Unsupervised Classification (Clustering)

Introduction to MAchine Learning

Mathilde Mougeot

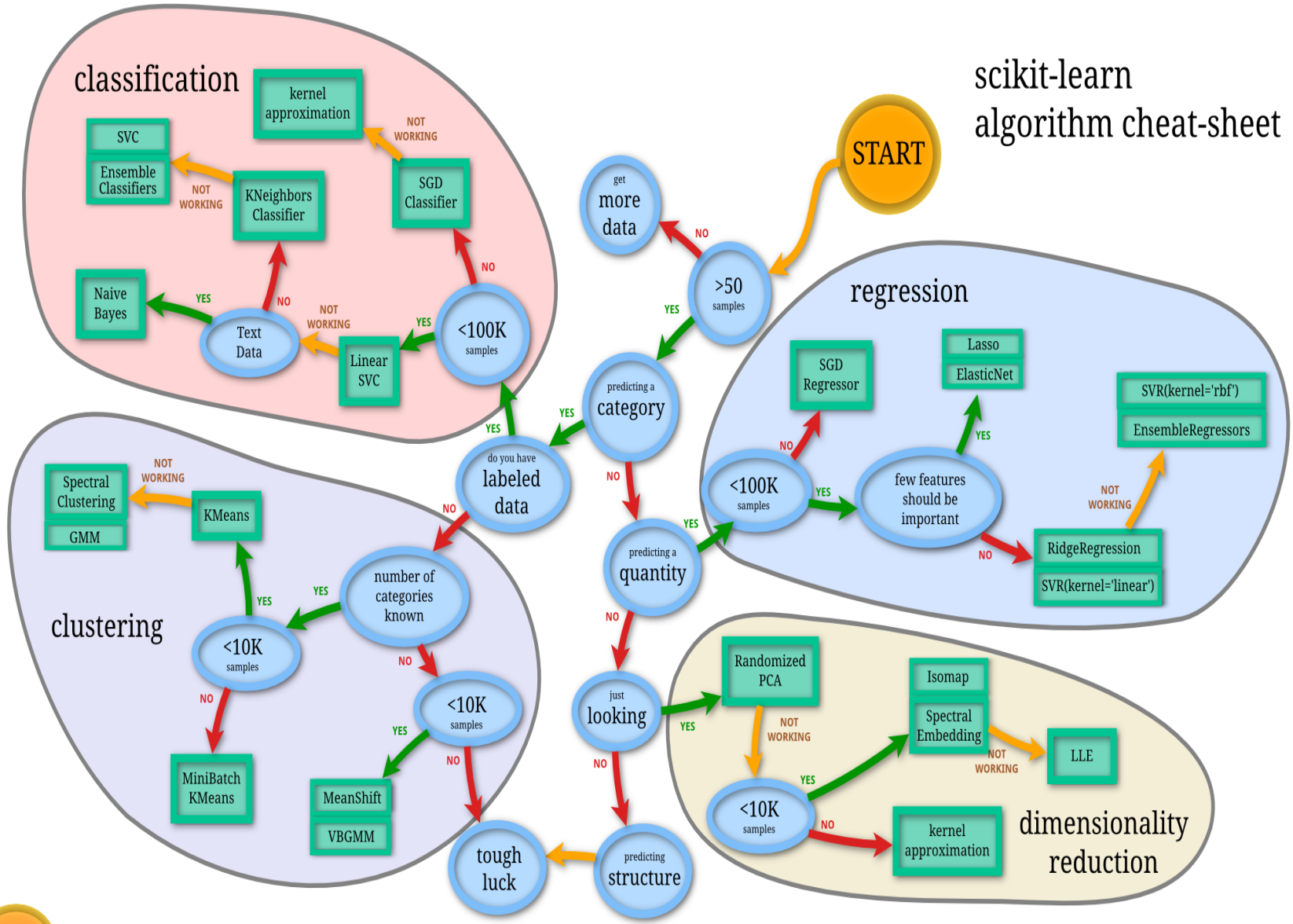
ENSIIE

2019

At this stage

- Outline
  - Observations $(X_1, \ldots, X_n)$ in $\mathbb{R}^d$, d may be large
  - Target $(Y_1, \ldots, Y_n)$ labels (classification) or continuous (regression)
  - Goal : Predictive modeling : understand the link between $X \mapsto Y$, reduce d.

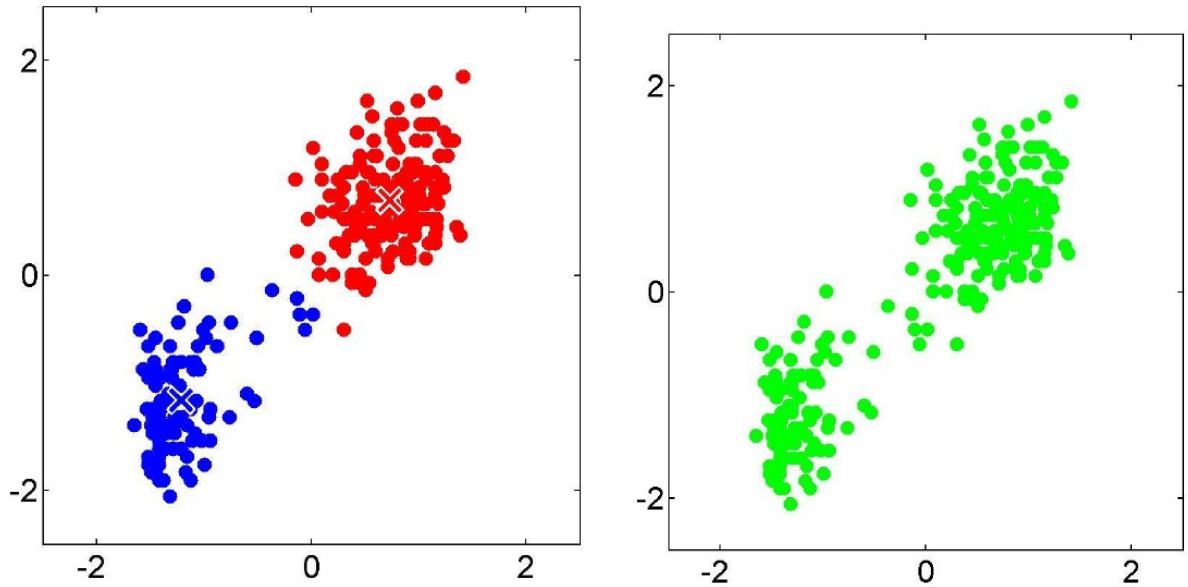
- Methods
  1. **Classification** models (Parametric/ Non Parametric)
     Linear Quadratic Discriminant Analysis, Logistic, KNN CART, Bagging, Random Forest
  2. **Regression** models (Parametric/ Non Parametric)...
     Linear models, Linear models with penalization : LASSO, Ridge, KNN, CaRt, Bagging, Random Forest...
  3. Clustering (unsupervised Classification)


