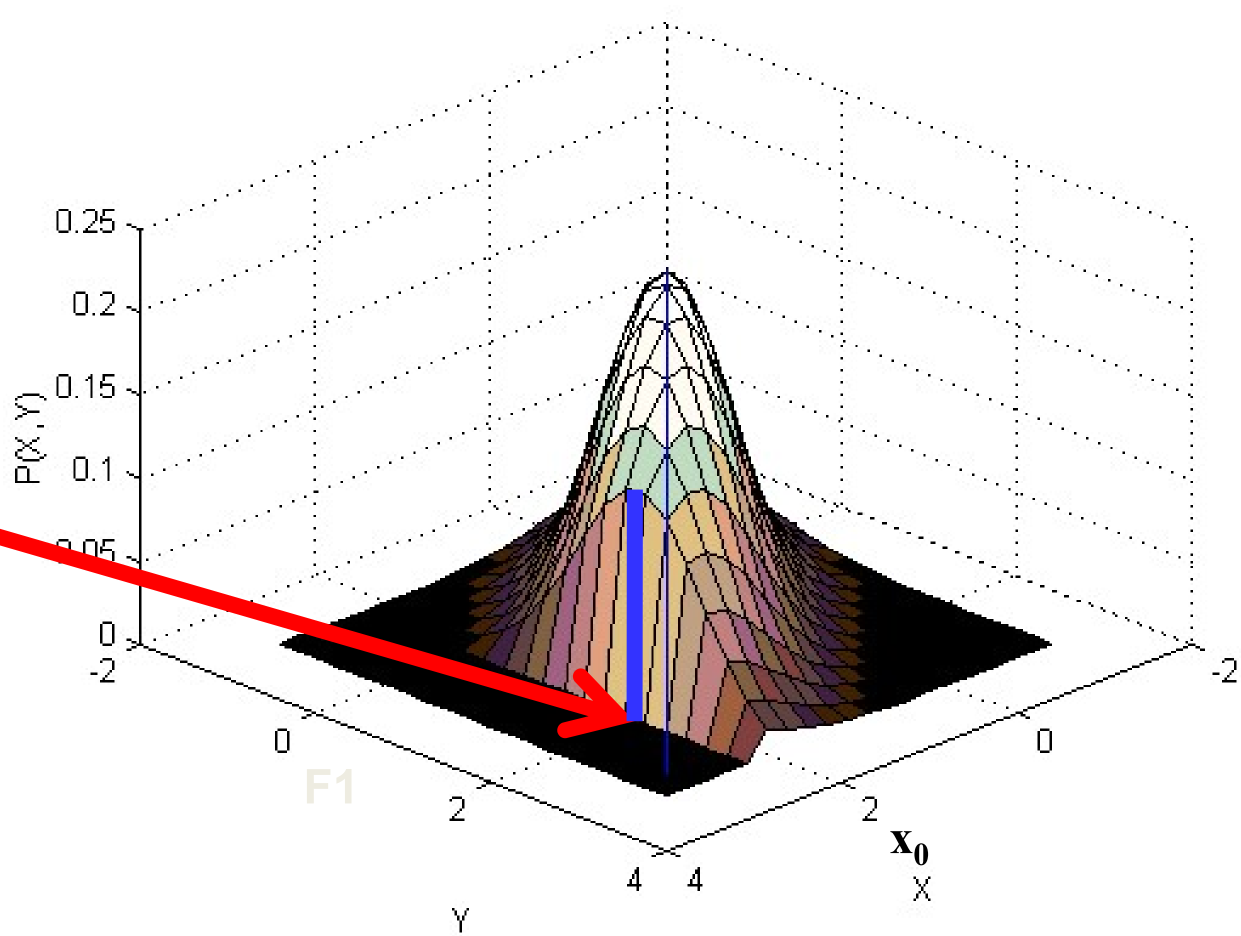
- The conditional pro
  - The slice in the fig
- The mean of this Ga
- The variance of y re
  - Uncertainty is redu

The variance of Y shrinks because we know X

Note that the actual value of X doesn't matter. Simply knowing X reduces the variance of Y if the two are correlated

MMSE estimation

Mean value



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:

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

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

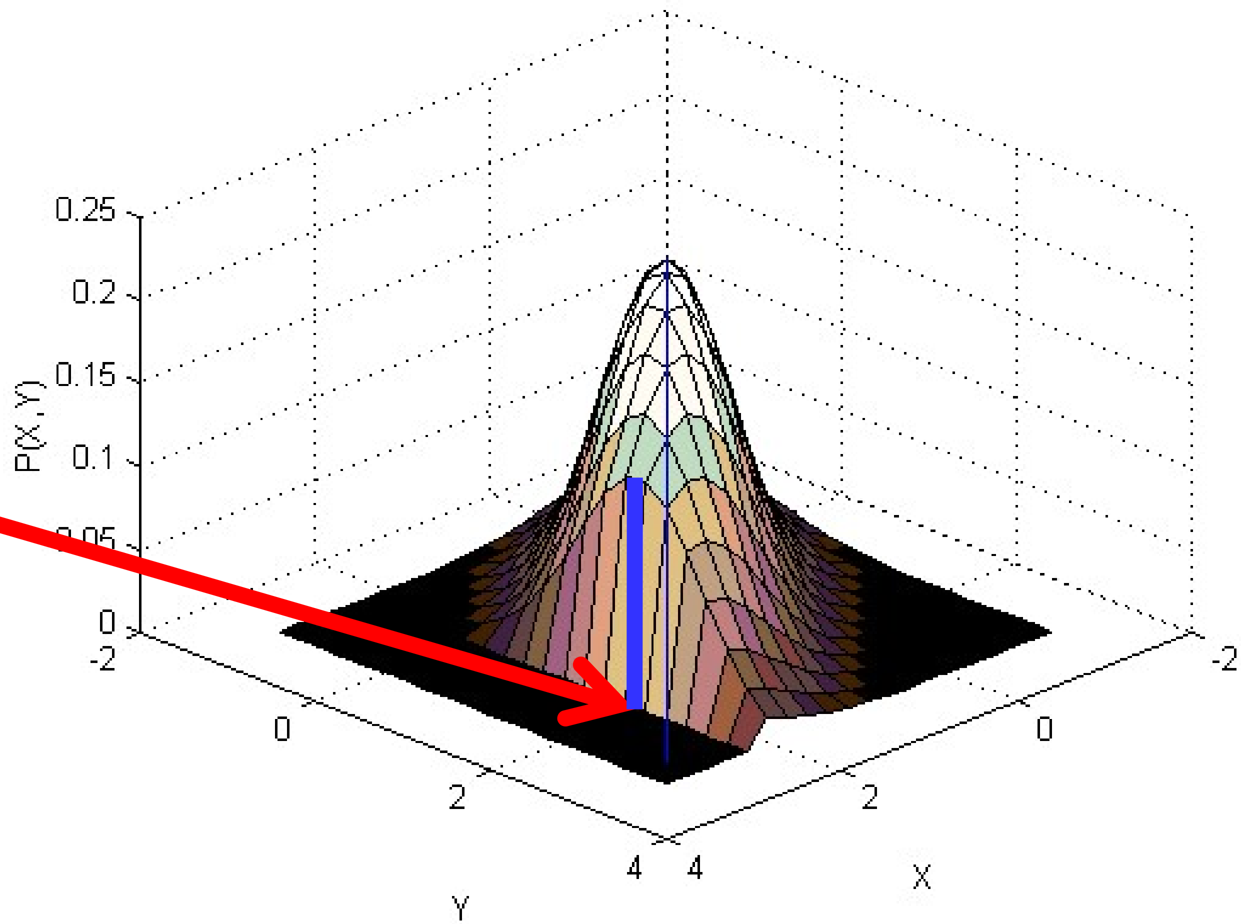
The MMSE estimate is the mean of the distribution

For the Gaussian: MAP = MMSE

Most likely value

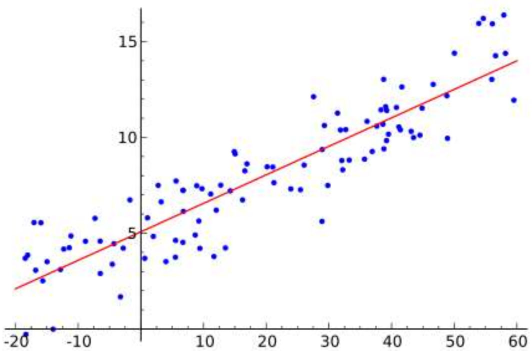
is also

The MEAN value

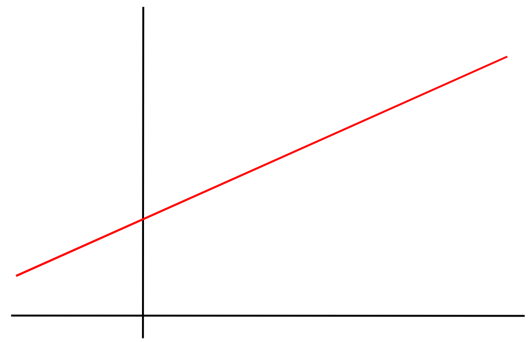


- Would be true of any symmetric distribution

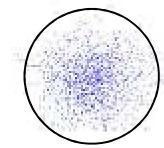
Linear Regression: A Likelihood Perspective



=



+



- $\mathbf{y}$ is a noisy reading of $\mathbf{a}^T\mathbf{x}$

$$\mathbf{y} = \mathbf{a}^T\mathbf{x} + \mathbf{e}$$

- Error $\mathbf{e}$ is Gaussian

$$\mathbf{e} \sim N(0, \sigma^2\mathbf{I})$$

- Estimate $\mathbf{A}$ from $\mathbf{Y} = [\mathbf{y}_1 \ \ \mathbf{y}_2 \ldots \mathbf{y}_N] \quad \mathbf{X} = [\mathbf{x}_1 \ \ \mathbf{x}_2 \ldots \mathbf{x}_N]$

The *Likelihood* of the data

$$\mathbf{y} = \mathbf{a}^T\mathbf{x} + \mathbf{e} \qquad\qquad \mathbf{e} \sim N(0, \sigma^2\mathbf{I})$$

- Probability of observing a specific $\mathbf{y}$, given $\mathbf{x}$, for a particular matrix $\mathbf{a}$

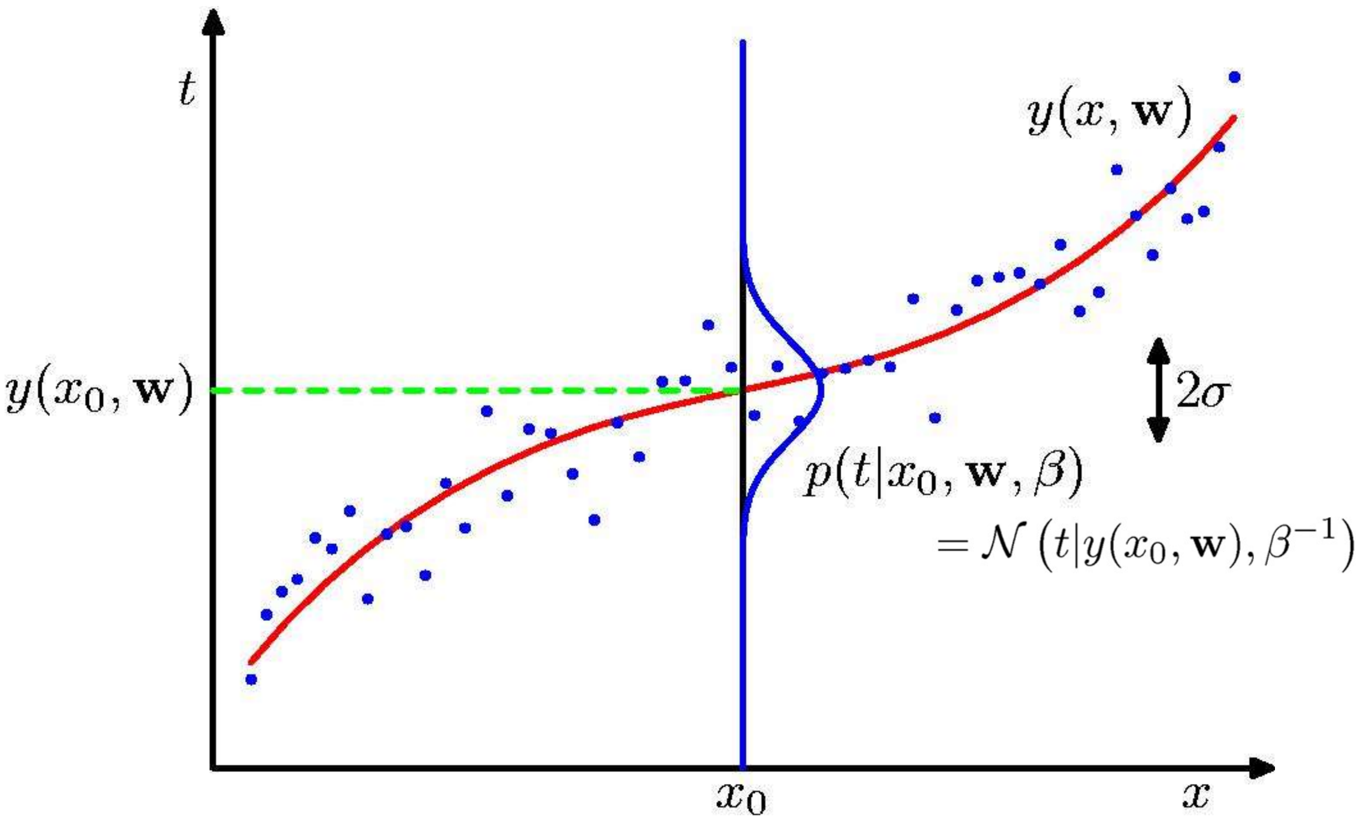
$$P(\mathbf{y} \mid \mathbf{x}; \mathbf{a}) = N(\mathbf{y}; \mathbf{a}^T\mathbf{x}, \sigma^2\mathbf{I})$$

