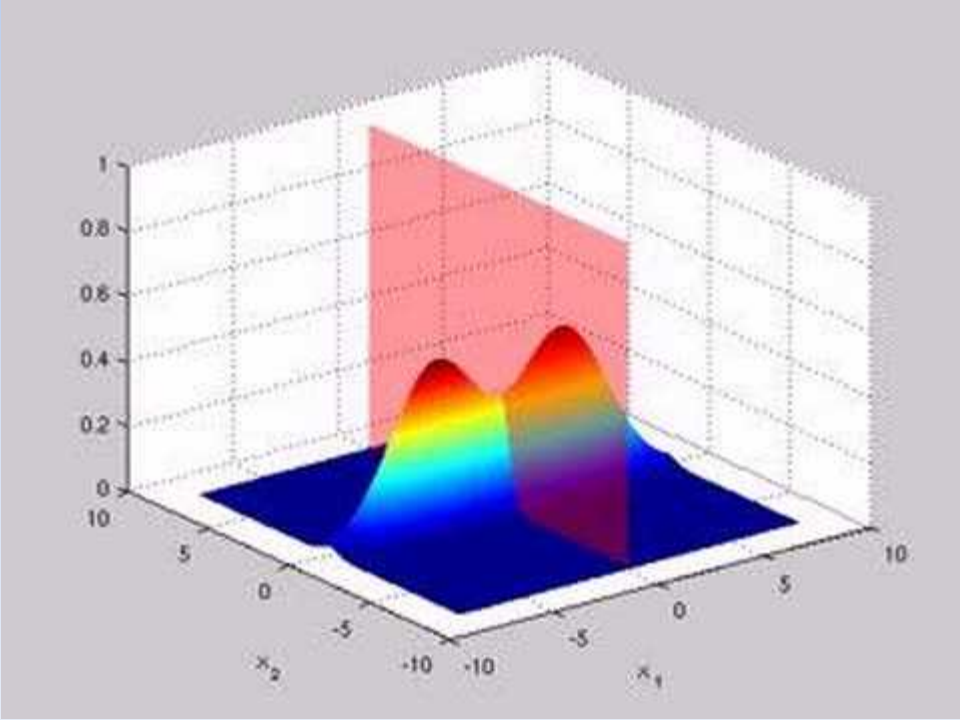
UCB - CS189
Introduction to Machine Learning
Fall 2015
Lecture 13: Linear Discriminant Analysis (LDA) and QDA

Isabelle Guyon

ChaLearn

Come to my office hours…
Wed 2:30-4:30 Soda 329

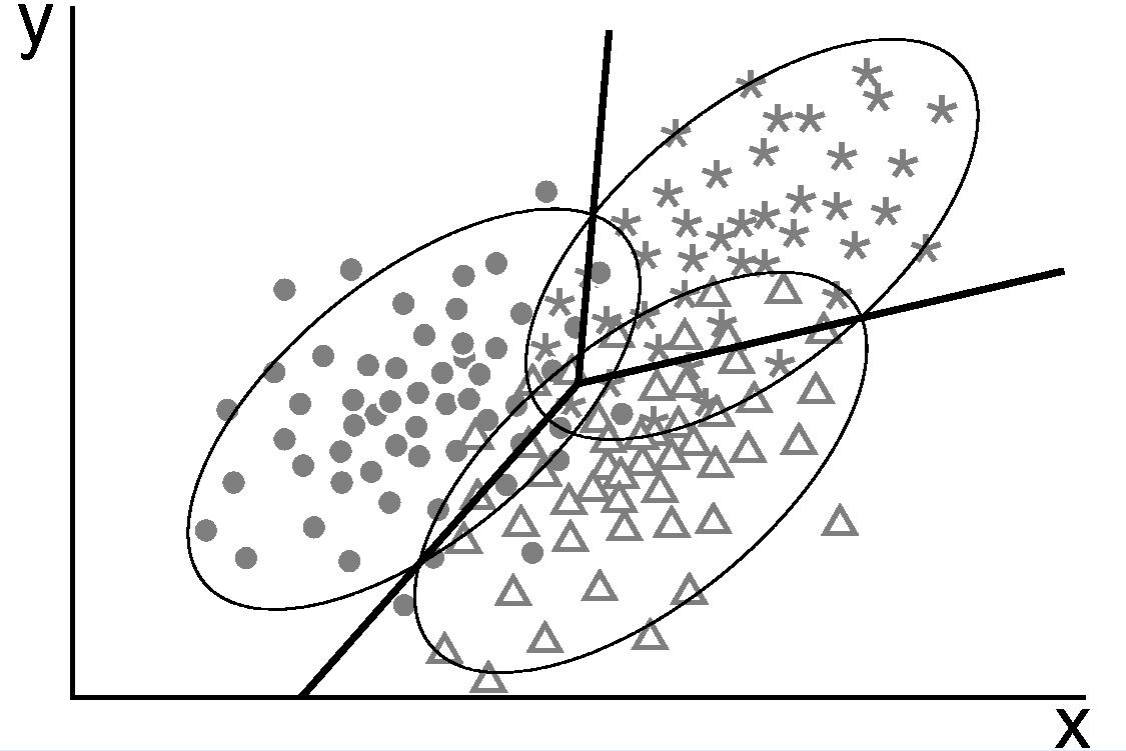
Last time: Gaussian classifier aka centroid method aka naïve Bayes aka Hebb's rule revised



Come to my office hours…

Wed 2:30-4:30 Soda 329

Today: LDA



Math prerequisites

Independence, covariance, correlation, and dot product

Independence: $P(X, Y) = P(X)P(Y)$

Mean:

$\mu_x = E(X)$ — Expected

$\mu_x = (1/N) \sum_{k=1:N} x^k$ — Empirical

Variance:

$\sigma_X^2 = E[(X - \mu_x)^2]$

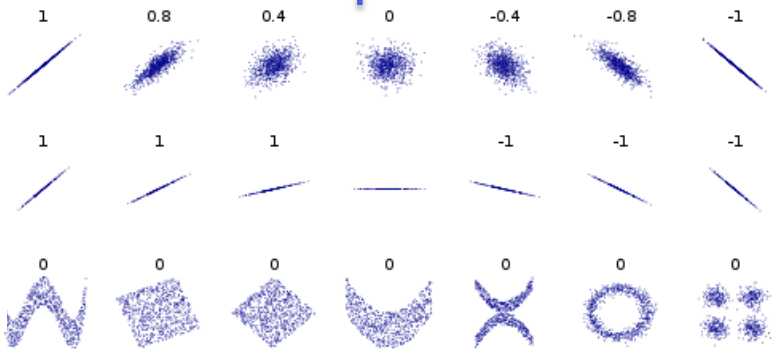
$\sigma_x^2 = (1/N) \sum_{k=1:N} (x^k - \mu_x)^2$

Covariance:

$\sigma_{XY} = E[(X - \mu_x)(Y - \mu_Y)]$

Covariance:

$\sigma_{XY} = (1/N) \sum_{k=1:N} (x^k - \mu_x)(y^k - \mu_Y)$



Pearson correlation:

$$\rho_{X,Y} = \frac{\sigma_{xy}}{\sigma_X \sigma_Y}$$

Pearson correlation:

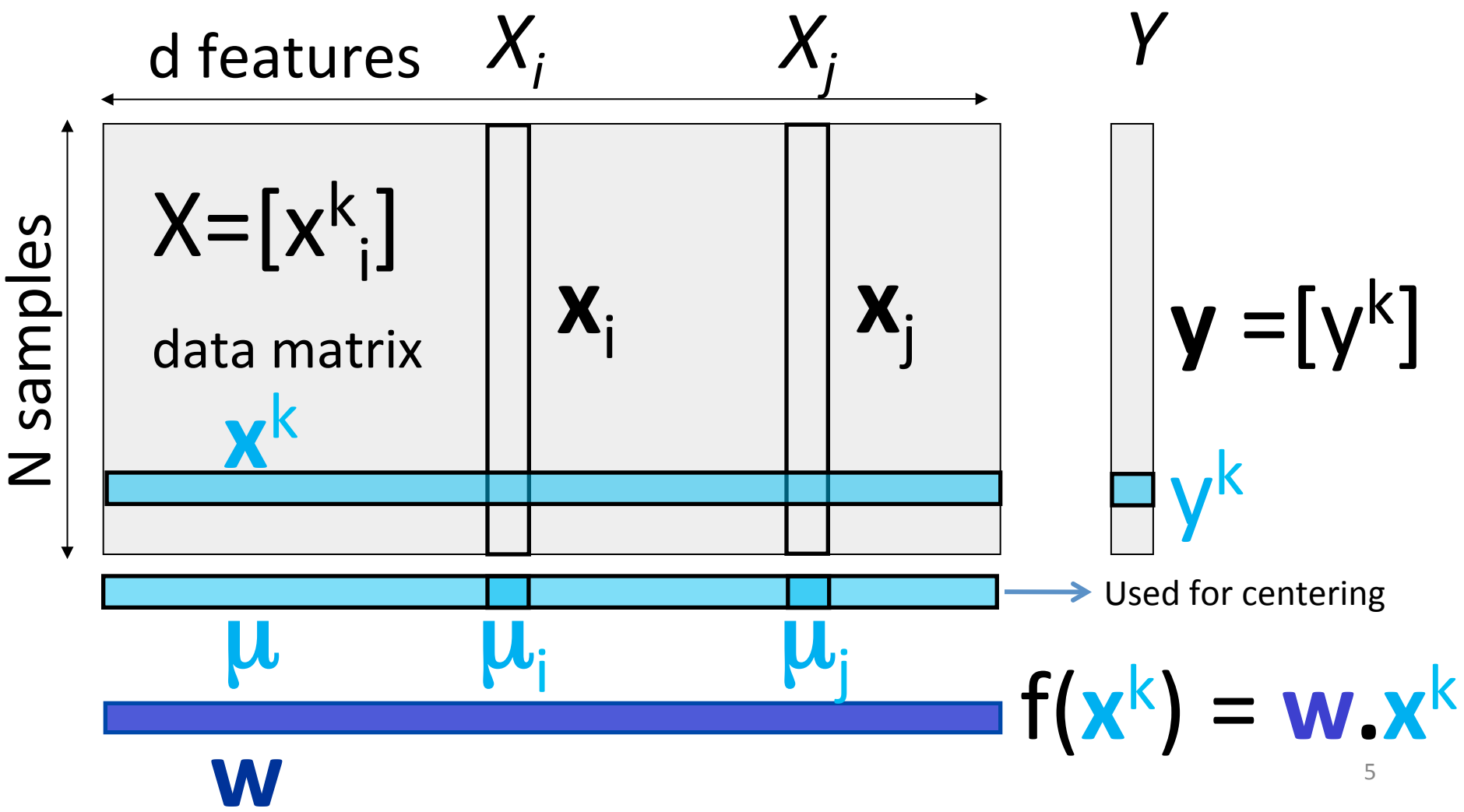
$$\rho_{X,Y} = \frac{\sigma_{xy}}{\sigma_X \sigma_Y}$$

Link to dot product:

$E(X\,Y)$

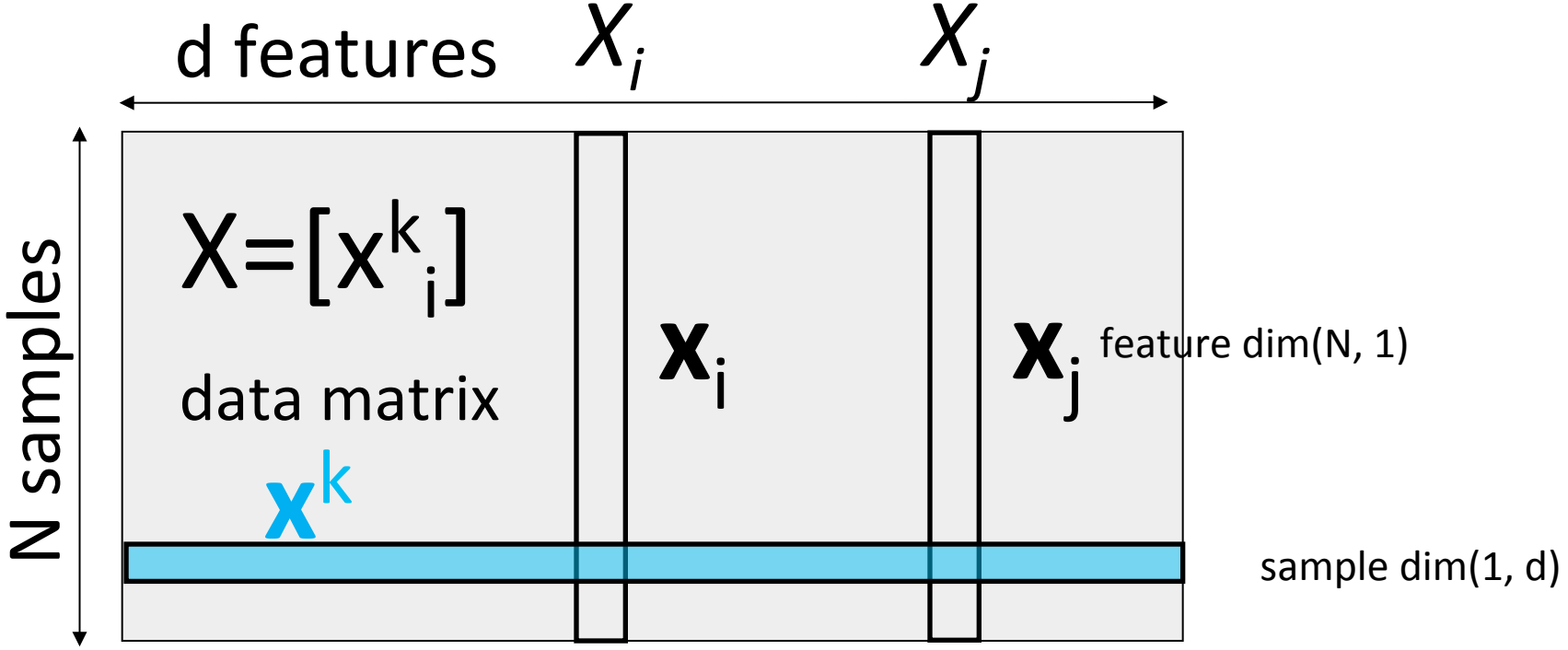
$(1/N)\ \mathbf{x} \cdot \mathbf{y} = (1/N) \sum_{k=1:N} x^k y^k$

What could be X and Y?