scikit-learn
algorithm cheat-sheet
START
>50 samples
NO
get more data
YES
predicting a category
YES
do you have labeled data
YES
<100K samples
NO
SGD Classifier
NOT WORKING
kernel approximation
NOT WORKING
KNeighbors Classifier
NOT WORKING
SVC
Ensemble Classifiers
NO
Linear SVC
NOT WORKING
NO
Text Data
YES
Naive Bayes
classification
NO
predicting a quantity
YES
<100K samples
YES
few features should be important
NO
RidgeRegression
SVR(kernel='linear')
NOT WORKING
SVR(kernel='rbf')
EnsembleRegressors
YES
Lasso
ElasticNet
NO
SGD Regressor
regression
NO
just looking
YES
Randomized PCA
NOT WORKING
<10K samples
YES
Isomap
Spectral Embedding
NOT WORKING
LLE
NO
kernel approximation
dimensionality reduction
NO
predicting structure
tough luck
NO
<10K samples
YES
MeanShift
VBGMM
NO
<10K samples
YES
MiniBatch KMeans
NO
number of categories known
YES
<10K samples
YES
KMeans
NOT WORKING
Spectral Clustering
GMM
NO
clustering
Back
scikit learn

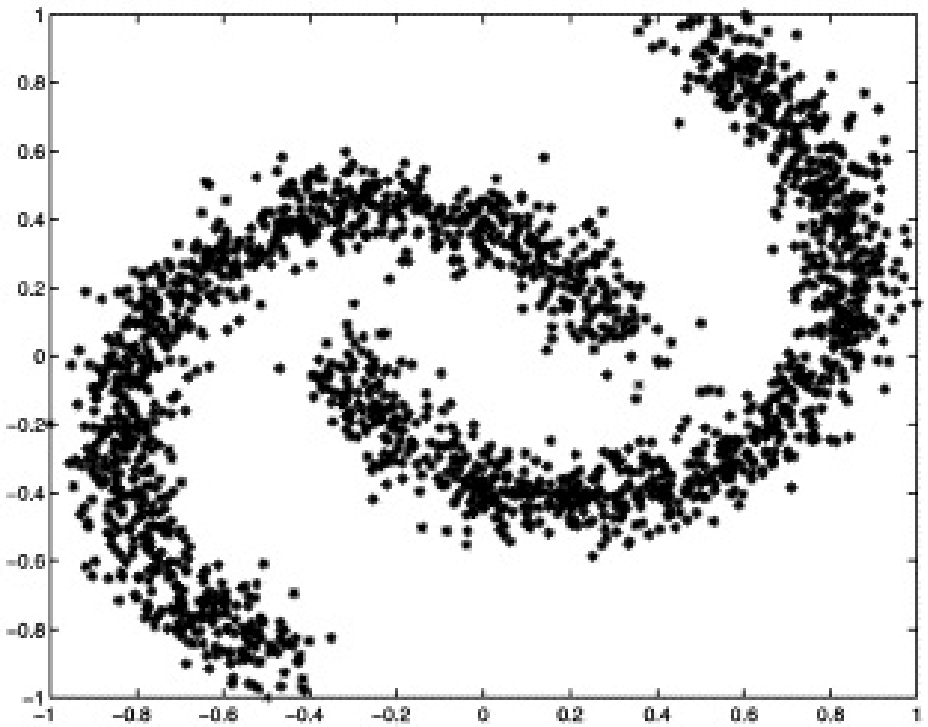
Outline

Motivations

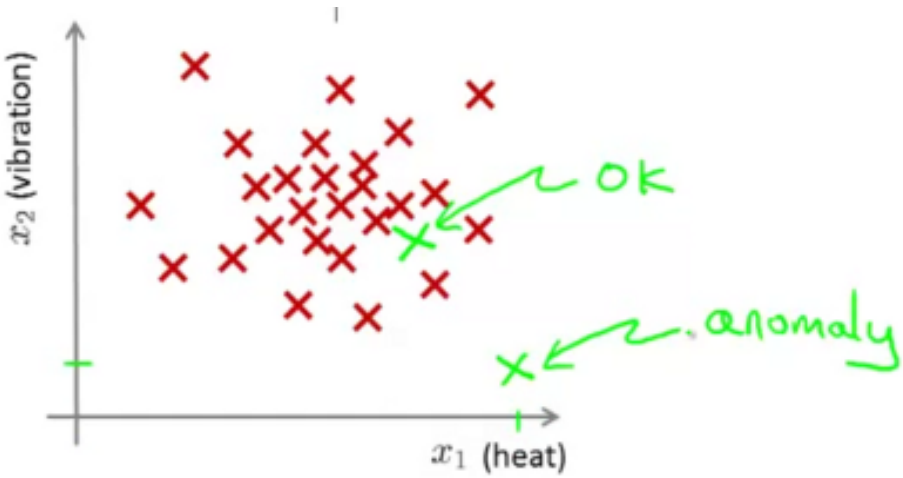
Unsupervised data (1) - Clustering

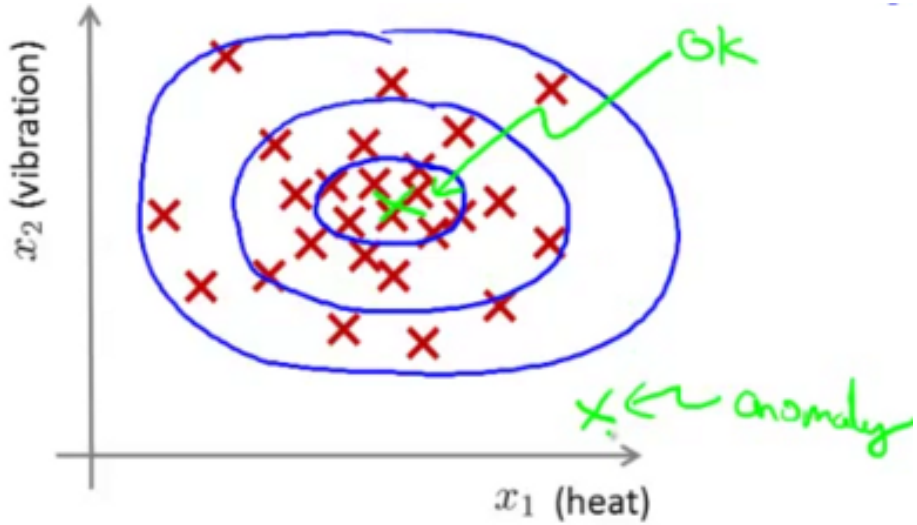


Clustering can be difficult !

