Experimental Data Analysis

in ©MATLAB

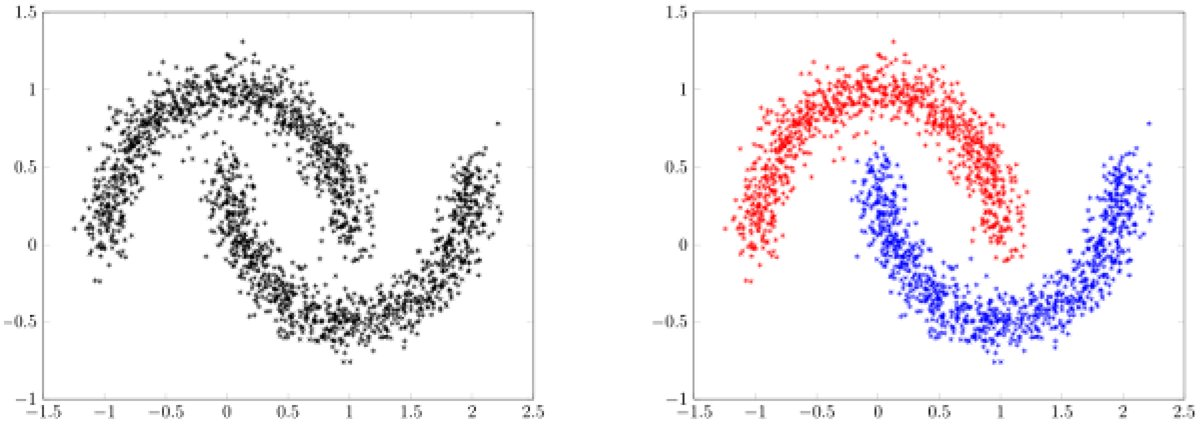
Lecture 11:

Clustering, K-means, EM-algorithm

Jan Rusz
Czech Technical University in Prague

What is data clustering?

- Data clustering is an **unsupervised learning** problem
  - given N unlabeled samples $x_1 \dots x_n$ and the number of partitions K
  - goal is to group the N examples into K partitions
- In the context of machine learning, **classification is supervised learning** whereas **clustering is unsupervised learning**

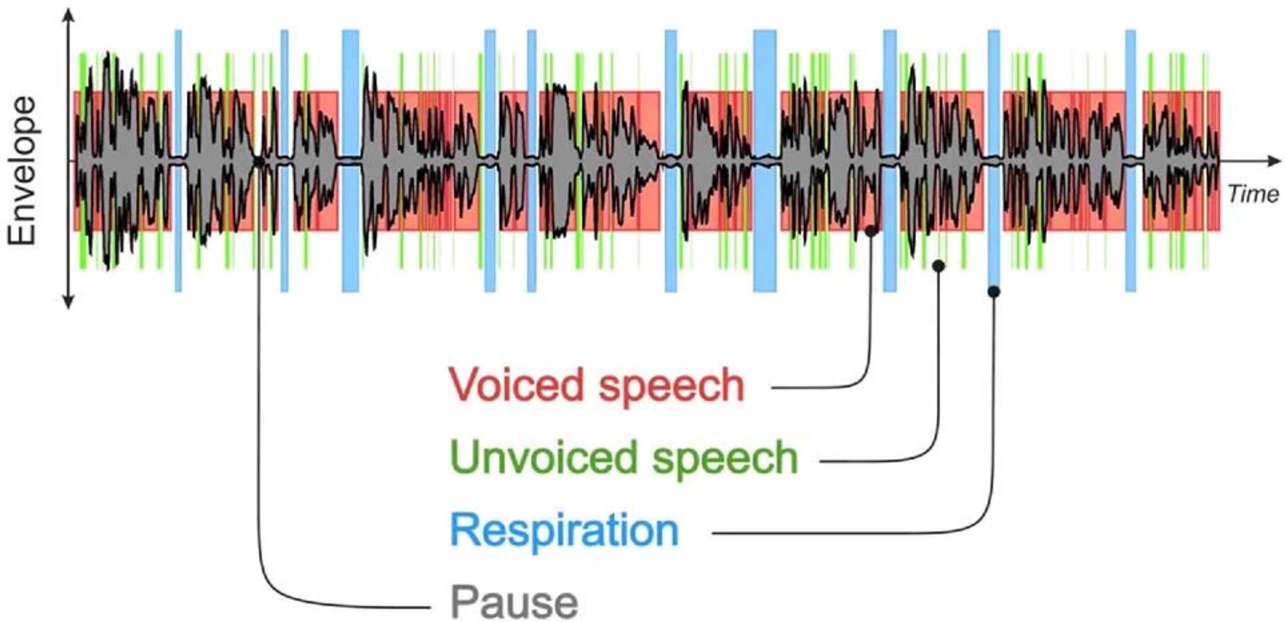


What is data clustering?

- Only information that is used by clustering is the **similarit**y between samples, i.e. groups examples based on their mutual similarities
- A good clustering is one that achieves **high within-cluster similarity** and **low inter-cluster similarity**

Data clustering: practical use

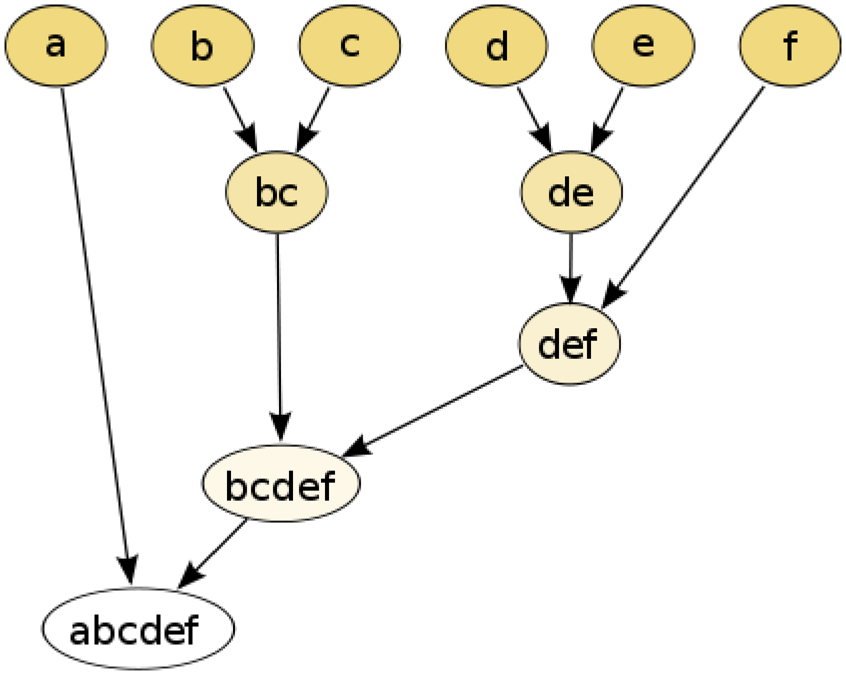
- Signal segmentation (clustering different physiological sources of speech)