Outer product

X is a line random vector $\boldsymbol{X} = [X_1\, X_2 \ldots\ldots X_i \ldots\ldots X_d\,]$



$\Sigma = E(X^T X)$ covariance matrix (if X has 0 mean), dim (d,d)

$\Sigma = (1/N)\, X^T X$ outer product = empirical covariance matrix , dim (d,d)

Σ is positive semi-definite and has therefore a square root $R = \Sigma^{(1/2)}$

$\Phi = XR^{-1}$ is the "whitened data" $\Phi^T \Phi = I$

Data transforms

$\mathbf{x}^k = [x^k_1\ x^k_2\ \ldots\ x^k_i\ \ldots\ x^k_d\ ]$ is a line of the data matrix X.

Mean:

$$\boldsymbol{\mu} = (1/N) \sum_{k=1:N} \mathbf{x}^k$$

Variance:

$$\sigma_i^2 = (1/N) \sum_{k=1:N} (x_i^k - \mu_i)^2$$

Covariance:

$$\sigma_{ij} = (1/N) \sum_{k=1:N} (x_i^k - \mu_i)(x_j^k - \mu_{xj})$$

(d,d) covariance matrix:

$$\Sigma = \sigma_{ij} = (1/N)\ \Xi^T \Xi$$

(N, d) centered data matrix:

$$\Xi = [\xi_i(\mathbf{x}^k)] = [x_i^k - \mu_i]$$

Centering: subtracting the mean of the features.

$$\xi(\mathbf{x}^k) = \mathbf{x}^k - \boldsymbol{\mu}$$

Standardizing or **"sphering":** subtracting the mean and dividing by the standard deviation (component-wise).

$$\phi(\mathbf{x}^k) = (\mathbf{x}^k - \boldsymbol{\mu})\ ./\ \boldsymbol{\sigma}$$

$$\phi_i(\mathbf{x}^k) = (x_i^k - \mu_i)/\sigma_i$$

Whitening: multiplying by the square root of the inverse coveriance matrix.

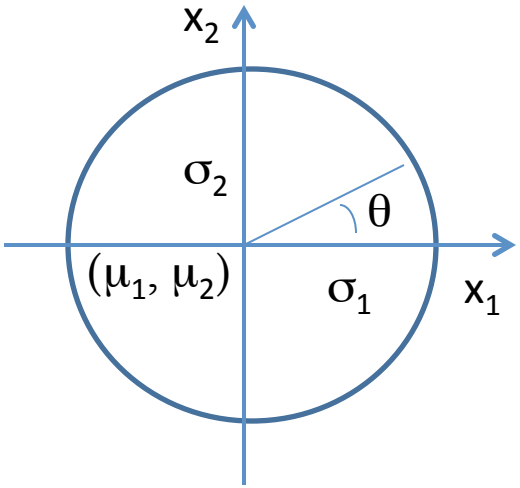
$$\Phi = \Xi\ \ \Sigma^{-1/2}$$

(N, d) = (N, d) (d, d)

Math prerequisites

Ellipses

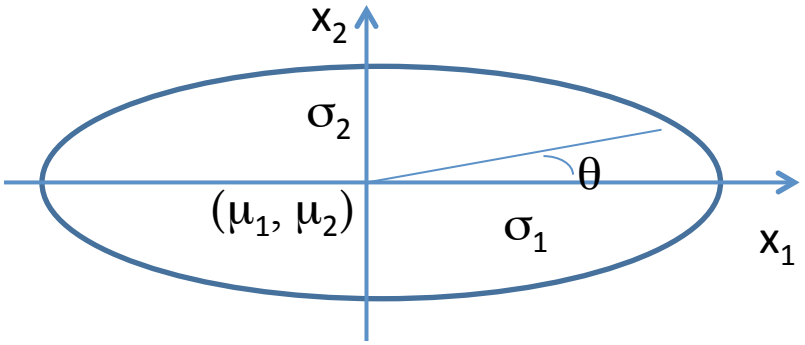
An ellipse in its principal axes is a flattened circle:



circle: $\sigma_1=\sigma_2$

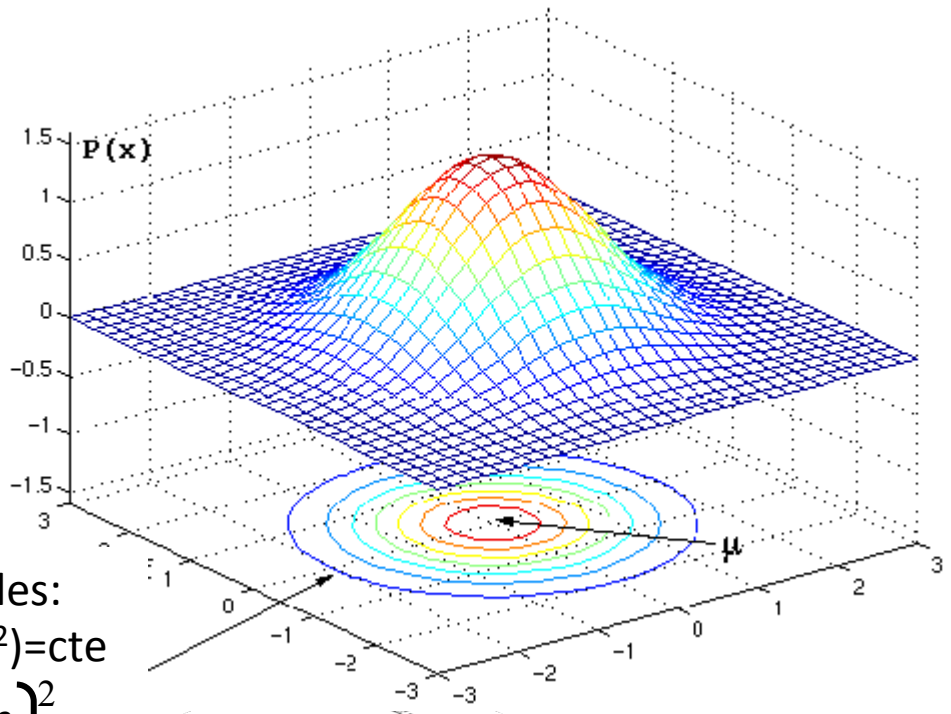
$$x_1 - \mu_1 = \sigma_1 \cos\theta$$
$$x_2 - \mu_2 = \sigma_2 \sin\theta$$

$$\left(\frac{x_1 - \mu_1}{\sigma_1}\right)^2 + \left(\frac{x_2 - \mu_2}{\sigma_2}\right)^2 = 1$$



ellipse: $\sigma_1 \neq \sigma_2$

Isotropic Gaussian ~exp(-$\|\mathbf{x}-\boldsymbol{\mu}\|^2/2\sigma^2$)

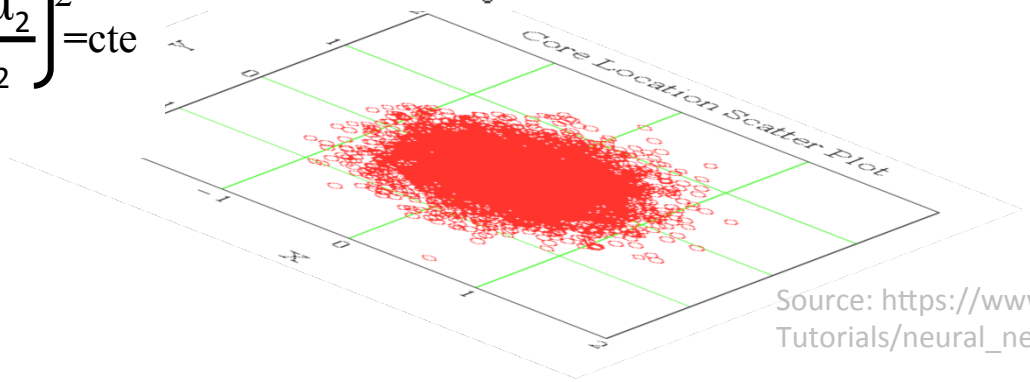


Contours = circles:

exp(-$\|\mathbf{x}-\boldsymbol{\mu}\|^2/2\sigma^2$)=cte

$$\left(\frac{x_1 - \mu_1}{\sigma_1}\right)^2 + \left(\frac{x_2 - \mu_2}{\sigma_2}\right)^2 = \text{cte}$$

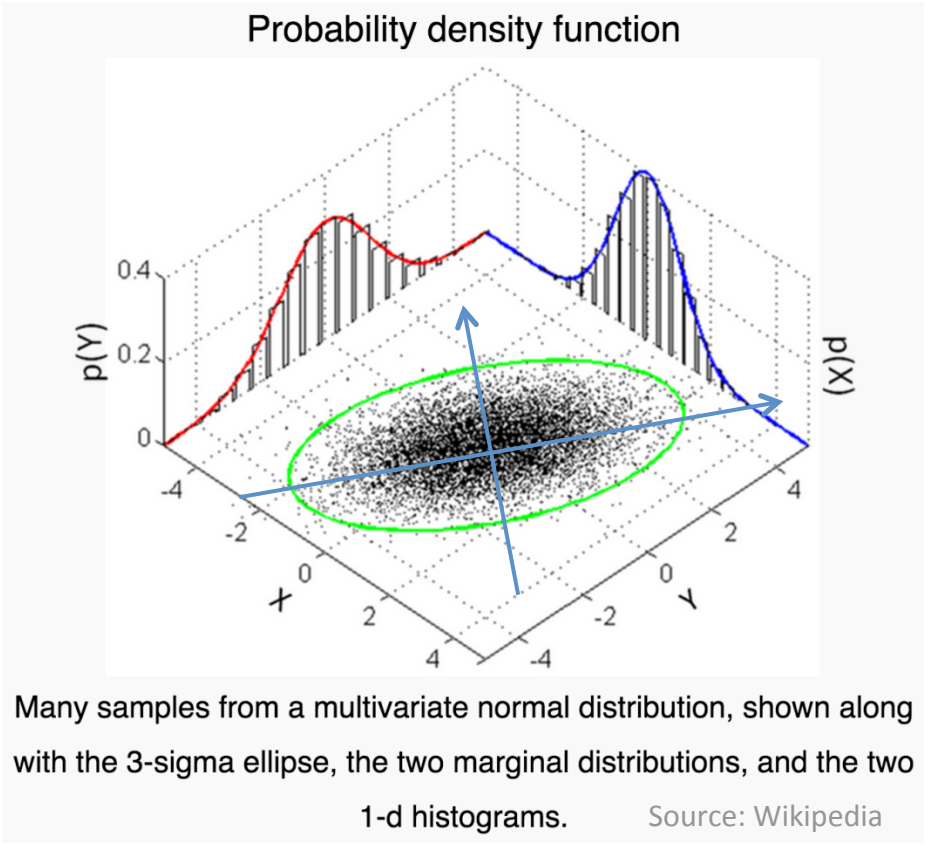
$\sigma_1=\sigma_2$



Source: https://www.byclb.com/TR/Tutorials/neural_networks/ch4_1.htm

Multivariate Gaussian

$$\sim \exp\left(-\frac{1}{2}\underset{(1,\,d)}{(\mathbf{x}-\boldsymbol{\mu})}\ \underset{(d,d)}{\boldsymbol{\Sigma}^{-1}}\underset{(d,\,1)}{(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}}\right) \quad =\exp(-(1/2)\phi\phi^{T})$$



Many samples from a multivariate normal distribution, shown along with the 3-sigma ellipse, the two marginal distributions, and the two 1-d histograms. Source: Wikipedia

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{21} & \sigma_2^2 \end{pmatrix}$$

Contours in the principal axes:

$$\left(\frac{\phi_1 - \mu_{\phi 1}}{\sigma_{\phi 1}}\right)^2 + \left(\frac{\phi_2 - \mu_{\phi 2}}{\sigma_{\phi 2}}\right)^2 = \text{cte}$$

How to get the principal axes?