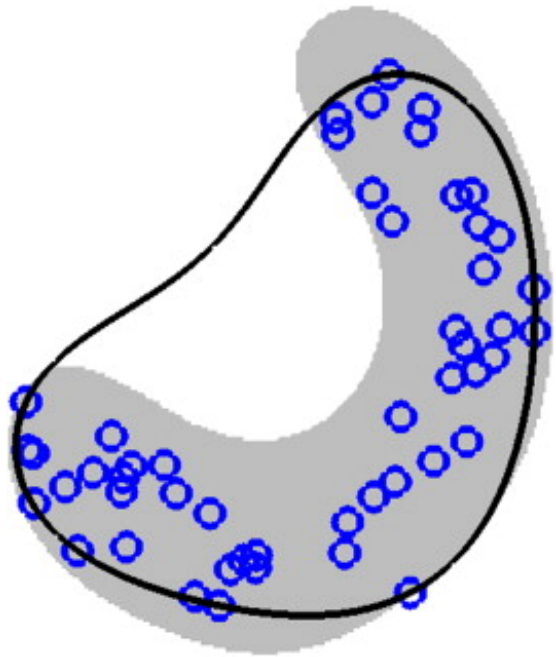


Unsupervised data (2) - Anomaly/Mode detection





Unsupervised data (3) - Novelty detection



Distance-based clustering

-

Notations

Clustering input : distance matrix

- Data matrix
  - Individual index $i \in \{1, \ldots, n\}$
  - Feature index $p \in \{1, \ldots, d\}$
  - Measurements x_{ip}
- Distance matrix
  - p-th feature distance between individuals i and $j = d_p(x_i, x_j)$
  - Distance between individuals i and j :

$$D(x_i, x_j) = \sum_{p=1}^{d} w_p d_p(x_i, x_j)$$

where w_p p-th feature importance, $w_p > 0$

Examples of distances

- Quantitative features
  - Squared distance or absolute difference
  - 1-correlation

Examples of distances

- Quantitative features
  - Squared distance or absolute difference
  - 1-correlation
- Discrete ordinal variables
  - Equidistance encoding

Examples of distances

- Quantitative features
  - Squared distance or absolute difference
  - 1-correlation
- Discrete ordinal variables
  - Equidistance encoding
- Categorical variables
  - Zero-one distance
- What if missing values ?

Cluster dispersion functions

- Encoder function $C : \{1, \ldots, n\} \mapsto \{1, \ldots, K\}$ (point to cluster)
- Within-cluster dispersion

$$W(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(j)=k} D(x_i, x_j)$$

Cluster dispersion functions

- Encoder function $C : \{1, \ldots, n\} \mapsto \{1, \ldots, K\}$ (point to cluster)
- Within-cluster dispersion

$$W(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(j)=k} D(x_i, x_j)$$

- Between-cluster dispersion

$$B(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(j)\neq k} D(x_i, x_j)$$

- Total dispersion : $T = W(C) + B(C)$

Clustering method #1 - Brute force

- Combinatorial assignment
- Number of possibilities for assigning n points to K clusters

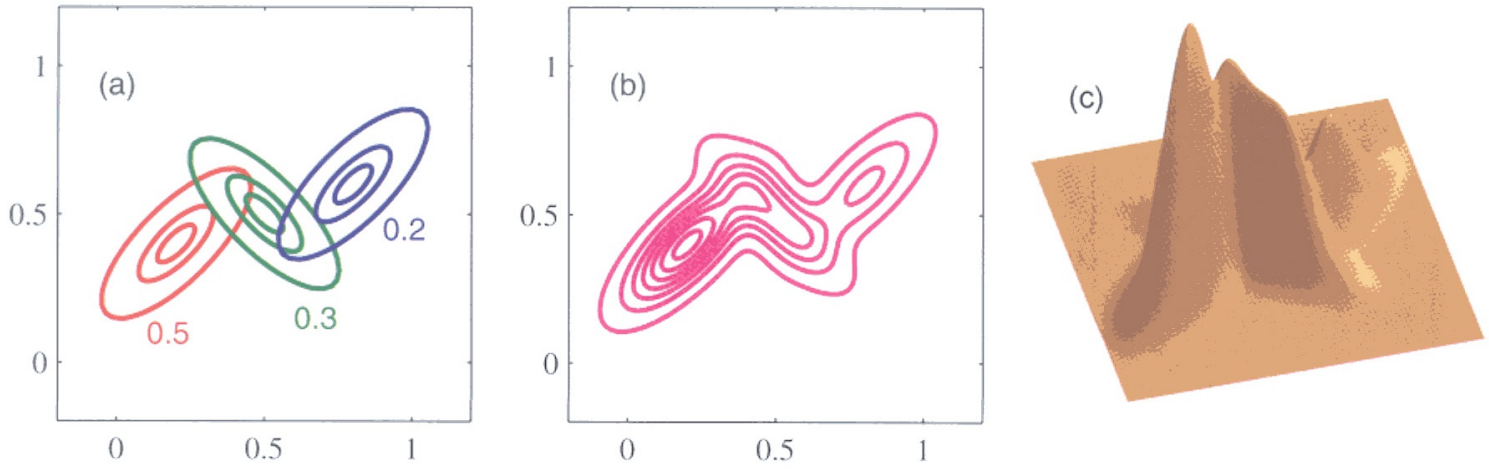
$$S(n, K) = \frac{1}{K!} \sum_{k=1}^{K} (-1)^{K-k} \left(\begin{array}{c} K \\ k \end{array} \right) k^n$$

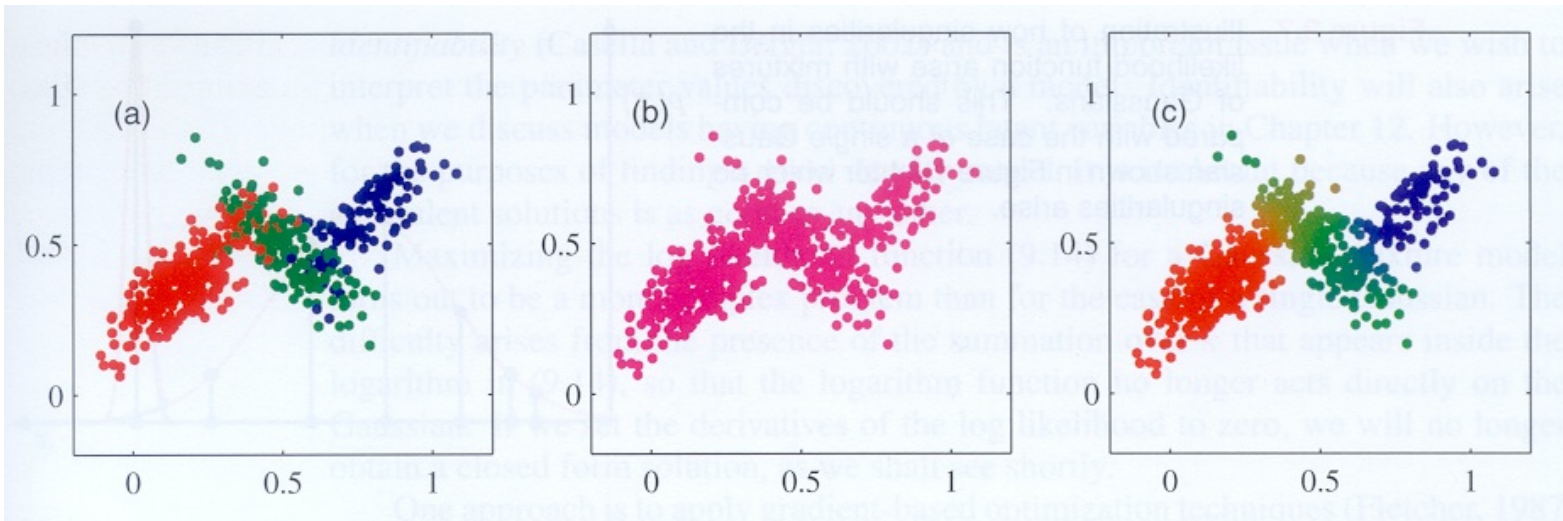
Example : $S(19, 4) \simeq 10^{10}$

- Question : Limited search vs. approximate solution

Parametric approach

Density estimation (Course #1)