- Probability of collection: $\mathbf{Y} = [\mathbf{y_1}, \mathbf{y_1} \ldots \mathbf{y_1}]$, $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2} \ldots \mathbf{x_N}]$

$$P(\mathbf{Y} \mid \mathbf{X}; \mathbf{a}) = \prod_i N(\mathbf{y}_i; \mathbf{a}^T\mathbf{x}_i, \sigma^2\mathbf{I})$$

- Assuming IID for convenience (not necessary)

Curve Fitting With Noise



A Maximum Likelihood Estimate

$$\mathbf{y} = \mathbf{a}^T\mathbf{x} + \mathbf{e} \qquad \mathbf{e} \sim N(0, \sigma^2\mathbf{I}) \qquad \mathbf{Y} = [\mathbf{y}_1 \;\; \mathbf{y}_2 \ldots \mathbf{y}_N] \quad \mathbf{X} = [\mathbf{x}_1 \;\; \mathbf{x}_2 \ldots \mathbf{x}_N]$$

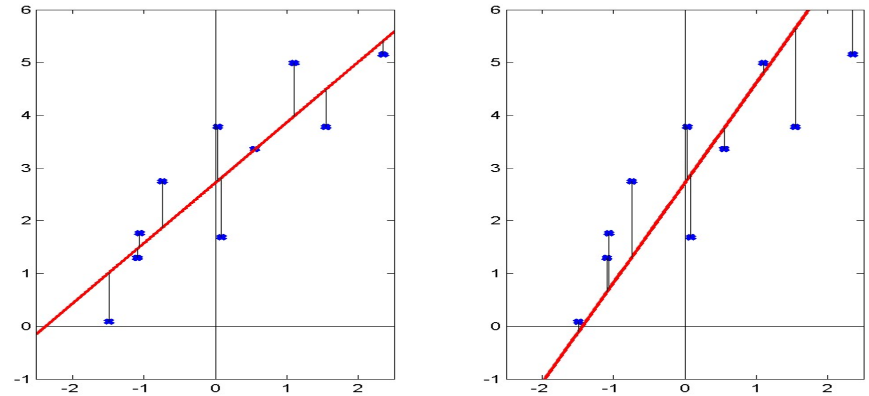
$$P(\mathbf{Y} \mid \mathbf{X}) = \prod_i \frac{1}{\sqrt{(2\pi\sigma^2)^D}} \exp\left(\frac{-1}{2\sigma^2} \left\| \mathbf{y}_i - \mathbf{a}^T\mathbf{x}_i \right\|^2 \right)$$

$$\log P(\mathbf{Y} \mid \mathbf{X}; \mathbf{a}) = C - \sum_i \frac{1}{2\sigma^2} \left\| \mathbf{y}_i - \mathbf{a}^T\mathbf{x}_i \right\|^2$$

$$\log P(\mathbf{Y} \mid \mathbf{X}, \mathbf{a}) = C - \frac{1}{2\sigma^2} trace\left((\mathbf{Y} - \mathbf{a}^T\mathbf{X})^T (\mathbf{Y} - \mathbf{a}^T\mathbf{X}) \right)$$

- Maximizing the log probability is identical to minimizing the squared error
  - Just L_2 based regression

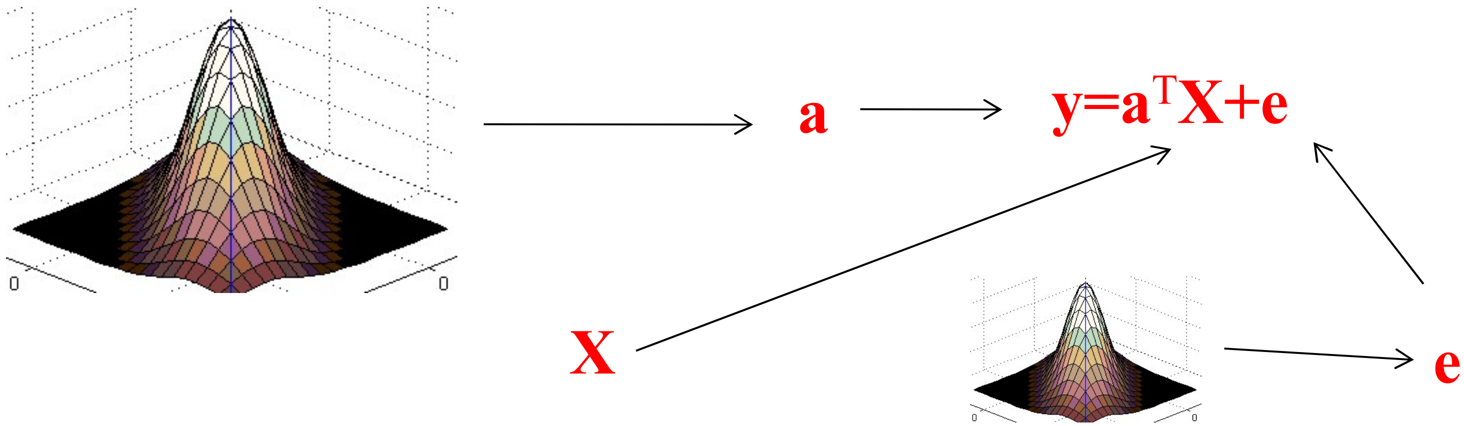
A problem with regressions



$$\mathbf{A} = \left(\mathbf{X}\mathbf{X}^T\right)^{-1}\mathbf{X}\mathbf{Y}^T$$

- ML fit is sensitive
  - Error is squared
  - Small variations in data → large variations in weights
  - Outliers affect it adversely
- Unstable
  - If dimension of $\mathbf{X}$ >= no. of instances
    - ($\mathbf{X}\mathbf{X}^T$) is not invertible

MAP estimation of weights



- Assume weights drawn from a Gaussian
  - $P(\mathbf{a}) = N(0, \sigma^2 I)$
- Max. Likelihood estimate

$$\hat{\mathbf{a}} = \arg\max_{\mathbf{a}} \log P(\mathbf{Y} \mid \mathbf{X}; \mathbf{a})$$

- Maximum *a posteriori* estimate

$$\hat{\mathbf{a}} = \arg\max_{\mathbf{a}} \log P(\mathbf{a} \mid \mathbf{Y}, \mathbf{X}) = \arg\max_{\mathbf{a}} \log P(\mathbf{Y} \mid \mathbf{X}, \mathbf{a}) P(\mathbf{a})$$

MAP estimation of weights

$$\hat{\mathbf{a}} = \arg\max_{\mathbf{A}} \log P(\mathbf{a} \mid \mathbf{Y}, \mathbf{X}) = \arg\max_{\mathbf{A}} \log P(\mathbf{Y} \mid \mathbf{X}, \mathbf{a}) P(\mathbf{a})$$

- P(**a**) = N(0, σ^2I)
- Log P(**a**) = C – log σ – 0.5σ^{-2} $||\mathbf{a}||^2$

$$\log P(\mathbf{Y} \mid \mathbf{X}, \mathbf{a}) = C - \frac{1}{2\sigma^2} trace\left((\mathbf{Y} - \mathbf{a}^T\mathbf{X})^T (\mathbf{Y} - \mathbf{a}^T\mathbf{X})\right)$$

$$\hat{\mathbf{a}} = \arg\max_{\mathbf{A}} C' - \log\sigma - \frac{1}{2\sigma^2} trace\left((\mathbf{Y} - \mathbf{a}^T\mathbf{X})^T (\mathbf{Y} - \mathbf{a}^T\mathbf{X})\right) - 0.5\sigma^2 \mathbf{a}^T\mathbf{a}$$

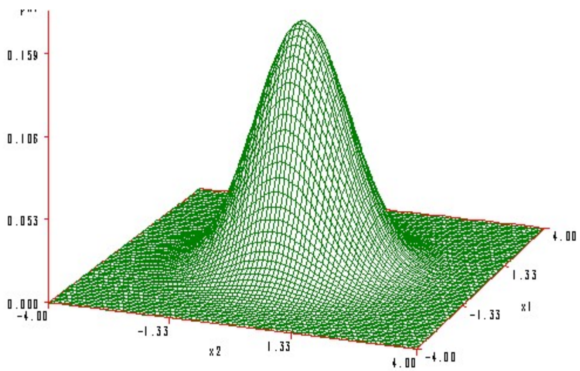
- Similar to ML estimate with an additional term

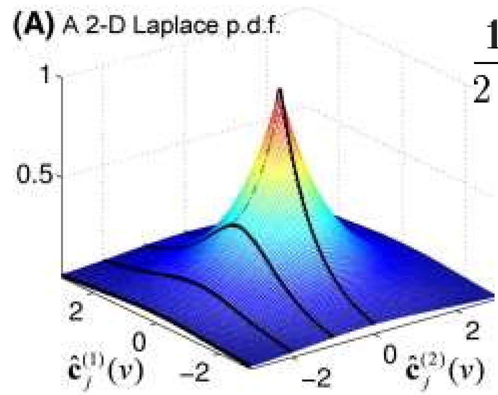
MAP estimate of weights

$$\mathbf{a} = \left(\mathbf{X}\mathbf{X}^T + \sigma^2\mathbf{I}\right)^{-1}\mathbf{X}\mathbf{Y}^T$$

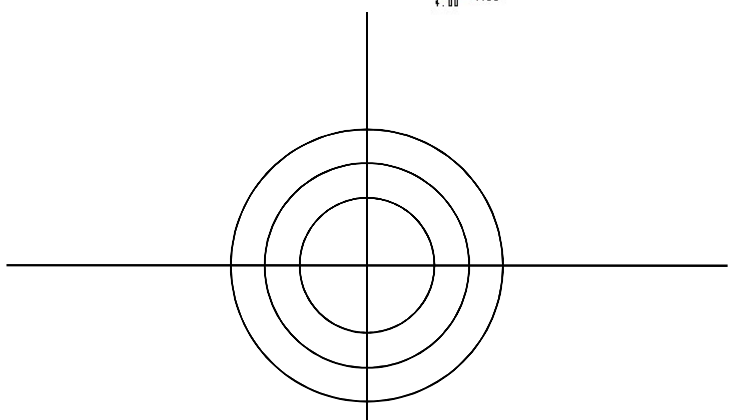
- Equivalent to *diagonal loading* of correlation matrix
  - Improves condition number of correlation matrix
    - Can be inverted with greater stability
  - Will not affect the estimation from well-conditioned data
  - Also called Tikhonov Regularization
    - Dual form: Ridge regression
- **MAP estimate of *weights***
  - **Not to be confused with MAP estimate of Y**

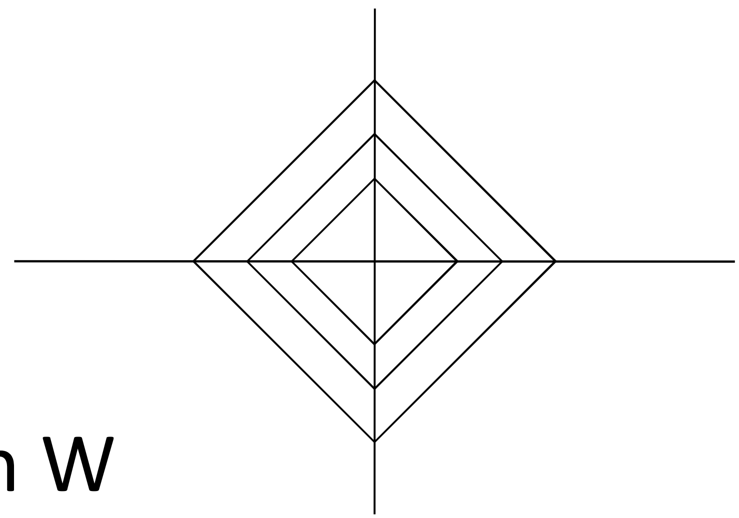
MAP estimate priors





$$\frac{1}{2\,b}\exp\left(-\frac{|x-\mu|}{b}\right)$$





- Left: Gaussian Prior on W
- Right: Laplacian Prior

MAP estimation of weights with laplacian prior

- Assume weights drawn from a Laplacian
  - $P(\mathbf{a}) = \lambda^{-1}\exp(-\lambda^{-1}|\mathbf{a}|_1)$
- Maximum *a posteriori* estimate

$$\hat{\mathbf{a}} = \arg\max_{\mathbf{a}} C' - trace\left((\mathbf{Y} - \mathbf{a}^T\mathbf{X})^T(\mathbf{Y} - \mathbf{a}^T\mathbf{X})^T\right) - \lambda^{-1}|\mathbf{a}|_1$$

- No closed form solution
  - Quadratic programming solution required
    - Non-trivial

MAP estimation of weights with laplacian prior

- Assume weights drawn from a Laplacian
  - $P(\mathbf{a}) = \lambda^{-1}\exp(-\lambda^{-1}|\mathbf{a}|_1)$
- Maximum *a posteriori* estimate
  - $\hat{\mathbf{a}} = \arg\max_{\mathbf{a}} C' - trace\left((\mathbf{Y} - \mathbf{a}^T\mathbf{X})^T(\mathbf{Y} - \mathbf{a}^T\mathbf{X})^T\right) - \lambda^{-1}|\mathbf{a}|_1$
- This is also L_1 regularized least-squares estimation

L_1-regularized LSE

$$\hat{\mathbf{a}} = \arg\max_{\mathbf{a}} C' - trace\left((\mathbf{Y} - \mathbf{a}^T\mathbf{X})^T (\mathbf{Y} - \mathbf{a}^T\mathbf{X})^T \right) - \lambda^{-1} |\mathbf{a}|_1$$

- No closed form solution
  - Quadratic programming solutions required

- Dual formulation

$$\hat{\mathbf{a}} = \arg\max_{\mathbf{a}} C' - trace\left((\mathbf{Y} - \mathbf{a}^T\mathbf{X})^T (\mathbf{Y} - \mathbf{a}^T\mathbf{X})^T \right) \quad \textbf{subject to} \quad |\mathbf{a}|_1 \leq t$$

- "LASSO" – Least absolute shrinkage and selection operator

LASSO Algorithms

- Various convex optimization algorithms
- LARS: Least angle regression
- Pathwise coordinate descent..
- Matlab code available from web

