UCB - CS189
Introduction to Machine Learning
Fall 2015
Lecture 14: Mixture models

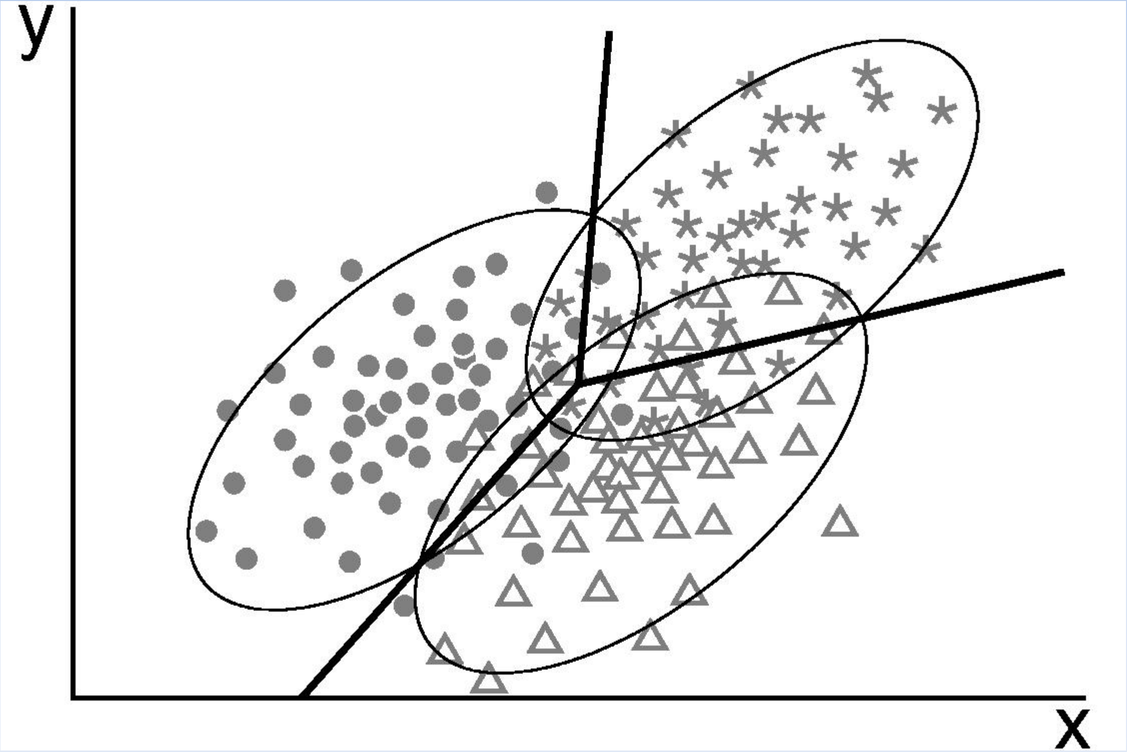
Isabelle Guyon

ChaLearn

Come to my office hours…

Wed 1:30-3:30 Soda 329

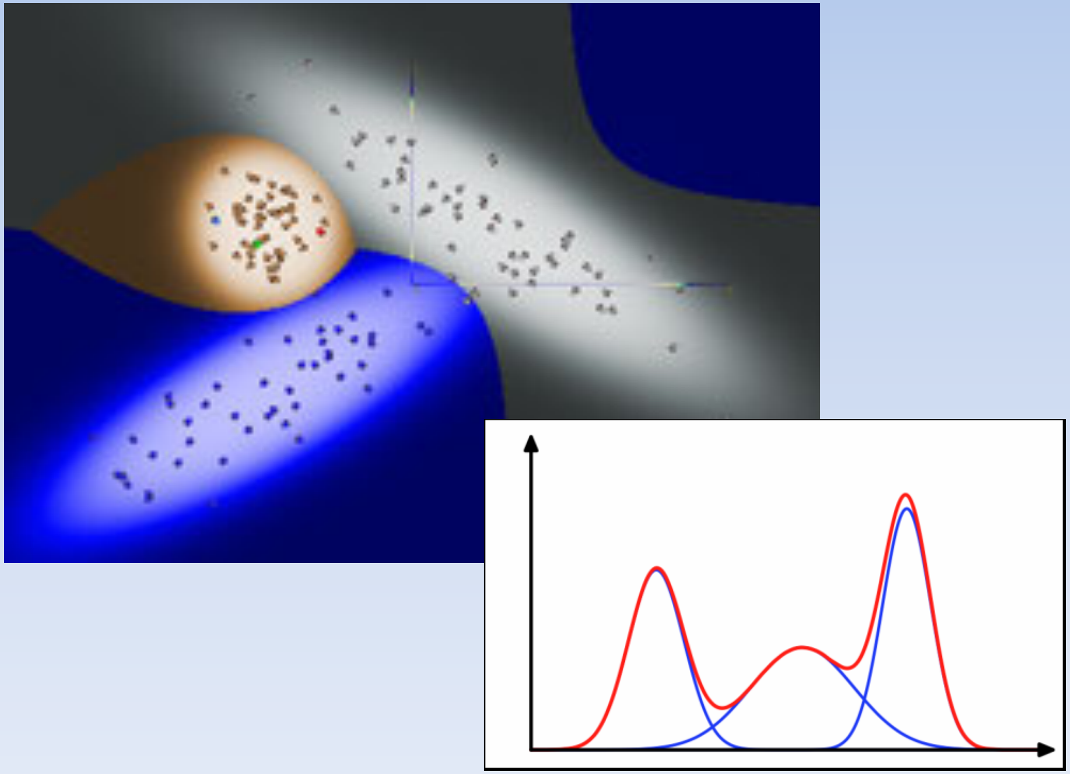
Last time: LDA



Come to my office hours…

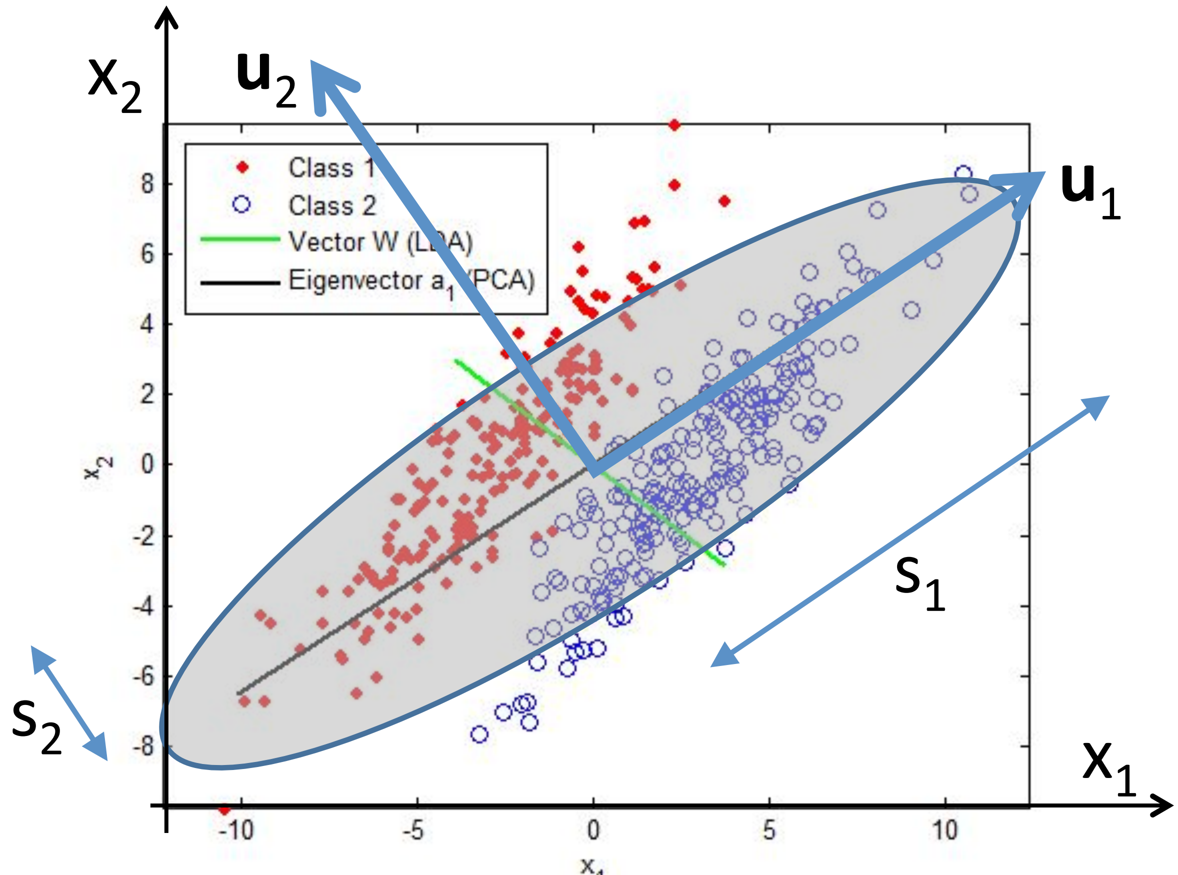
Wed 1:30-3:30 Soda 329

Today: Mixture models



Review

Covariance matrix $\Sigma = X^T X$



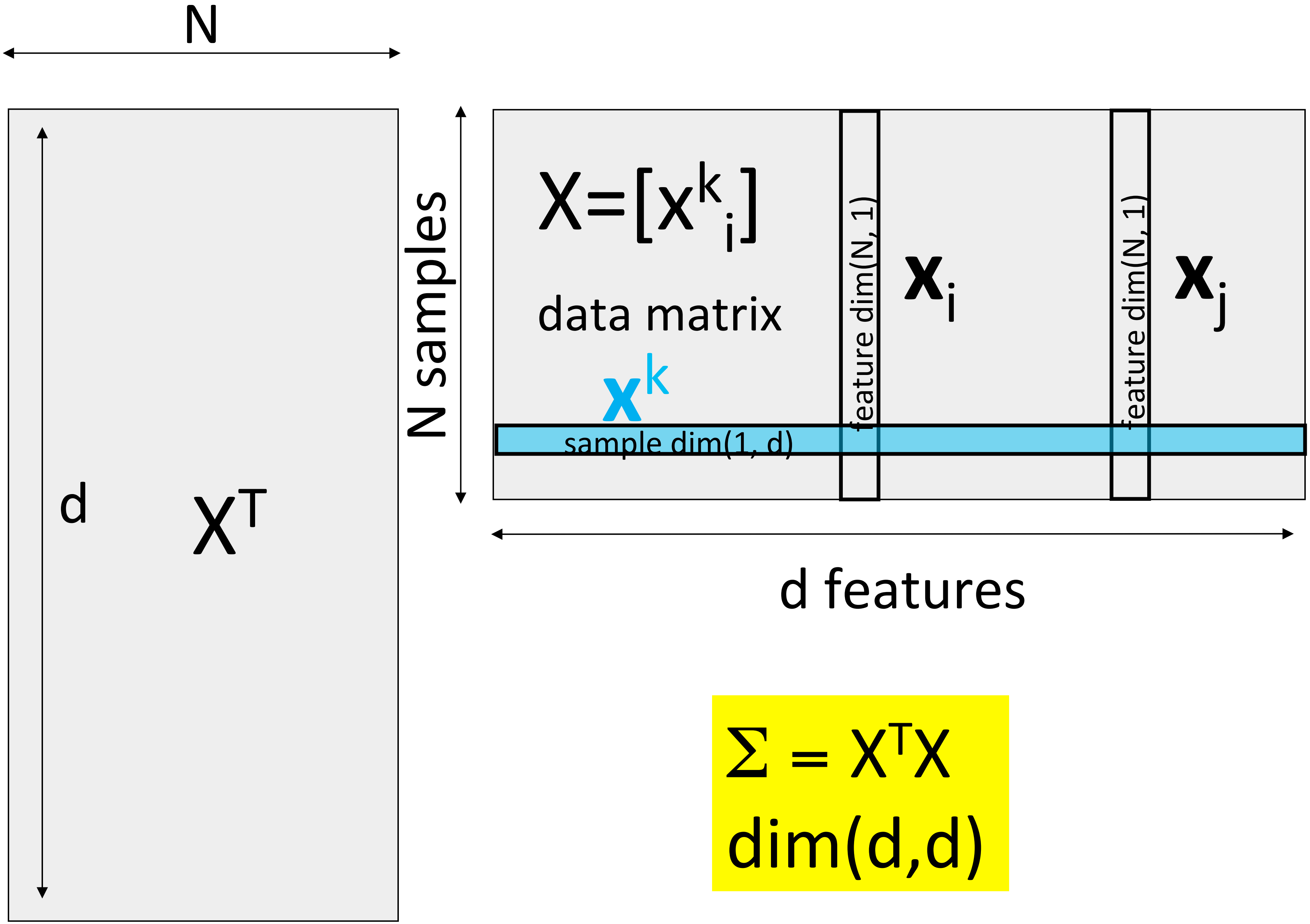
Source: www.intechopen.com

$\Sigma = X^T X$ covariance matrix (centered data) $\sigma_{ij}=(1/N)\Sigma_{k=1:N}\,(x_i^k-\mu_i)(x_j^k-\mu_{xj})$

$\Sigma = US^2U^T$

$U = [\mathbf{u}_i$ eigen vectors], $S = [s_i$ diagonal singular values]

$U^T U = I$



$$\Sigma = X^T X$$

$\dim(d,d)$

Σ^{-1} for BIG data

X^TX dim(d,d)
XX^T dim(N,N)
which one do I rather invert?