Reminder on gaussian mixture models

- Random vector X on $\mathbb{R}^d$ with K components
- Gaussian densities f_k , $k = 1, \ldots, K$,
- Component parameters (μ_k, Σ_k),
- Mixture parameter $p = (p_1, \ldots, p_K)$ in the simplex
- Distribution of X = Mixture density

$$f_X(x) = \sum_{k=1}^{K} p_k f_k(x) \quad , \quad \forall x \in \mathbb{R}^d$$

- For estimation, use EM algorithm...

Complexity of the problem depends on...

- The dimension d

Complexity of the problem depends on...

- The dimension d
- The number of clusters K

Complexity of the problem depends on...

- The dimension d
- The number of clusters K
- The number of samples n

Complexity of the problem depends on...

- The dimension d
- The number of clusters K
- The number of samples n
- The smallest mixture coefficient "$\min_j p_j$"

Complexity of the problem depends on...

- The dimension d
- The number of clusters K
- The number of samples n
- The smallest mixture coefficient "$\min_j p_j$"
- How separated the clusters are...

High dimension

Distance between observations

–	X^1	X^2	...	X^j	...	X^d
1	x_{11}		...	x_{1j}		x_{1d}
2						
...						
$\rightarrow i$	x_{i1}		...	x_{ij}		x_{id}
...						
n	x_{n1}		...	x_{nj}		x_{nd}

- For two observations (x_i, x_k), $x_i \in \mathbb{R}^d$, $x_k \in \mathbb{R}^d$

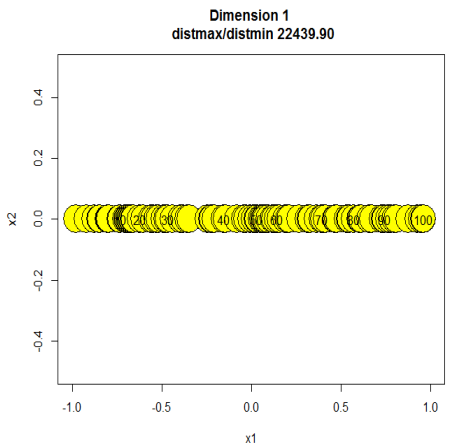
$\rightarrow$ Euclidian distance ℓ_2 between two observations

$$||x_i - x_k||_{\ell_2} = \sqrt{\sum_{j=1}^{d} (x_i(j) - x_k(j))^2}$$

Dimension curse

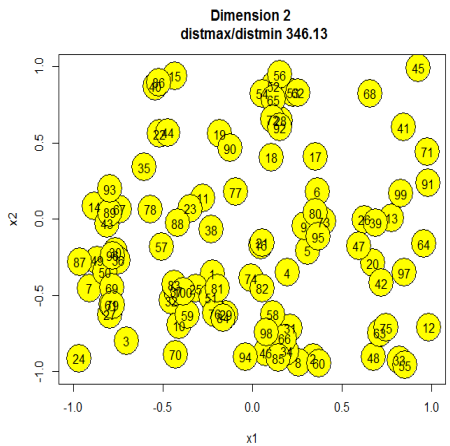
- Evaluation of the distance between two observations in dimension d
- Illustrations : $n = 100$ observations uniformly distributed, 1, 2, 3, ...
- Indicator : $\frac{max_{i \neq j} ||x_i - x_k||_{\ell_2}}{min_{i \neq j} ||x_i - x_k||_{\ell_2}}$

$d = 1$



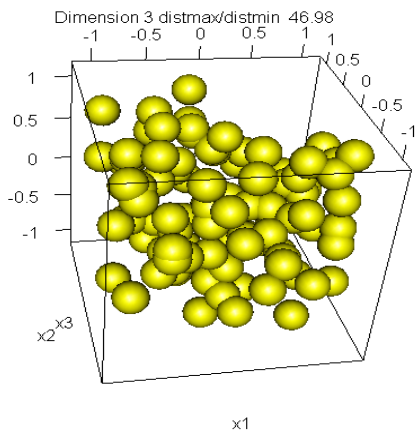
22 435

$d = 2$



346

$d = 3$

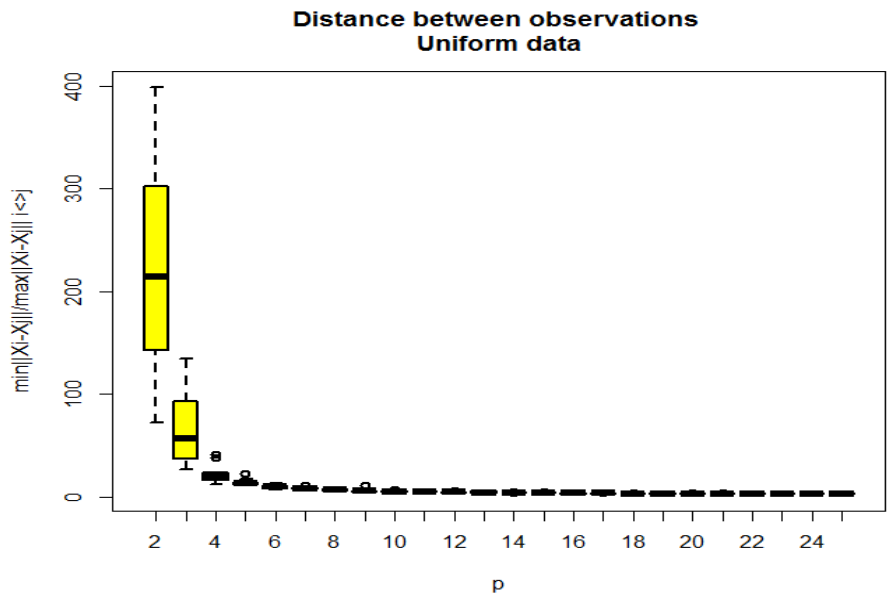


47

Dimension curse

Ratio study $\frac{max_{i \neq j} ||x_i - x_j||_{\ell_2}}{min_{i \neq j} ||x_i - x_j||_{\ell_2}}$ function of the dimension d

Illustration : $n = 100$ observations uniformly distributed ($K = 100$ repetitions)



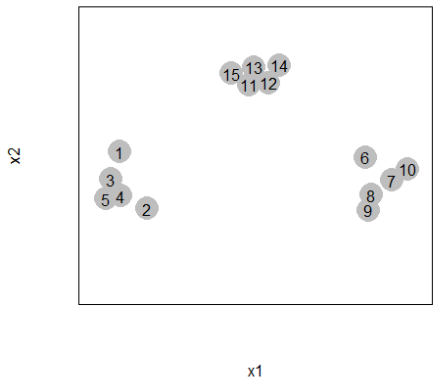
$\rightarrow$ The value of the ratio tends to ~ 1 when d increases.