Regularized least squares

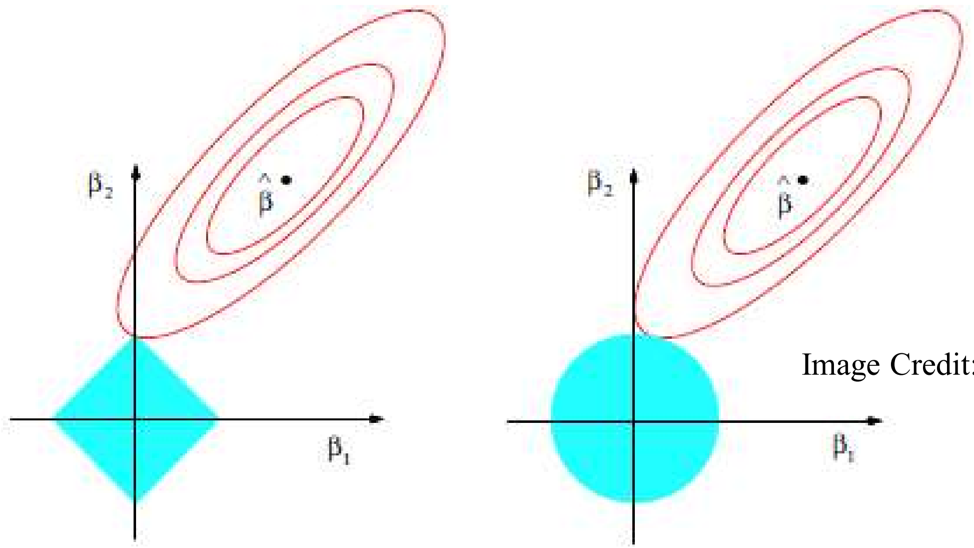
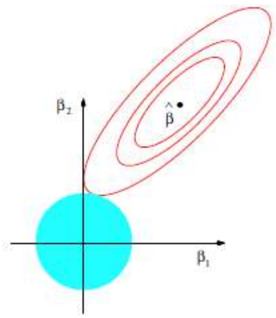


Image Credit: Tibshirani

- Regularization results in selection of suboptimal (in least-squares sense) solution
  - One of the loci outside center
- Tikhonov regularization selects ***shortest*** solution
- L_1 regularization selects ***sparsest*** solution

The different formalisms in L_2



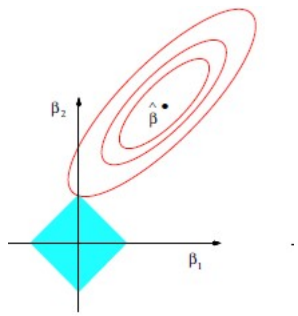
Tikhnov regularization (Gaussian prior)

$$\hat{\mathbf{a}} = \arg\max_{\mathbf{a}} C' - trace\left((\mathbf{Y} - \mathbf{a}^T\mathbf{X})^T(\mathbf{Y} - \mathbf{a}^T\mathbf{X})^T\right) - \lambda^{-1}\|\mathbf{a}\|^2$$

$$\hat{\mathbf{a}} = \arg\max_{\mathbf{a}} C' - trace\left((\mathbf{Y} - \mathbf{a}^T\mathbf{X})^T(\mathbf{Y} - \mathbf{a}^T\mathbf{X})^T\right)$$ **subject to** $$\|\mathbf{a}\|^2 \le t$$

- Expand both the ball and the ellipses till the both just meet
- Fix the ball, expand the ellipse till it meets the ball

The different formalisms in L_1



L_1 regularization (Laplacian prior)

$$\hat{\mathbf{a}} = \arg\max_{\mathbf{a}} C' - trace\left((\mathbf{Y} - \mathbf{a}^T\mathbf{X})^T(\mathbf{Y} - \mathbf{a}^T\mathbf{X})^T\right) - \lambda^{-1}|\mathbf{a}|_1$$

$$\hat{\mathbf{a}} = \arg\max_{\mathbf{a}} C' - trace\left((\mathbf{Y} - \mathbf{a}^T\mathbf{X})^T(\mathbf{Y} - \mathbf{a}^T\mathbf{X})^T\right)$$ **subject to** $$|\mathbf{a}|_1 \leq t$$

- Expand both the diamond and the ellipses till the both just meet
- Fix the diamond, expand the ellipse till it meets the ball

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..

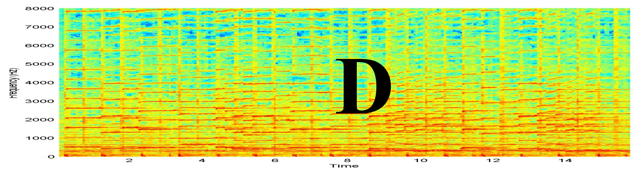
Gaussians and more Gaussians..

- Linear Gaussian Models..
- But first a recap

Linear Gaussian Models

- MAP and MMSE prediction with Gaussian models
  - Estimation
  - Regularization
- Representation
  - PCA
  - Probabilistic PCA
- Gaussian Classifier

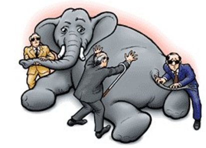
A Brief Recap



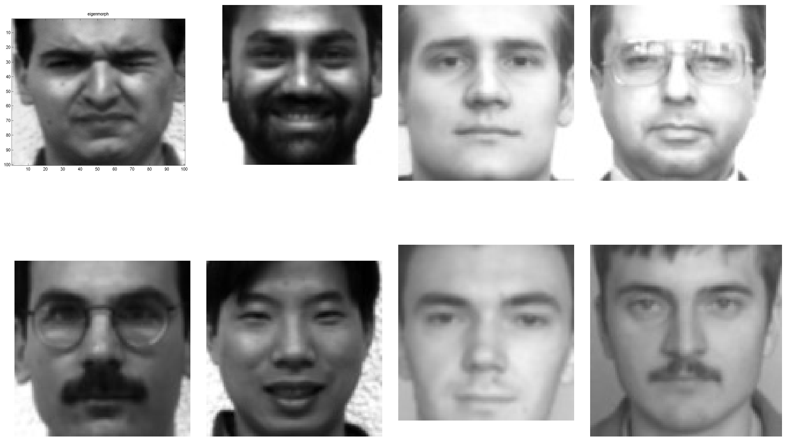
$\mathbf{C}$

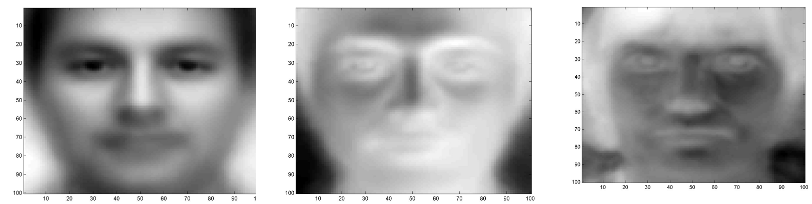
$\mathbf{B}$ $\quad$ $\mathbf{D} \approx \mathbf{BC}$

- 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

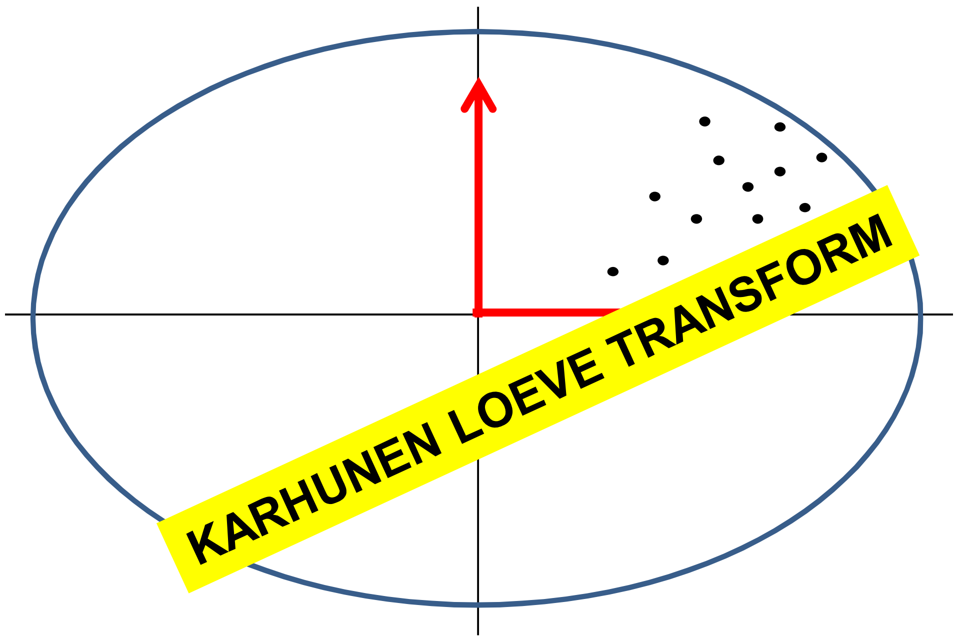


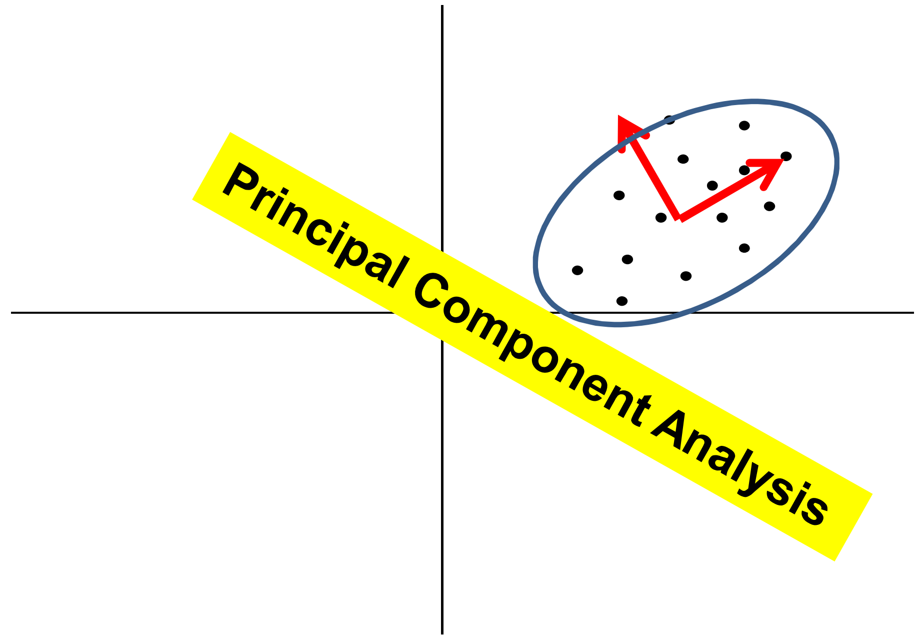


- Approximate every face f as
  $f = w_{f,1} V_1 + w_{f,2} V_2 + w_{f,3} V_3 + .. + w_{f,k} V_k$
- Estimate V to minimize the squared error

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

Karhunen Loeve vs. PCA

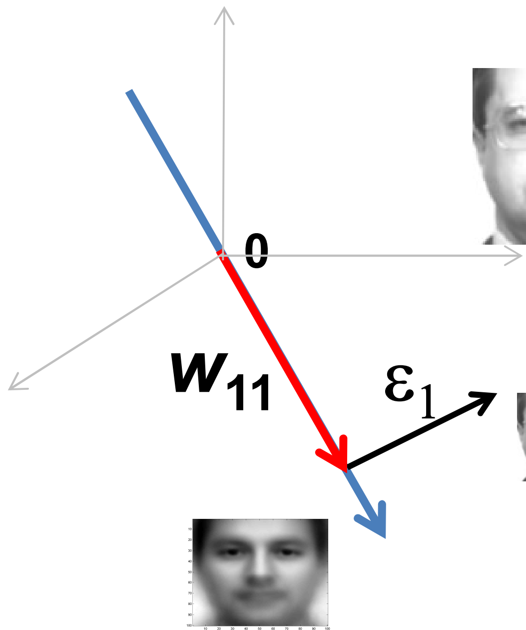




- Eigenvectors of the *Correlation* matrix:
  - Principal directions of tightest ellipse ***centered on origin***
  - Directions that retain maximum ***<u>energy</u>***

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

Eigen Representation



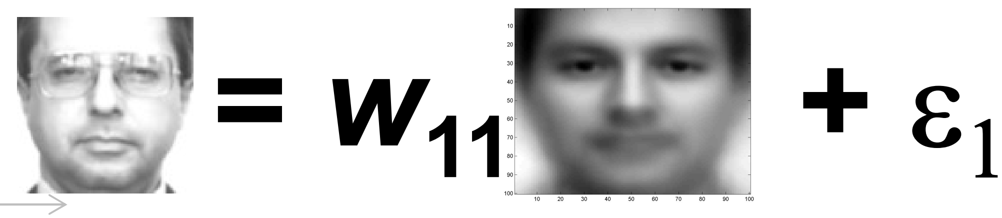
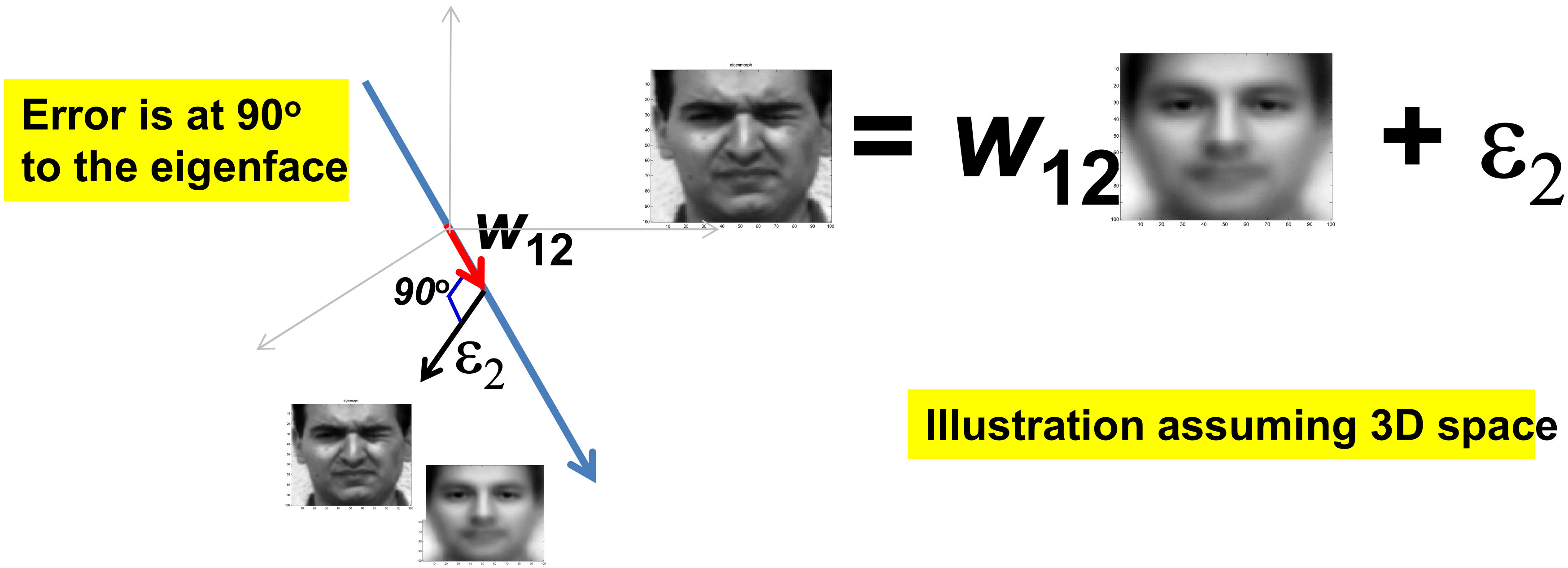