- $X = [x_i^k]$ is our data matrix (N lines = samples, d columns = feature).
- Assume the sample means $\mu_i = (1/N) \Sigma_k x_i^k$ were subtracted from columns x_i already.
- Then $X^T X$ is the (d, d) covariance matrix of the features (up to a factor N).

- We can diagonalize it as: $\Sigma = X^T X = U S^2 U^T$

With U and orthogonal matrix of column eigenvectors, such that $U^T U = I$.

S^2 is a diagonal matrix of positive eigenvalues s_i^2.

Matrix $R = USU^T = \Sigma^{-1/2}$ is the square root of $X^T X$: $R^2 = X^T X$.

- So to obtain a diagonal covariance matrix S, one just needs to apply matrix U to rotate the axes (and get $\Phi^T \Phi = S^2$, that is a diagonal covariance matrix):

  $\Phi = XU$

  or apply R^{-1} for data **whitening** (and get $\Phi^T \Phi = I$, that is a diagonal covariance matrix, with all the feature variances equal to 1):

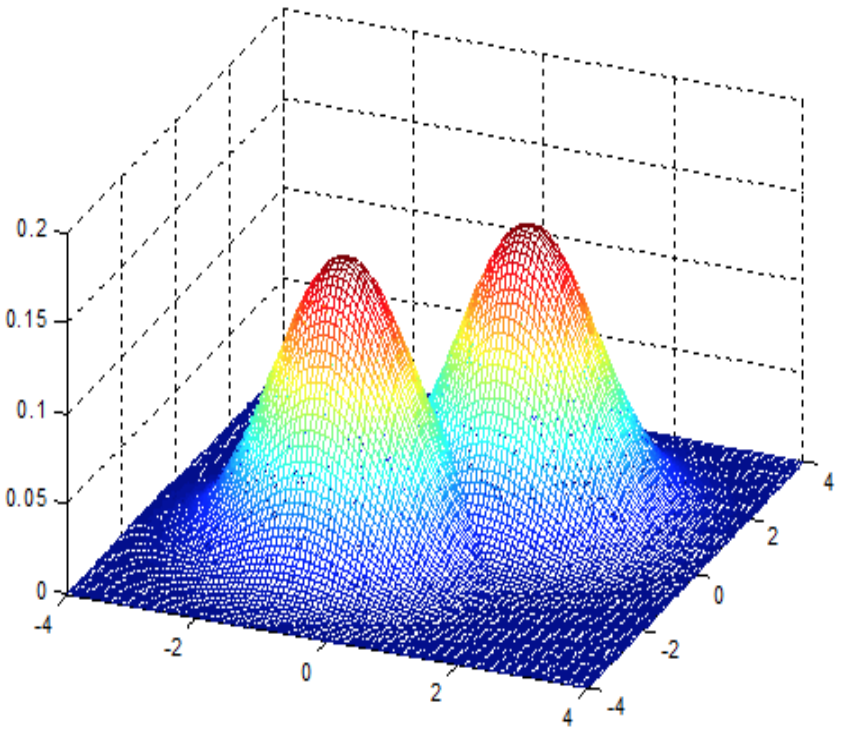
  $\Phi = XR^{-1} = X \Sigma^{-1/2}$

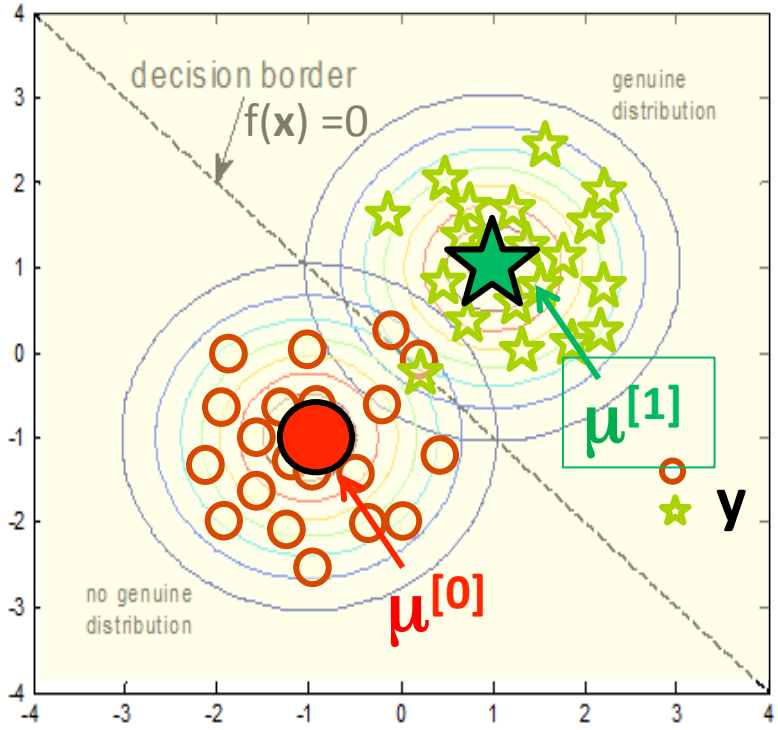
Gaussian classifier

$\hat{y} = \text{argmax}_y\ P(Y=y|X=\mathbf{x}) \sim P(Y=y)\ P(X=\mathbf{x}|Y=y)$

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

Source: http://www.intechopen.com/source/html/17742/media/image25.png



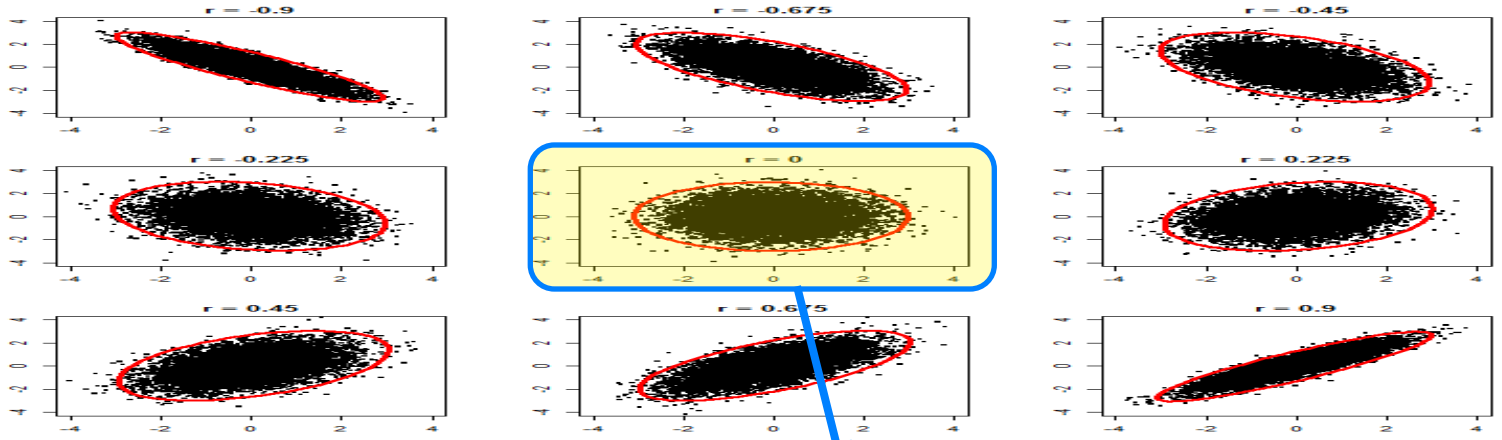


$f(\mathbf{x}) = \underbrace{(\mu^{[1]} - \mu^{[0]})}_{\mathbf{w}} \bullet x + b$

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

Multivariate Gaussian and covariance

$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} \qquad \operatorname{cov}(X,Y) = \mathrm{E}[(X - \mu_X)(Y - \mu_Y)]$$



Independence: P(X, Y)=P(X)P(Y) Zero correlation or covariance

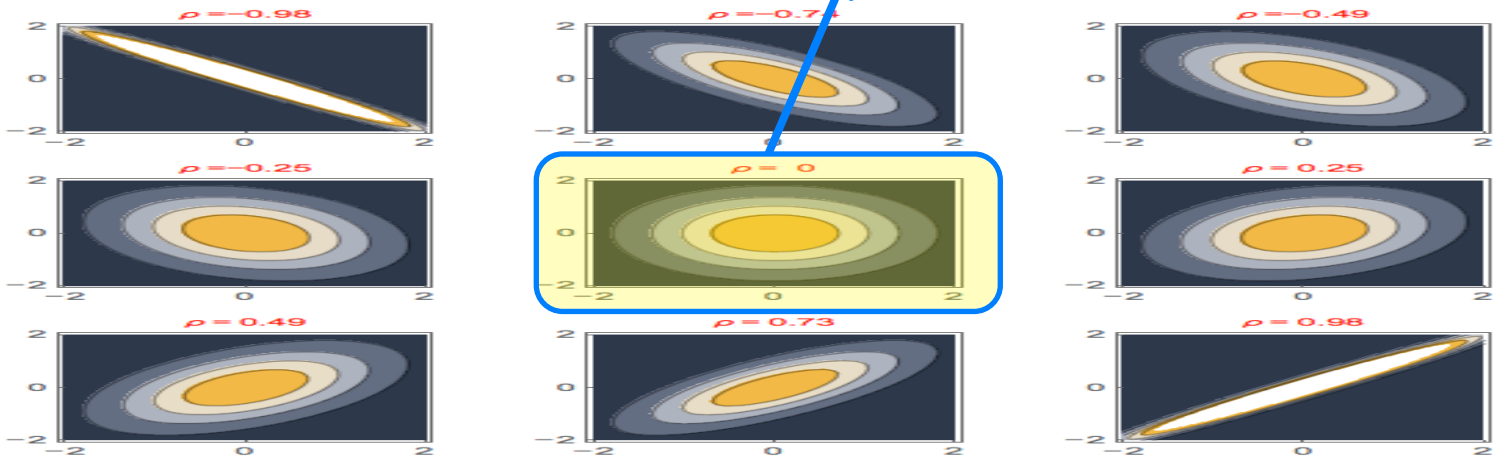
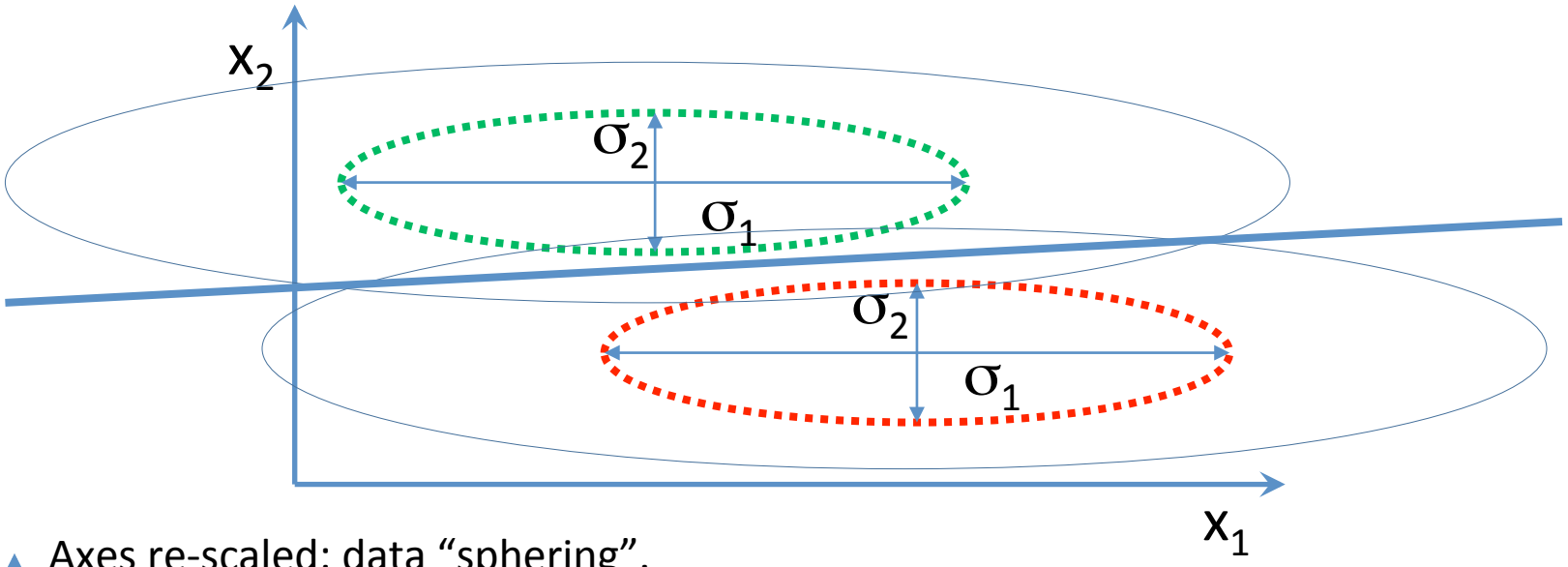
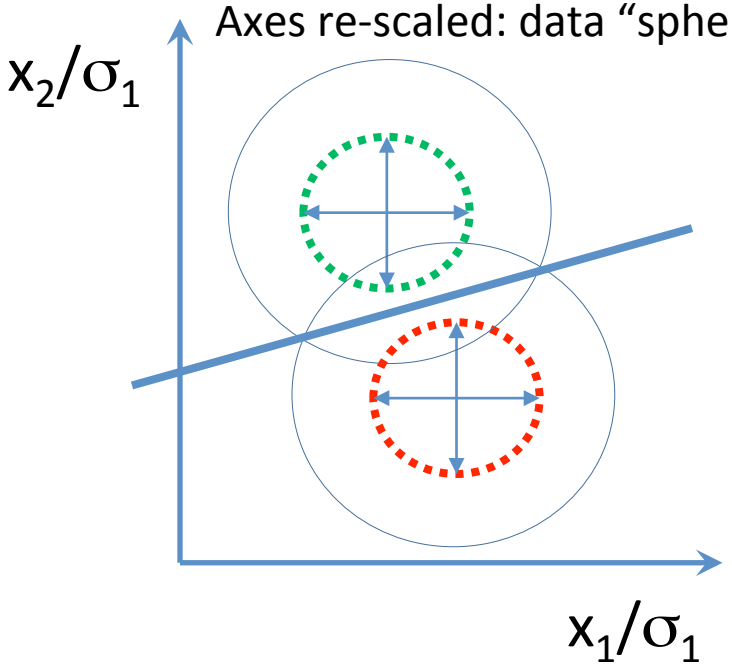


Fig. 12: Contour plots of the bivariate Normal pdf, for different values of ρ

Different σ_i



Axes re-scaled: data "sphering".