$\rightarrow$ The euclidian distance loses its discrimination ability in high dimension

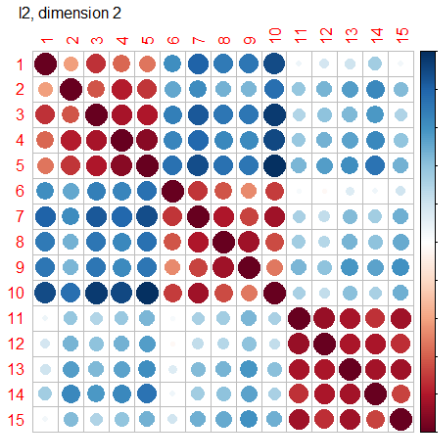
$\rightarrow$ Serious problem especially for segmentation tasks...

Data segmentation (d=2)

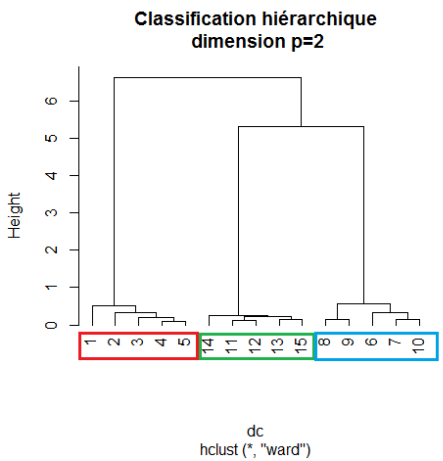
Observations



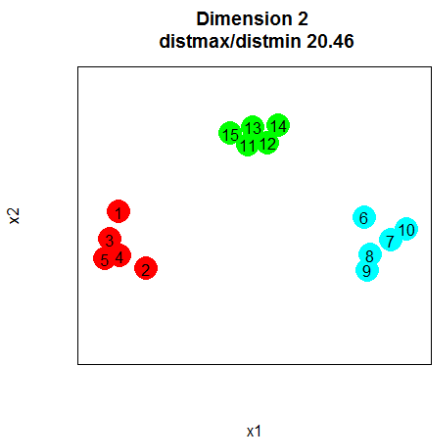
distance matrix



HAC



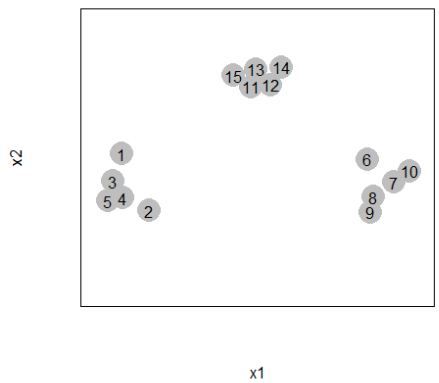
3 classes Clustering



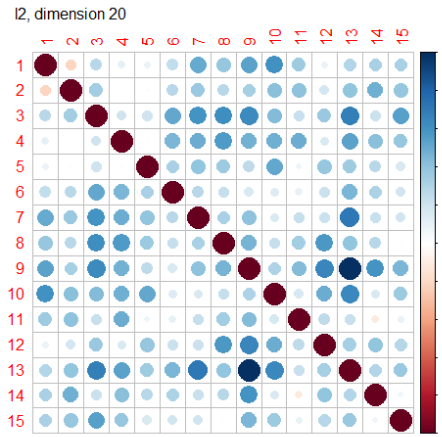
Data segmentation (d=20)

data are embedded in a high dimensional space $d = 20 = 2 + 18$

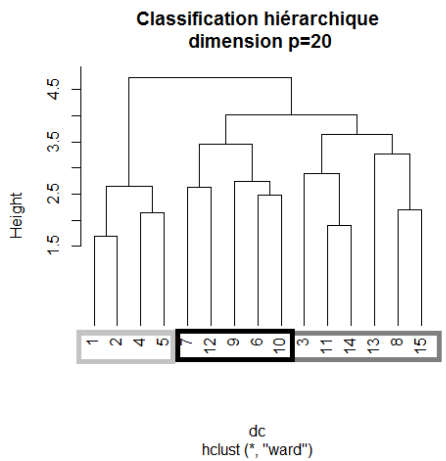
Observations



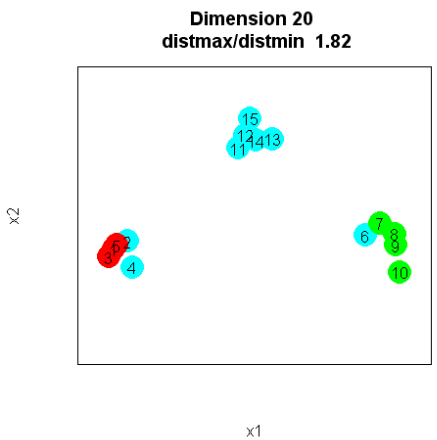
distance matrix



HAC



3 classe Clustering



Dimension reduction

Find good representations of the data initially coded in large dimensions

- Features : a small number of discriminant features
  based on data expertise or automatic extraction.
- Compress Sensing : sparse representation (S) of x based on a linear combinaison of p vectors.
- Manifold estimation : x is represented in a low-dimensional space using the Laplacian eigenvectors on the variety, estimated from a graph of neighborhoods using the examples

$\rightarrow$ Mathematical tools at the interface of harmonic analysis, geometry, probability and statistics.

Model based Clustering

Model-based clustering

Set of observations $\{x_i, x_i \in \mathbb{R}^d, 1 \leq i \leq N\}$

Assumptions : Mixture of K gaussian : $f_X(x) = \sum_{k=1}^{K} \pi_k f_k(x)$
μ_k (means), Σ_k (covariances), π_k (mixing coefficients)
$1 \leq k \leq K$

Find ? : μ_k, Σ_k, π_k

using the EM Algorithm :

1. **Initialization**
2. **E Step :** Expectation Step
3. **M Step :** Maximization Step
4. **LogLikelihood computation**

EM for gaussian mixture (1/4)

1. **Initialization** :
   μ_k (means), Σ_k (covariances), π_k (mixing coefficients)
   Compute Log Likelihood

$$\ln p(X|\mu, \Sigma, \pi) = \sum_{n=1}^{N} \ln\{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)\}$$

2. **E Step :**
3. **M Step :**
4. **Evaluate the log likelihood :**

EM for gaussian mixture (2/4)

1. **Initialization** :μ_k, Σ_k, π_k, Compute Log Likelihood
2. **E Step :** Evaluate the responsibilities using current parameter values