- Image segmentation (clustering images based upon their perceptual similarities)
- Clustering web pages based on their content
- Clustering web-search results
- Clustering people in social networks based upon their preferences
- and many others…

Types of clustering

- **Flat or partitional clustering** (K-means, Gaussian mixture models, etc.)
    - partitions are independent of each other
- **Hierarchical clustering** (agglomerative clustering, divisive clustering, etc.)
    - partitions can be visualized using a tree structure
    - does not need the number of clusters as input



K-means clustering algorithm

- Input: number of clusters K, set of points $x_1 \dots x_n$
- Place centroids $c_1 \dots c_n$ at random (or logical) locations
- Repeat until convergence:
  - for each point xi:
    - find nearest centroid c_j
    - assign the point x_i to cluster j

Euclidean distance D between instance x_i and cluster center c_j:

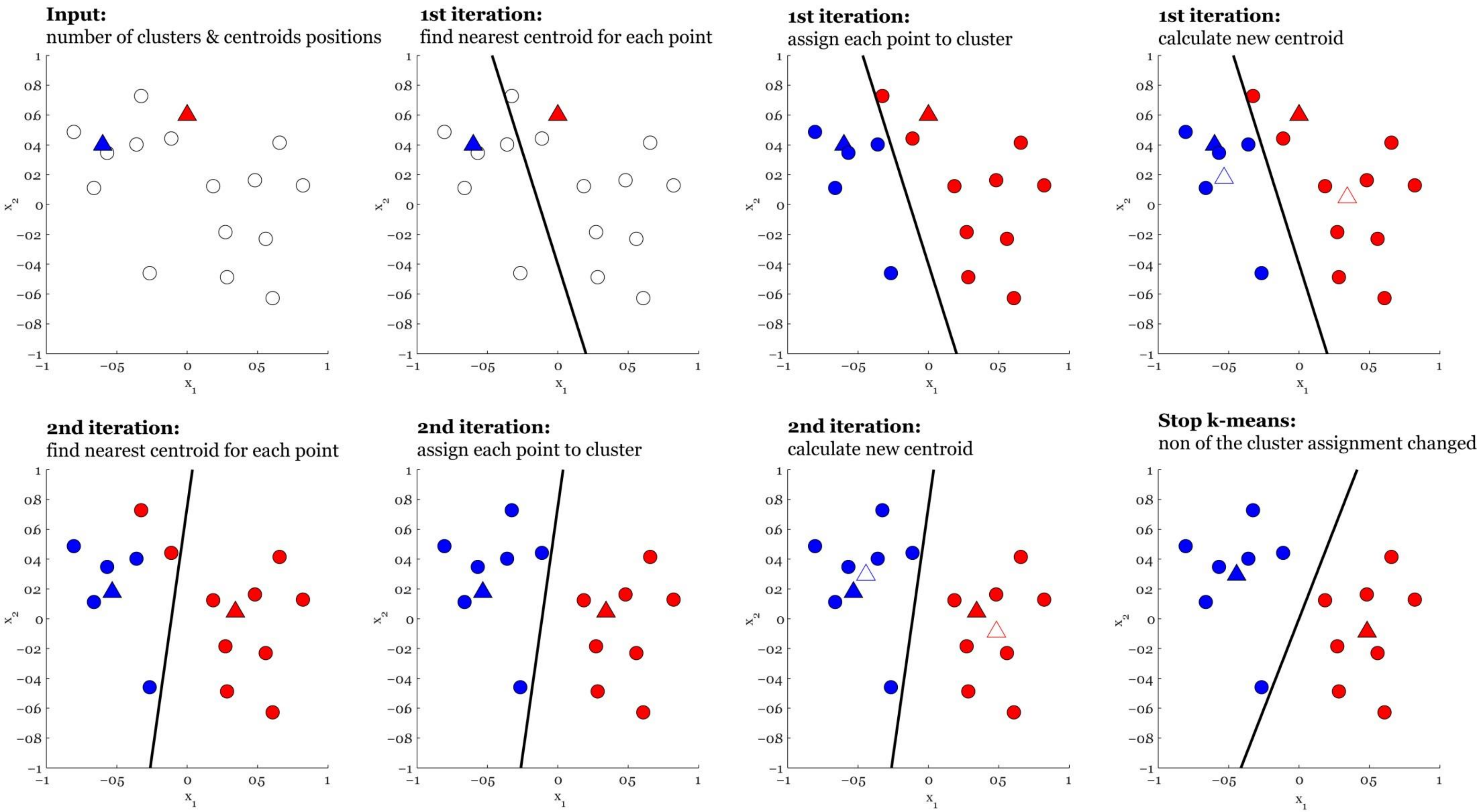
$$arg \min_j D(x_i, c_j)$$

  - for each cluster j = 1 … K
    - new centroid c_j = mean of all points x_i assigned to cluster j in previous step

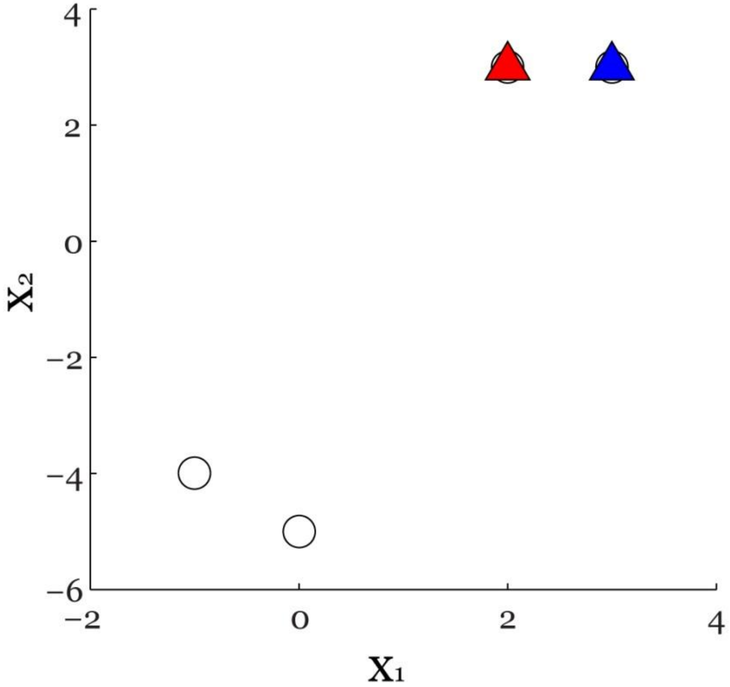
$$c_j = \frac{1}{n_j} \sum_{x_i \rightarrow c_j} x_i$$

- Stop when none of the cluster assignment changed

K-means principle



How to compute K-means?