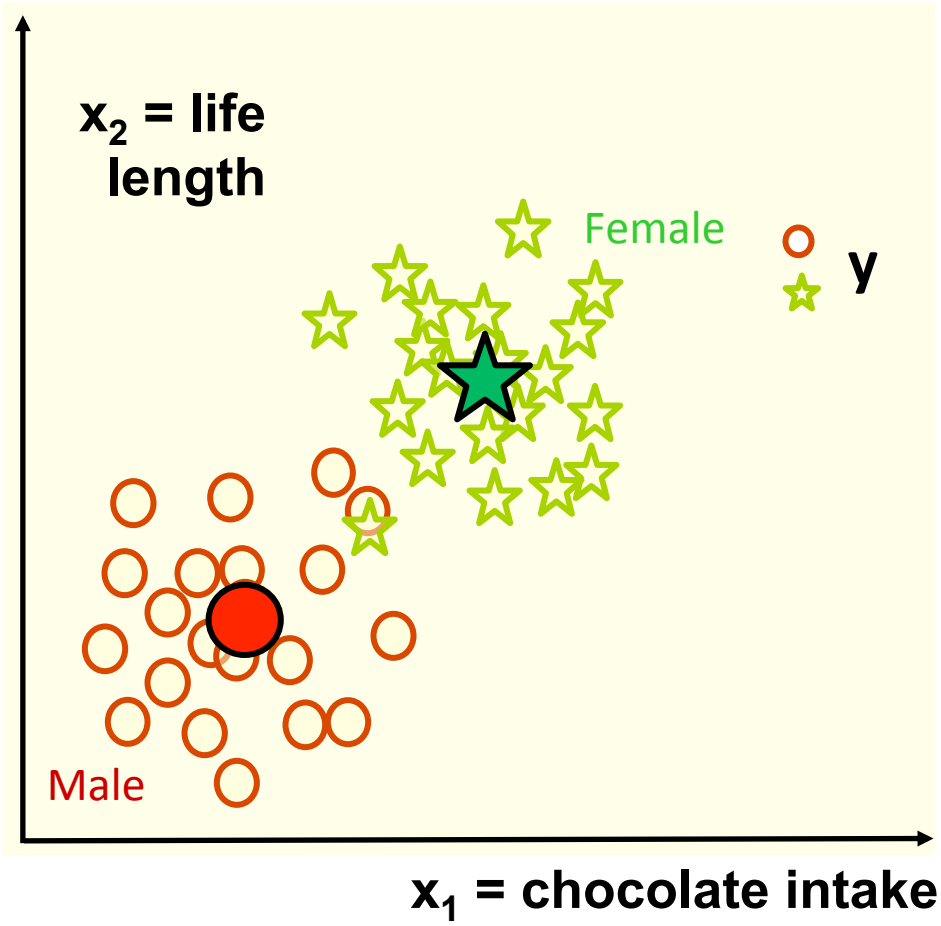


$$f(\mathbf{x}) = \sum_i [(\mu_i^{[1]} - \mu_i^{[0]})/\sigma_i^2]\, x_i + b$$

$$f(\mathbf{x}) = \sum_i [\underbrace{\mu_i^{[1]}/\sigma_i} - \underbrace{\mu_i^{[0]}/\sigma_i}]\, \underbrace{x_i/\sigma_i} + b$$

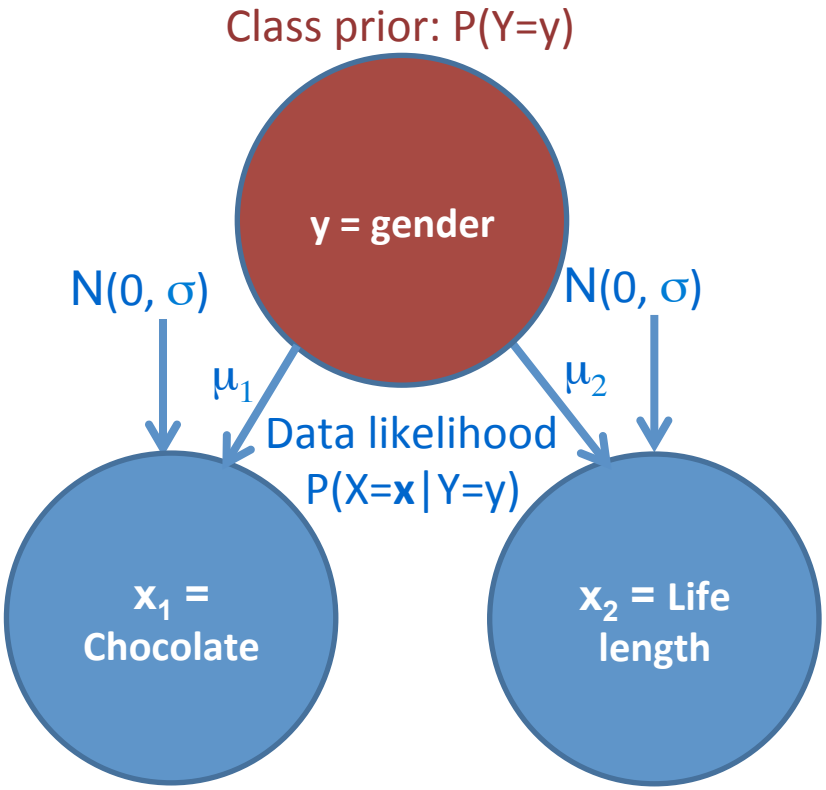
$$f(\mathbf{x}) = \sum_i [\ \mu_i^{[1]\prime} - \mu_i^{[0]\prime}]\ x'_i + b$$

Class conditional independence



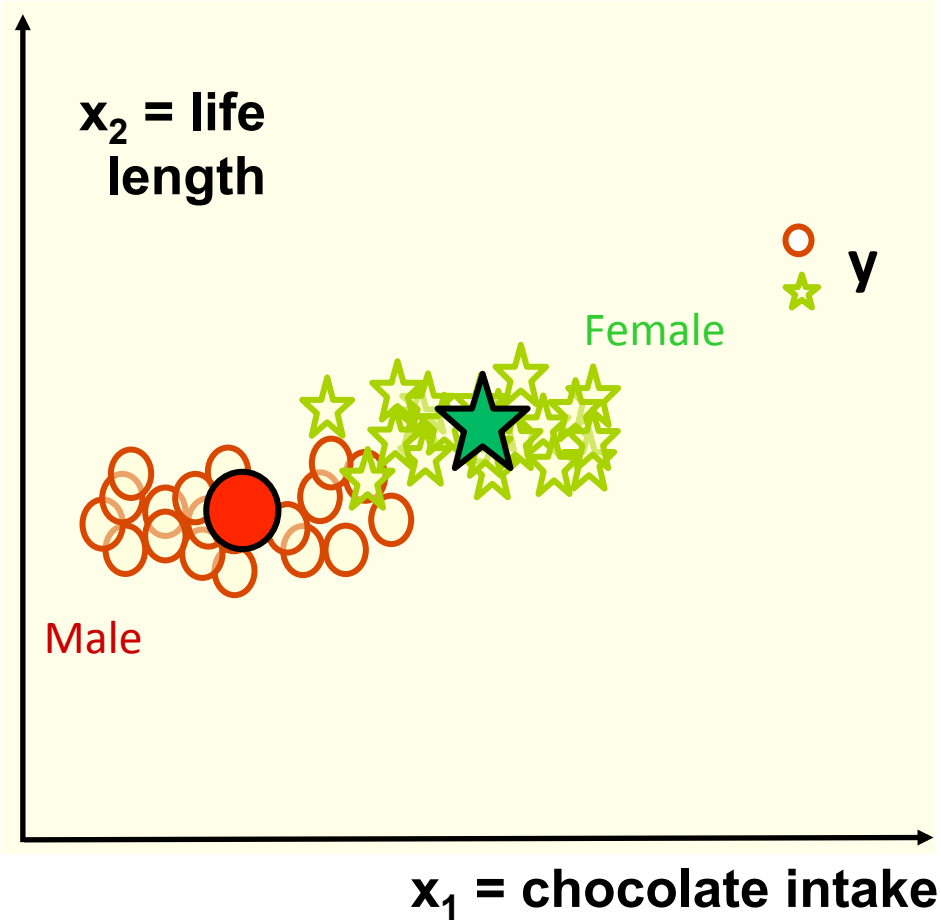
$$x_1 \perp x_2 \mid y$$

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



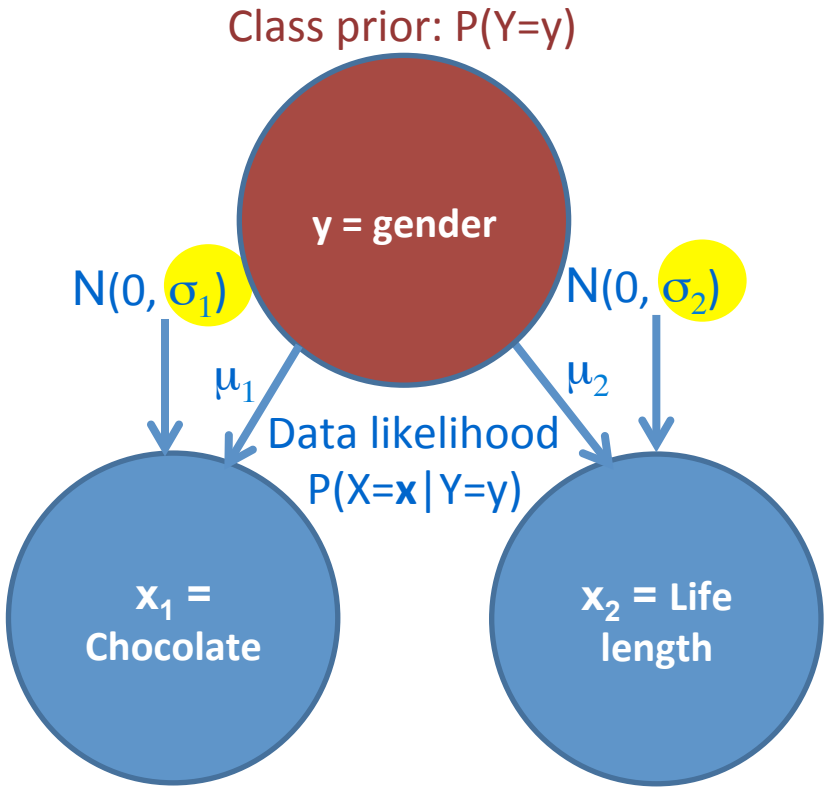
$$P(X=\mathbf{x} \mid Y=y) \sim \exp(-\|\mathbf{x}-\mu\|^2/2\sigma^2)$$
$$= \exp(-\textstyle\sum_{i=1:N} (x_i-\mu_i)^2/2\sigma^2)$$
$$= \textstyle\prod_{i=1:N} \exp(-(x_i-\mu)^2/2\sigma^2)$$
$$\sim \textstyle\prod_{i=1:N} P(X_i \mid Y)$$