Singular value decomposition:

$X=VSU^T$, with $U^TU=1$ and $V^TV=1$

S diagonal dim(r, r): singular values, r = rank(X)≤min(d,N)

$X^TX=US^2U^T$ and $XX^T=VS^2V^T$ dim(U)=(d, r) dim(V)=(N, r)

cheap $\Xi = XU$, $\Sigma^{1/2} = USU^T$, $\Sigma^{-1} = US^{-2}U^T$, etc.

r is small + keep only largest singular values.

Kernel trick:

$d>>N \longrightarrow XX^T = VS^2V^T$ $XX^T \longrightarrow K$ (kernel trick)

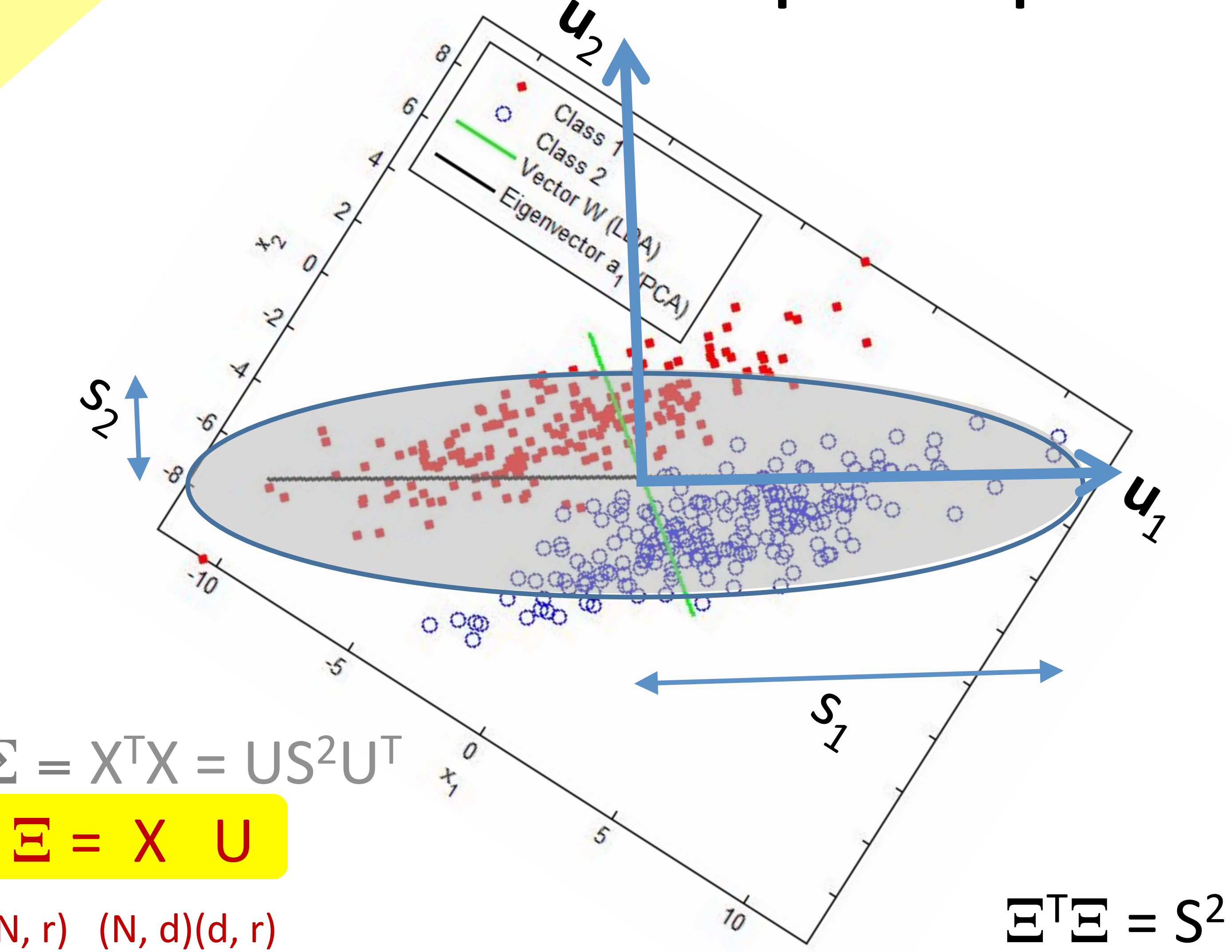
Instead of $\Xi = XU$ use $\Xi = VS$, much cheaper!

$\Xi_{new} = X_{new}\ U$? $X=VSU^T$ so $U = X^TVS^{-1}$

$\Xi_{new} = X_{new}\, X^TVS^{-1}$ $X_{new}\, X^T \longrightarrow [k(x^h, x^k)]$ (kernel trick)

Review

Rotation in principal axes



$\Sigma = X^T X = U S^2 U^T$

$\Xi = X \quad U$

(N, r) (N, d)(d, r)

$\Xi^T \Xi = S^2$

U = [$\mathbf{u}_i$ eigen vectors]

PCA consists in keeping only the axes with largest eigen values

Review

Benefits of rotation in principal axes

$\Sigma = X^T X = U S^2 U^T$

$\Xi = X\,U$ dim(N, r)

$\Xi^T \Xi = S^2$

U = [$\mathbf{u}_i$ eigen vectors]

Review

Benefits of rotation in principal axes

$\Sigma = X^T X = U S^2 U^T$

$\Xi = X\,U$ dim(N, r)

$\Xi^T \Xi = S^2$

U = [$\mathbf{u}_i$ eigen vectors]

- In the new Ξ-space the covariance matrix $\Xi^T\Xi$ is **diagonal**.
- **Inverting** a diagonal matrix (or taking its **square root**) is trivial.

Review

Benefits of rotation in principal axes

$\Sigma = X^T X = US^2U^T$

$\Xi = X\,U$ dim(N, r) $\Xi^T\Xi = S^2$

U = [$\mathbf{u}_i$ eigen vectors]

- In the new Ξ-space the covariance matrix $\Xi^T\Xi$ is **diagonal**.
- **Inverting** a diagonal matrix (or taking its **square root**) is trivial.

<u>Application to ridge regression:</u>

$f(\mathbf{x}) = \mathbf{x}\,\mathbf{w}^T$

$$\underset{(d,1)}{\mathbf{w}^T} = (\underset{(d,N)}{X^T}\underset{(N,d)}{X} + \underset{(d,d)}{\lambda I})^{-1}\underset{(d,N)}{X^T}\,\underset{(N,1)}{\mathbf{y}} \qquad \lambda>0$$

In the rotated space:

$f(\mathbf{x}) = \xi\,\boldsymbol{\omega}^T \quad \xi = \mathbf{x}\,U$

$$S^2+\lambda I = \begin{pmatrix} s_1^2+\lambda & & & \\ & s_2^2+\lambda & & \\ & & \ddots & \\ & & & s_d^2+\lambda \end{pmatrix}$$

$\boldsymbol{\omega}^T = (S^2+\lambda I)^{-1}\,\Xi^T\,\mathbf{y}$

$\mathbf{w}^T = U\,(S^2+\lambda I)^{-1}\,U^T X^T\,\mathbf{y}$

Review

How to transform new samples

$\Sigma = X^T X = U S^2 U^T$

$\Xi = X \ U$ dim(N, r)

(N, r) (N, d)(d, r)

$\Xi^T \Xi = S^2$

U = [$\mathbf{u}_i$ eigen vectors]

Now we get a new sample $\mathbf{x}_{new}$ not in the training set.

$\phi_{new} = \mathbf{x}_{new} U$

Likewise on a test matrix X' of dim(N', d)

$\Xi' = X' \ U$ dim(N', r)

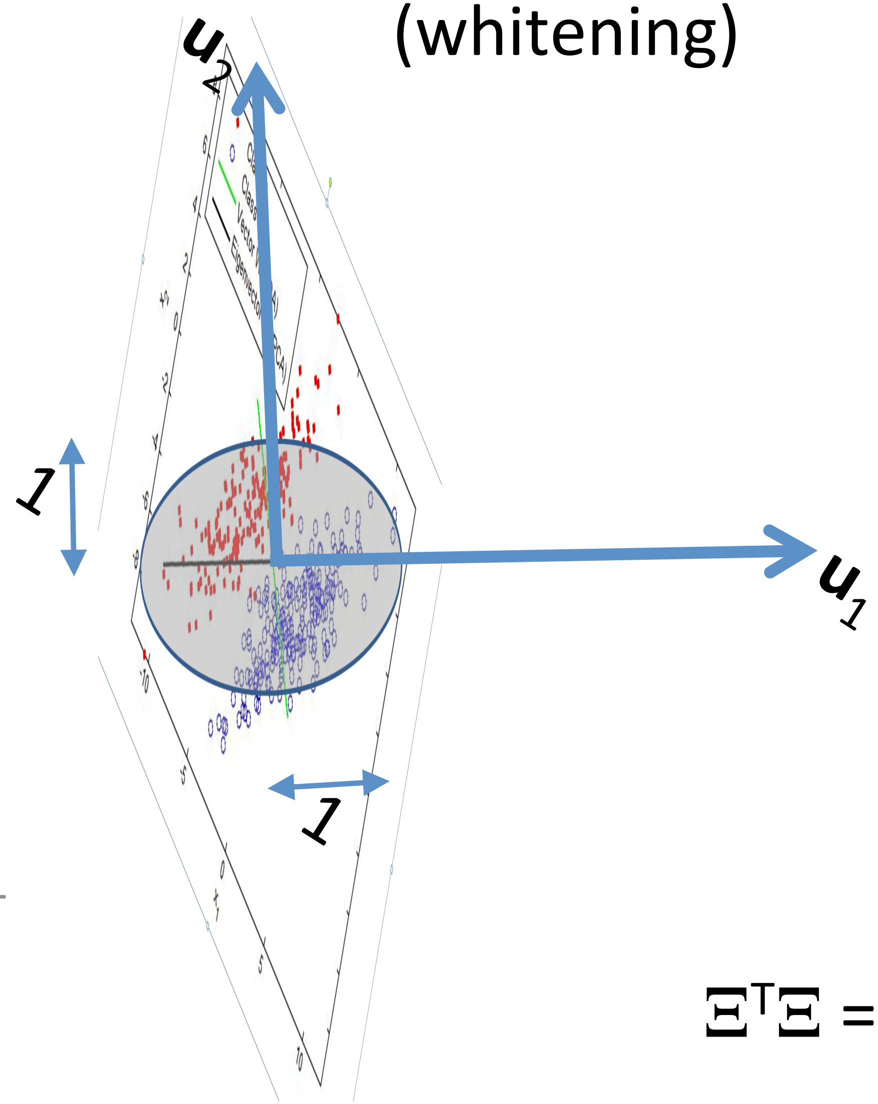
(N', r) (N', d)(d, r)

Do NOT compute U on all the data [X ; X'] and do [X ; X']U.

Review

Rotation and scaling

(whitening)



$$\Sigma = X^T X = U S^2 U^T$$

$$\Xi = X \; \Sigma^{-1/2}$$

(N, r) (N,d)(d, r)

$$\Sigma^{1/2} = U S U^T$$

$$\Xi^T \Xi = I$$

Review

Benefits of whitening

$\Sigma = X^T X = US^2U^T$

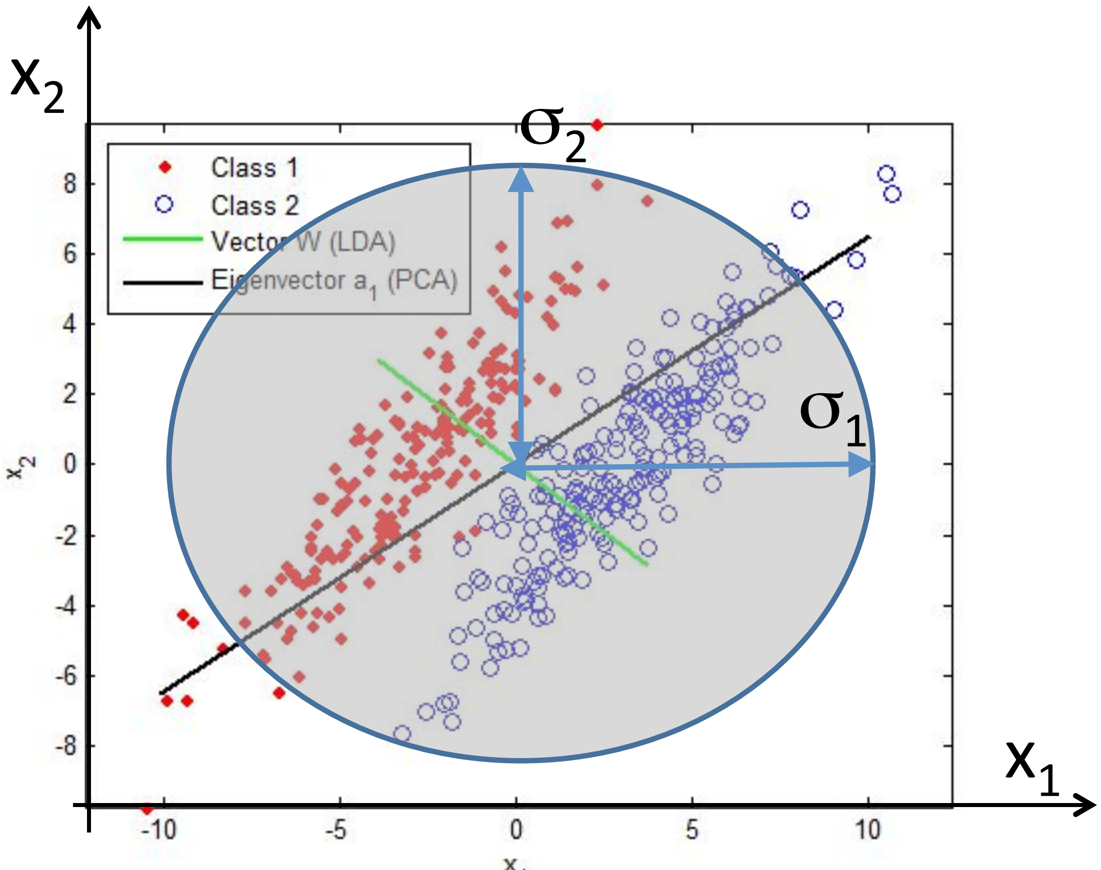
$\Xi = X\,\Sigma^{-1/2}$ dim(N, r) $\Xi^T\Xi = I$

$\Sigma^{1/2} = USU^T$

Same as rotating in principal axes +
The weights of any linear method in Ξ-space
are on the same scale.