The data are:

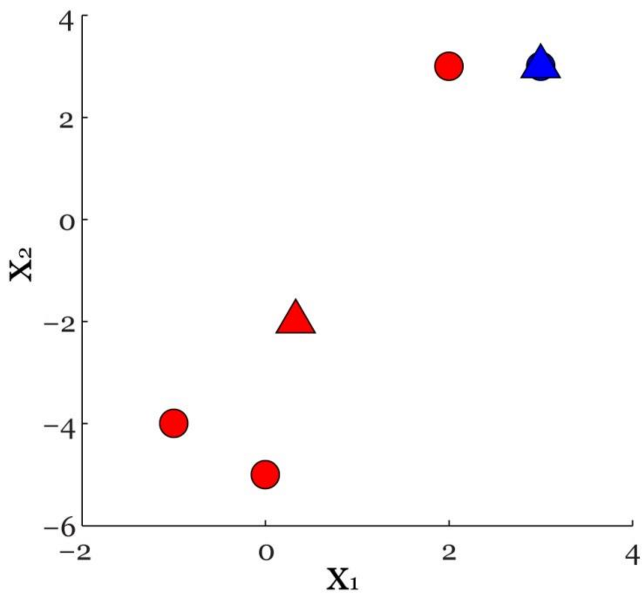
$X=(x_1,x_2)=\{(3,3),(-1,-4),(2,3),(0,-5)\}$

Centroids are:

$c_1=(3,3)$
$c_1=(2,3)$

x_1	x_2	$\sqrt{(x_i-c_1)^2}$	$\sqrt{(x_i-c_2)^2}$	$c_1=(3,3)$	$c_2=(2,3)$
3	3	$\sqrt{(3-3)^2+(3-3)^2}=\sqrt{0}$	$\sqrt{(3-2)^2+(3-3)^2}=\sqrt{1}$	0	1
-1	-4	$\sqrt{(-1-3)^2+(-4-3)^2}=\sqrt{65}$	$\sqrt{(-1-2)^2+(-4-3)^2}=\sqrt{58}$	8.1	7.6
2	3	$\sqrt{(2-3)^2+(3-3)^2}=\sqrt{1}$	$\sqrt{(2-2)^2+(3-3)^2}=\sqrt{0}$	1	0
0	-5	$\sqrt{(0-3)^2+(-5-3)^2}=\sqrt{73}$	$\sqrt{(0-2)^2+(-5-3)^2}=\sqrt{68}$	8.5	8.2

How to compute K-means?



The data are:

$X=(x_1,x_2)=\{(3,3),(-1,-4),(2,3),(0,-5)\}$

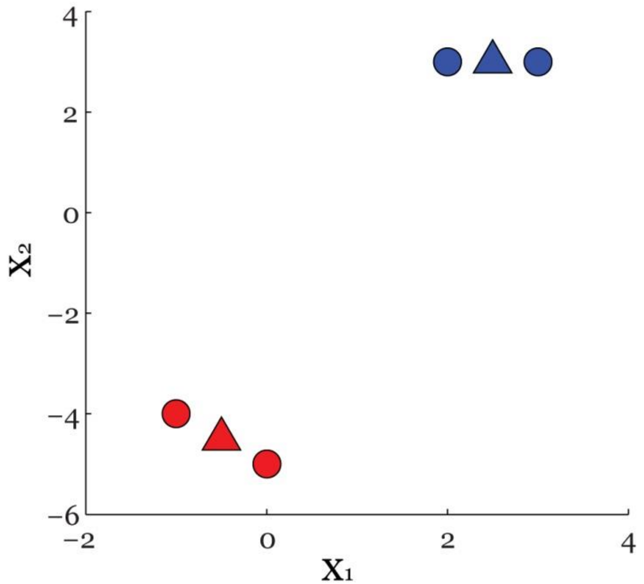
New centroids are (calculated from data with assigned labels):

$c_1=(3,3)$
$c_1=(0.33,-2)$

, i.e. (-1+2+0/3,-4+3-5/3)

x_1	x_2	$\sqrt{(x_i-c_1)^2}$	$\sqrt{(x_i-c_2)^2}$	$c_1=(3,3)$	$c_2=(0.33,-2)$
3	3	$\sqrt{(3-3)^2+(3-3)^2}=\sqrt{0}$	$\sqrt{(3-0.33)^2+(3-(-2))^2}=\sqrt{32.1}$	0	5.7
-1	-4	$\sqrt{(-1-3)^2+(-4-3)^2}=\sqrt{65}$	$\sqrt{(-1-0.33)^2+(-4-(-2))^2}=\sqrt{5.8}$	8.1	2.4
2	3	$\sqrt{(2-3)^2+(3-3)^2}=\sqrt{1}$	$\sqrt{(2-0.33)^2+(3-(-2))^2}=\sqrt{5.3}$	1	5.3
0	-5	$\sqrt{(0-3)^2+(-5-3)^2}=\sqrt{73}$	$\sqrt{(0-0.33)^2+(-5-(-2))^2}=\sqrt{9.1}$	8.5	3

How to compute K-means?



The data are:

$X=(x_1,x_2)=\{(3,3),(-1,-4),(2,3),(0,-5)\}$

New centroids are (calculated from data with assigned labels):

$c_1=(2,5,3)$
$c_1=(-0.5,-4.5)$

LABELS ARE UNCHANGED, NO FURTHER STEP IS REQUIRED!

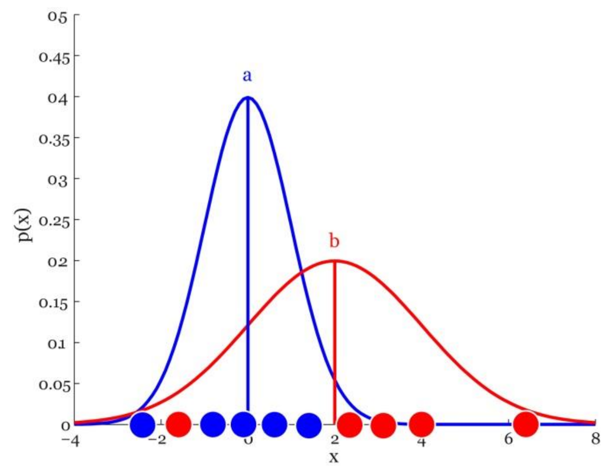
x_1	x_2	$\sqrt{(x_i-c_1)^2}$	$\sqrt{(x_i-c_2)^2}$	$c_1=(2.5,3)$	$c_2=(-0.5,-4.5)$
3	3	$\sqrt{(3-2.5)^2+(3-3)^2}=\sqrt{0.25}$	$\sqrt{(3-(-0.5))^2+(3-(-4.5))^2}=\sqrt{68.5}$	0.5	8.3
-1	-4	$\sqrt{(-1-2.5)^2+(-4-3)^2}=\sqrt{61.3}$	$\sqrt{(-1-(-0.5))^2+(-4-(-4.5))^2}=\sqrt{0.5}$	7.8	0.7
2	3	$\sqrt{(2-2.5)^2+(3-3)^2}=\sqrt{0.25}$	$\sqrt{(2-(-0.5))^2+(3-(-4.5))^2}=\sqrt{62.5}$	0.5	7.9
0	-5	$\sqrt{(0-2.5)^2+(-5-3)^2}=\sqrt{70.3}$	$\sqrt{(0-(-0.5))^2+(-5-(-4.5))^2}=\sqrt{0.5}$	8.4	0.7