Illustration assuming 3D space

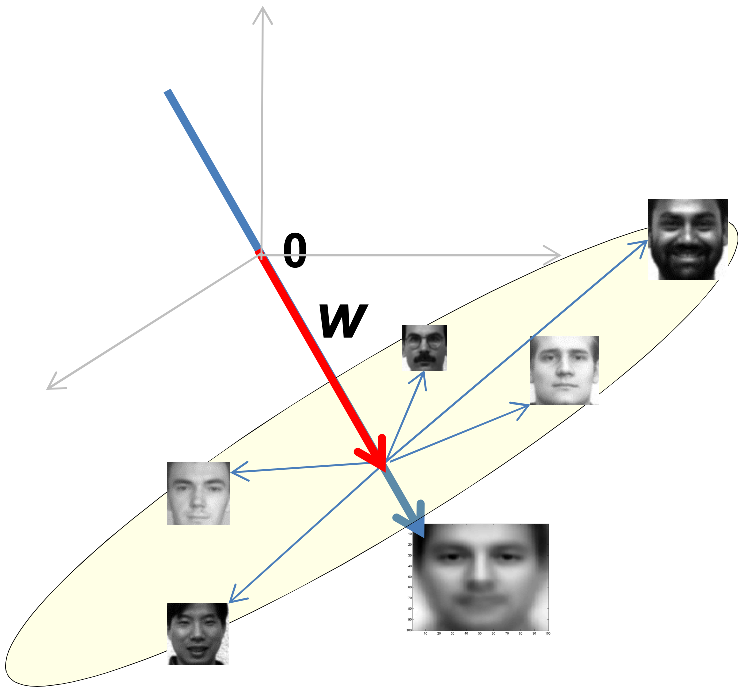
- K-dimensional representation
  - Error is orthogonal to representation
  - Weight and error are specific to data instance

Representation



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

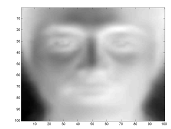
Representation



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

 $= w_{11}$  $+ w_{21}$  $+ \varepsilon_1$

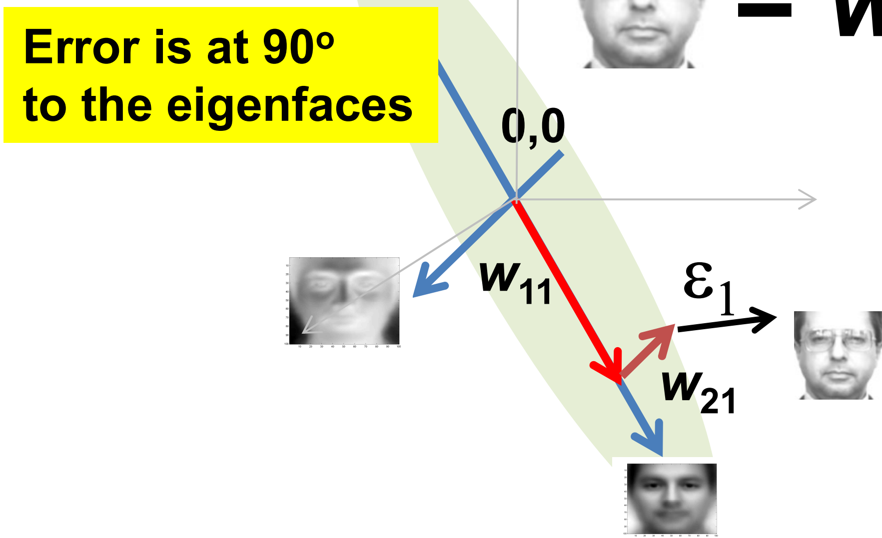
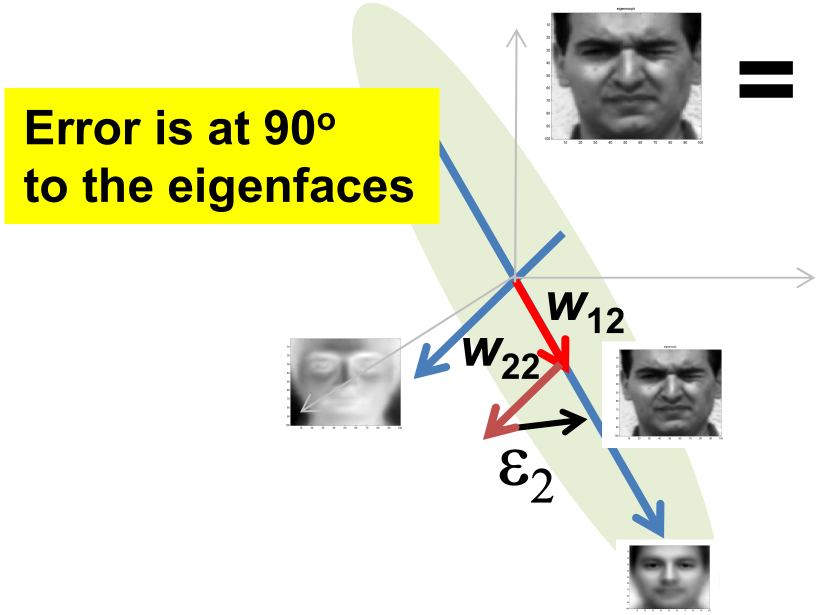
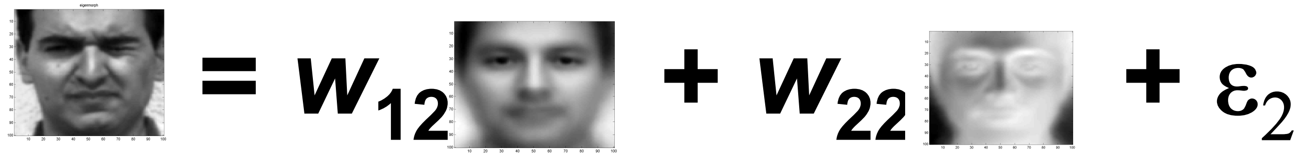


Illustration assuming 3D space

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

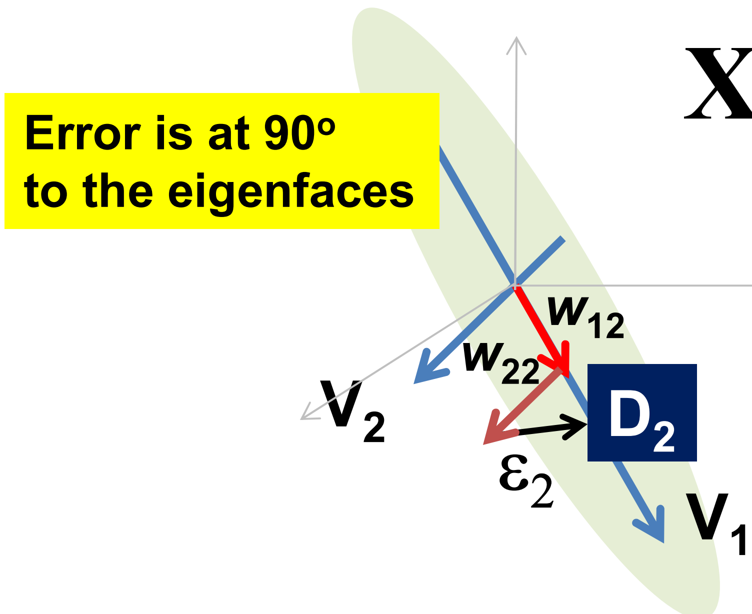
With 2 bases





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

In Vector Form

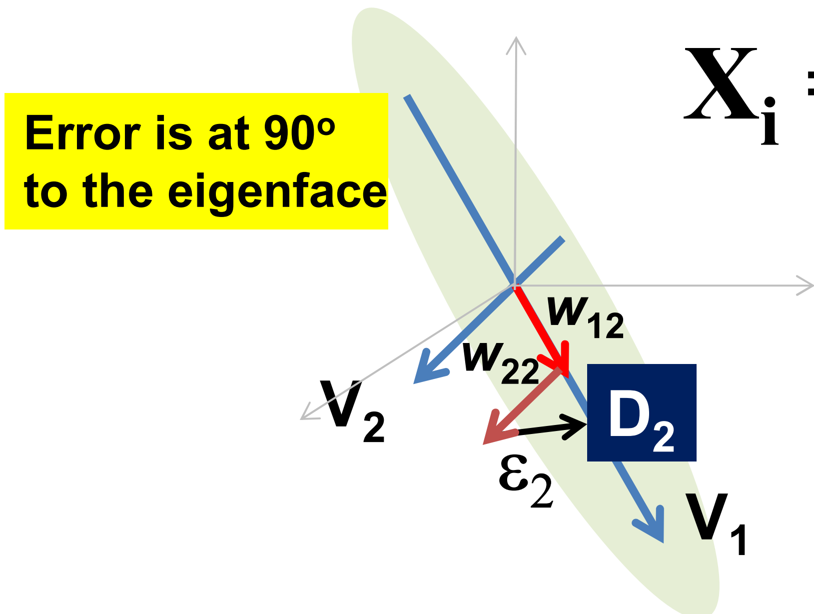


$$\mathbf{X_i} = w_{1\mathbf{i}}\mathbf{V}_1 + w_{2\mathbf{i}}\mathbf{V}_2 + \varepsilon_\mathbf{i}$$

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

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

In Vector Form



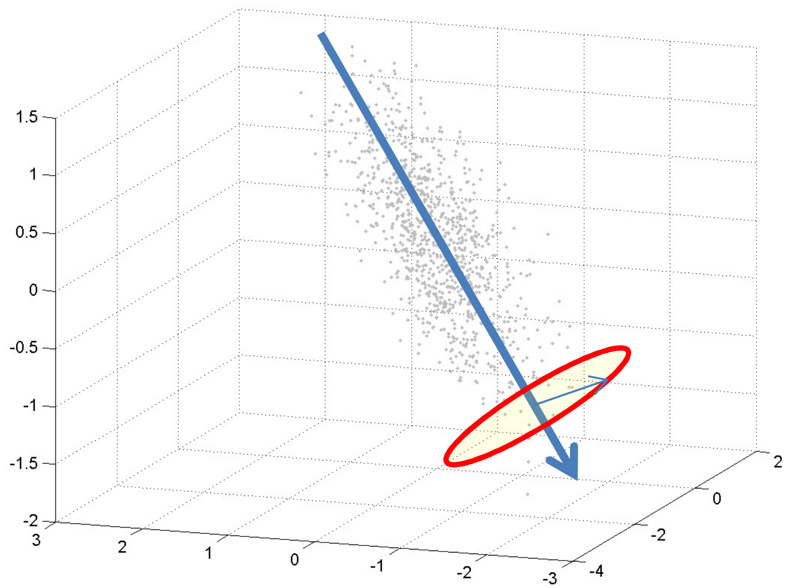
$$\mathbf{X_i} = w_{1i}\mathbf{V}_1 + w_{2i}\mathbf{V}_2 + \varepsilon_{\mathbf{i}}$$

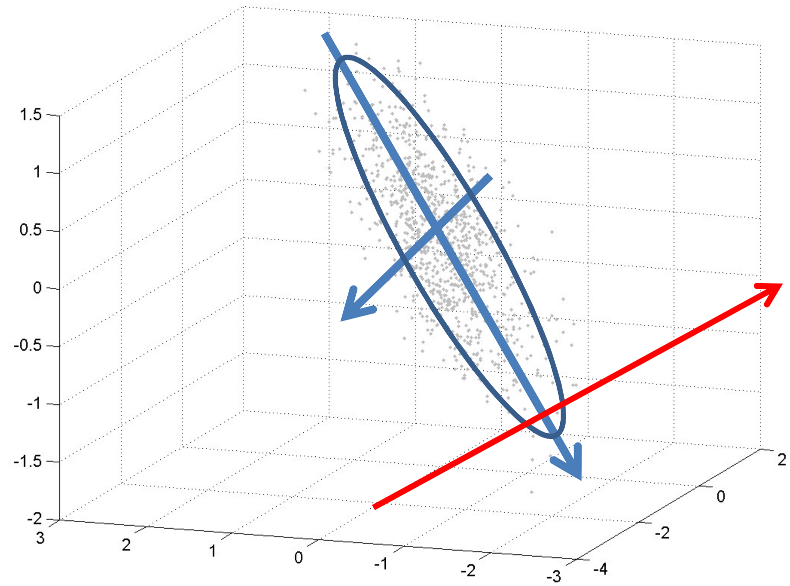
$$\mathbf{x} = \mathbf{V}\mathbf{w} + \mathbf{e}$$