Review

Sphering

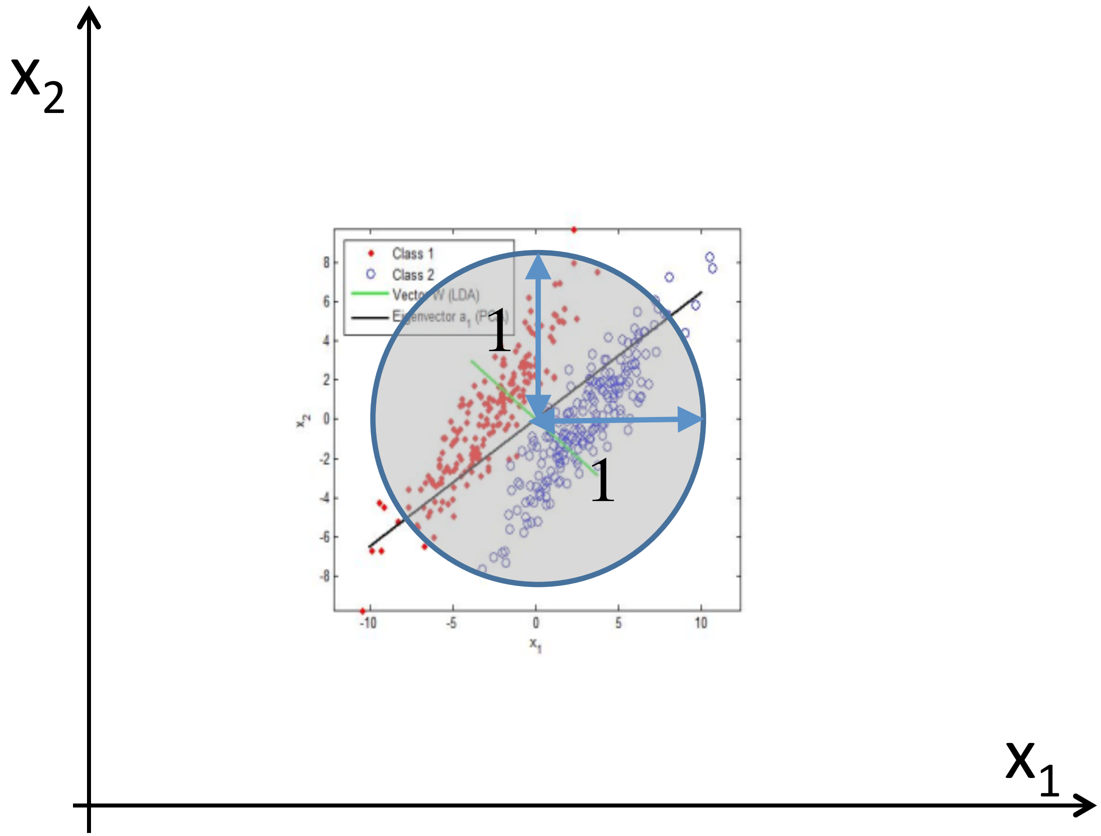


Source: www.intechopen.com

Hypothesize no covariance (even if there is some).
Rescale all axes by the standard deviation or the samples.

Review

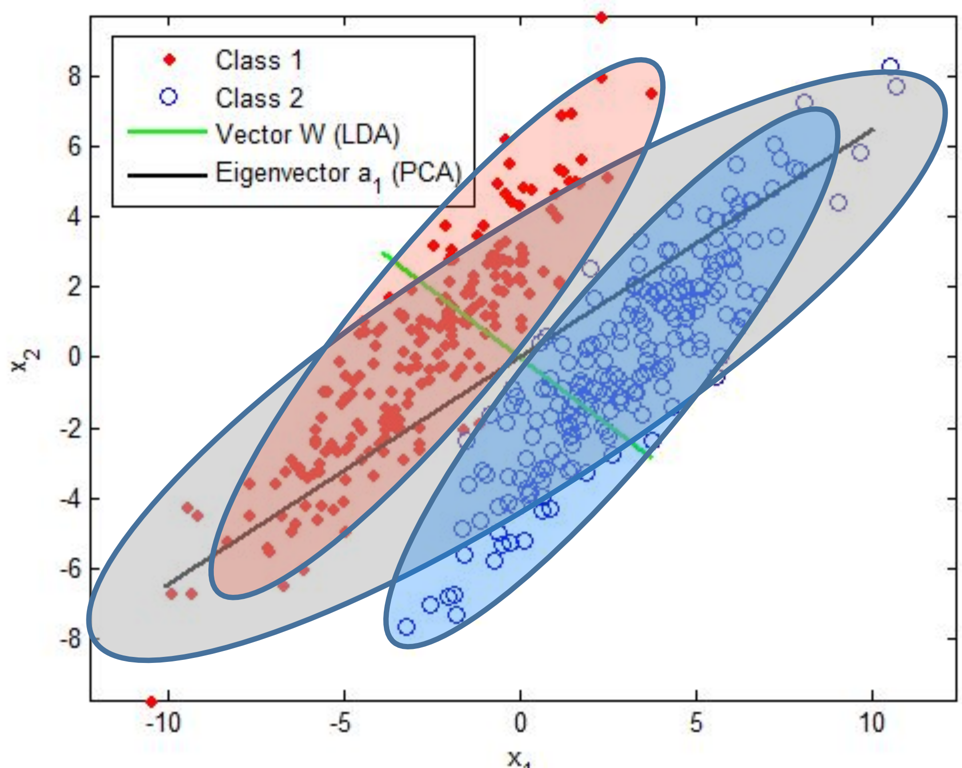
Sphering



Benefit: The weights of any linear method are on the same scale.
But we still have some covariance between features…

Review

LDA and PCA difference



Source: www.intechopen.com

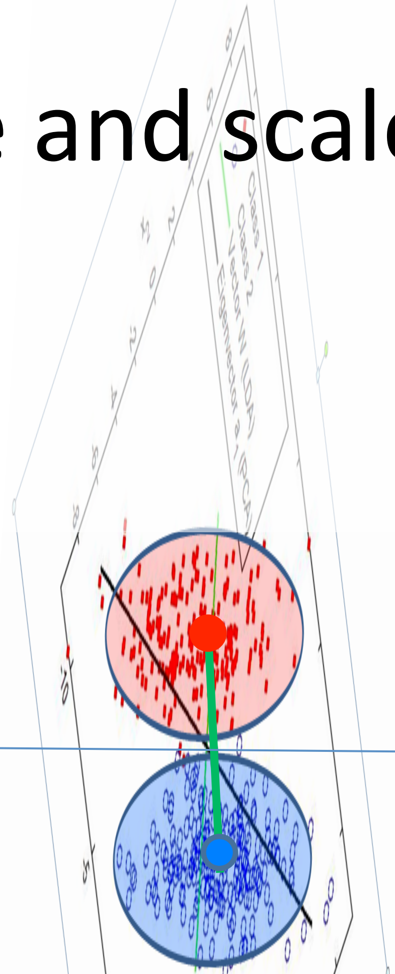
Apply whitening in both case to get $\Xi = X \; \Sigma^{-1/2}$

PCA: Σ is the TOTAL covariance matrix.

LDA: Σ is the POOLED WITHIN CLASS covariance matrix.

Review

Rotate and scale in LDA directions



Now we can use the plain centroid method.