Gaussian mixture models (GMM)

- Types of clustering problems
  - hard clustering: clusters do not overlap
  - soft clustering: clusters may overlap
- Mixture models
  - probabilistically-grounded way of doing soft clustering
  - each cluster is a generative model (e.g. Gaussian)
  - parameters (e.g. mean/covariance are unknown)

Gaussian mixture models in 1D

- Observations $x_1 \ldots x_n$
  - K = 2 Gaussians with known μ, σ^2
  - Estimation is trivial if we know the source of each observation

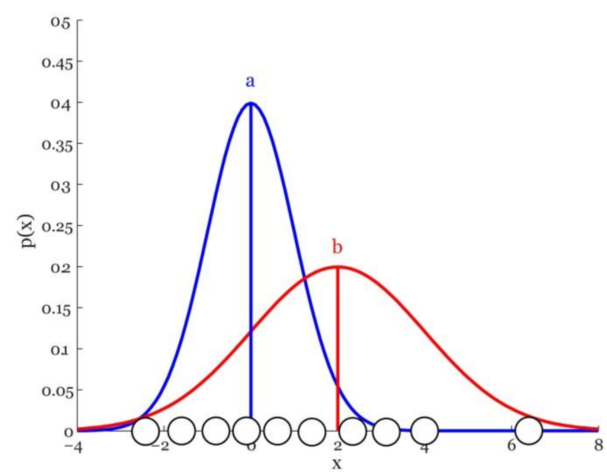


$$\mu_a = \frac{x_1 + x_2 + \cdots + x_n}{n_a} \quad \sigma_a^2 = \frac{(x_1 - \mu_1)^2 + \cdots + (x_n - \mu_n)^2}{n_a}$$

- What if we do not know the source (μ, σ^2)?
  - We can guess whether point is more likely to be "a" or "b"

$$P(a|x_i) = \frac{P(x_i|a)P(a)}{P(x_i|a)P(a) + P(x_i|b)P(b)}$$

$$P(x_i|a) = \frac{1}{\sqrt{2\pi\sigma_a^2}} exp\left(-\frac{(x_i - \mu_a)}{2\sigma_a^2}\right)$$

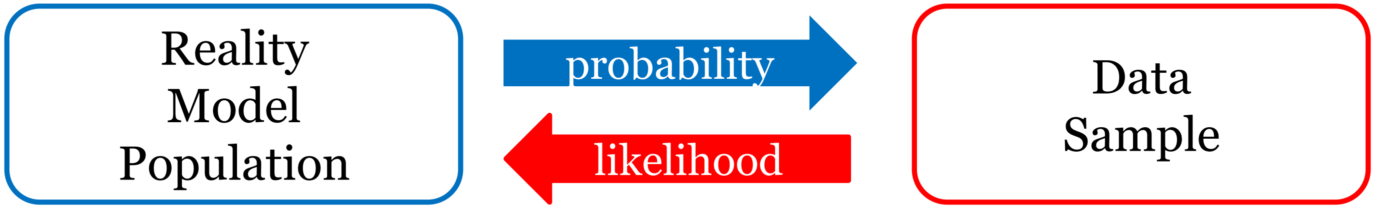


Bayes' theorem

$$P(A|x) = \frac{P(x|A)P(A)}{P(x|A)P(A) + P(x|B)P(B)}$$

- P(A|x) = Chance of having *disease* (A) given a positive test (x). This is our main question: How likely is it to have disease with a positive result?
- P(x|A) = Chance of positive test (x) given that you have *disease* (A), i.e. TRUE POSITIVE.
- P(A) = Chance of having *disease* (A).
- P(B) = Chance of *not having disease*.
- P(x|B) = Chance of positive test (x) given that you *do not have the disease*, i.e. FALSE POSITIVE.

What is Likelihood?

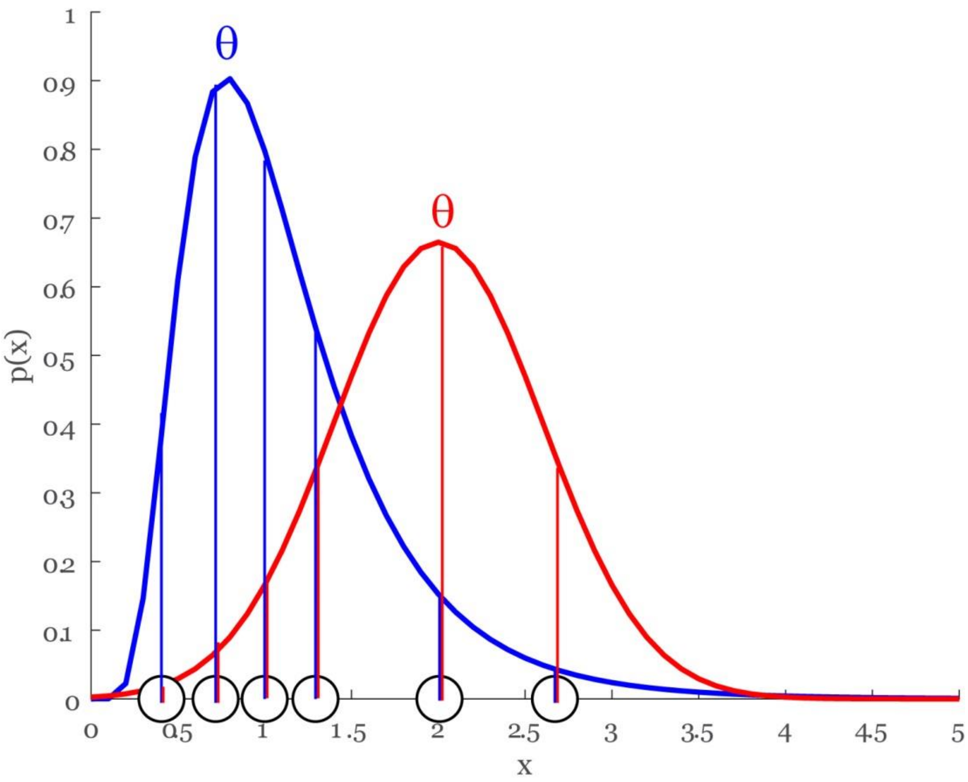


- **Probability**: What is the chance of observing particular data or sample given a specific model of population?
- If the true prevalence of Parkinson's disease (PD) is 0.3% in population, what is the chance of finding 1 subject at high risk of PD in sample of 100 healthy controls?
- If probability distribution is normal (μ, σ), what is the chance of observing x?

- **Likelihood**: Given observed data, what is the chance that a given reality or model is true?
- If we found 1 subject at high risk of PD in sample of 100 healthy controls, what is the probability that true prevalence of PD in population is 0.3%?
- If you observe x, what is the best normal distribution (μ, σ)?