Class conditional independence



$$x_1 \perp x_2 \mid y$$

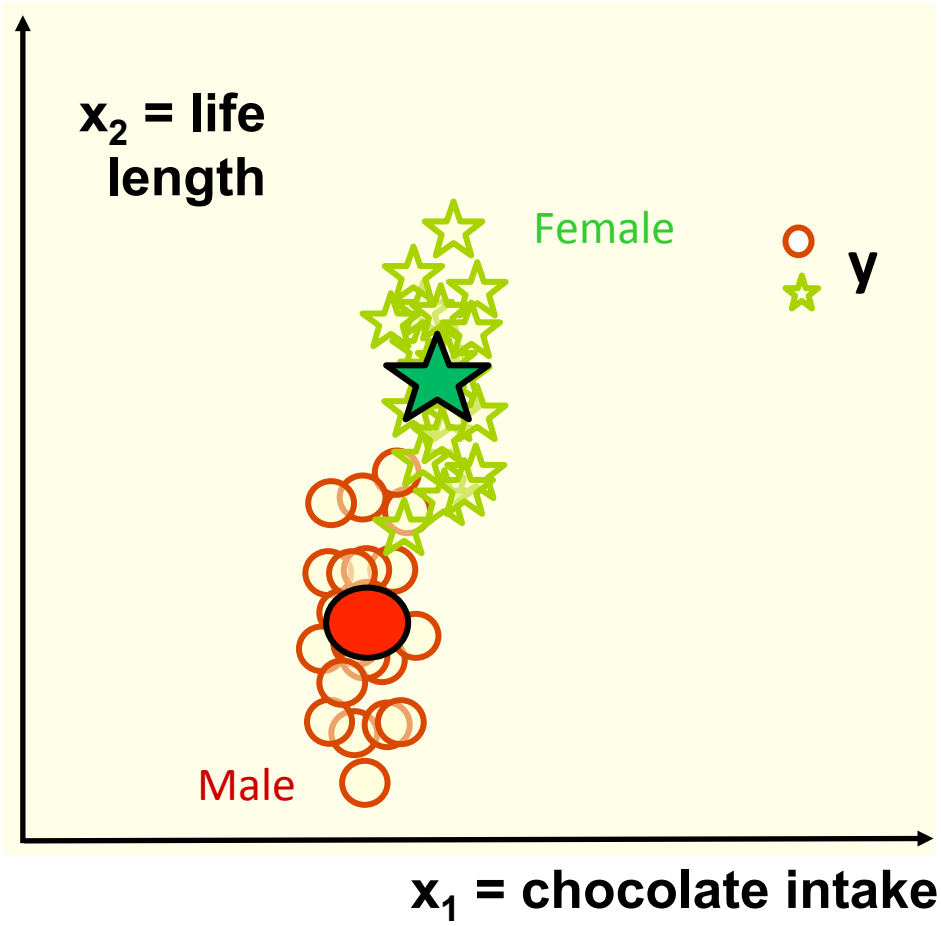
$$P(Y=y|X=\mathbf{x}) \sim P(Y=y)\, P(X=\mathbf{x}|Y=y)$$

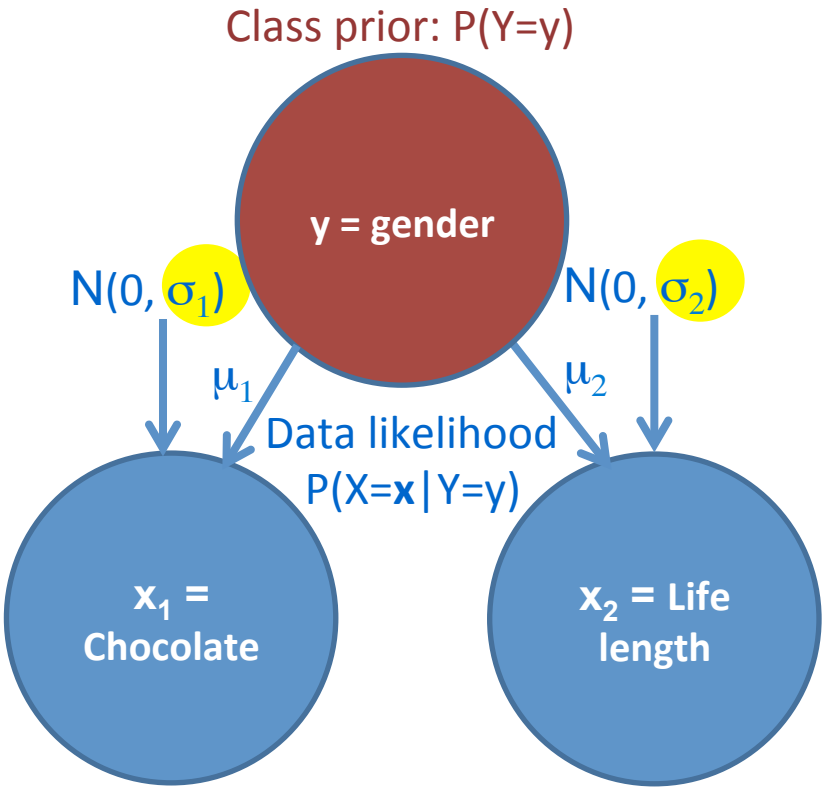


$$P(X=\mathbf{x}|Y=y) \sim \exp(-\textstyle\sum_{i=1:N} (x_i-\mu_i)^2/2\sigma_i^2)$$
$$= \textstyle\prod_{i=1:N} \exp(-(x_i-\mu)^2/2\sigma_i^2)$$
$$\sim \textstyle\prod_{i=1:N} P(X_i|Y)$$

Class conditional independence

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

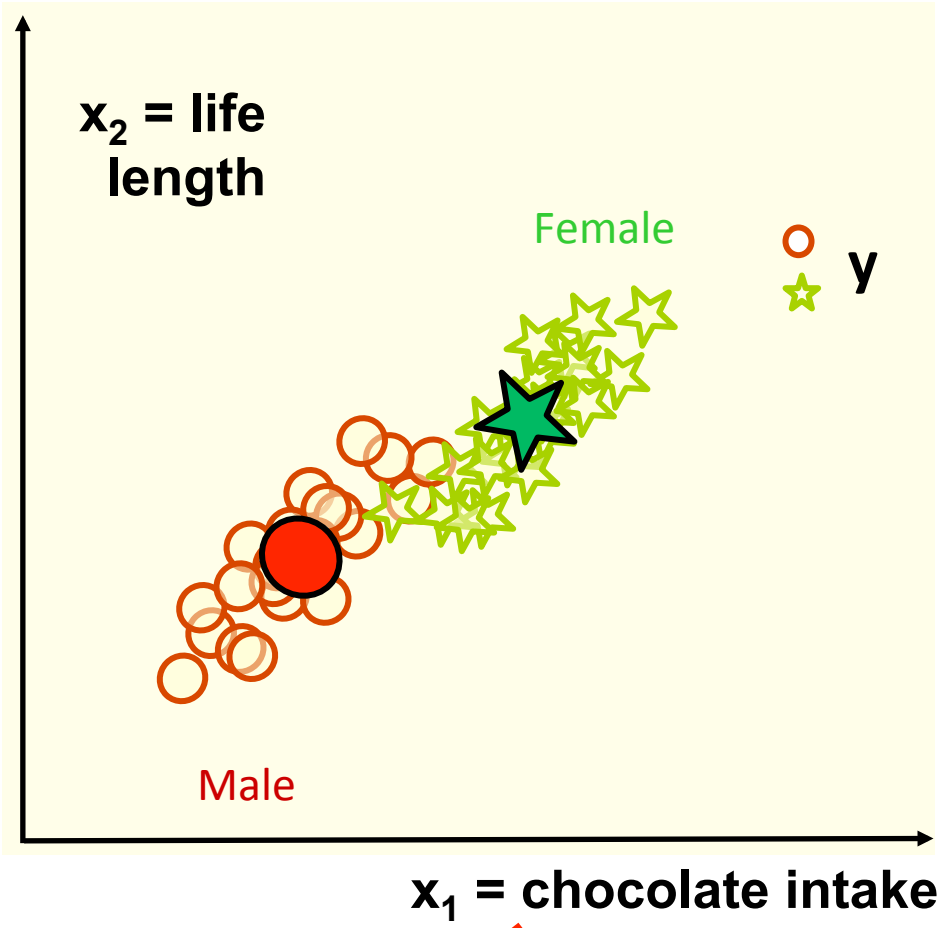




$$x_1 \perp x_2 \mid y$$

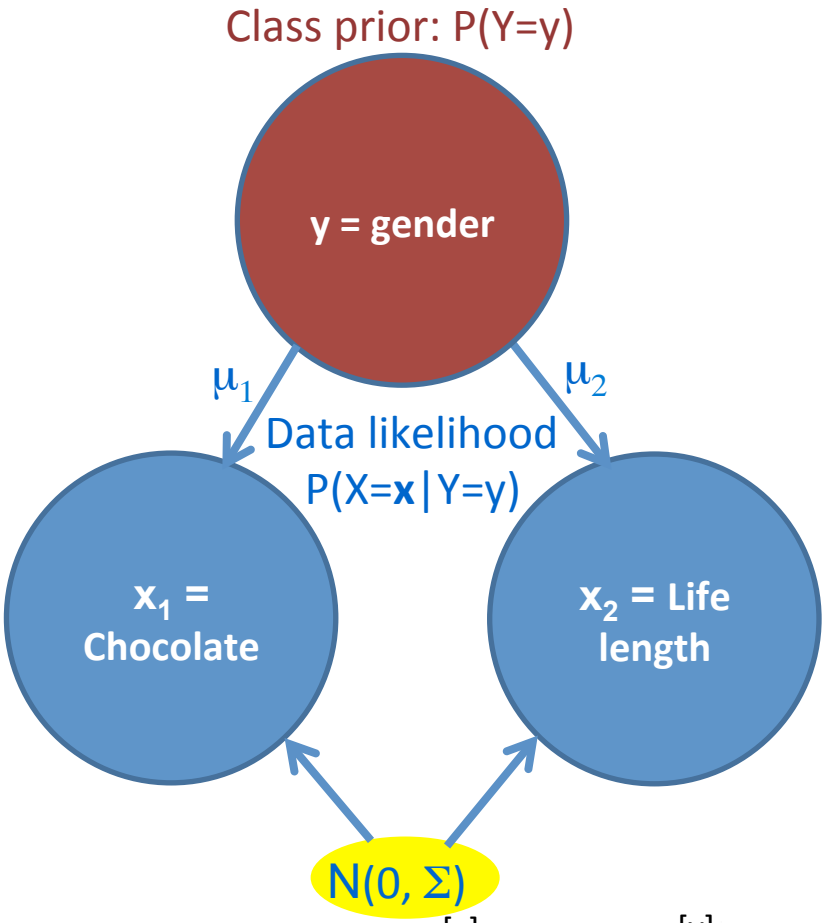
$$P(X=\mathbf{x}|Y=y) \sim \exp(-\textstyle\sum_{i=1:N} (x_i-\mu_i)^2/2\sigma_i^2)$$
$$= \textstyle\prod_{i=1:N} \exp(-(x_i-\mu)^2/2\sigma_i^2)$$
$$\sim \textstyle\prod_{i=1:N} P(X_i|Y)$$

No independence



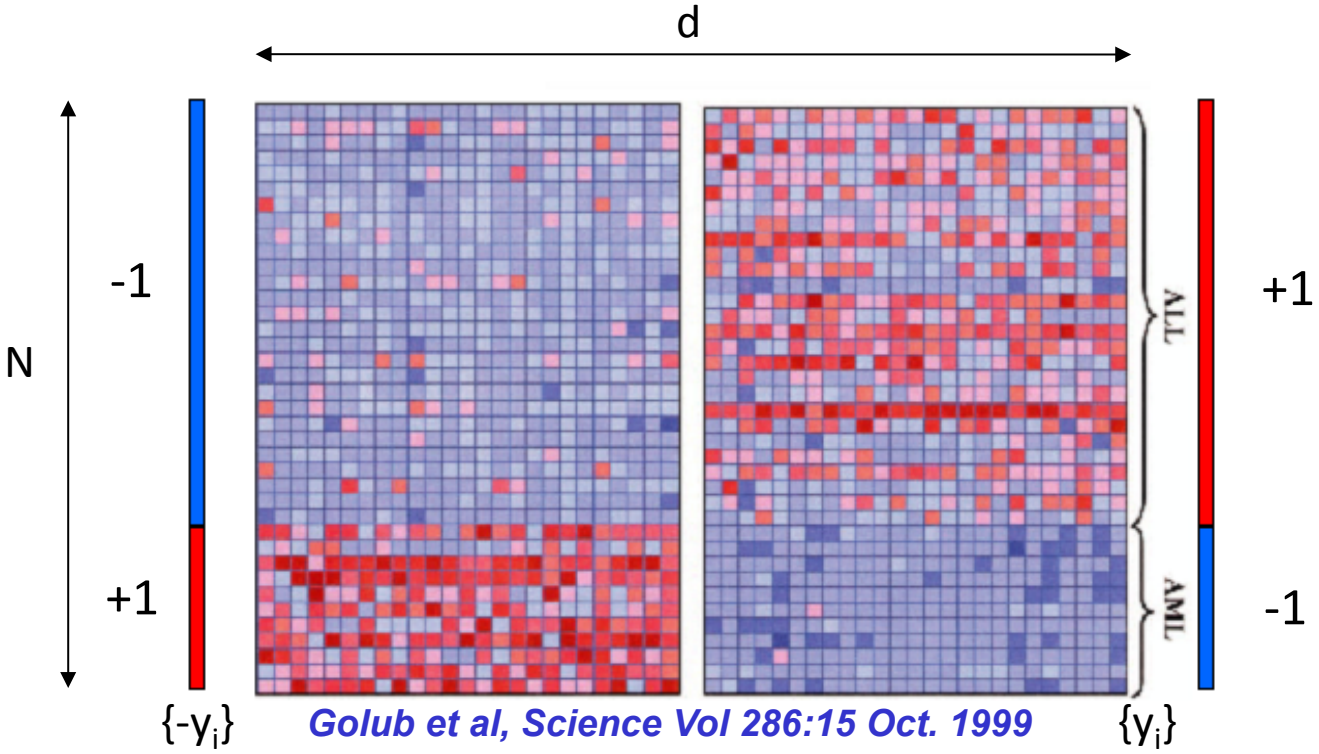
$x_1 \perp x_2 \mid y$

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

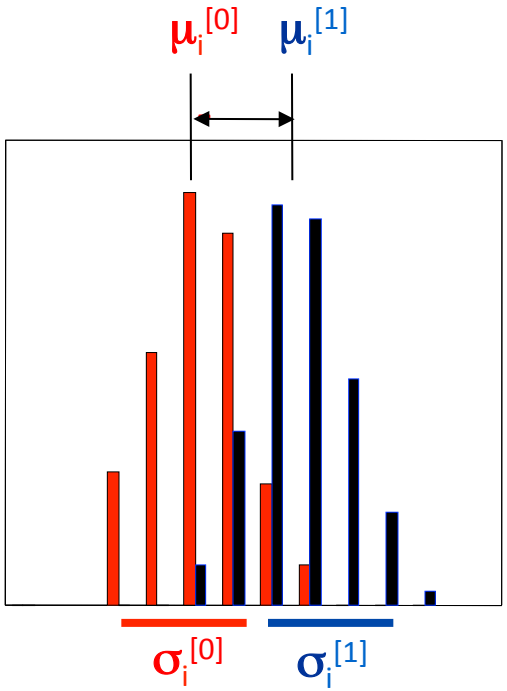


$$P(X=\mathbf{x}|Y=y) \sim (\mathbf{x}-\mu^{[y]})^T \Sigma^{-1} (\mathbf{x}-\mu^{[y]})$$