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

- *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

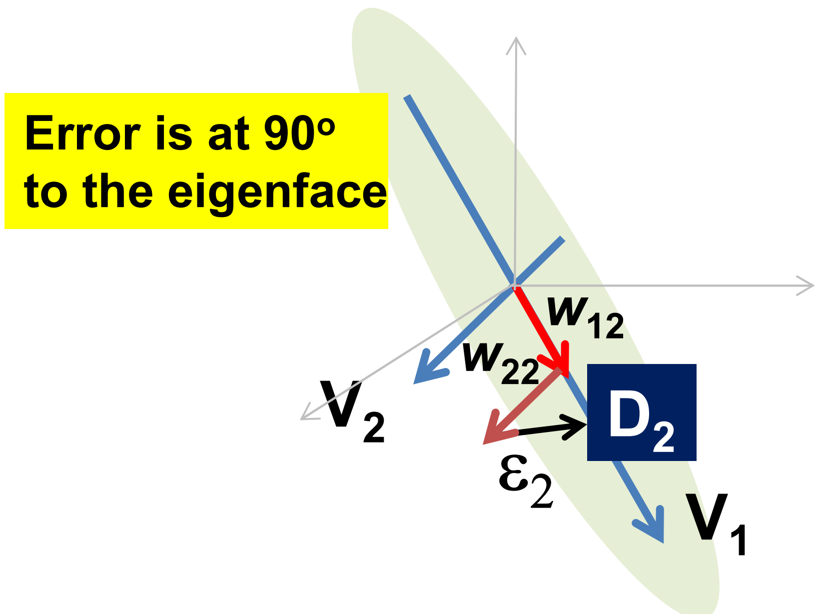
Learning PCA





- 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

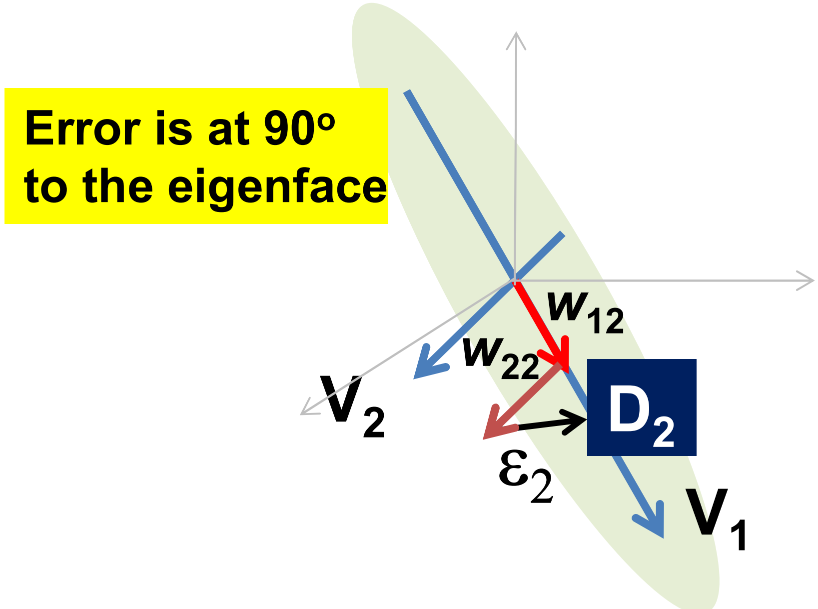
Constraints



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

- $\mathbf{V}^T\mathbf{V} = \mathbf{I}$: 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 $\mathbf{w}\mathbf{w}^T = \mathbf{Diagonal}$: Eigen representations are uncorrelated
  - $\mathbf{e}^T\mathbf{e}$ = minimum: Error variance is minimum
    - Mean of error is 0

A Statistical Formulation of PCA



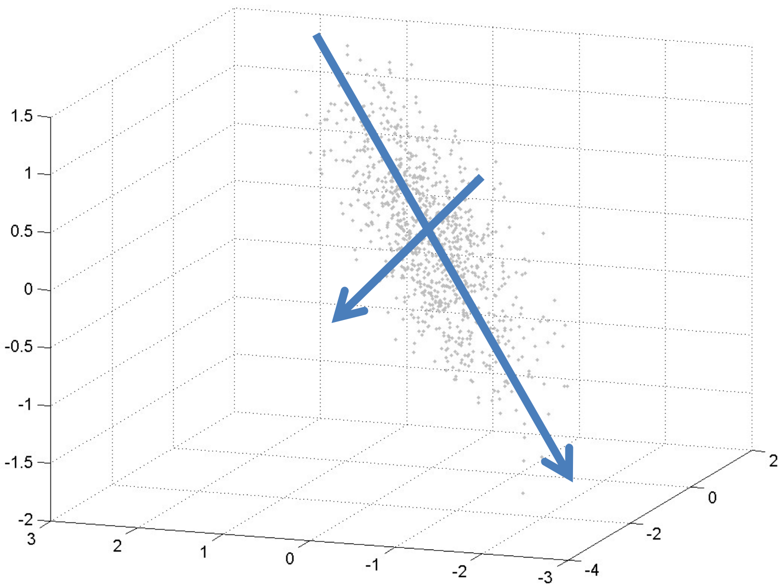
$$\mathbf{x} = \mathbf{V}\mathbf{w} + \mathbf{e}$$

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

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

- $\mathbf{x}$ 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!!



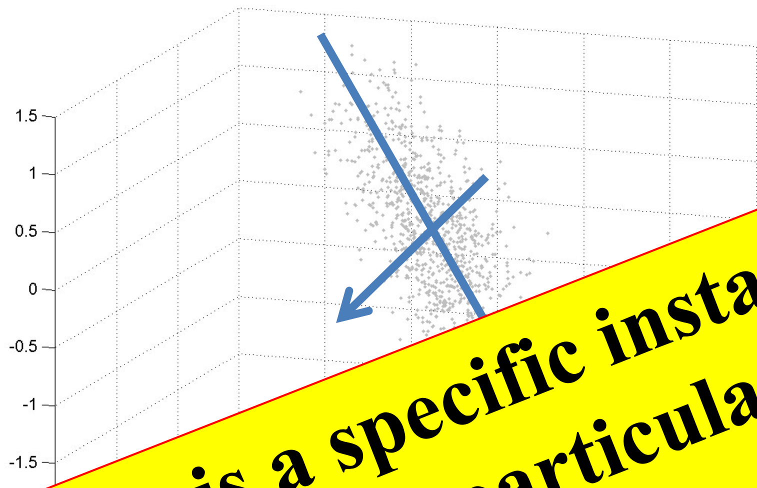
$$\mathbf{x} = \mathbf{V}\mathbf{w} + \mathbf{e}$$

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

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

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

Linear Gaussian Models!!



X

...g to a linear relation

-

- ... nean Gaussian

- Es... 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***
  - **$V^TV = I$**
- ***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

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

- 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 μ, $\mathbf{V}$, 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 μ, $\mathbf{V}$, and E

The Maximum Likelihood Estimate

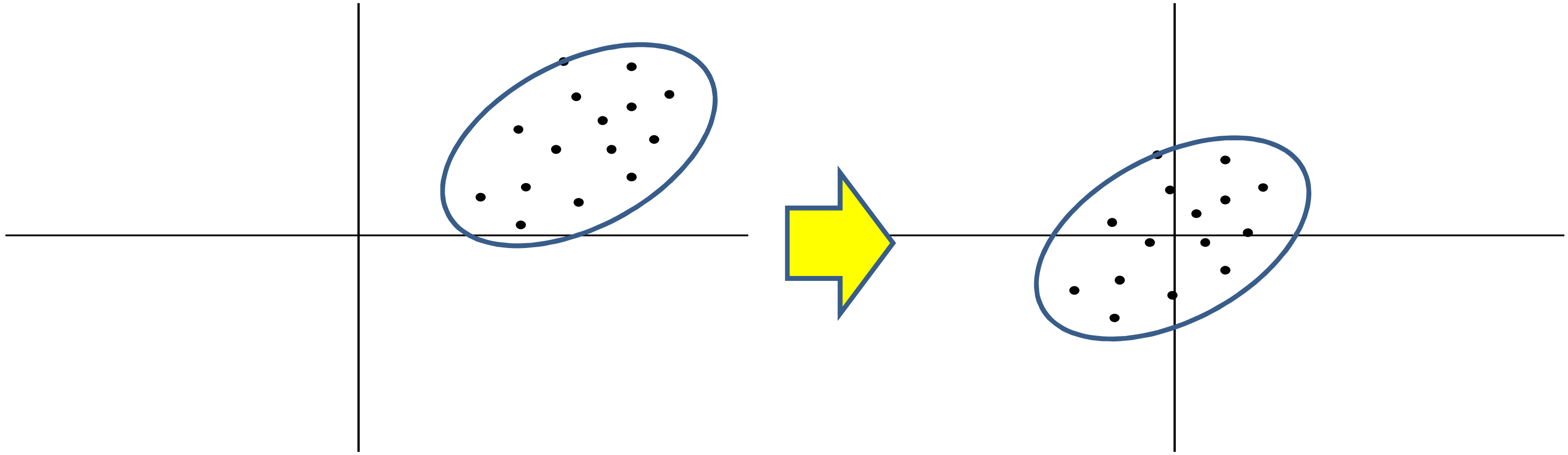
$$\boxed{\mathbf{x} \sim N(\boldsymbol{\mu}, \mathbf{V}\mathbf{V}^T + E)}$$

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

- The ML estimate of μ 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 $\mathbf{x}_i$ 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

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

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

$$\boxed{P(\mathbf{x} \mid \mathbf{w}) = N(\mathbf{V}\mathbf{w}, E)}$$

Reminder: x and w are jointly Gaussian

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

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

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

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

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

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

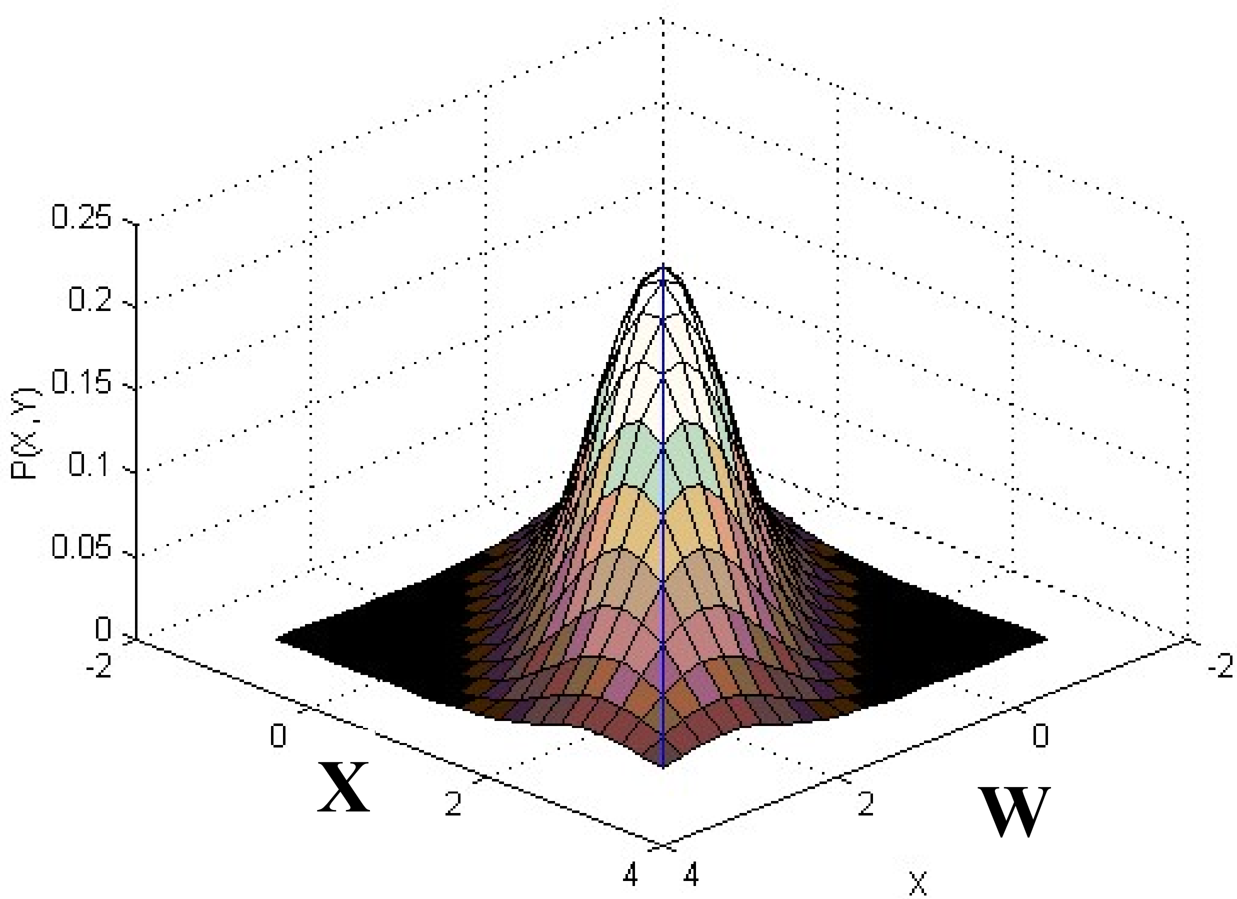
$$C_{\mathbf{zz}} = \begin{bmatrix} C_{\mathbf{xx}} & C_{\mathbf{xw}} \\ C_{\mathbf{wx}} & C_{\mathbf{ww}} \end{bmatrix}$$