Maximum Likelihood Estimation

- Determine best model parameters that fit given data by maximizing log-likelihood function to estimate those parameters



- $P(x_1, x_2, \ldots, x_n|\theta)$ is probability observing $x_i \ldots x_n$ given parameters of distribution θ
- $L(\theta|x) = P(x_1|\theta) \,.\, P(x_2|\theta). \ldots P(x_n|\theta) = \prod P(x_i|\theta)$

Expectation Maximization (EM)

- Chicken and egg problem
  - we need (μ_a, σ_a^2) and (μ_b, σ_b^2) to guess source of points
  - we need to know source of points to estimate (μ_a, σ_a^2) and (μ_b, σ_b^2)
- EM algorithm

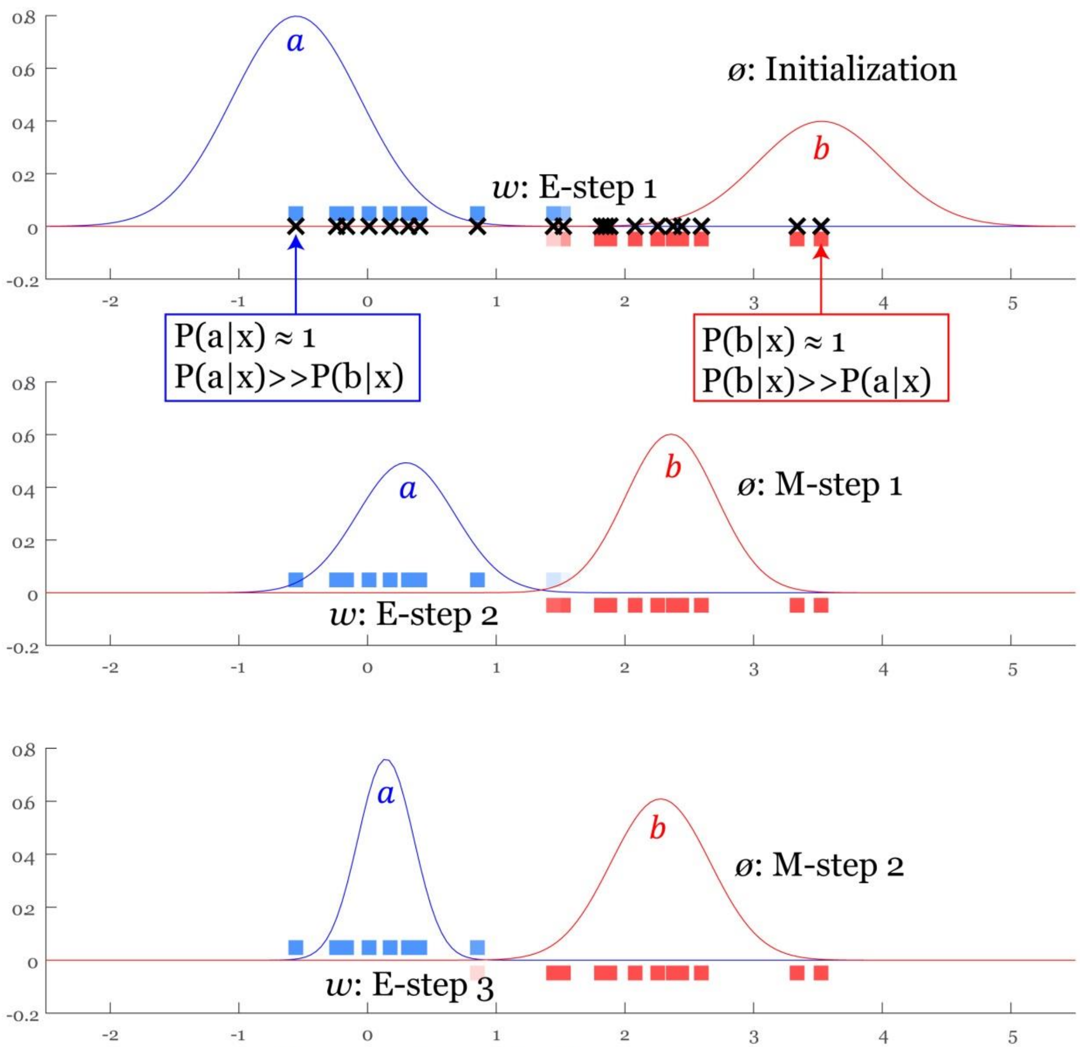
Init • start with two randomly placed Gaussians (μ_a, σ_a^2) and (μ_b, σ_b^2)

E-step • for each point estimate $P(a|x_i)$ (i.e., does x_i looks like it came from a?)

M-step • adjust (μ_a, σ_a^2) and (μ_b, σ_b^2) to fit points assigned to them

• iterate to convergence

EM algorithm: 1D example



ø: M-step 3 no further step is required

Initialization

$$\mu_a = -0.5, \sigma_a = 1 \; (\phi_a = 1)$$

$$\mu_b = 3.5, \sigma_b = 1 \; (\phi_b = 0.5)$$

Expectation

$$P(x_i|a) = \frac{1}{\sqrt{2\pi\sigma_a^2}} exp\left(-\frac{(x_i - \mu_a)}{2\sigma_a^2}\right)$$

Probability **D**ensity **F**unction

$$a_i = P(a|x_i) = \frac{P(x_i|a)P(a)}{P(x_i|a)P(a) + P(x_i|b)P(b)}$$

ϕ_a

weights w

$$b_i = P(b|x_i) = 1 - P(a|x_i)$$

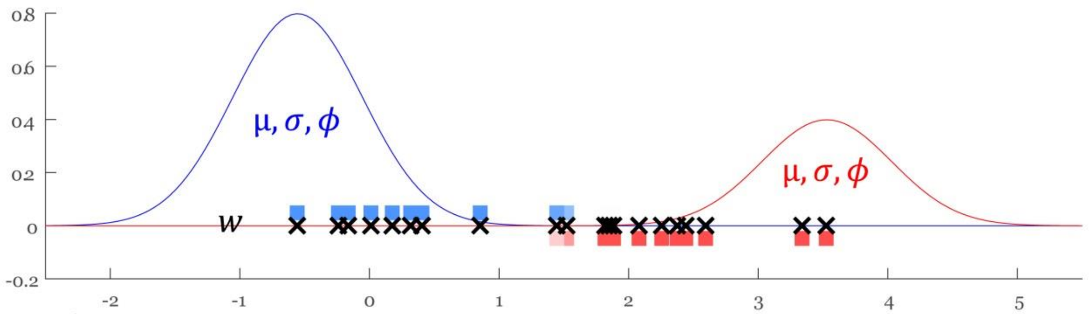
Maximization

$$\mu_a = \frac{a_1x_1 + a_2x_2 + \cdots + a_nx_n}{a_1 + a_2 + \cdots + a_n}$$

K-means: a = 0 or 1

$$\sigma_a^2 = \frac{a_1(x_1 - \mu_1)^2 + \cdots + a_n(x_n - \mu_n)^2}{a_1 + a_2 + \cdots + a_n}$$