Data sphering



Golub et al, Science Vol 286:15 Oct. 1999



$$w_i = \frac{\mu_i^{[1]} - \mu_i^{[0]}}{\sigma_i^2} \qquad \sigma_i \cong \tfrac{1}{2}\left(\sigma_i^{[0]} + \sigma_i^{[1]}\right)$$

Rank features according to "signal-to-noise" ratio:

$S2N = |\, \sigma_i w_i \,| \cong |\, \text{corrcoef} \,| \sim |\, \mathbf{x} \bullet \mathbf{y} \,|$

For "balanced classes and after "standardization" $\mathbf{x} \leftarrow (\mathbf{x}-\mu_x)/\sigma_x$

Matrix notations

- For the examples of a given class y, *within class variance*: $\sigma_i^2 = E[(x_i - \mu_i^{[y]})^2]$ (same for all classes).
- Independence assumption: $x_i \perp x_j \mid y \Rightarrow$

no *within class* covariance: $E[(x_i - \mu_i^{[y]})(x_i - \mu_i^{[y]})] = 0$

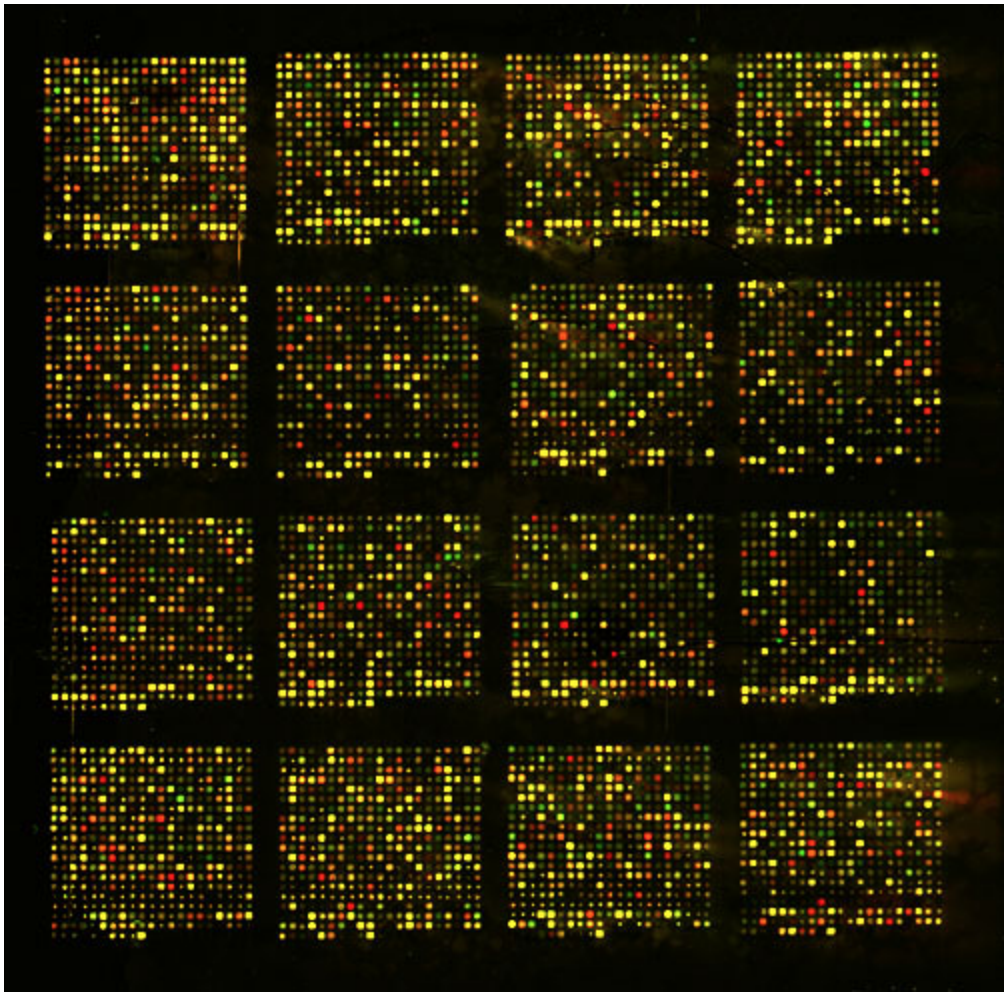
- Diagonal covariance matrix: $\Sigma = \begin{pmatrix} \sigma_1^2 & & & 0 \\ & \sigma_2^2 & & \\ & & \sigma_3^2 & \\ 0 & & & \ddots \\ & & & & \sigma_d^2 \end{pmatrix}$

- Trivial to invert, for each class y:

$$\underset{(1,\,d)}{(\mathbf{x} - \boldsymbol{\mu}^{[y]})}\ \underset{(d,d)}{\boldsymbol{\Sigma}^{-1}}\ \underset{(d,\,1)}{(\mathbf{x} - \boldsymbol{\mu}^{[y]})^T} = \sum_{i=1:d} (x_i - \mu_i^{[y]})^2/\sigma_i^2$$

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

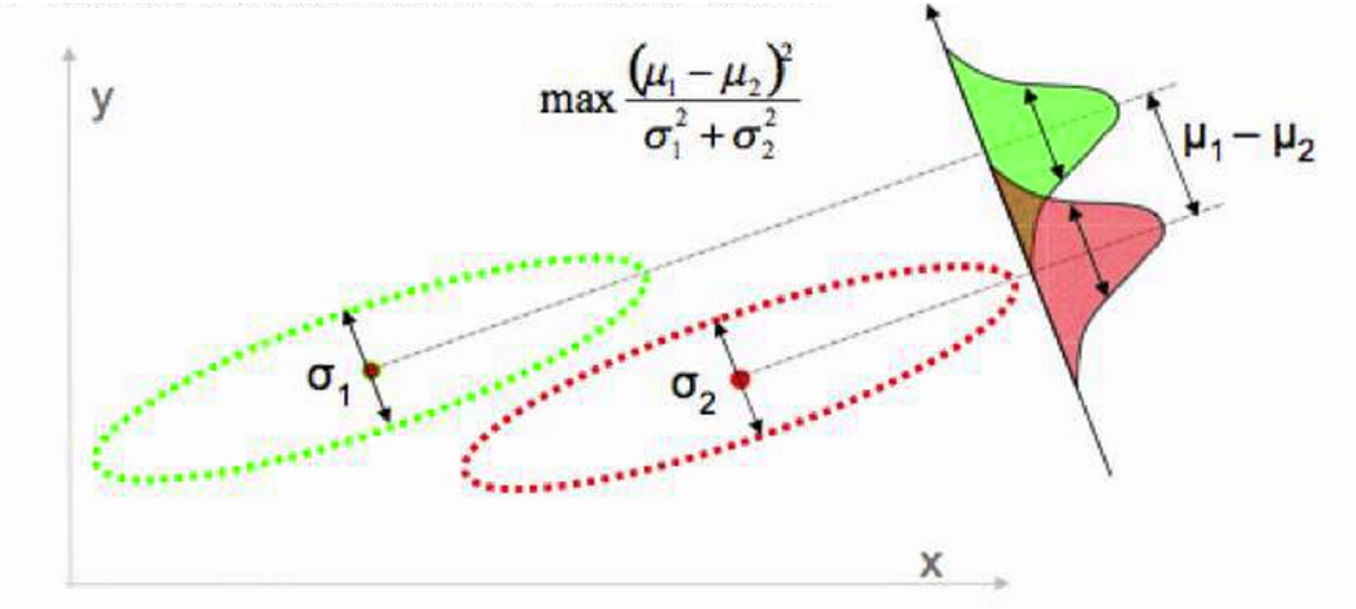
Correlated noise



Source: http://www.uv.es/~arandaa/2002-07-03_WO-vs-WA.jpg

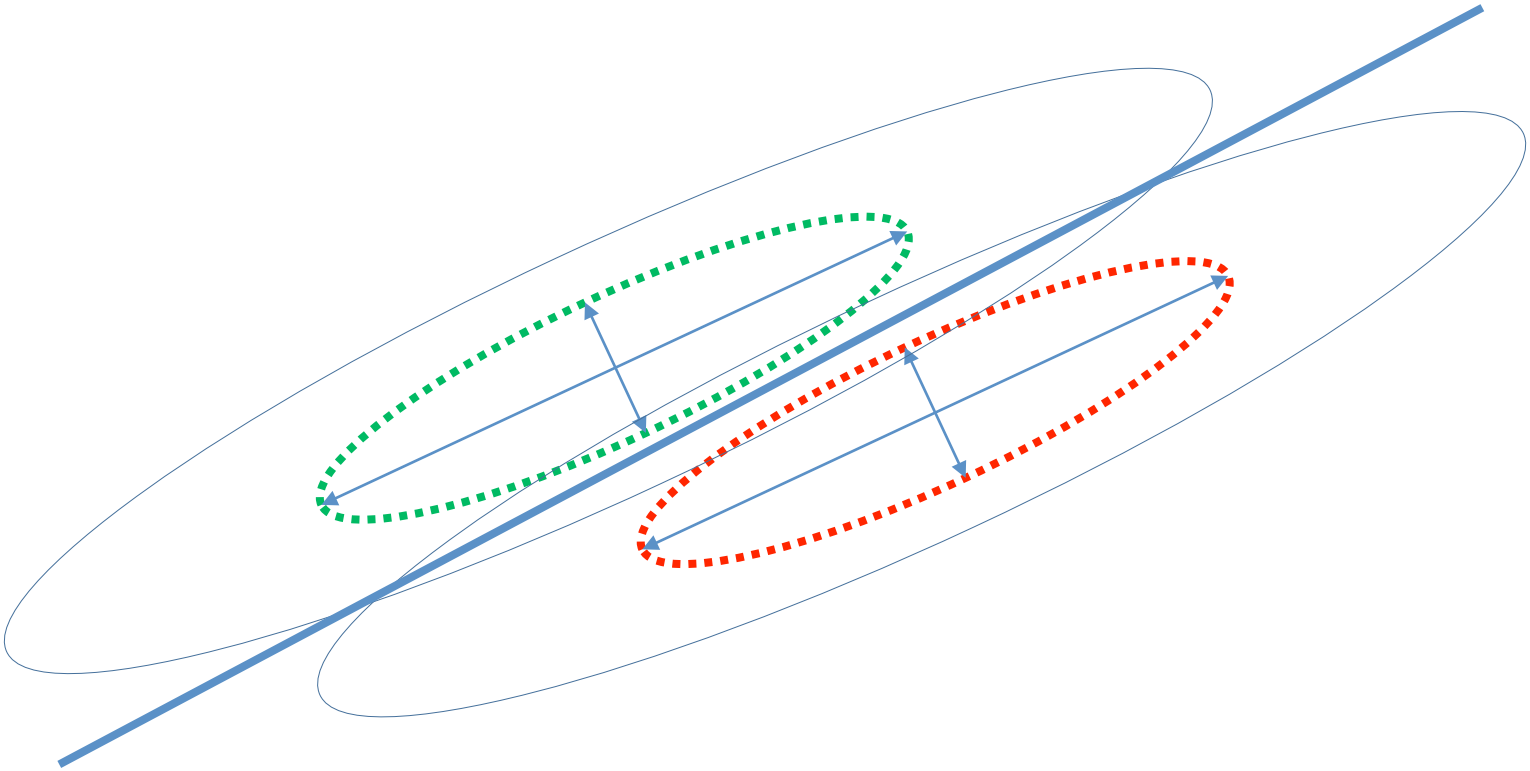
Linear Discriminant Analysis (LDA)

- Rotate input space.
- Find the direction that maximizes the ratio of the between class variance over the within class variance.