$\Sigma = X^T X = U S^2 U^T$

$\Xi = X\, \Sigma^{-1/2}$ $\qquad \Xi^T \Xi = I$

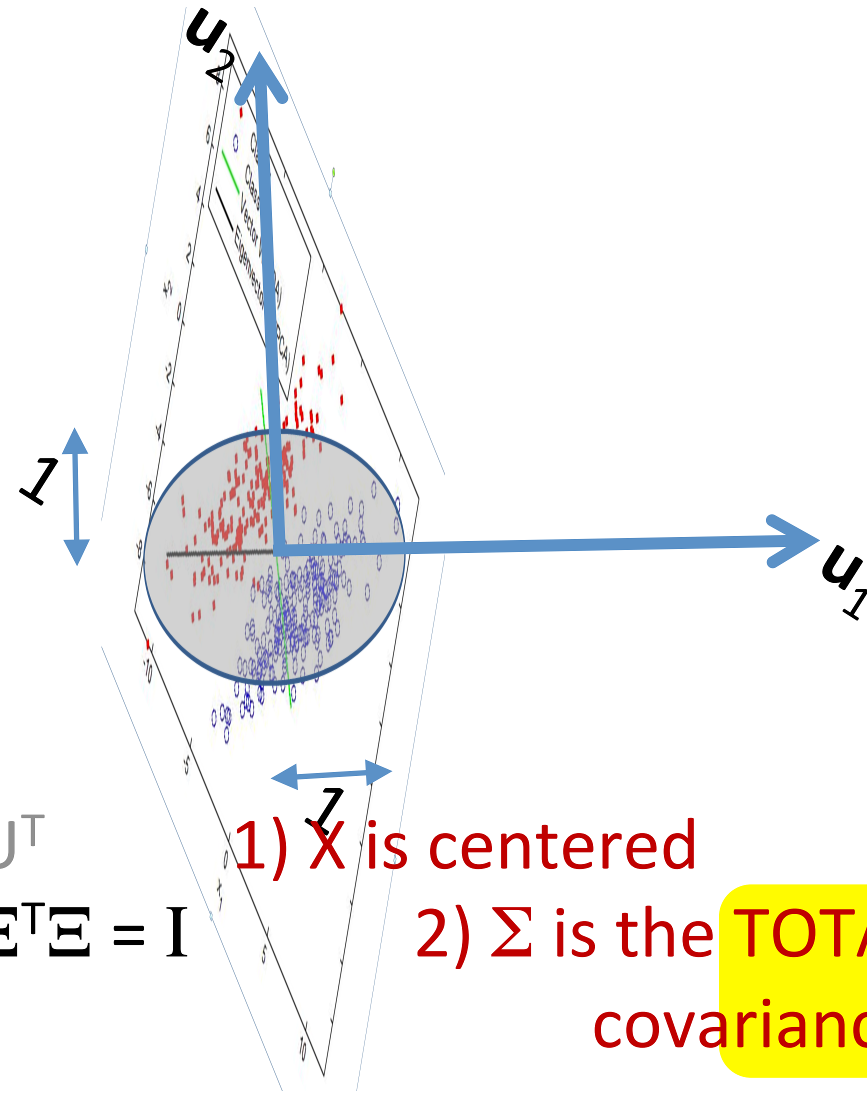
$\Sigma^{1/2} = U S U^T$

1) X is centered

2) Σ is the POOLED within class covariance

Review

Rotate and scale in PCA directions



$\Sigma = X^T X = U S^2 U^T$

$\Xi = X\, \Sigma^{-1/2}$ $\quad \Xi^T \Xi = I$

$\Sigma^{1/2} = U S U^T$

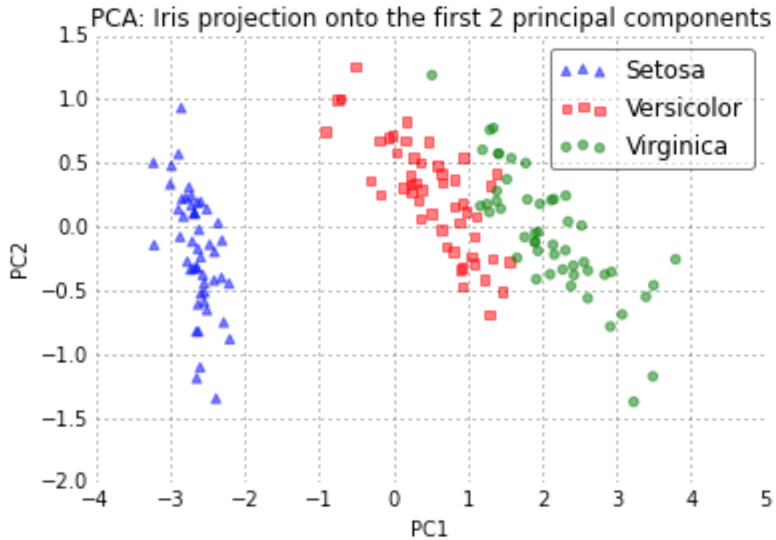
1) X is centered

2) Σ is the TOTAL overall covariance

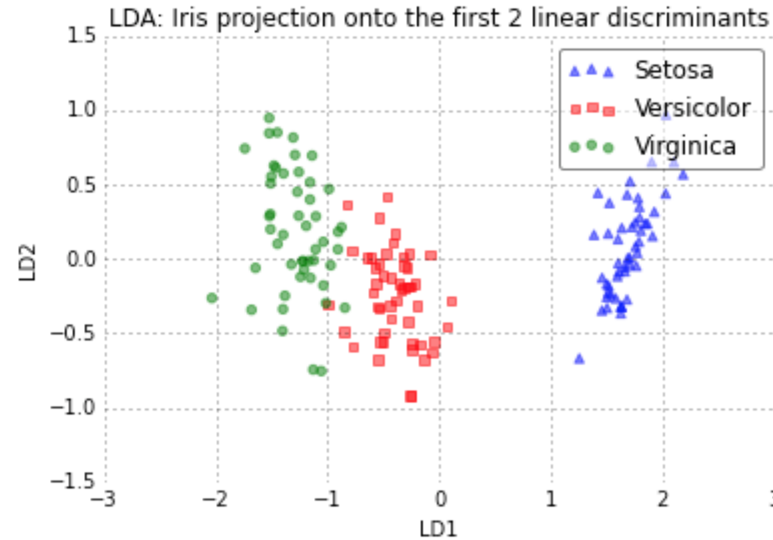
Review

LDA and PCA visualization

PCA



LDA



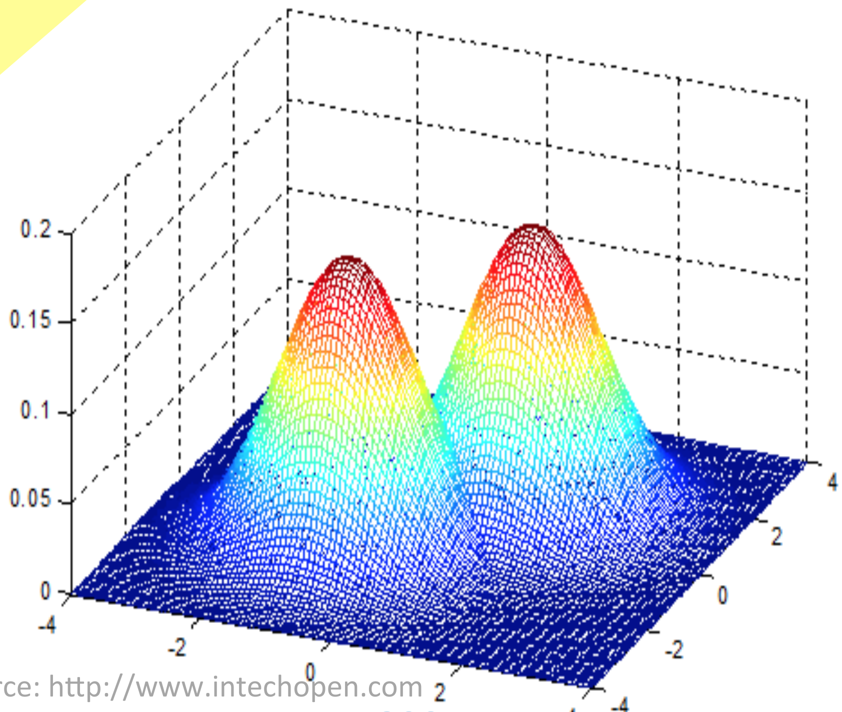
http://sebastianraschka.com/Images_old/2014_intro_supervised_learning/iris_pca_lda.png

Review

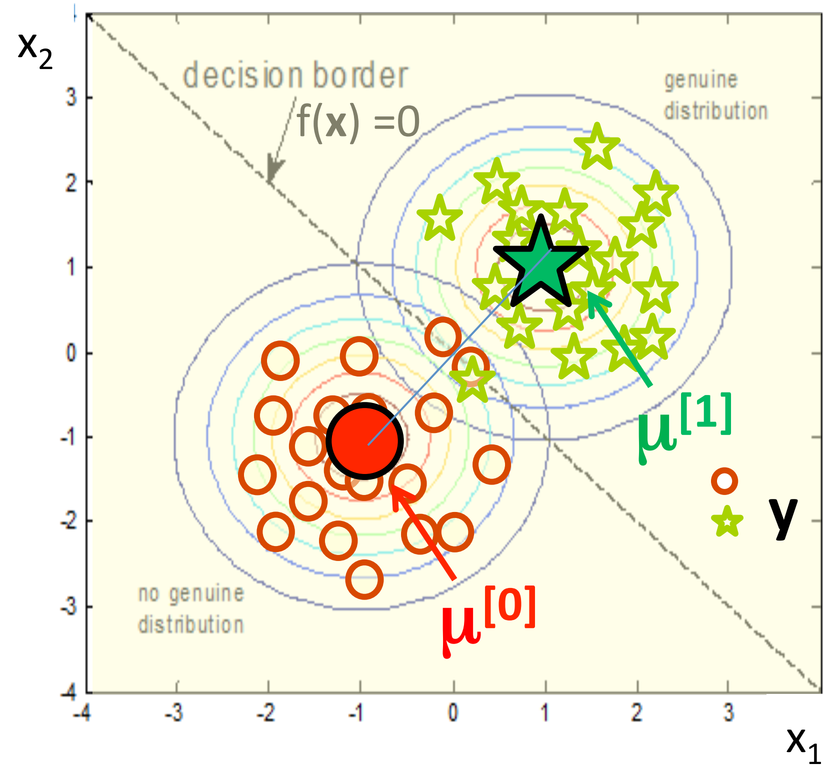
Gaussian classifier

$$\hat{y} = \operatorname{argmax}_y P(Y=y \mid X=\mathbf{x}) \sim P(Y=y)\, P(X=\mathbf{x} \mid Y=y)$$

$$\sim P(Y=y) \exp(-\|\mathbf{x}-\mu^{[y]}\|^2/2\sigma^2)$$



Source: http://www.intechopen.com



$$f(\mathbf{x}) = \underbrace{(\mu^{[1]} - \mu^{[0]})}_{\mathbf{w}} \bullet \mathbf{x} + b \qquad b = (\mu^{[0]2} - \mu^{[1]2})/2 + \log(N_1/N_0)$$

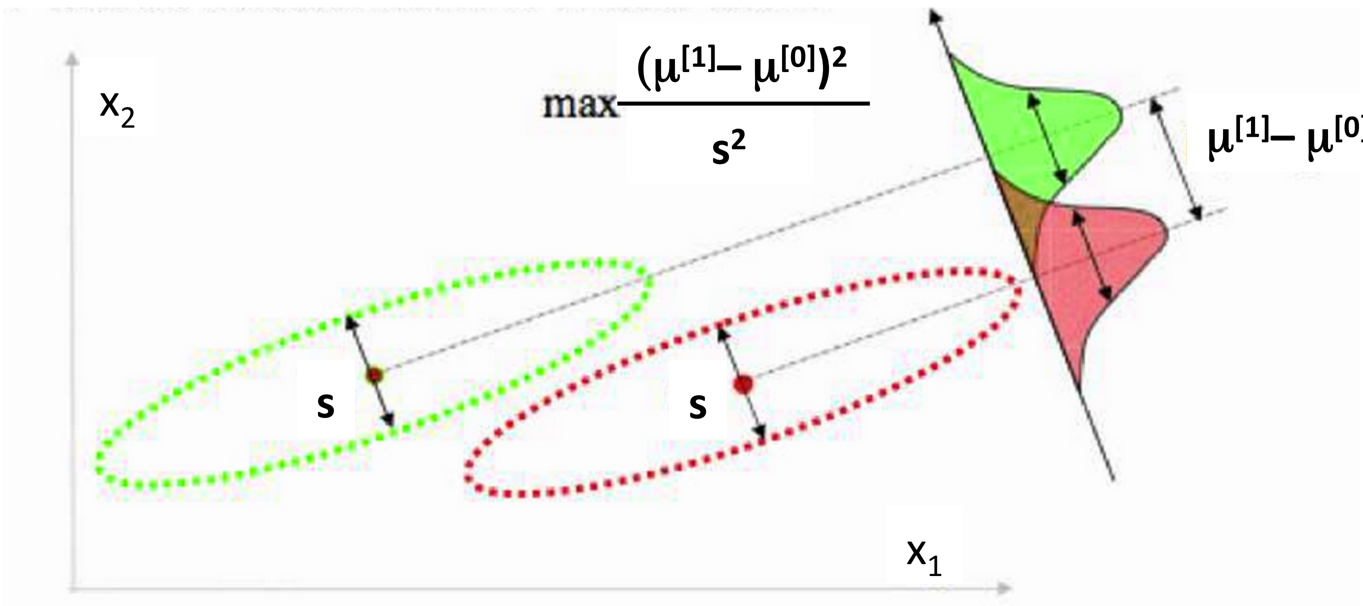
$$f(\mathbf{x}) = (\mu^{[1]}./\sigma^{[1]2} - \mu^{[0]}./\sigma^{[0]2}) \bullet \mathbf{x} + b \qquad b = [(\mu^{[0]}./\sigma^{[0]})^2 - (\mu^{[1]}./\sigma^{[1]})^2)/2 + \log(N_1/N_0)$$

$$f(\mathbf{x}) = \Sigma^{-1}(\mu^{[1]} - \mu^{[0]}) \bullet \mathbf{x} + b \qquad b = [(\mu^{[0]T}\Sigma^{-1}\mu^{[0]} - \mu^{[1]T}\Sigma^{-1}\mu^{[1]})/2 + \log(N_1/N_0)$$

Review

LDA is a "glorified" Gaussian classifier

$$P(X=\mathbf{x}\,|\,Y=y) \sim \exp\left(-\frac{1}{2}\underbrace{(\mathbf{x}-\boldsymbol{\mu}^{[y]})}_{(1,\,d)}\underbrace{\boldsymbol{\Sigma}^{-1}}_{(d,d)}\underbrace{(\mathbf{x}-\boldsymbol{\mu}^{[y]})^{T}}_{(d,\,1)}\right)$$



Source: Victor Lavrenko

$$f(\mathbf{x}) = \overbrace{\Sigma^{-1}(\mu^{[1]}-\mu^{[0]})}^{\mathbf{w}} \bullet \mathbf{x} + b$$

$$b = [(\mu^{[0]T}\Sigma^{-1}\mu^{[0]} - \mu^{[1]T}\Sigma^{-1}\mu^{[1]})/2+\log(N_1/N_0)$$

Which assumption is right for you?

$$P(X, Y) = P(X)P(Y|X) = P(Y)P(X|Y)$$

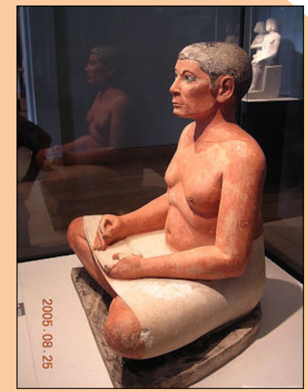
P(X)

P(Y|X)

96 98 78 6 63 79…

P(Y)

5

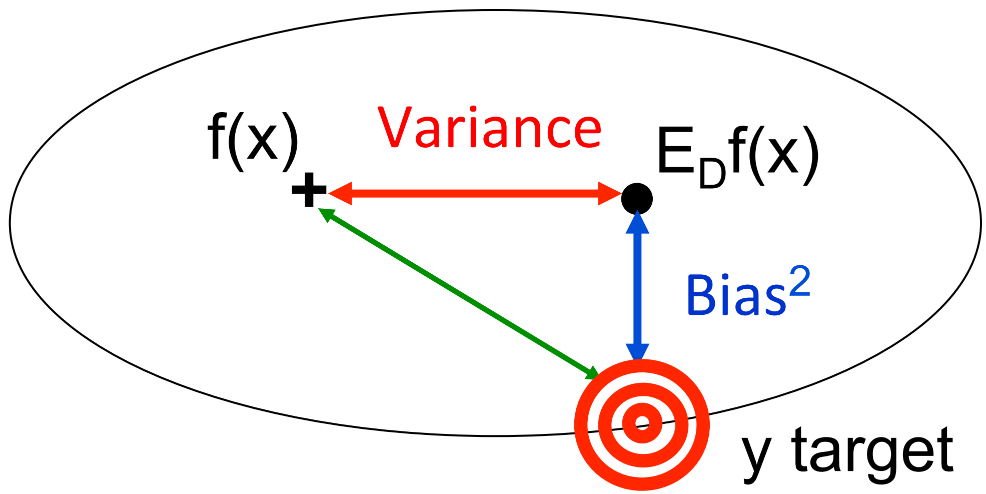


P(X|Y)