GMM EM calculation



Initialization

μ, σ, ϕ

Expectation

$$g_j(x) = \frac{1}{\sqrt{(2\pi)^n |\sigma_j|}} exp\left(-\frac{1}{2}(x-\mu_j)^T \sigma^{-1}(x-\mu_j)\right)$$

$$w_j^{(i)} = \frac{g_j(x)\phi_j}{\sum_{l=1}^{k} g_l(x)\phi_l}$$

Maximization

$$\phi_j = \frac{1}{m}\sum_{i=1}^{m} w_j^{(i)}$$

$$\mu_j = \frac{\sum_{i=1}^{m} w_j^{(i)} x^{(i)}}{\sum_{i=1}^{m} w_j^{(i)}}$$

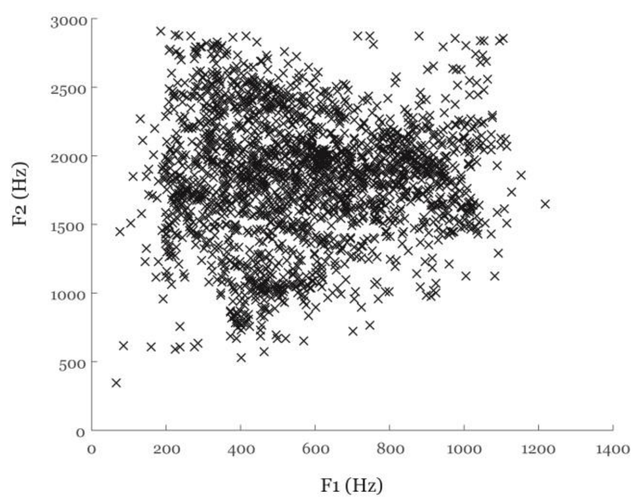
$$\sigma_j = \frac{\sum_{i=1}^{m} w_j^{(i)}(x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T}{\sum_{i=1}^{m} w_j^{(i)}}$$

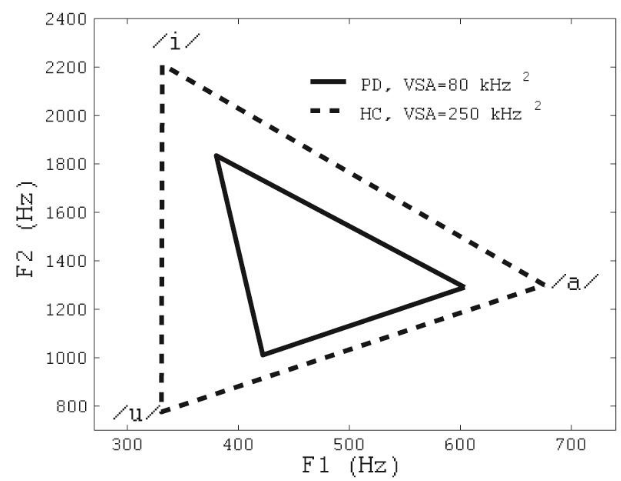
Symbol	**Definition**
$g_j(x)$	PDF of the multivariate Gaussian for cluster j
j	Cluster number
x	Input vector (a column vector)
n	Input vector length
σ_j	n × n covariance matrix for cluster j
$\lvert\sigma_j\rvert$	Ddeterminant of covariance matrix
σ_j^{-1}	Inverse of covariance matrix
w_j	Probability that example "i" belongs to cluster j
ϕ_j	Prior probability of cluster j
k	Number of clusters

Data

Matlab example 1

The quality and intelligibility of each vowel can be determined primarily by the distinctive acoustic energy peak of the first (F1) and second (F2) formant frequencies. The F1 and F2 frequencies particularly reflect tongue position, with the acoustic-articulatory relationship defined such that the F1 frequency varies inversely with tongue height and the F2 frequency varies directly with tongue advancement. Thus, limited articulatory range of motion due to Parkinson's disease (PD), resulting in the overall reduction of working space for vowels, can be captured well by a reduced size of the vowel space area (VSA), which is constructed by the Euclidean distances between the F1 and F2 coordinates of the corner vowels /a/, /i/, and /u/ in the triangular F1-F2 vowel space.





Gentle introduction to covariance matrix

by

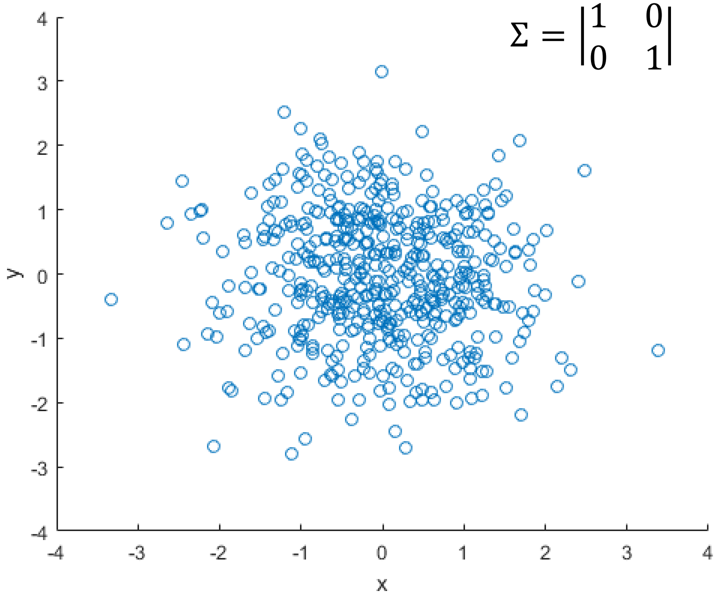
Jan Hlavnička, 2018

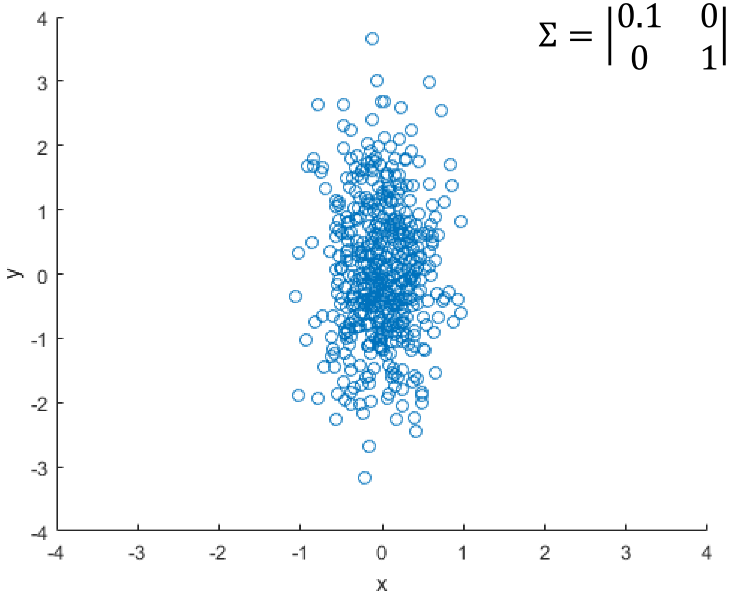
Covariance matrix

$$\Sigma = \begin{vmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{yx} & \sigma_{yy} \end{vmatrix} \quad ?$$