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)}$$

3. **M Step :**
4. **Evaluate the log likelihood :**

EM for gaussian mixture (3/4)

1. **Initialization** :μ_k, Σ_k, π_k, Compute Log Likelihood
2. **E Step :** $\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(x_n|\mu_k,\Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j,\Sigma_j)}$
3. **M Step :** Re-estimate the parameters using the current responsibilities :
   - $\mu_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n$
   - $\sum_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k^{new})(x_n - \mu_k^{new})^T$
   - $\pi_k^{new} = \frac{N_k}{N}$ where $N_k = \sum_{n=1}^{N} \gamma(z_{nk})$
4. **Evaluate the log likelihood :**

EM for gaussian mixture (4/4)

1. **Initialization** :μ_k, Σ_k, π_k, Compute Log Likelihood
2. **E Step :** $\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(x_n|\mu_k,\Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j,\Sigma_j)}$
3. **M Step :** Re-estimate the parameters using the current responsibilities :
   - $\mu_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n$
   - $\sum_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k^{new})(x_n - \mu_k^{new})^T$
   - $\pi_k^{new} = \frac{N_k}{N}$ where $N_k = \sum_{n=1}^{N} \gamma(z_{nk})$
4. **Evaluate the log likelihood :**

$$\ln p(X|\mu, \Sigma, \pi) = \sum_{n=1}^{N} \ln\{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)\}$$

REPEAT 2,3,4 UNTIL CONVERGENCE

Model-based clustering

- The mixture density assumption (gaussian) should be valid
- How to chose the number of clusters ? (value of K)
  penalization of the likelihood

Illustration

Image segmentation based on model-based clustering

Graph-based clustering

Basics on graphs (1)

- Undirected graph $G = (V, E)$ with vertex set $V = \{v_1, \ldots, v_n\}$
- Weighted adjacency matrix $W = (w_{ij})_{ij}$ with positive coefficients
- If $w_{ij} = 0$ then vertices v_i and v_j are not connected
- Undirected graph means W symmetric

Basics on graphs (2)

- Degree of vertex of index i

$$\deg_i = \sum_{j=1}^{n} w_{ij}$$

- Degree matrix : $D = \mathrm{diag}(\deg_1, \ldots, \deg_n)$
- Let A and B two subsets of $\{1, \ldots, n\}$

  The Mincut distance is defined by :

$$W(A, B) = \sum_{i \in A, j \in B} w_{ij}$$

- Measuring the cluster size with $A \subset \{1, \ldots, n\}$:

$$\begin{aligned} |A| &= \text{cardinality of } A \\ \mathrm{vol}(A) &= \textstyle\sum_{i \in A} \deg_i \end{aligned}$$

Graph cut formulation

- Want : edges between groups to have low weights and edges within group to have high weights
- MinCut criterion :

$$\text{MinCut}(A_1, \ldots, A_K) = \frac{1}{2} \sum_{i=1}^{K} W(A_i, \overline{A}_i)$$

  where $\overline{A}_i = V - A_i$

- Drawback : Often leads to a cluster such that $|A_1| = 1$ if $K = 2$.

Alternatives to MinCut

- Other criteria :

$$\text{RatioCut}(A_1, \ldots, A_K) = \frac{1}{2} \sum_{i=1}^{K} \frac{W(A_i, \overline{A}_i)}{|A_i|}$$

$$\text{NCut}(A_1, \ldots, A_K) = \frac{1}{2} \sum_{i=1}^{K} \frac{W(A_i, \overline{A}_i)}{\text{vol}(A_i)}$$

- Idea : Guarantee that clusters are large enough
- Drawback : NP-hard problems

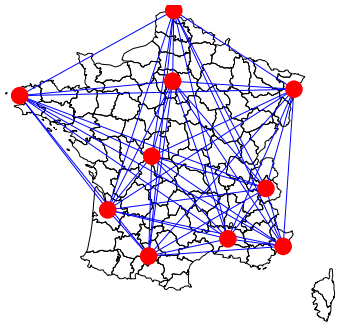
Idea of spectral clustering

- Relaxations of RatioCut and Ncut minimization

$\rightarrow$ Eigenvectors of the Graph Laplacian operator approximate the solution of RatioCut

Spectral clustering

Full connected graph with n nodes.



Weight between two nodes (Z_i, Z_j) :

$w_{i,j} = e^{\frac{-||z_i - z_j||_2^2}{2\mu^2}}$,

μ heat parameter

Normalized Graph Laplacian :

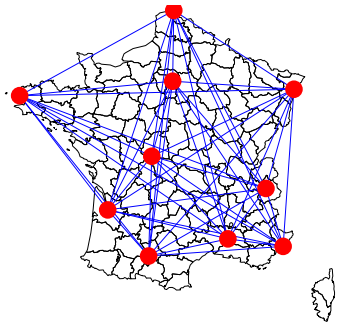
$$\boxed{L = I - D^{-1/2} W D^{-1/2}}$$

$L \in \mathbb{R}^{N \times N}$,

W adjacency matrix, $D_{i,i} = \sum_j w_{i,j}$.

Spectral clustering

Full connected graph with n nodes.



Weight between two nodes (Z_i, Z_j) :

$w_{i,j} = e^{\frac{-||z_i - z_j||_2^2}{2\mu^2}}$,

μ heat parameter

Normalized Graph Laplacian :

$$\boxed{L = I - D^{-1/2} W D^{-1/2}}$$

$L \in \mathbb{R}^{N \times N}$,

W adjacency matrix, $D_{i,i} = \sum_j w_{i,j}$.

Ng et al. Algorithm (2002) :

Input : Fix k nb . clusters

1. Compute the first k eigenvectors $u_1, \ldots u_k$ of L corresponding to the "k" smallest eigenvalues,
2. let $U \in \mathbb{R}^{n \times k}$ be the matrix of column vectors $u_1, \ldots, u_k$.
3. Form the matrix $T \in \mathbb{R}^{n \times k}$ $t_{i,j} = u_{i,j} / (\sqrt{\sum_k u_{ik}^2})$. Let $y_i \in R^k$ i^{th} row of T.
4. Cluster $\{y_i\}$, $1 \leq i \leq n$ with the k-means into clusters $C_1, \ldots, C_k$

Output : Clusters $A_1, \ldots, A_k$ with $A_i = \{y_i \in C_i\}$

Graph Laplacian (1)

- Definition - Unnormalized graph Laplacian matrix

$$L = D - W$$

- Property 1 - For any vector $f \in \mathbb{R}^n$

$$f^T L f = \frac{1}{2} \sum_{i,j} w_{ij} (f_i - f_j)^2$$

- Property 2 - L symmetric, positive
- Property 3 - Smallest eigenvalue of L is 0
- Property 4 - Relation between number of connected components and the spectrum of L

Clustering method #4b - Unnormalized Spectral Clustering Algorithm

Input : number K of clusters, Similarity matrix S

Preprocessing :
Build a similarity graph with adjacency matrix W
Compute the unnormalized Laplacian L