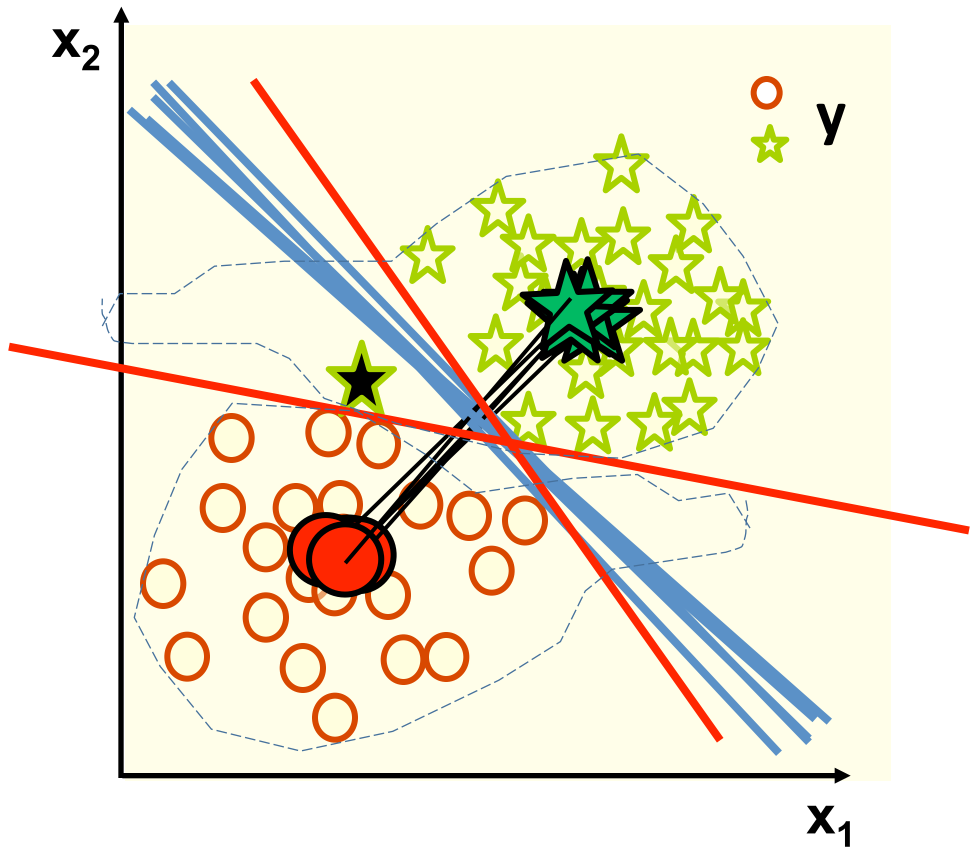
It does not really matter!
Bias *vs.* variance tradeoff

- f trained on a training set D of size m (m fixed)
- For the square loss:

$$\underbrace{E_D[f(\mathbf{x})-y]^2}_{\substack{\text{Expected value}\\ \text{of the loss over}\\ \text{datasets D of}\\ \text{the same size}}} = \underbrace{[E_D f(\mathbf{x})-y]^2}_{\text{Bias}^2} + \underbrace{E_D[f(\mathbf{x})-E_D f(\mathbf{x})]^2}_{\text{Variance}}$$

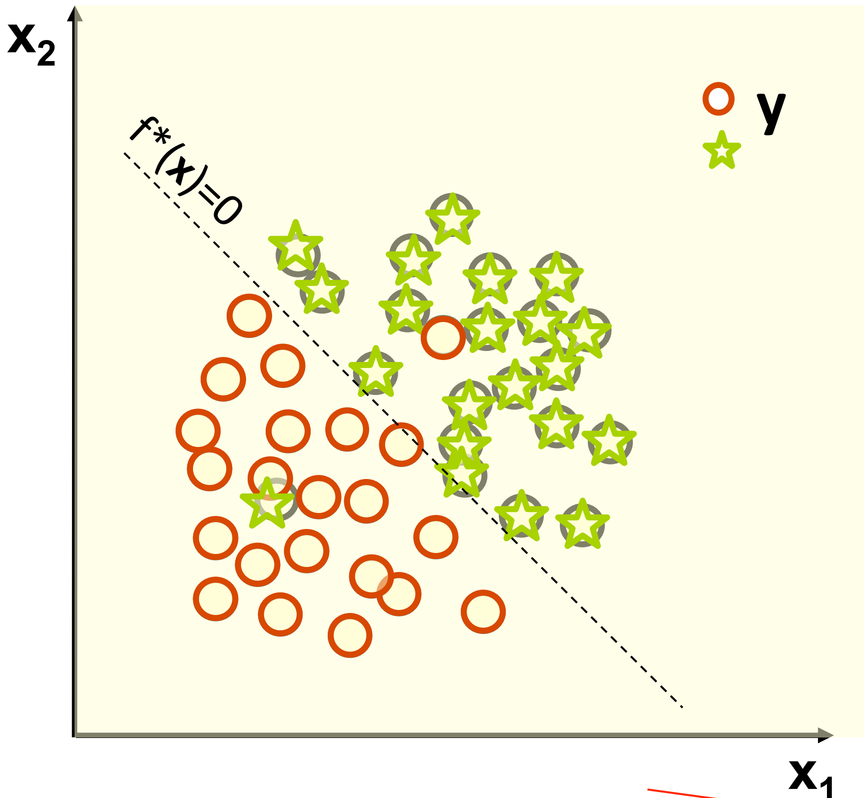


Violated assumption (1): Gaussianity
Centroid *vs.* SVM



Noisy data or lack of training data: Even if the class clusters have a tailed distribution, you might be better off making Gaussian assumptions.

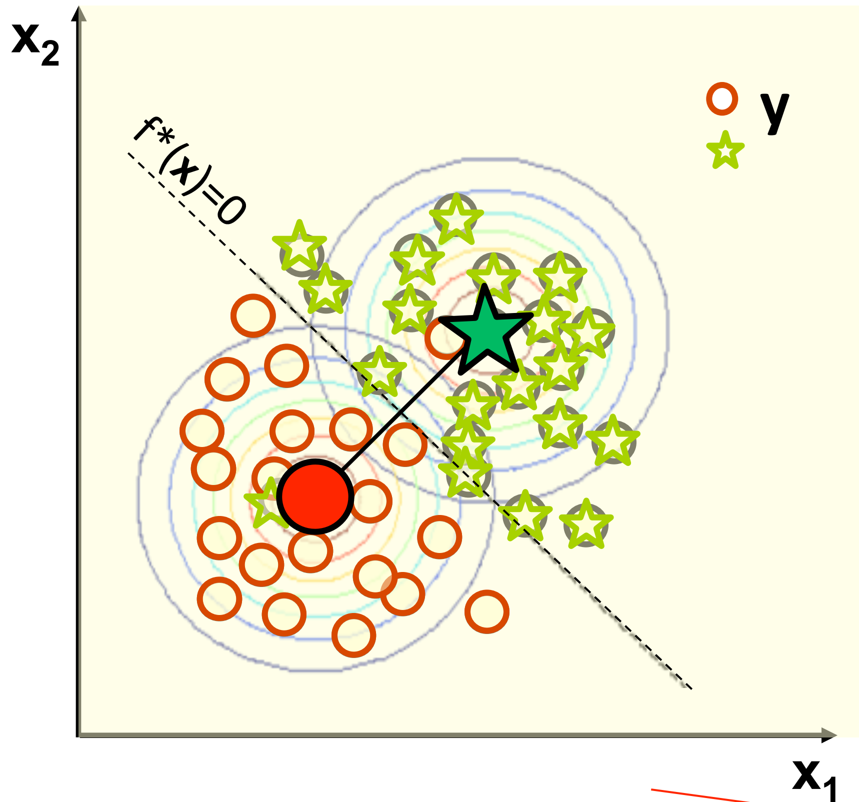
Violated assumption (2): wrong data generating "direction"



"Wrong" data generating assumption: ~~P(Y=y) then P(X=**x**|Y=y)~~

In reality: P(X=**x**) then P(Y=y|X=**x**)=1 if f*(**x**)>0, or 0 if f*(**x**)<0.

Violated assumption (2): P(X=**x**|Y=y) not Gaussian



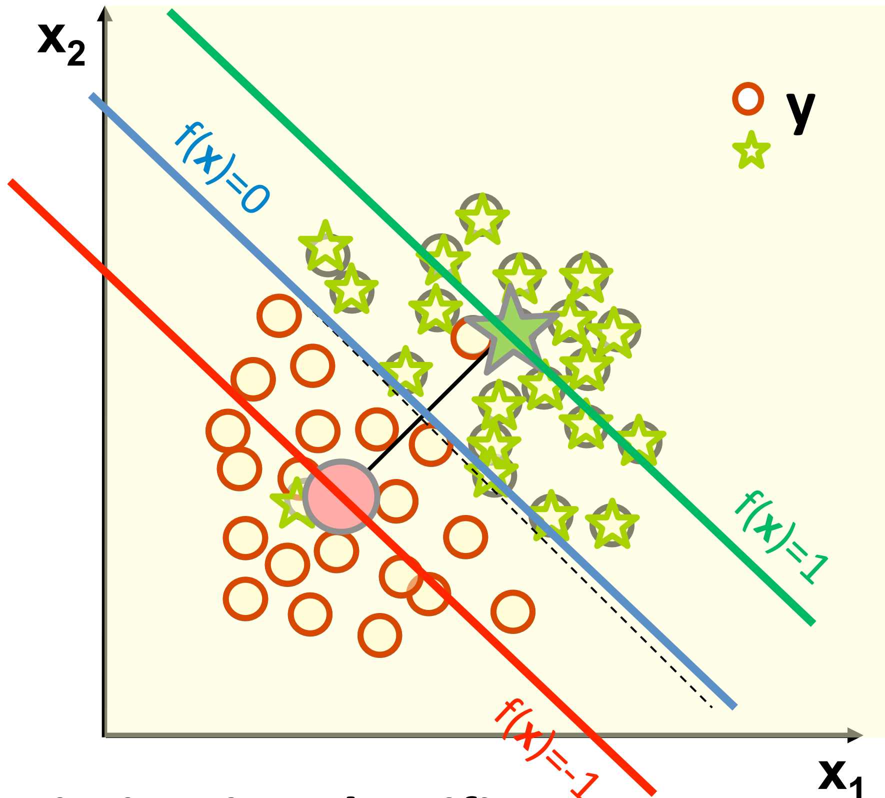
"Wrong" data generating assumption: ~~P(Y=y) then P(X=**x**|Y=y)~~

In reality: P(X=**x**) then P(Y=y|X=**x**)=1 if f*(**x**)>0, or 0 if f*(**x**)<0.

The Gaussian model assumptions are violated.

Would the centroid method still be OK?

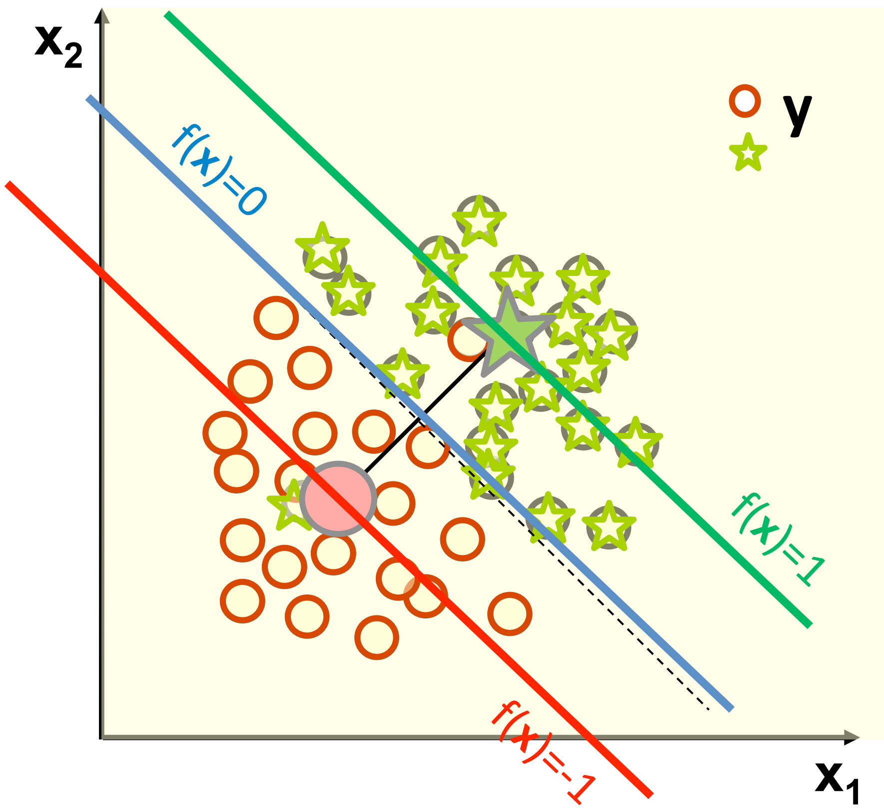
Let's compare with the regression solution



Regression = **discriminative classifier**:

- "Correct" data generating assumption: P(X=**x**) then P(Y=y|X=**x**)
- but "wrong" loss function (f(**x**) – y)^2 for classification? (y=±1)

Explanation: Regression for classification ≡ LDA / Fisher linear discriminant



LDA:

$\mathbf{w} = \Sigma^{-1}(\mu^{[1]} - \mu^{[0]})$

RIDGE REGRESSION:

$\mathbf{w} = \Sigma^{-1} X^T \mathbf{y}$

Regularized Σ^{-1}

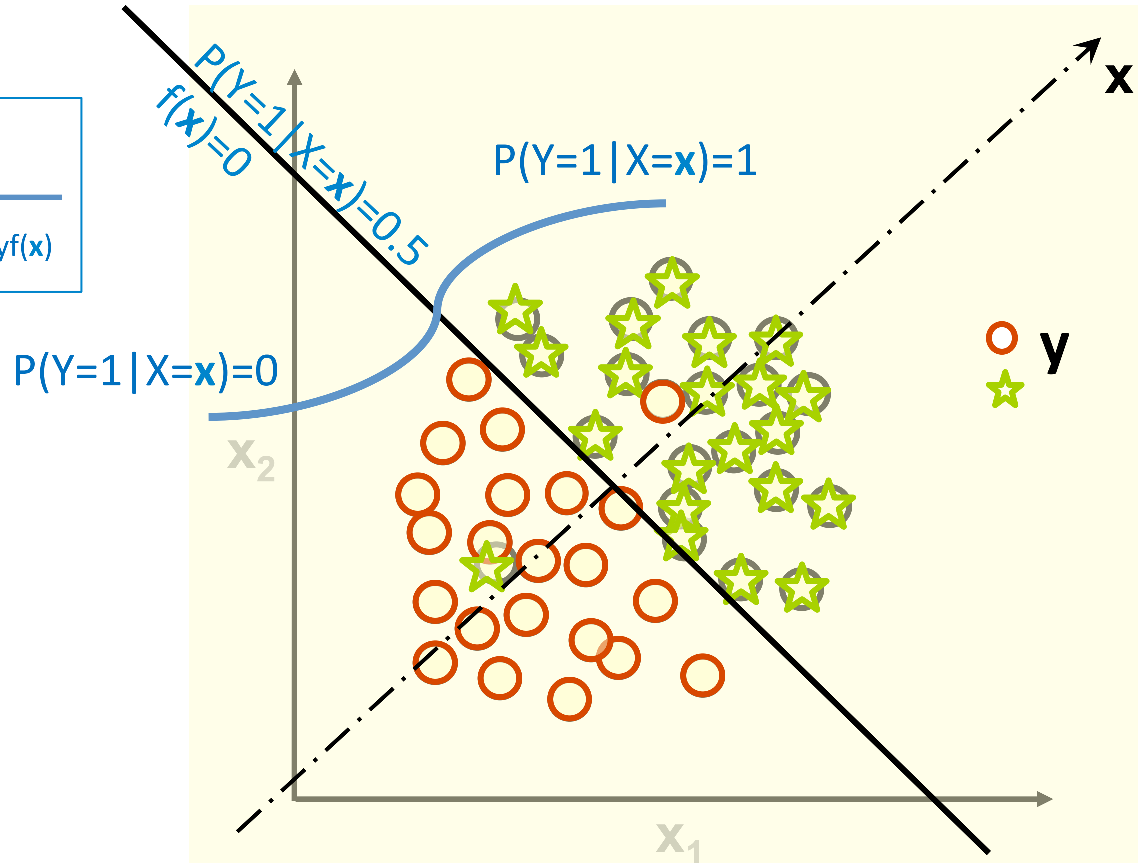
$\Sigma^{-1} = (X^T X + \lambda I)^{-1}$

If:

- covariance matrix ≡ pooled within class covariance, and
- $y \in \{+1/N_1, -1/N_0\}$.

What about logistic regression?

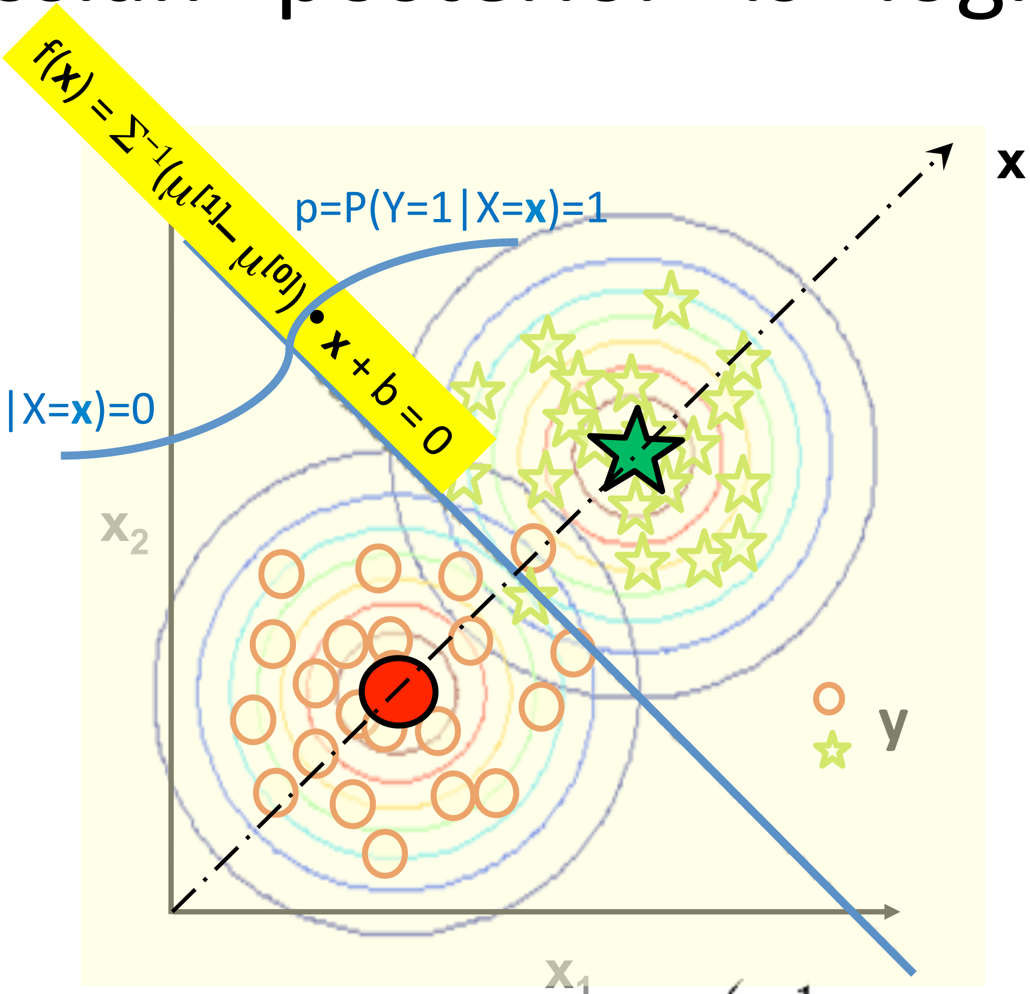
$$P(Y=y|X=\mathbf{x}) = \frac{1}{1+e^{-yf(\mathbf{x})}}$$



Logistic regression = **discriminative classifier**:

- "Correct" data generating assumption: $P(X=\mathbf{x})$ then $P(Y=y|X=\mathbf{x})$
- "Correct" loss function $-\log P(Y=y|X=\mathbf{x}) = \log(1+e^{-yf(\mathbf{x})})$ $(y=\pm 1)$

The Gaussian "posterior" is "logistic"



The sigmoid comes from the logistic link:

$$\log \frac{P(Y=1|X=\mathbf{x})}{P(Y=-1|X=\mathbf{x})} = f(x)$$

$\log((p/(1-p)) = f(x)$

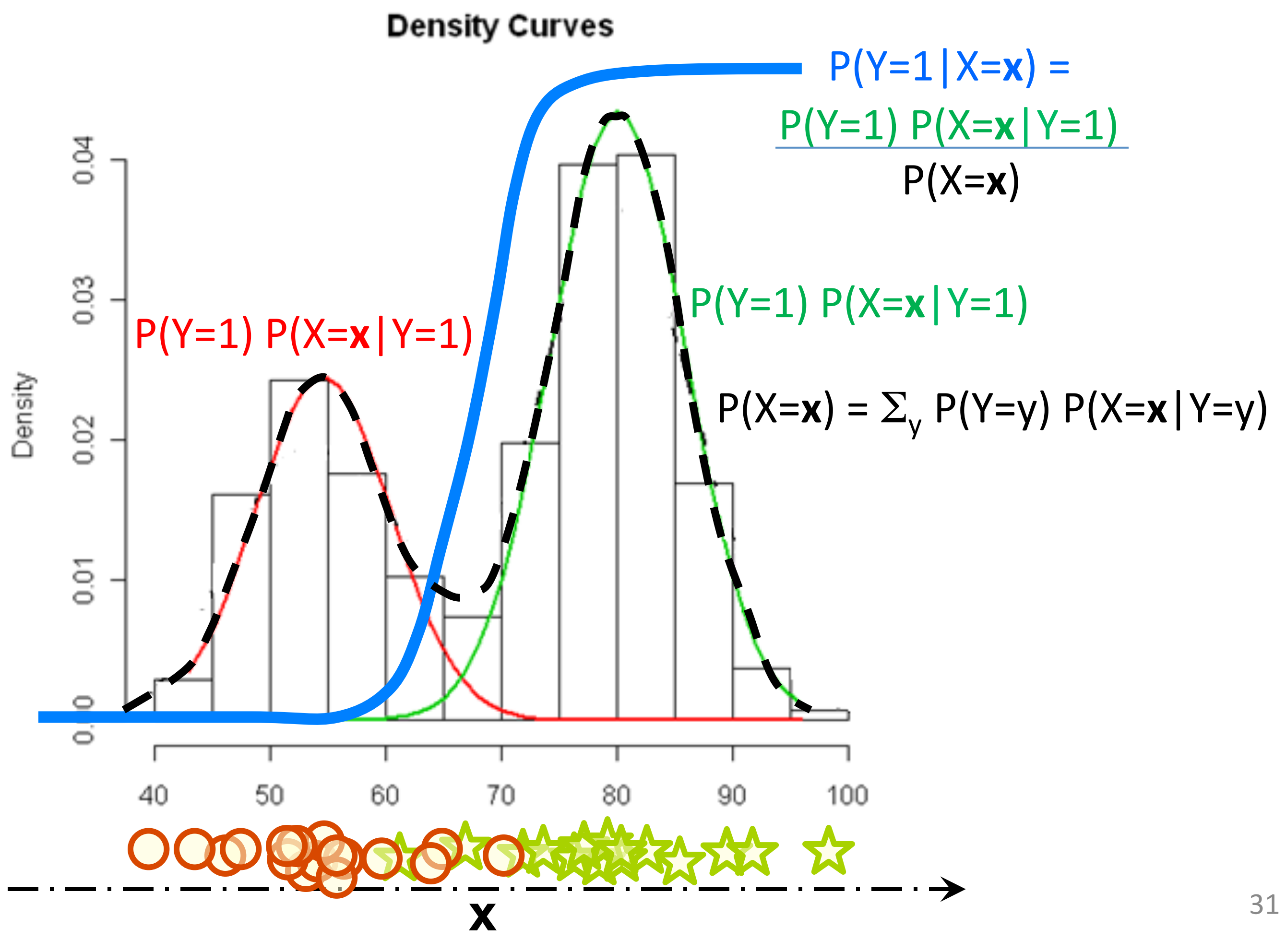
$p = 1/(1+ e^{-f(\mathbf{x})})$

Generative model: P(Y=y) then P(X=**x**|Y=y)~ $\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}^{[y]})^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}^{[y]})\right)$

We have seen that the log odds-ratio is a linear model.