Diagonal covariance matrix

$$\Sigma = \begin{vmatrix} \sigma_{xx} & 0 \\ 0 & \sigma_{yy} \end{vmatrix}$$





Correlation is "normalized covariance"

$$\Sigma = \begin{vmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{yx} & \sigma_{yy} \end{vmatrix}$$

σ_{xx} … variance

σ_{xy} … covariance

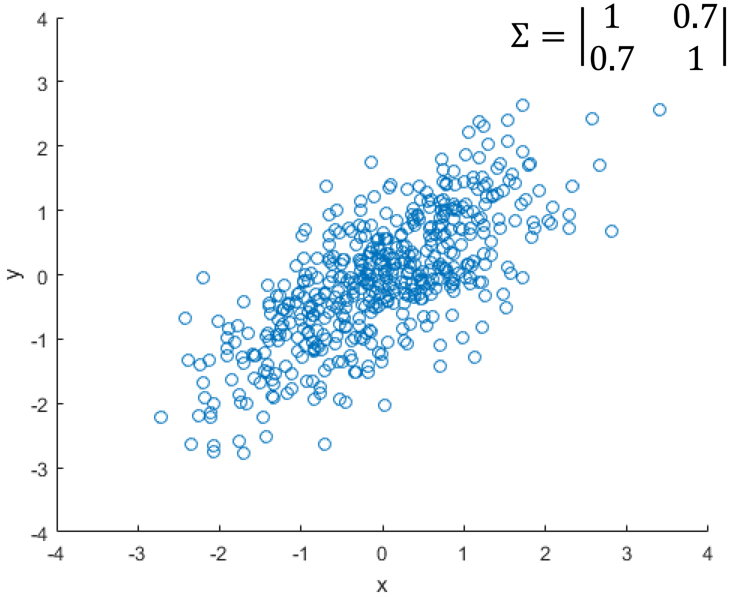
$$R_{xy} = \frac{\sigma_{xy}}{\sigma_x \cdot \sigma_y}$$

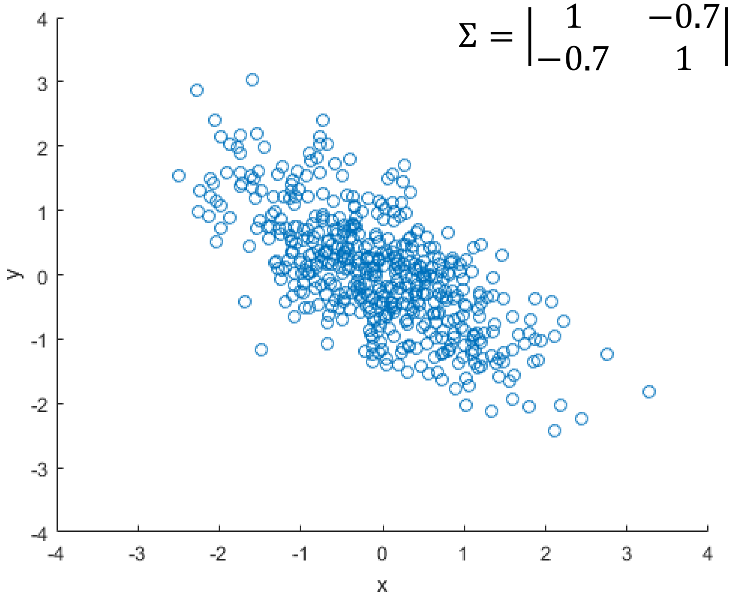
σ_x … standard deviation

$$R_{xy} = \frac{\sigma_{xy}}{\sqrt{\sigma_{xx} \cdot \sigma_{yy}}}$$

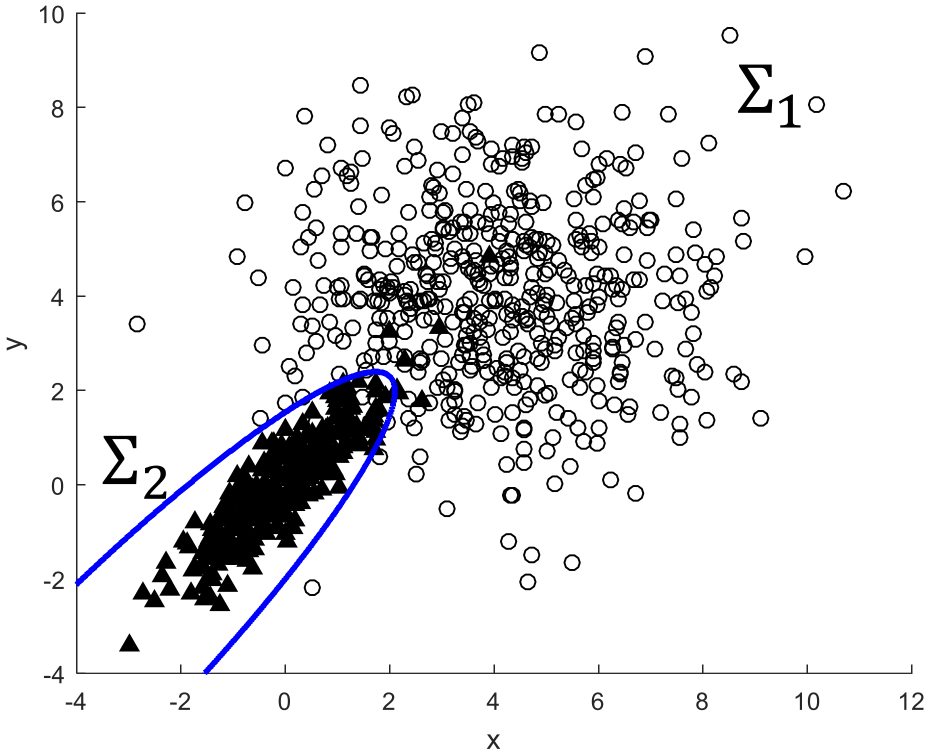
Geometric attributes

$$\Sigma = \begin{vmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{yx} & \sigma_{yy} \end{vmatrix}$$