Solve eigenvalue problem : compute the first K eigenvectors of L

Clustering in feature space : Let $U \in \mathbb{R}^{n \times K}$ be the matrix containing the vectors $u_1, \ldots, u_K$ as columns

- For $i = 1, \ldots, n$, let $y_i \in \mathbb{R}^K$ be the vector corresponding to the i-th row of U
- Cluster the points $(y_i)_{i=1,\ldots,n}$ in $\mathbb{R}^K$ with the K-means algorithm into clusters encoded by a partition $A_1, \ldots, A_K$

Building a similarity graph

In practice :

Input : Similarities $s(x_i, x_j)$ or distances $D(x_i, x_j)$, $\forall i, j$

- Following options are often applied on the similarity matrix, and appeared to be very useful (introduction of thresholding or non-linearities)

$\rightarrow$ Option 1 - ϵ-neighborhood graph

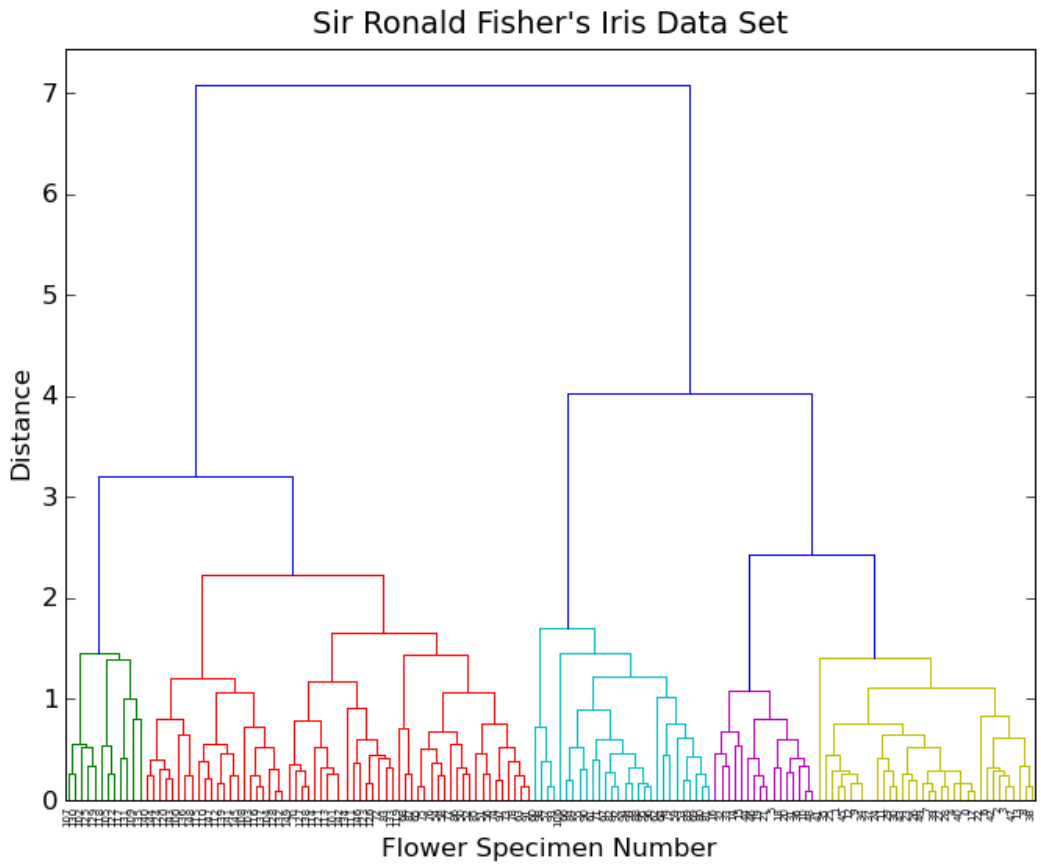
$\rightarrow$ Option 2- k-nearest neighbor graph

$\rightarrow$ Option 3 - Fully connected graph

$$s(x_i, x_j) = \exp(-\|x_i - x_j\|^2/(2\sigma^2))$$

Hierarchical clustering

Dendrogram


Sir Ronald Fisher's Iris Data Set
Distance
0
1
2
3
4
5
6
7
Flower Specimen Number

Clustering method #2a - Bottom-up heuristic

Input : number K of clusters, distance matrix $D(x_i, x_j)$, cluster distance Δ

Initial step : find the pair (x_i, x_j) the minimal element in the distance matrix and form cluster $A_1 = \{i, j\}$, the remaining x_k's form clusters with one element $A_2, \ldots, A_{n-1}$

Agglomeration step : Consider $A_1, \ldots, A_{n-1}$ clusters and find the pair (k^*, ℓ^*) such that

$$(k^*, l^*) = \arg\min_{k \neq l} \Delta(A_k, A_l)$$

and merge these clusters into $A = A_{k^*} \cup A_{l^*}$.

Stopping criterion : Iterate until the target number of clusters is reached.

Linkage distance

- $A, B \subset \{1, \ldots, n\}$
- Single linkage

$$\Delta(A, B) = \min_{i \in A, j \in B} \{D(x_i, x_j)\}$$

- Complete linkage

$$\Delta(A, B) = \max_{i \in A, j \in B} \{D(x_i, x_j)\}$$

- Centroid linkage

$$\Delta(A, B) = D(\overline{x}_A, \overline{x}_B)$$

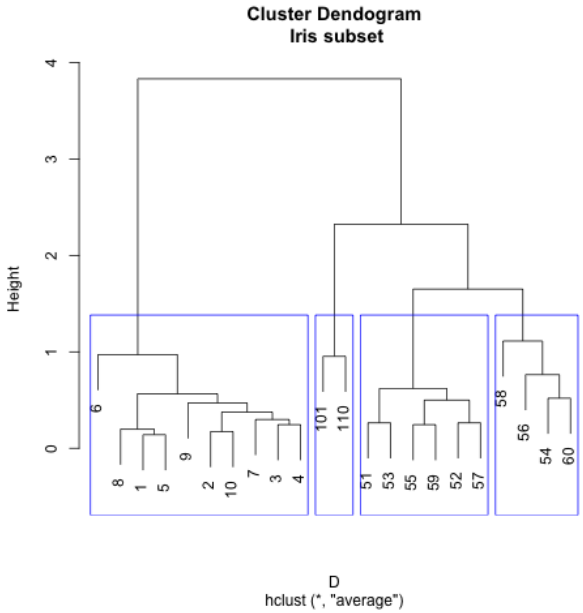
- Average linkage

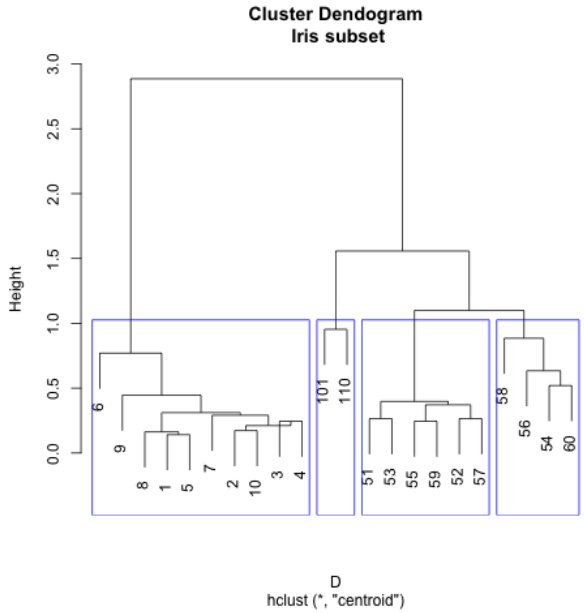
$$\Delta(A, B) = \frac{1}{|A| \cdot |B|} \sum_{i \in A} \sum_{j \in B} D(x_i, x_j)$$