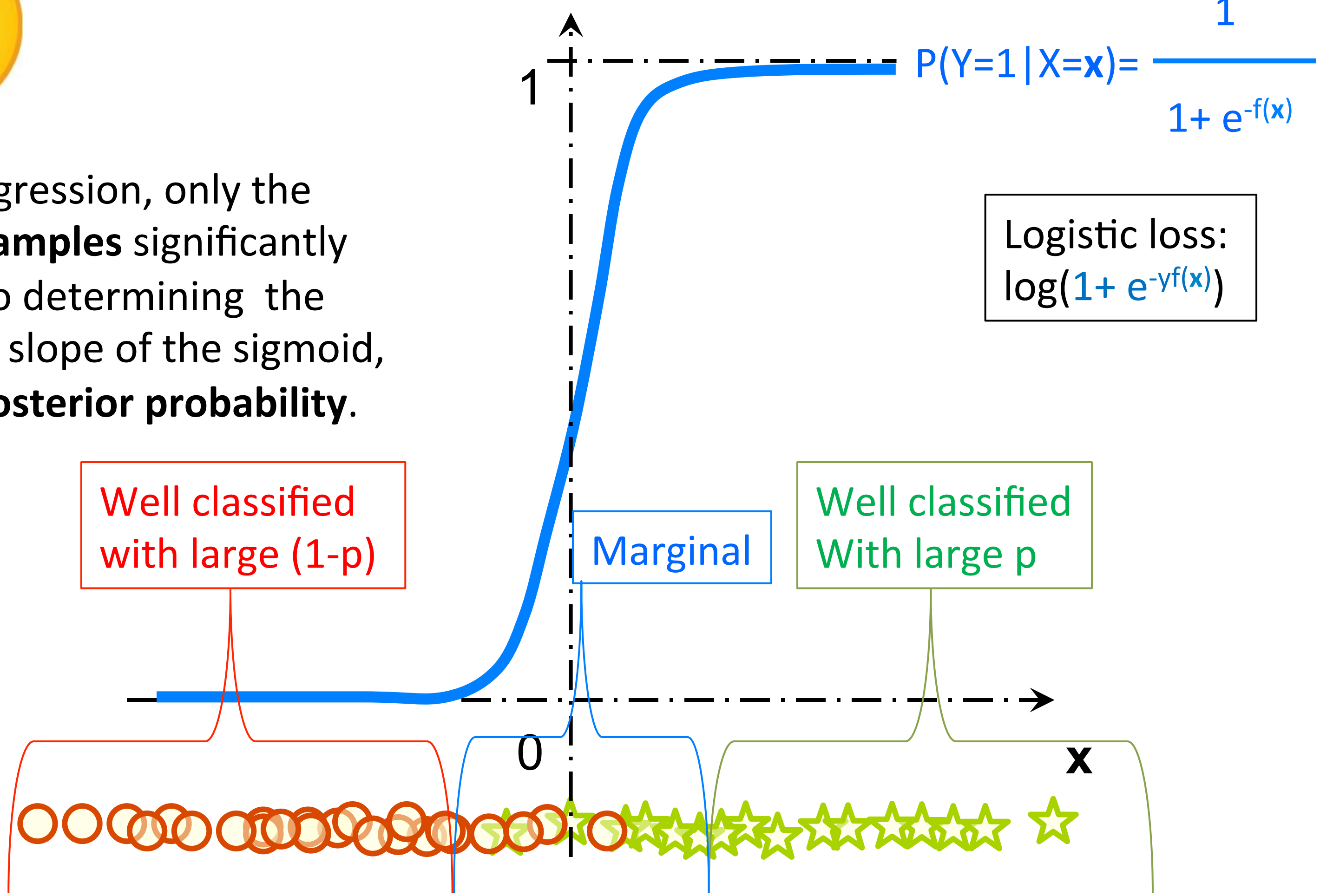
So... The posterior P(Y=1|X=**x**) of the Gaussian model is logistic.

The posterior $P(Y=1|X=\mathbf{x})$ of the Gaussian model is logistic



But the logistic model does not make any Gaussian assumption!

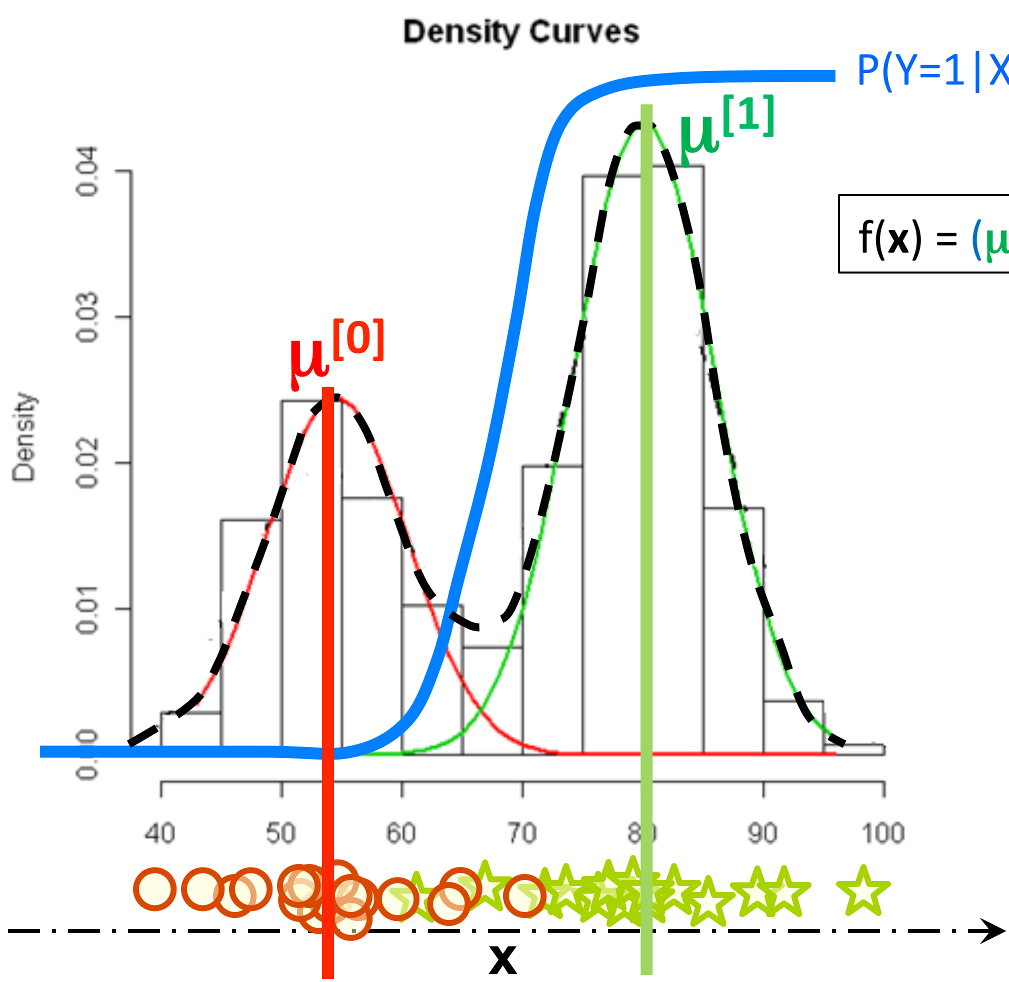
In logistic regression, only the **marginal examples** significantly contribute to determining the position and slope of the sigmoid, hence the **posterior probability**.



The Gaussian model posterior estimation uses the examples

$$P(Y=1|X=\mathbf{x}) = \frac{1}{1+e^{-f(\mathbf{x})}}$$

$$\boxed{f(\mathbf{x}) = (\mu^{[1]} - \mu^{[0]}) \bullet \mathbf{x} + b}$$



Do we want to bother with LDA?

My own cookbook:

1) Preprocessing:

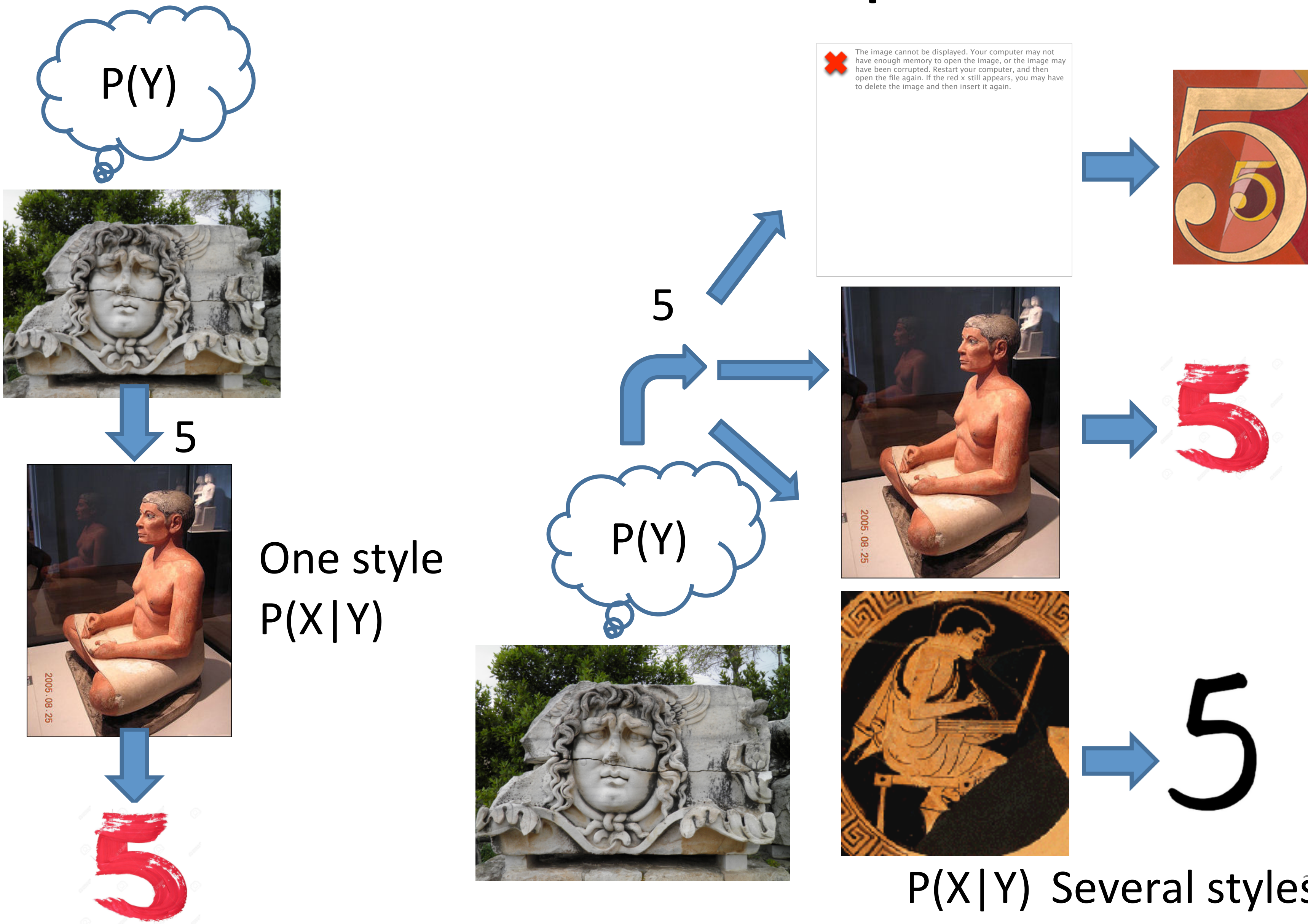
- Pattern normalization: $\mathbf{x}^k \leftarrow \mathbf{x}^k / \|\mathbf{x}^k\|$ (optional)
- Feature centering: $\mathbf{x}_i \leftarrow \mathbf{x}_i - \boldsymbol{\mu}_i$
- Singular value decomposition: $X=VSU^T$
  $U^TU=1$, $V^TV=1$, S diagonal dim(r, r), r =rank(X)≤min(d,N)
- Visualize data is 2 dim (top 2 s values) $\Xi = X \quad [\mathbf{u}^1, \mathbf{u}^2]$
  - Further preprocess (e.g. non-linear scaling) (N, d) (d,2)
  - Eventually replace $\boldsymbol{\Sigma}$ by pooled $\boldsymbol{\Sigma}$
  - Reduce space dimension with PCA or LDA $\Xi = X\, [\mathbf{u}^1, \mathbf{u}^2,..\, \mathbf{u}^{max}]$

2) Linear methods:

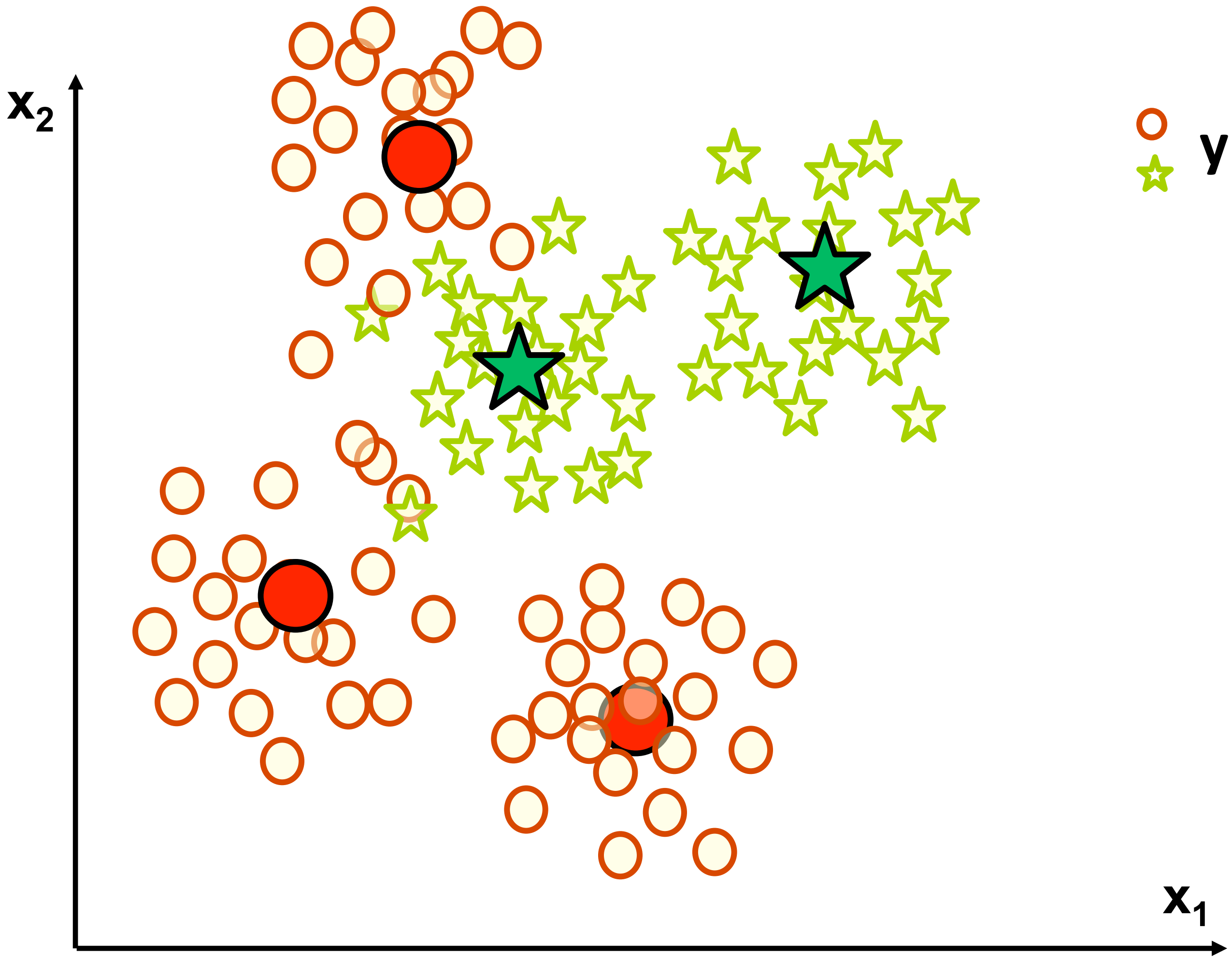
1. Centroid (with targets $y \in \{+1/N_1, -1/N_0\}$ or $y=\pm 1$)
2. Soft margin SVM (~ ridge regression for large λ)
3. Post-fit SVM output to sigmoid (skip logistic regression)