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

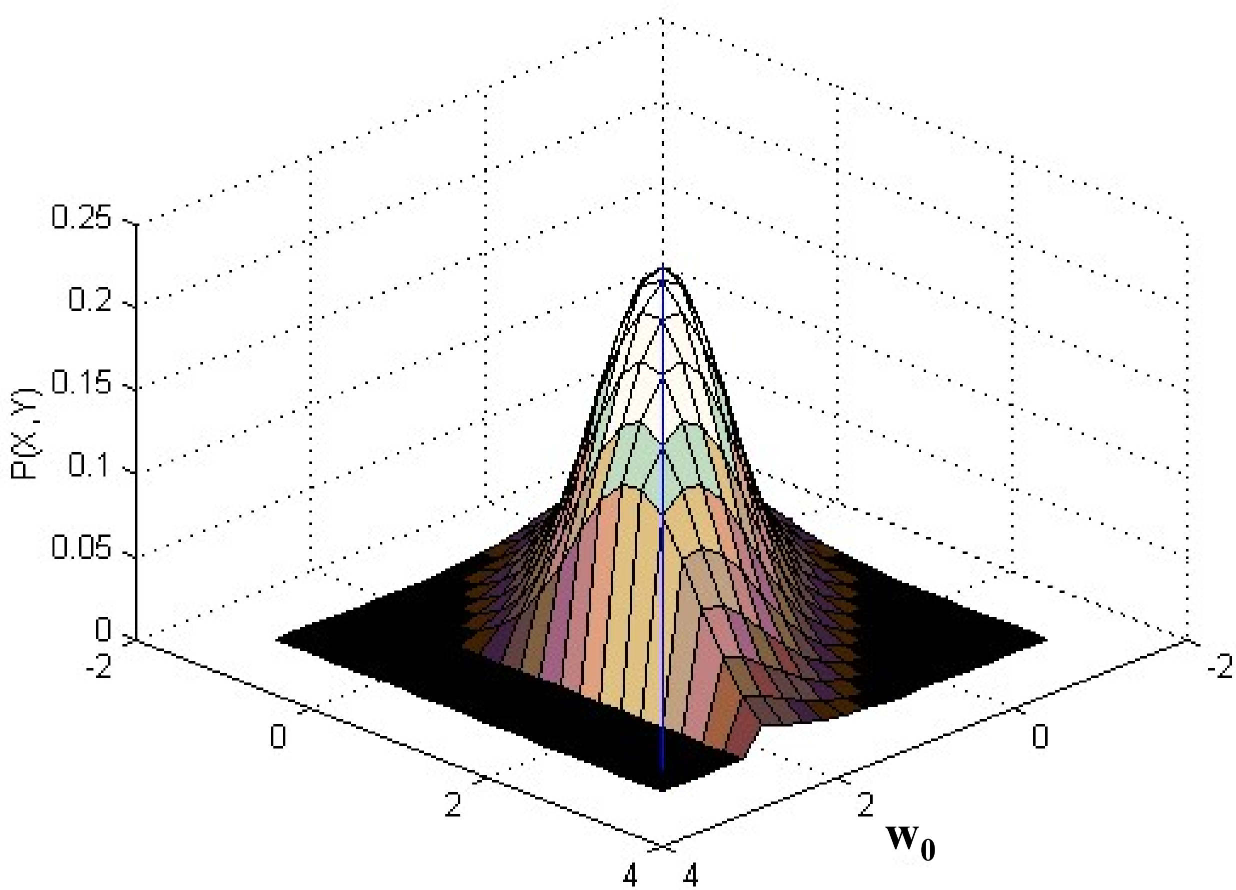
$$C_{\mathbf{zz}} = \begin{bmatrix} \mathbf{V}\mathbf{V}^T + E & \mathbf{V} \\ \mathbf{V}^T & I \end{bmatrix}$$

- $\mathbf{x}$ and $\mathbf{w}$ are jointly Gaussian!

MAP estimation: Gaussian PDF



MAP estimation: The Gaussian at a particular value of X



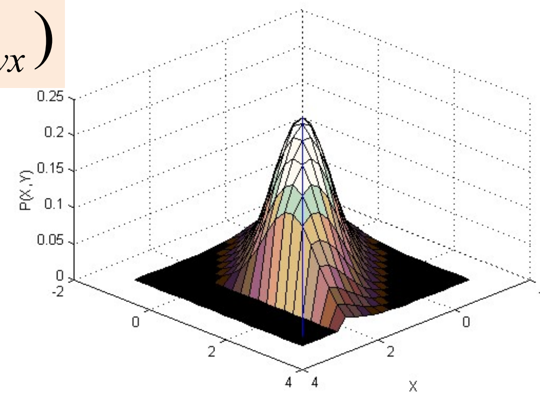
Conditional Probability of x|w

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

$$= N(C_{xw}C_{ww}^{-1}\mathbf{w}, C_{xx} - C_{xw}C_{ww}^{-1}C_{wx})$$

$$E_{x|w}[\mathbf{x}] = C_{xw}C_{ww}^{-1}\mathbf{w}$$

$$Var(\mathbf{x}\,|\,\mathbf{w}) = C_{xx} - C_{xw}C_{ww}^{-1}C_{wx}$$



- Comparing to

$$\boxed{P(\mathbf{x}\,|\,\mathbf{w}) = N(\mathbf{V}\mathbf{w}, E)}$$

- We get:

$$\mathbf{V} = C_{xw}C_{ww}^{-1} \qquad E = C_{xx} - C_{xw}C_{ww}^{-1}C_{wx}$$

Or more explicitly

$$C_{ww} = \frac{1}{N}\sum_i \mathbf{w}_i \mathbf{w}_i^T$$

$$C_{xw} = \frac{1}{N}\sum_i \mathbf{x}_i \mathbf{w}_i^T$$

$$\mathbf{V} = C_{xw} C_{ww}^{-1}$$

$$E = C_{xx} - C_{xw} C_{ww}^{-1} C_{wx}$$

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

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

Estimating LGMs: If we know *w*

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

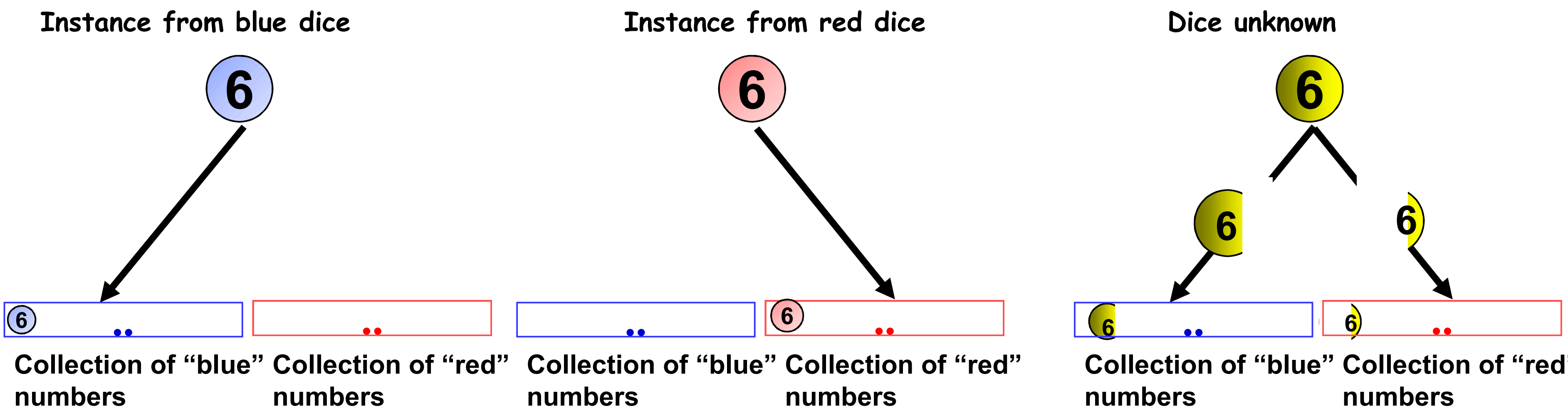
$$\mathbf{V} = \left(\sum_i \mathbf{x}_i \mathbf{w}_i^T\right)\left(\sum_i \mathbf{w}_i \mathbf{w}_i^T\right)^{-1}$$

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

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

- EM..

Recall EM



- 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

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

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

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

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

Flipping the problem

$$\boxed{E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]} \qquad \boxed{E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}\mathbf{w}^T]}$$

- How do we estimate the above terms?
- MAP to the rescue!!

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

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

Recall: w and x are jointly Gaussian

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

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

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

$$C_{xx} = \mathbf{V}\mathbf{V}^T + E \qquad C_{ww} = \mathbf{I}$$

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

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

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

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

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

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

Recall: w and x are jointly Gaussian

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

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

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

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

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

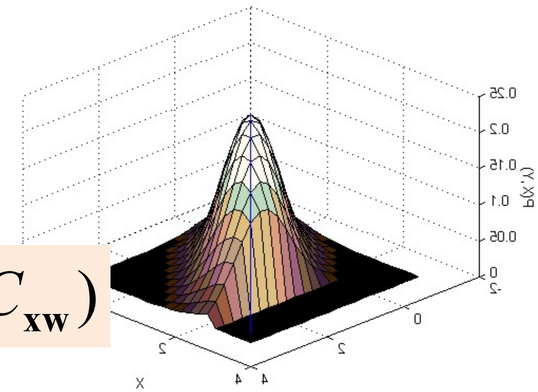
$$C_{xx} = \mathbf{V}\mathbf{V}^T + E \qquad C_{ww} = \mathbf{I}$$

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

- $\mathbf{x}$ and $\mathbf{w}$ are jointly Gaussian!

P(w | z)

- P(w|z) is a Gaussian



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

$$= N(C_{\mathbf{wx}}C_{\mathbf{xx}}^{-1}\mathbf{x}, C_{\mathbf{ww}} - C_{\mathbf{wx}}C_{\mathbf{xx}}^{-1}C_{\mathbf{xw}})$$

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

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

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

$$E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}\mathbf{w}^T] = Var(\mathbf{w}\,|\,\mathbf{x}) + E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]E_{\mathbf{w}|\mathbf{x}_i}[\mathbf{w}]^T$$

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

LGM: The complete EM algorithm

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

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

- 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:

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

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

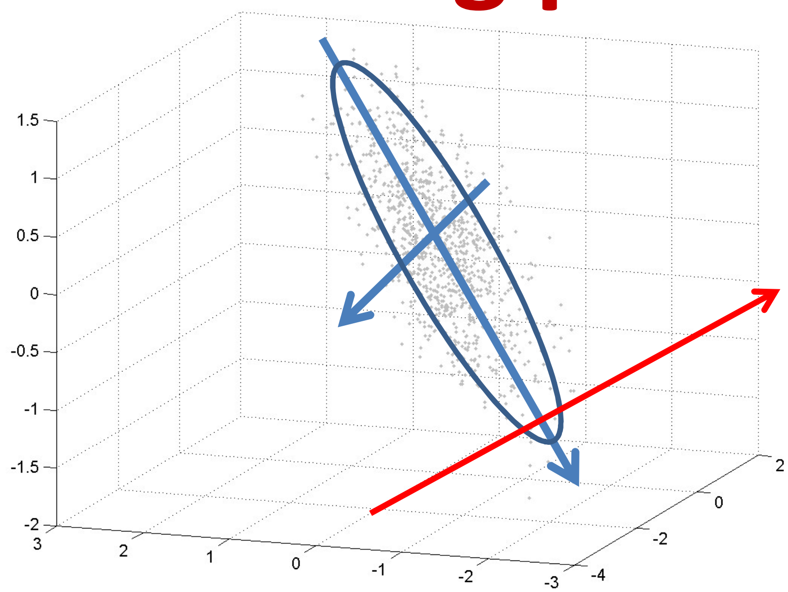
So what have we achieved

- Employed a complicated EM algorithm to learn a *Gaussian* PDF for a variable x
- What have we gained???
  – PCA
    - Sensible PCA
    - EM algorithms for PCA (Probabilistic PCA)

- Next class:
  – Factor Analysis
    - FA for feature extraction

LGMs : Application 1

Learning principal components



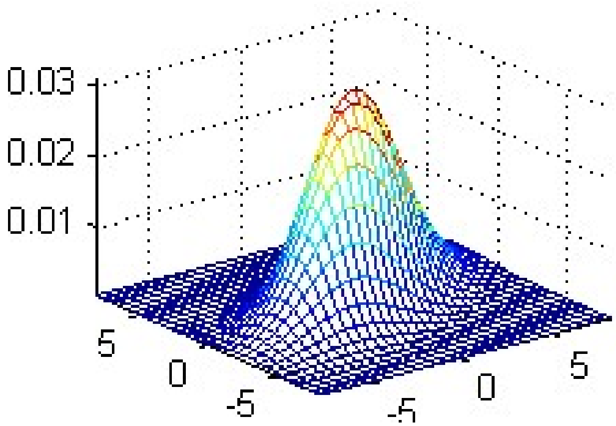
$$\mathbf{x} = \mathbf{V}\mathbf{w} + \mathbf{e}$$

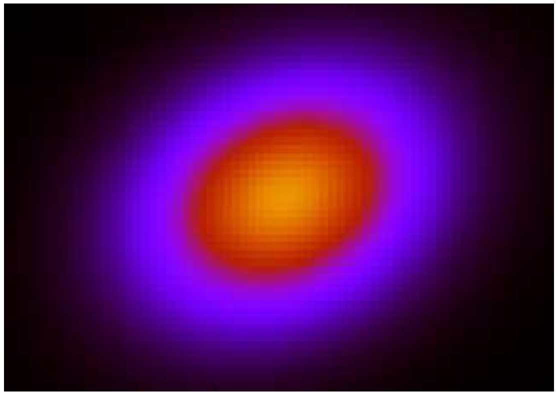
$$\mathbf{w} \sim N(0, I)$$

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

- 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**

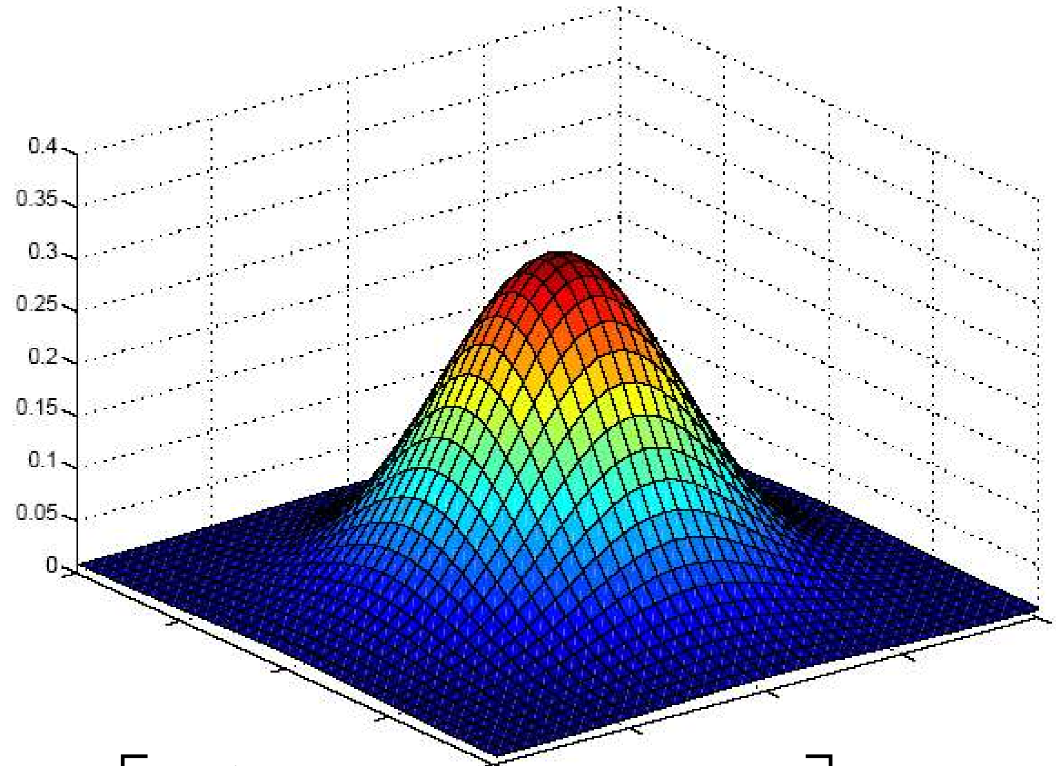
To be continued..