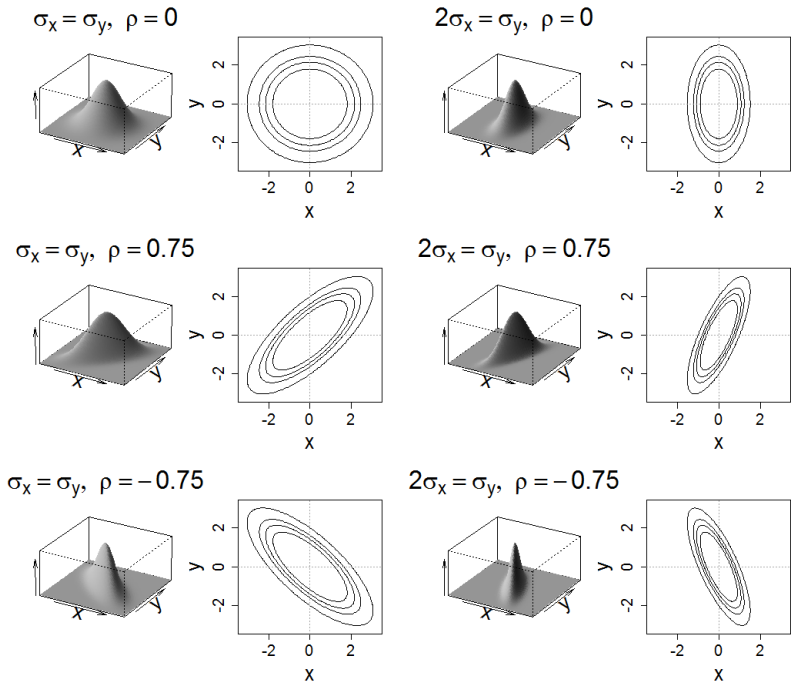
Source: Victor Lavrenko

This is still a linear classifier
(assumes identical covariance matrices)



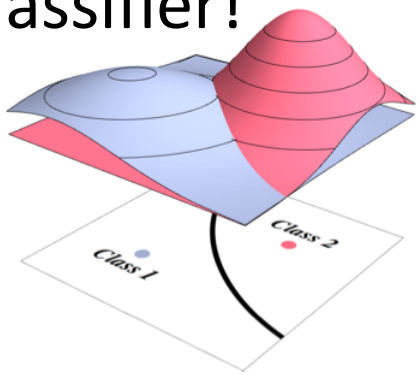
General case

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



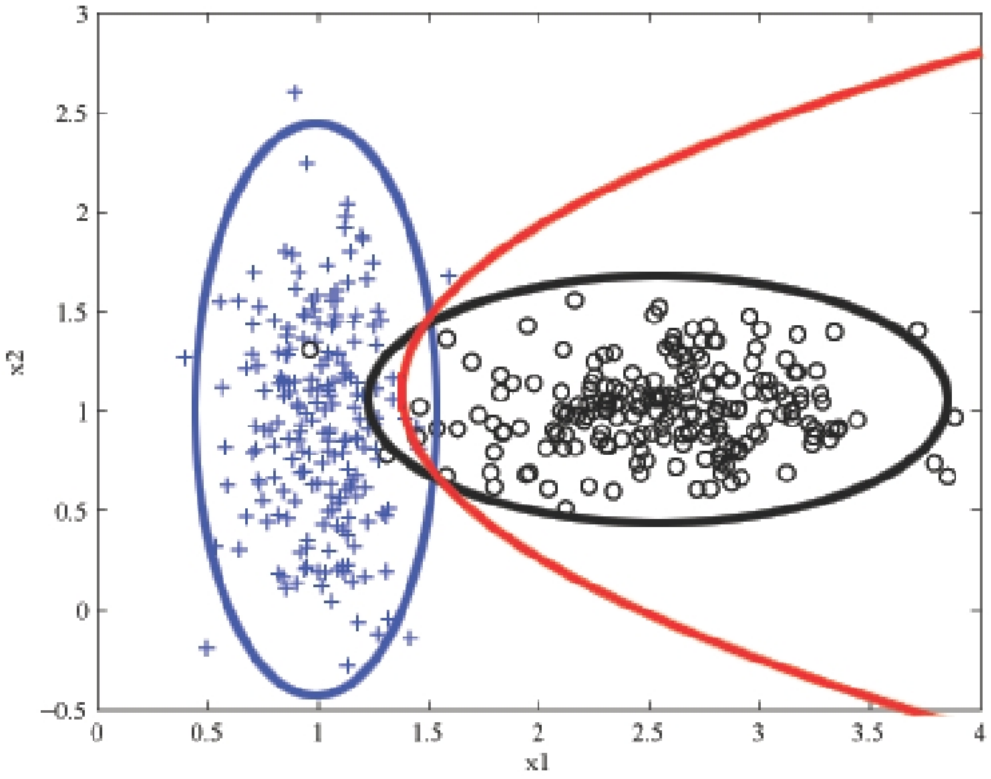
$$\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \sigma_{13} & \cdots & \sigma_{1d} \\ \sigma_{21} & \sigma_2^2 & \sigma_{23} & \cdots & \sigma_{2d} \\ \sigma_{31} & \sigma_{32} & \sigma_3^2 & \cdots & \sigma_{3d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sigma_{d1} & \sigma_{d2} & \sigma_{d3} & \cdots & \sigma_d^2 \end{pmatrix}$$

Not necessarily
a linear classifier!



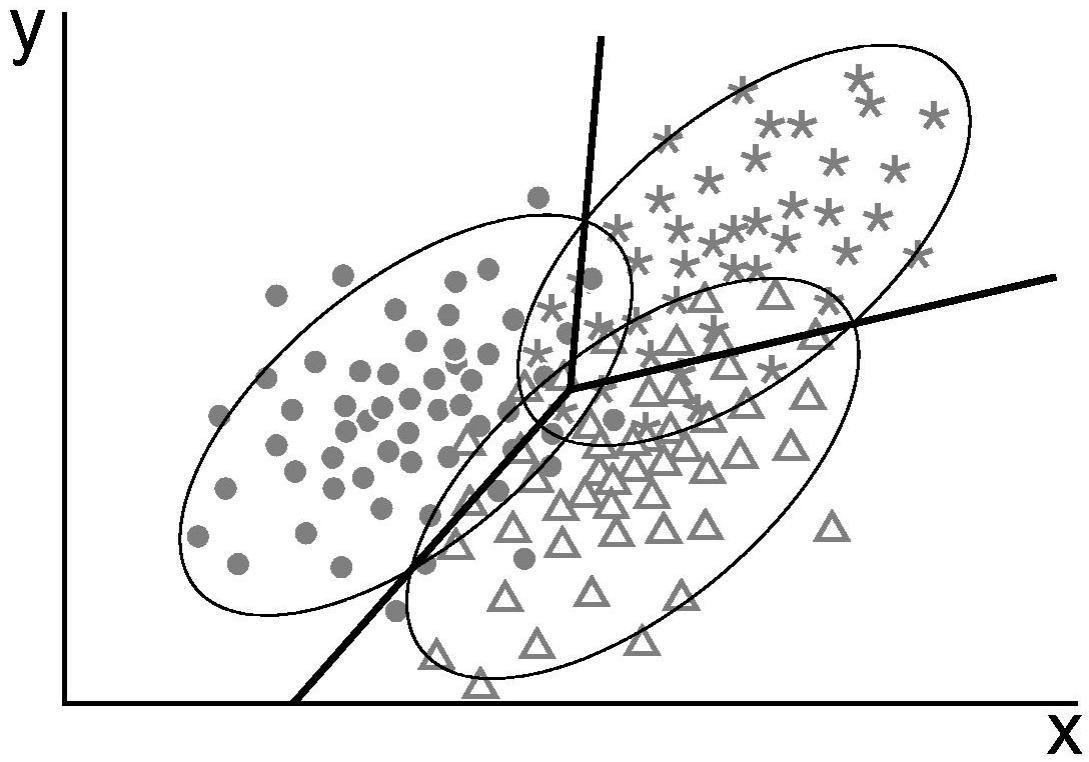
Source: http://www.stat.cmu.edu/

QDA

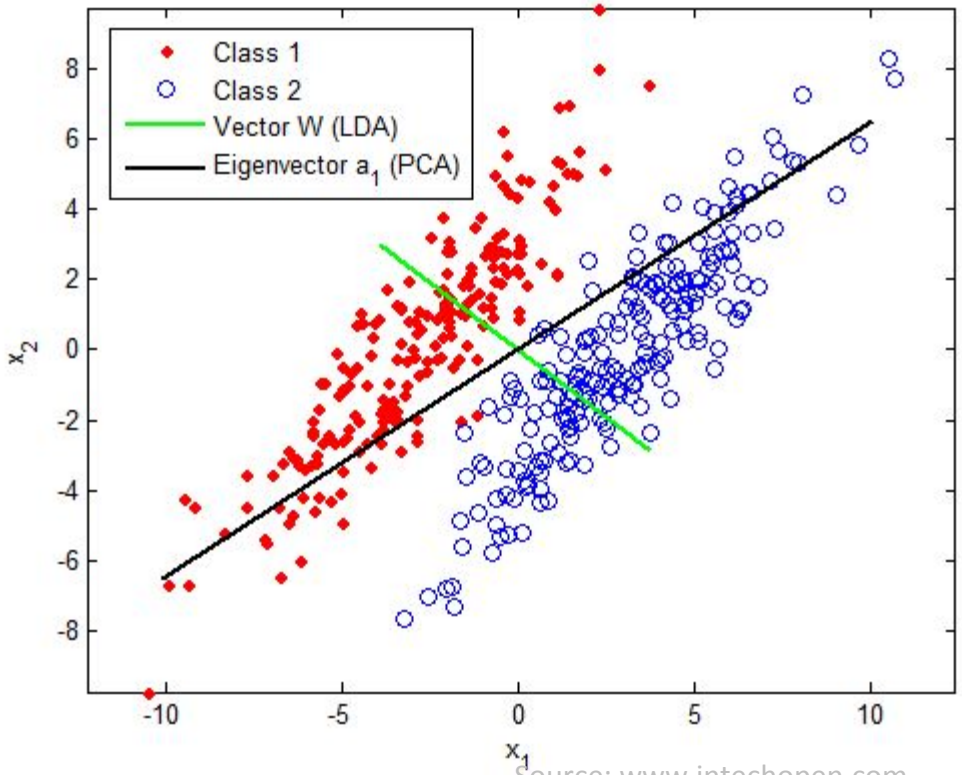


http://learning.cis.upenn.edu/cis520_fall2009/index.php?n=Lectures.NaiveBayes

Multiclass LDA



LDA and PCA

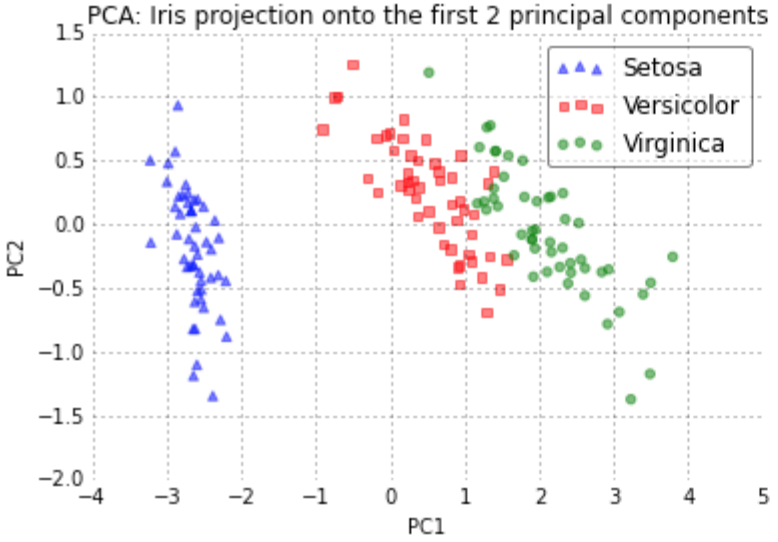


Source: www.intechopen.com

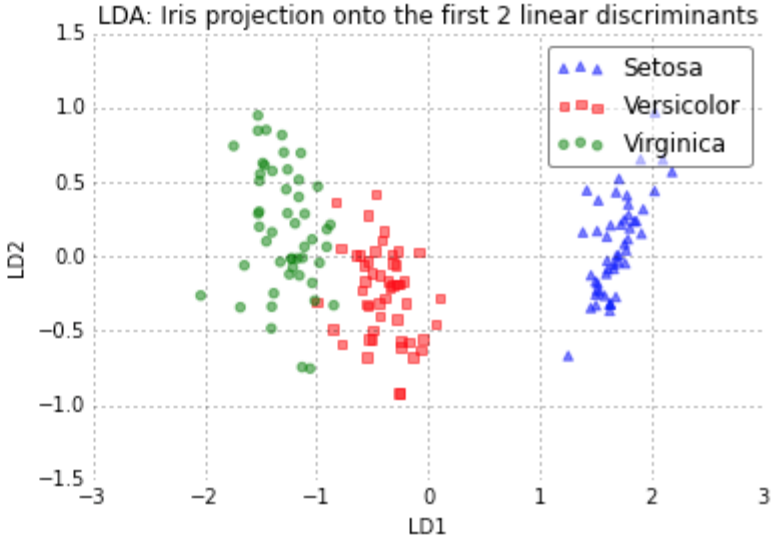
Apply whitening in both case with $\Phi = X \Sigma^{-1/2}$
PCA: Σ is the TOTAL covariance matrix.
LDA: Σ is the POOLED WITHIN CLASS covariance matrix.
Then use the centroid method.