3) Non-linear methods

Double random process



Multiple clusters per class



Mixture models

$$\overbrace{P(Y=1|X=\mathbf{x}) \sim}$$

$$\underbrace{P(X=\mathbf{x}|Y=y)}_{\text{likelihood}}\ \underbrace{P(Y=y)}_{\text{prior}} = \underbrace{\Sigma_k}_{\text{sum over different styles}} P(X=\mathbf{x}, S=s_k|Y=y)\ P(Y=y)$$

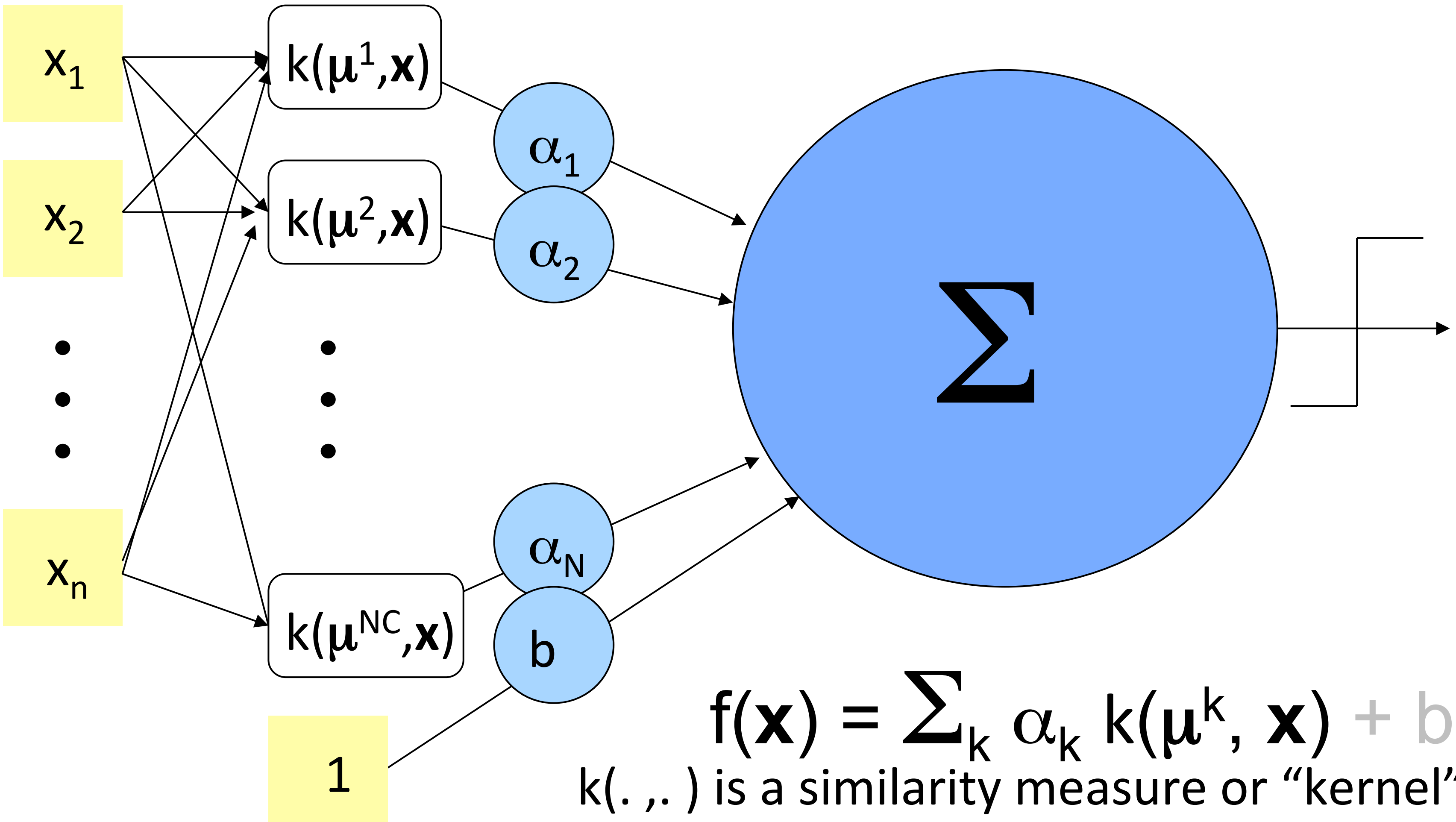
$$= \Sigma_k \underbrace{P(X=\mathbf{x}|S=s_k, Y=y)}_{\sim \exp(-\|\mathbf{x}-\mu_k\|^2/2\sigma^2)}\ \underbrace{P(S=s_k|Y=y)}_{\sim \alpha_k}\ P(Y=y)$$

$$f(\mathbf{x}) = P(Y=1|X=\mathbf{x}) - P(Y=-1|X=\mathbf{x})$$

$$\sim \Sigma_{k=1:Nc}\, \alpha_k \exp(-\|\mathbf{x}-\mu_k\|^2/2\sigma^2) + b$$

RBF network

RBF = radial basis function



$$f(\mathbf{x}) = \sum_k \alpha_k\, k(\boldsymbol{\mu}^k, \mathbf{x}) + b$$

k(. ,.) is a similarity measure or “kernel”.

Parameter estimation

- $f(\mathbf{x}) = \sum_{k=1:Nc} \alpha_k \, k(\mu^k, \mathbf{x}) + b$

  $k(\mu^k, \mathbf{x}) = \exp(-\|\mathbf{x}-\mu_k\|^2/2\sigma^2)$ Gaussian kernel

- **Parameters:**
  - Nc = number of clusters
  - μ_k = cluster centers
  - α_k = cluster weights
  - σ = kernel width

- **Simple way:**
  - Fix σ.
  - Fix Nc and take Nc/2 in each class.
  - In each class run k-means clustering to get the cluster centers μ_k.
  - Treat $f(\mathbf{x})$ as a model linear in its parameters and fit the α_k by gradient descent.
  - Optimize σ and Nc by cross-validation.

Or, yet simpler: Parzen windows

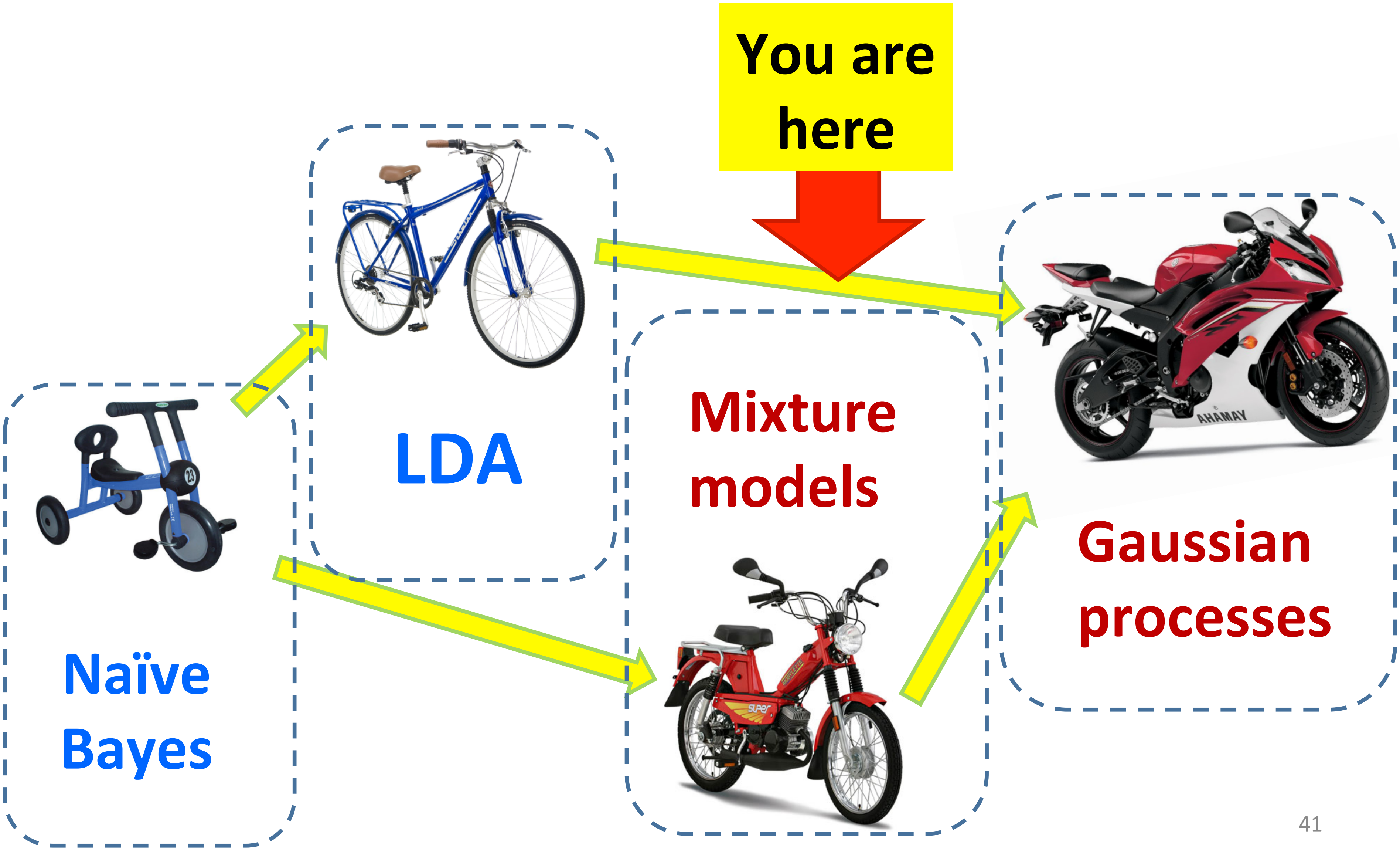
- $f(\mathbf{x}) = \Sigma_{k=1:N}\, y_k\, k(\mathbf{x}^k, \mathbf{x}) + b$

$$k(\mathbf{x}^k, \mathbf{x}) = \exp(-\|\mathbf{x}-\mathbf{x}_k\|^2/2\sigma^2)$$ Gaussian kernel

- **Parameters:**
  - ~~Nc = number of clusters~~
  - ~~μ_k = cluster centers~~
  - ~~α_k = cluster weights~~
  - σ = kernel width

- **Simpler way:**
  - There is one "cluster" per example (Nc = N, $\boldsymbol{\mu}_k=\mathbf{x}^k$).
  - The α_k are also fixed: $\alpha_k = y_k$
  - Optimize σ by cross-validation.

Generative models



Summary

- We are trying to make sense of the models we are working with from a **data generating process** point of view. Useful for gaining insight, and potentially gaining performance advantages (by injecting "prior knowledge"):
  - **Visualization** allows us to evaluate our hypotheses.
  - Even if the hypotheses are wrong, we may get good results if the **data are noisy** or if we have **few training samples**, because of the **"bias/variance" tradeoff**.
  - SVM and logistic regression do not make data generating assumptions. But, ridge regression applied to classification problems is similar to LDA.
- **Gaussian mixtures:** What if the single cluster per class hypothesis is violated? We can introduce multiple clusters per class; this is the idea of Gaussian mixtures and several "Radial Basis Function" (RBF) algorithms. This bridges the gap with kernel methods.

Come to my office hours…

Wed 1:30-3:30 Soda 329

Next time: Gaussian processes