- Other applications..
- Next class

Linear Gaussian Models

- Recap
- Representation
  - PCA
  - Probabilistic PCA
- Gaussian Classifier

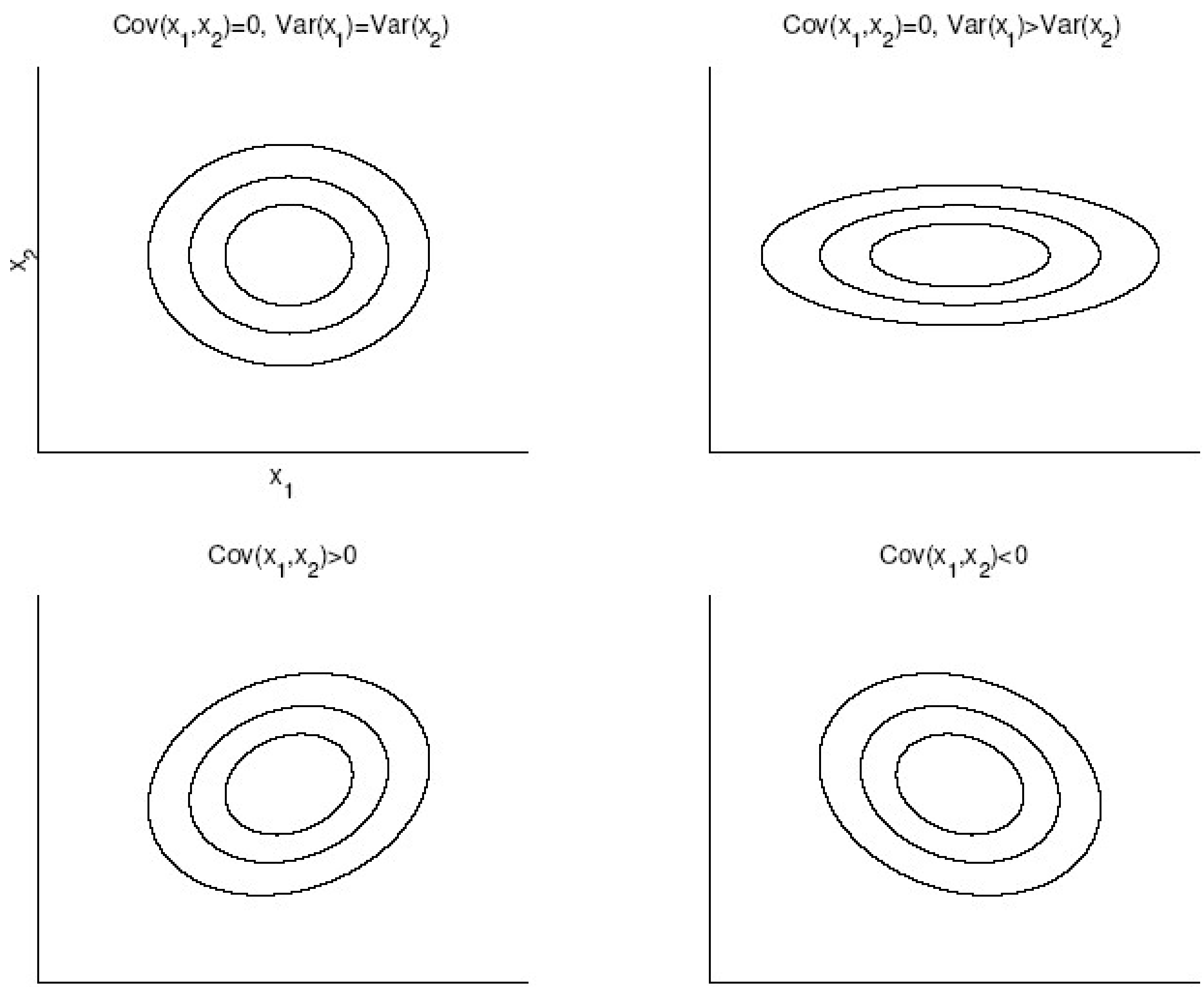
Multivariate Normal Distribution

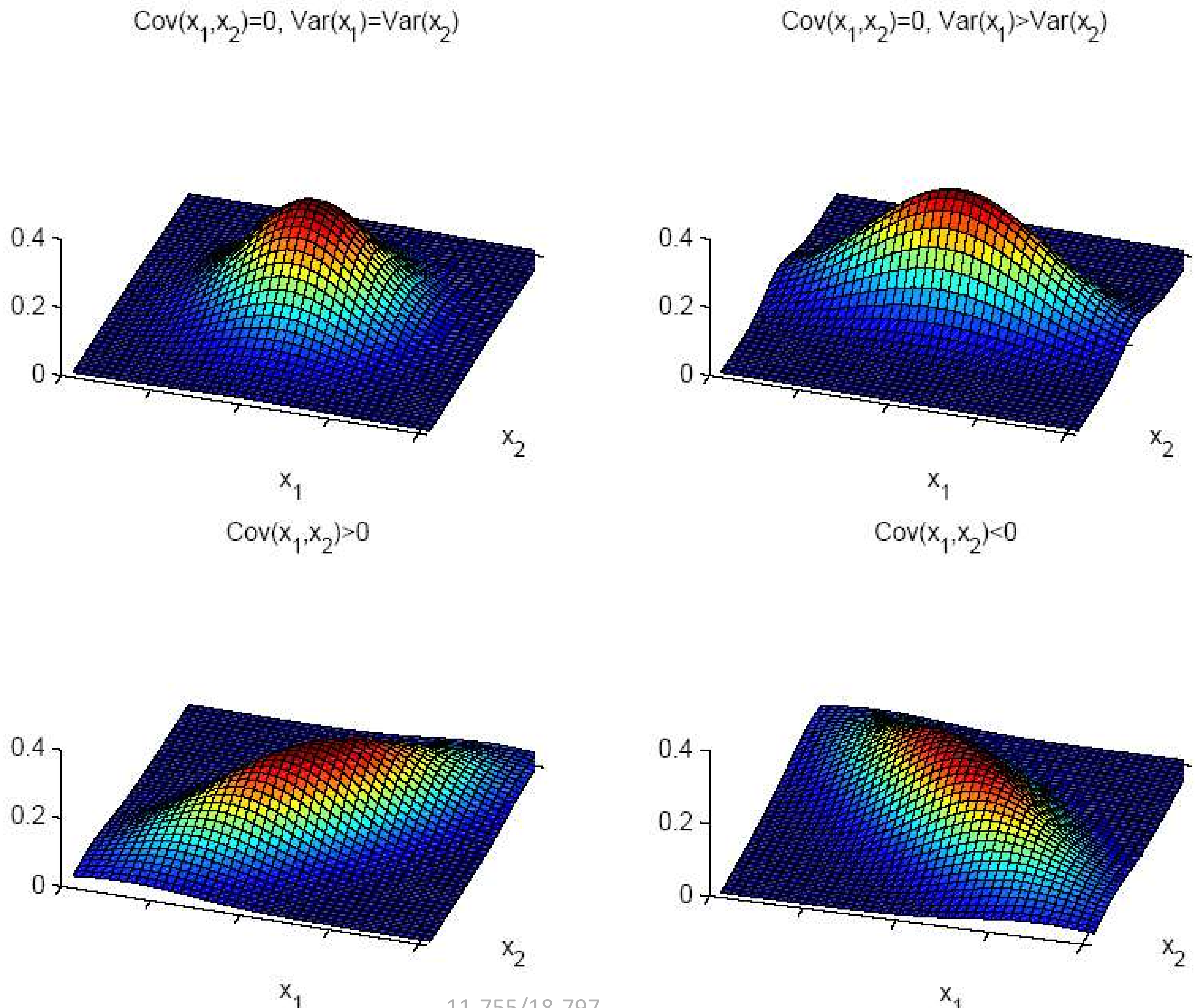


$$\mathbf{x} \sim \mathcal{N}_d(\boldsymbol{\mu}, \Sigma)$$

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x}-\boldsymbol{\mu})\right]$$

Bivariate Normal




Cov(x1,x2)=0, Var(x1)=Var(x2)
Cov(x1,x2)=0, Var(x1)>Var(x2)
0.4
0.2
0
x1
x2
Cov(x1,x2)>0
Cov(x1,x2)<0

Parametric Classification

- If $p\,(\boldsymbol{x} \mid C_i) \sim N\,(\boldsymbol{\mu}_i\,, \Sigma_i)$

$$p(\mathbf{x} \mid C_i) = \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T \Sigma_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)\right]$$

- Discriminant functions

$$\begin{aligned} g_i(\mathbf{x}) &= \log p(\mathbf{x} \mid C_i) + \log P(C_i) \\ &= -\frac{d}{2}\log 2\pi - \frac{1}{2}\log|\Sigma_i| - \frac{1}{2}(\mathbf{x}-\mu_i)^T \Sigma_i^{-1}(\mathbf{x}-\mu_i) + \log P(C_i) \end{aligned}$$

Estimation of Parameters

$$\hat{P}(C_i) = \frac{N_i}{N}$$

$$\mathbf{m}_i = \frac{\sum_{t \in class\, i} \mathbf{x}^t}{N_i}$$

$$\mathbf{S}_i = \frac{\sum_{t \in class\, i} \left(\mathbf{x}^t - \mathbf{m}_i\right)\left(\mathbf{x}^t - \mathbf{m}_i\right)^T}{N_i}$$

$$g_i(\mathbf{x}) = -\frac{1}{2}\log\left|\mathbf{S}_i\right| - \frac{1}{2}(\mathbf{x} - \mathbf{m}_i)^T \mathbf{S}_i^{-1}(\mathbf{x} - \mathbf{m}_i) + \log \hat{P}(C_i)$$

Different S_i

- Quadratic discriminant

$$g_i(\mathbf{x}) = -\frac{1}{2}\log|\mathbf{S}_i| - \frac{1}{2}\left(\mathbf{x}^T\mathbf{S}_i^{-1}\mathbf{x} - 2\mathbf{x}^T\mathbf{S}_i^{-1}\mathbf{m}_i + \mathbf{m}_i^T\mathbf{S}_i^{-1}\mathbf{m}_i\right) + \log\hat{P}(C_i)$$

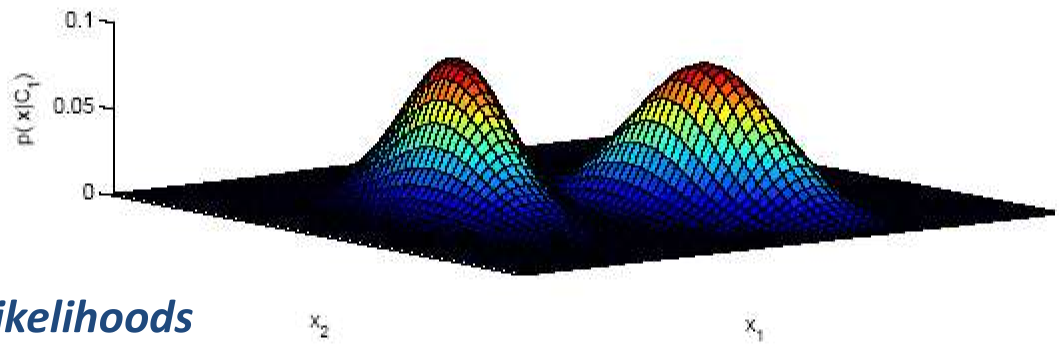
$$= \mathbf{x}^T\mathbf{W}_i\mathbf{x} + \mathbf{w}_i^T\mathbf{x} + w_{i0}$$

where

$$\mathbf{W}_i = -\frac{1}{2}\mathbf{S}_i^{-1}$$

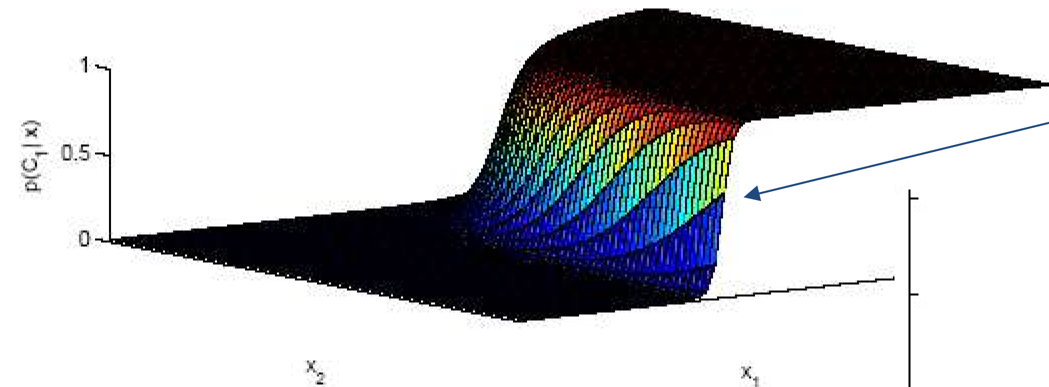
$$\mathbf{w}_i = \mathbf{S}_i^{-1}\mathbf{m}_i$$