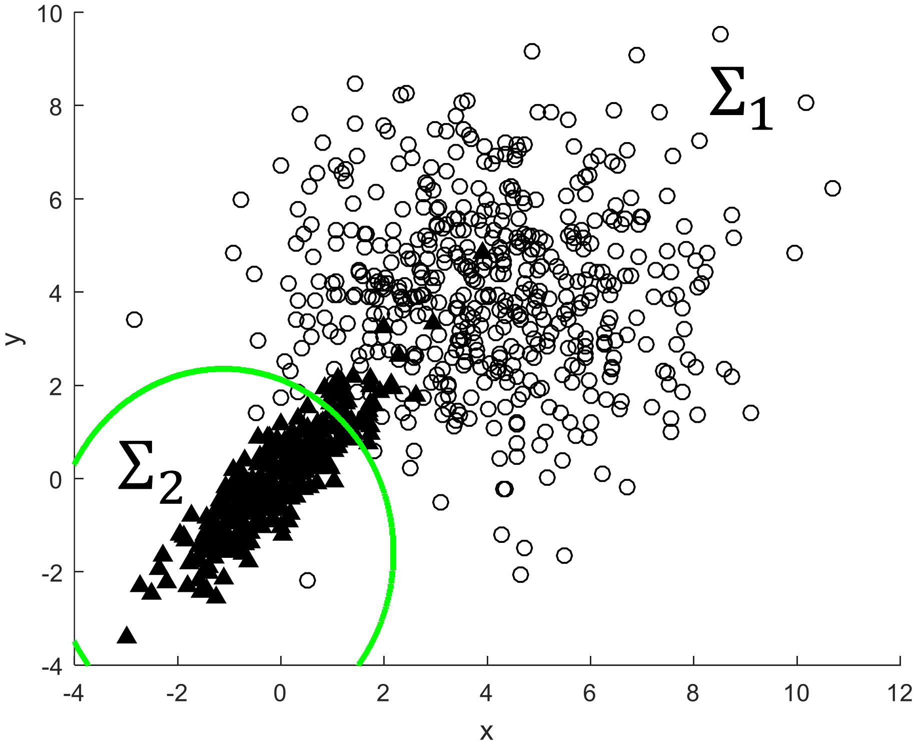
Decision by full covariance matrix



$$\Sigma = \begin{vmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{yx} & \sigma_{yy} \end{vmatrix}$$

$$\Sigma_1 \neq \Sigma_2$$

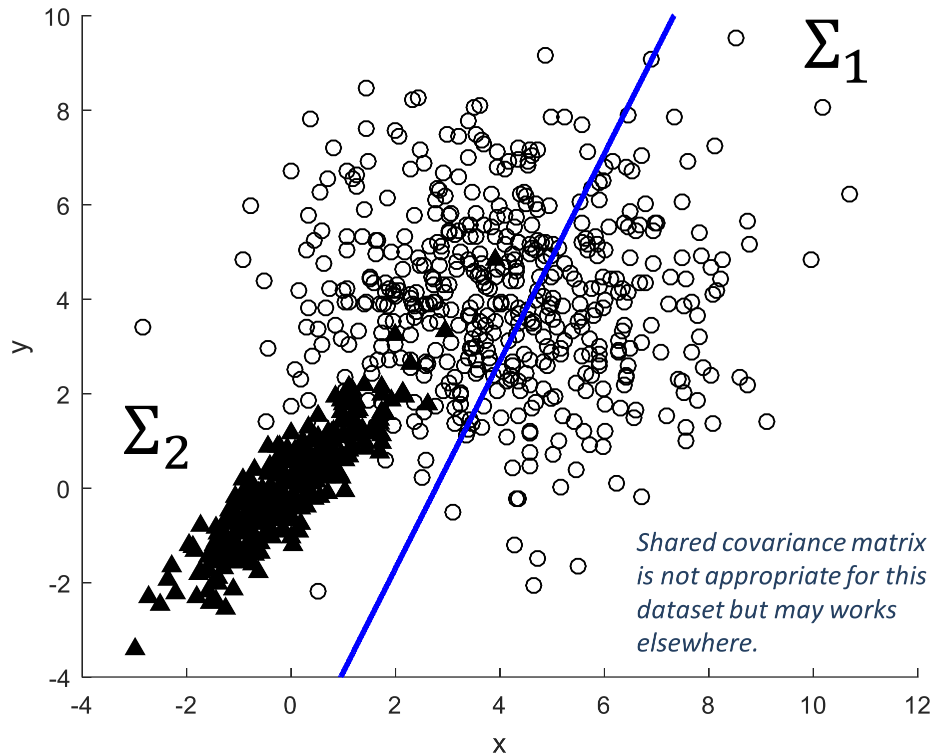
Decision by diagonal covariance matrix



$$\Sigma = \begin{vmatrix} \sigma_{xx} & 0 \\ 0 & \sigma_{yy} \end{vmatrix}$$

$$\Sigma_1 \neq \Sigma_2$$

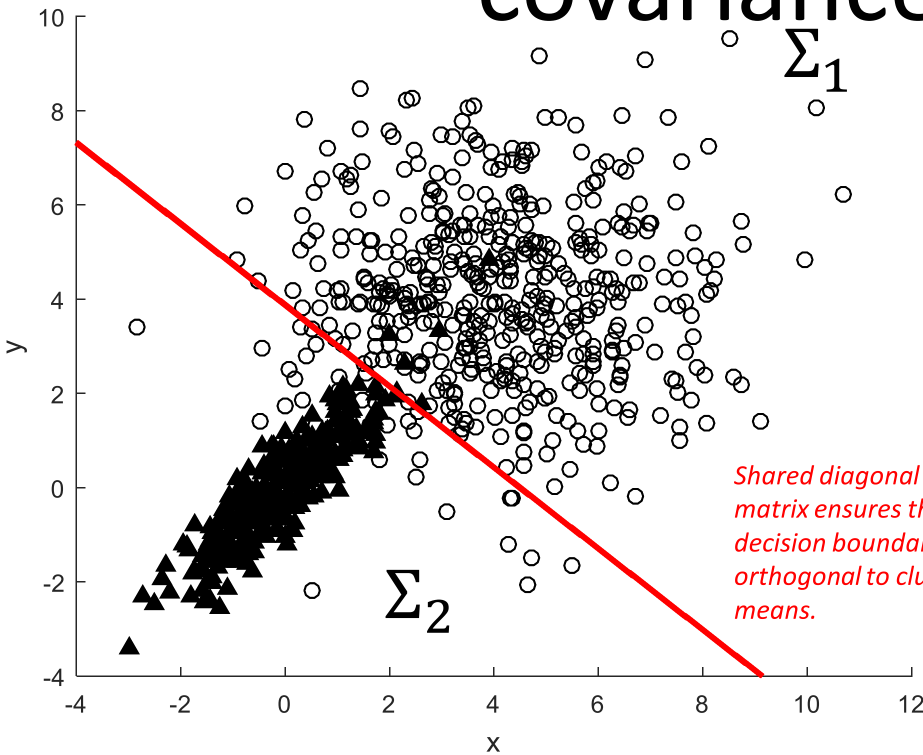
Decision by shared full covariance matrix



$$\Sigma = \begin{vmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{yx} & \sigma_{yy} \end{vmatrix}$$

$$\Sigma_1 = \Sigma_2$$

Decision by shared diagonal covariance matrix



Shared diagonal covariance matrix ensures that decision boundary will be orthogonal to clusters' means.

$$\Sigma = \begin{vmatrix} \sigma_{xx} & 0 \\ 0 & \sigma_{yy} \end{vmatrix}$$

$$\Sigma_1 = \Sigma_2$$