LDA and PCA (continued)

PCA



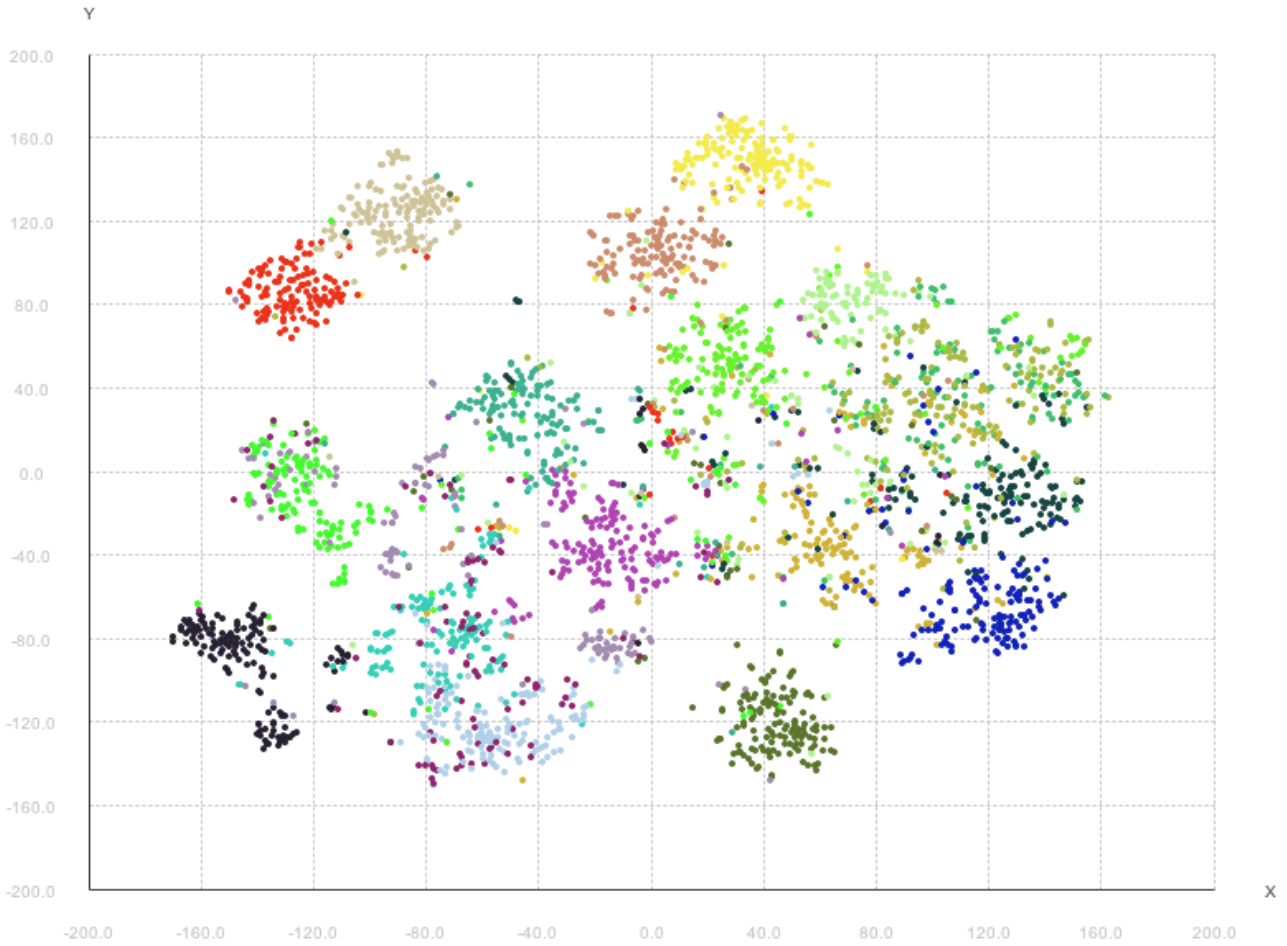
LDA



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

LDA for data visualization

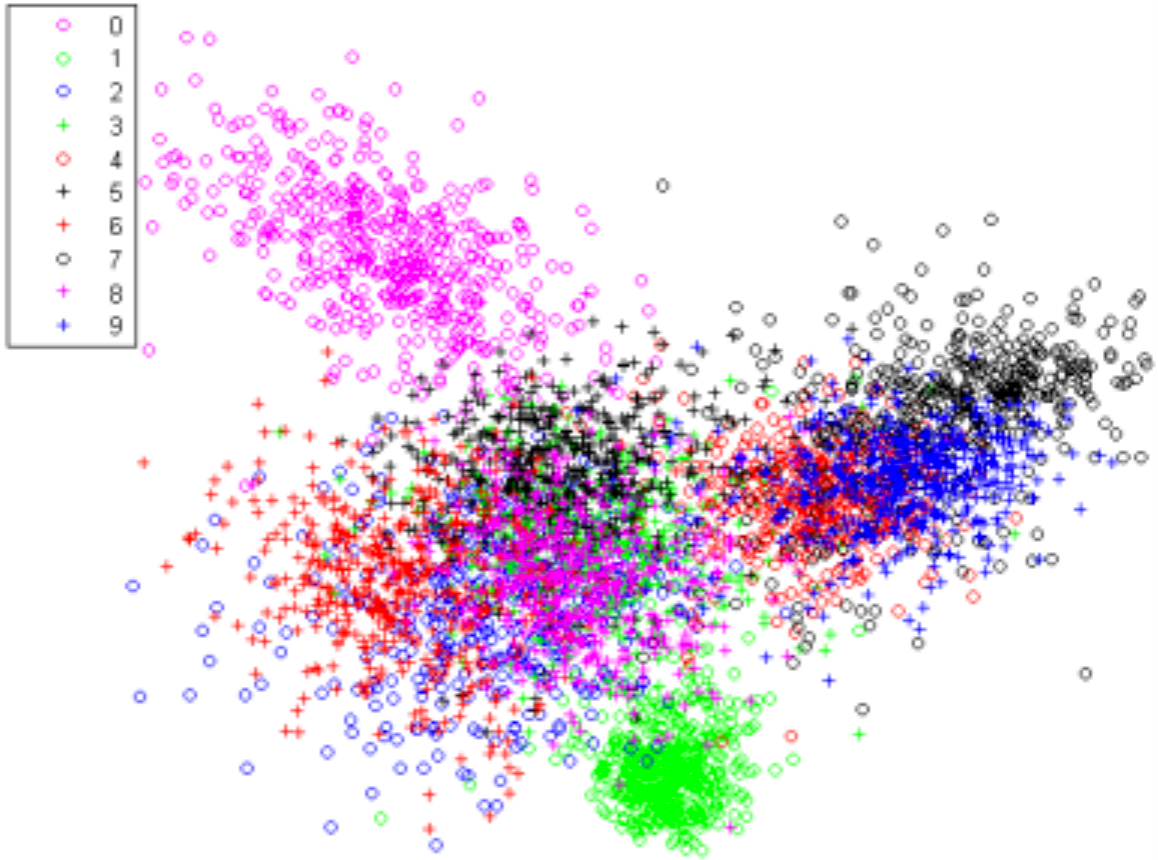
20 Newsgroups



Source http://lejon.github.io/

LDA for data visualization

MNIST



Source: Wang et al. 2014 Generalized Autoencoder

Shrinkage (comparison with ridge regression)

- Remember capacity control: Monitor C/N. C = d_{eff}

Model family	m models	Linear (d feat.)	Kernel method	SVM
Capacity	≤ log m	≤ d+1	≤ N	≤ Nsv

- Minimize $\|\mathbf{w}\|^2$: this boils down for regression and SVM to adding a small value $\lambda>0$ to X^TX!
- Do the same for LDA before inverting $\Sigma = X^TX$.

  $\Sigma \leftarrow \Sigma + \lambda I$
- Then whitening: $\Phi = X\,\Sigma^{-1/2}$
- **Compare with ridge regression:** $f(\mathbf{x}) = \mathbf{x} \cdot \mathbf{w}^T$

  $\mathbf{w}^T = (X^TX)^{-1}\,X^T\,\mathbf{y} = X^+\mathbf{y}$
- **Whitening:** $\Phi = X\,(X^TX)^{-1/2}$ is the whitened data. $\Phi^+ = \Phi^T \quad (\Phi^T\,\Phi = I)$

  $\phi = \mathbf{x}\,(X^TX)^{-1/2}$

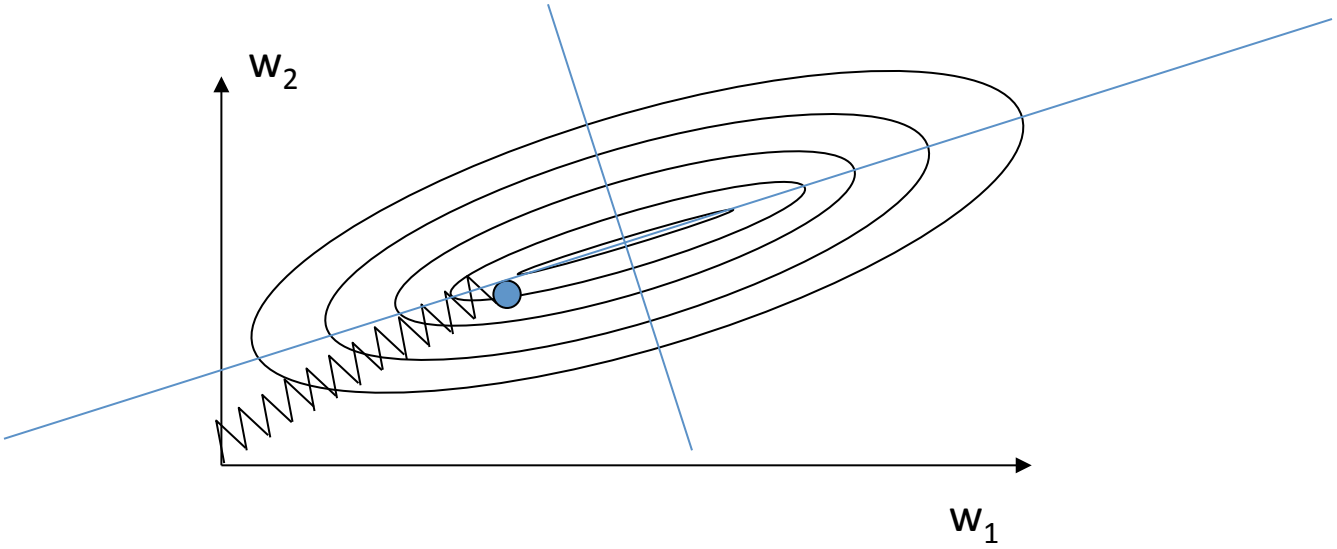
  In Φ-space, $\mathbf{w}^T = \Phi^T\,\mathbf{y}$

  $\mathbf{f(x)} = \phi \cdot \mathbf{w}^T$
- LDA does the same thing, with target values $1/N_1$ and $-1/N_0$ and using the pooled within class covariance matrix.

How do ridge regression and regularized LDA control capacity?

- For f(**x**) = **w.x** the VC dimension is C = d (the dimension of input space).
- $R_{reg} = RSS + \lambda \| \mathbf{w} \|^2 \qquad RSS = \| X\mathbf{w}^T - \mathbf{y} \|^2$

  $= \mathbf{w}X^TX\mathbf{w}^T - 2\mathbf{w}X^T\mathbf{y} + \mathbf{y}^T\mathbf{y}$
- Gradient: $\nabla_{\mathbf{w}} R = 2 (X^TX\mathbf{w}^T - X^T\mathbf{y})$
- Hessian: $H = 2X^TX$



Generative models

You are here

Naïve Bayes

LDA

Mixture models

Gaussian processes

Summary

- LDA is a generalization of the Gaussian classifier for cases in which the input variables are not statistically independent, but all classes have the **same covariance matrix Σ**.
- Once we rotate input space into the "principal axes" of Σ and rescale by the eigen values, LDA is like the isotropic Gaussian classifier, a.k.a. **centroid method**.
- PCA and ridge regression use the covariance matrix of all the data. LDA uses the **"pooled" within class covariance**.
- Two-class LDA is also called Fisher's linear discriminant and is similar to least square regression with $1/N_1$ and $-1/N_0$ target values.
- LDA is useful for multi-class classification and data visualization. Other classifiers have multi-class versions.
- LDA and logistic regression estimate $P(Y=y|X=\mathbf{x})$ in different ways.

Come to my office hours…
Wed 2:30-4:30 Soda 329

Next time: Mixture models