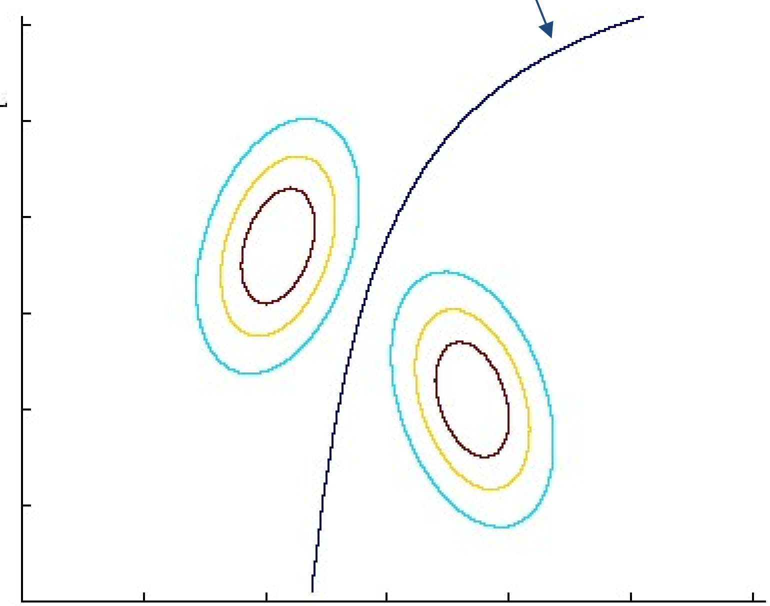
$$w_{i0} = -\frac{1}{2}\mathbf{m}_i^T\mathbf{S}_i^{-1}\mathbf{m}_i - \frac{1}{2}\log|\mathbf{S}_i| + \log\hat{P}(C_i)$$



likelihoods

discriminant:
$P\,(C_1|x\,) = 0.5$





posterior for C_1

Common Covariance Matrix S

- Shared common sample covariance **S**

$$S = \sum_i \hat{P}(C_i) S_i$$

- Discriminant reduces to

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \mathbf{m}_i)^T S^{-1} (\mathbf{x} - \mathbf{m}_i) + \log \hat{P}(C_i)$$

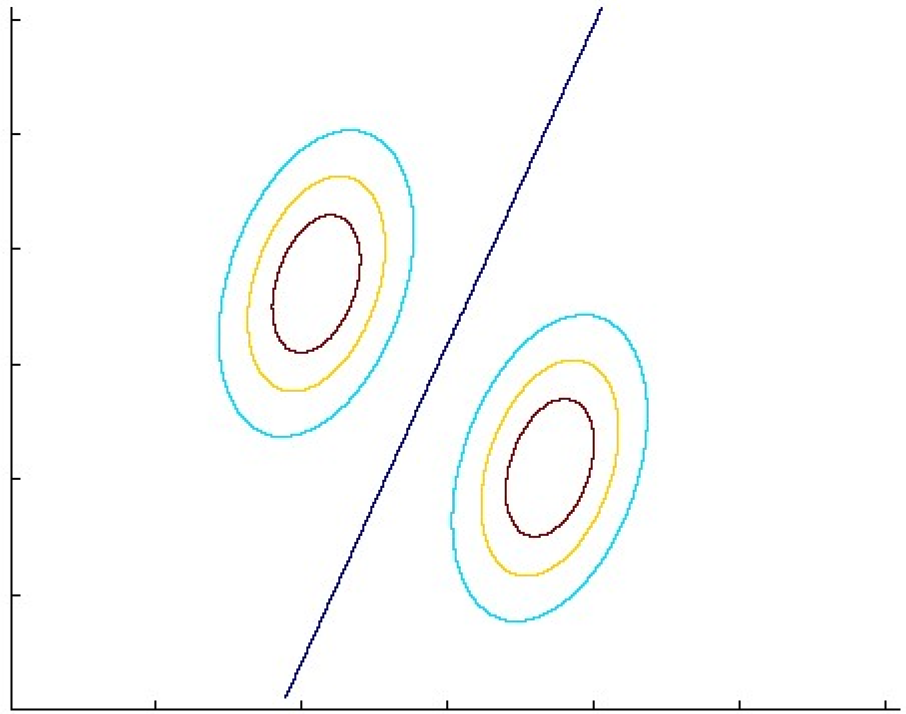
which is a linear discriminant

$$g_i(\mathbf{x}) = \mathbf{w}_i^T \mathbf{x} + w_{i0}$$

where

$$\mathbf{w}_i = \mathbf{S}^{-1}\mathbf{m}_i \quad w_{i0} = -\frac{1}{2}\mathbf{m}_i^T \mathbf{S}^{-1}\mathbf{m}_i + \log \hat{P}(C_i)$$

Common Covariance Matrix S



Diagonal S

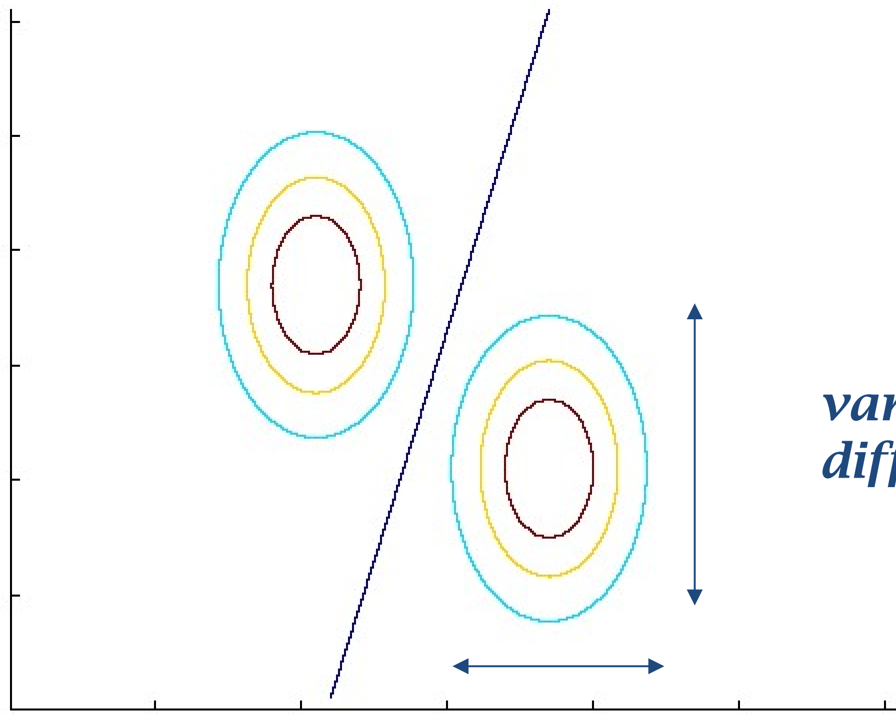
- When x_j j = 1,..d, are independent, ∑ is diagonal

  $p(\mathbf{x}|C_i) = \prod_j p(x_j | C_i)$ (Naive Bayes' assumption)

$$g_i(\mathbf{x}) = -\frac{1}{2}\sum_{j=1}^{d}\left(\frac{x_j^t - m_{ij}}{s_j}\right)^2 + \log \hat{P}(C_i)$$

Classify based on weighted Euclidean distance (in s_j units) to the nearest mean

Diagonal S



variances may be different

Diagonal S, equal variances

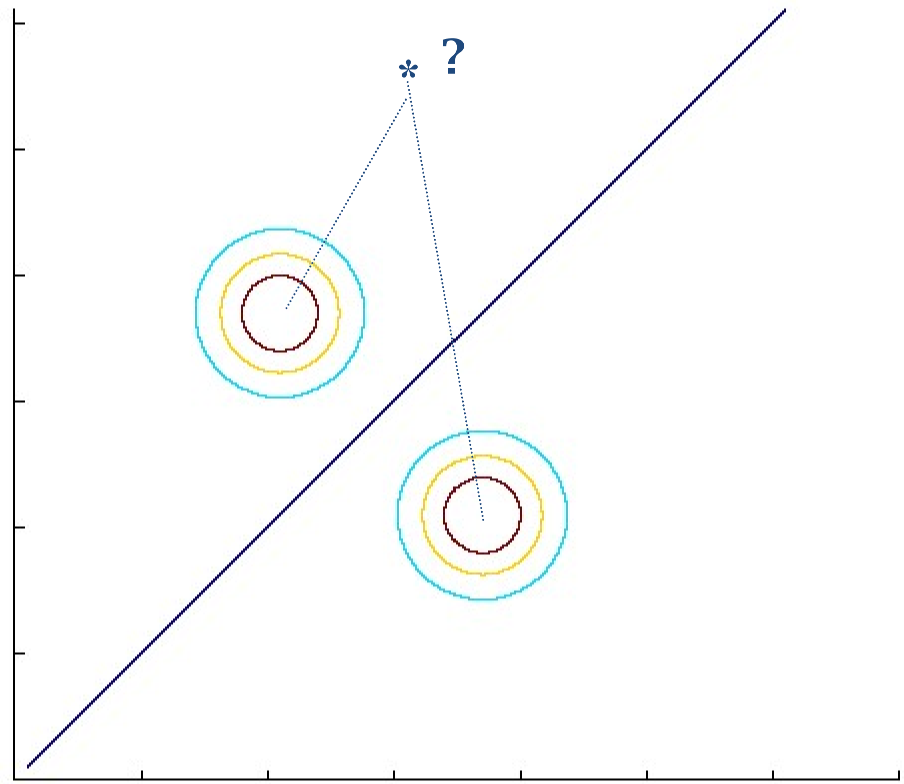
- Nearest mean classifier: Classify based on Euclidean distance to the nearest mean

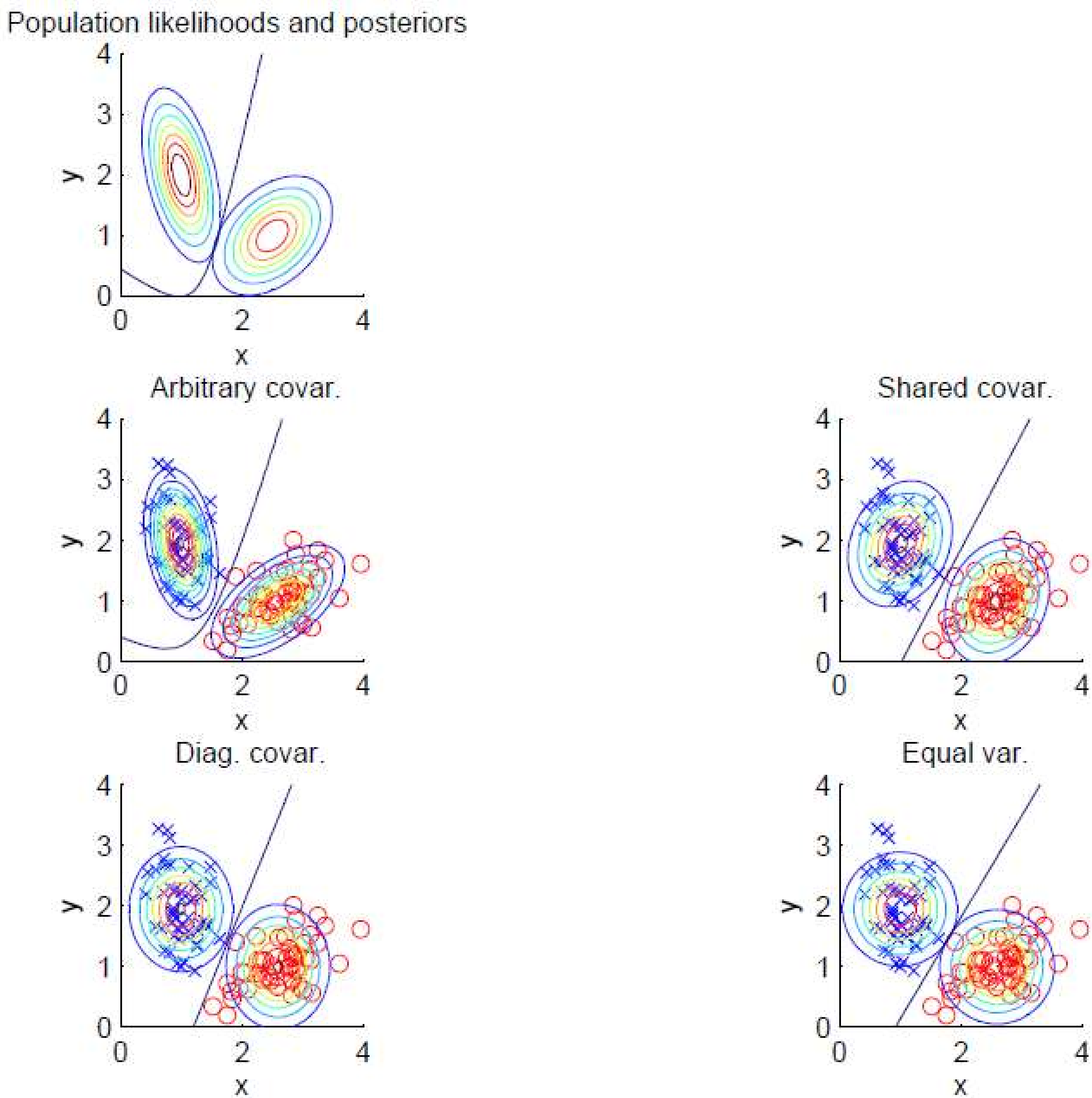
$$g_i(\mathbf{x}) = -\frac{\left\|\mathbf{x}-\mathbf{m}_i\right\|^2}{2s^2} + \log \hat{P}(C_i)$$

$$= -\frac{1}{2s^2}\sum_{j=1}^{d}\left(x_j^t - m_{ij}\right)^2 + \log \hat{P}(C_i)$$

- Each mean can be considered a prototype or template and this is template matching

Diagonal S, equal variances




Population likelihoods and posteriors
Arbitrary covar.
Shared covar.
Diag. covar.
Equal var.
x
y
0
2
4
1
3