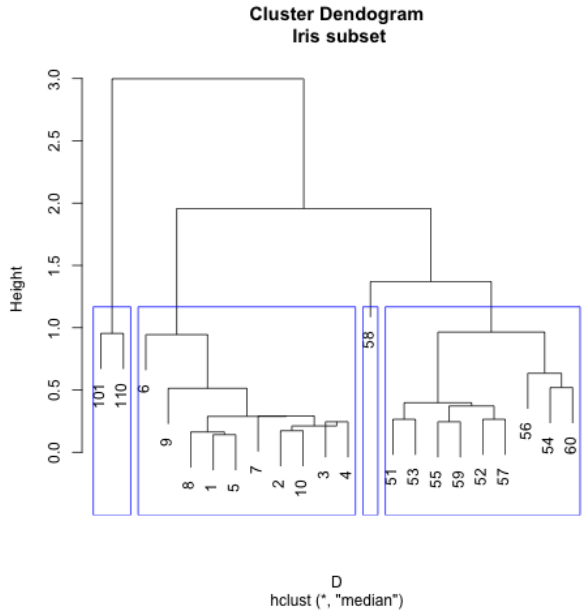
- Ward linkage

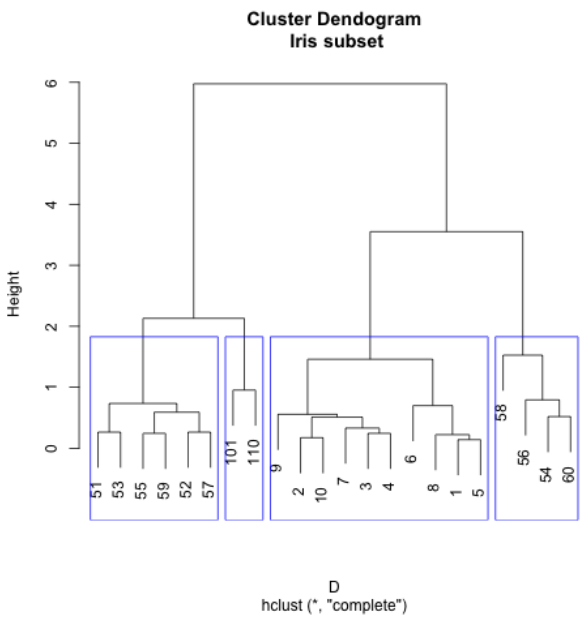
$$\Delta(A, B) = \frac{|A| + |B|}{|A| \cdot |B|} D(\overline{x}_A, \overline{x}_B)$$

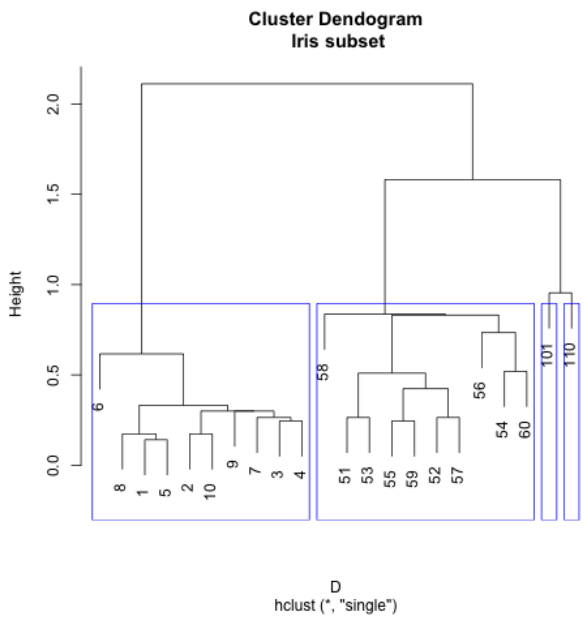
Hierachical Clustering. Impact of Linkage. Illustration

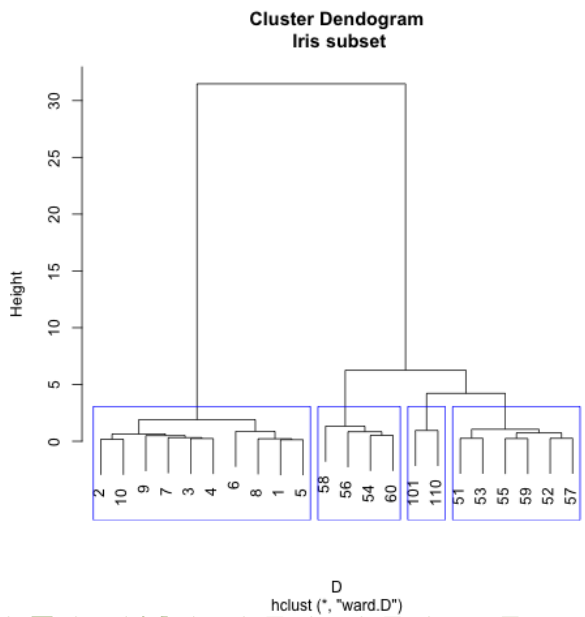




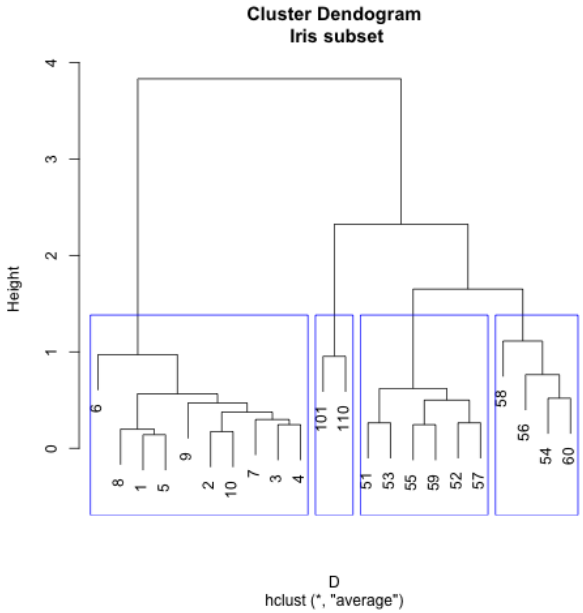


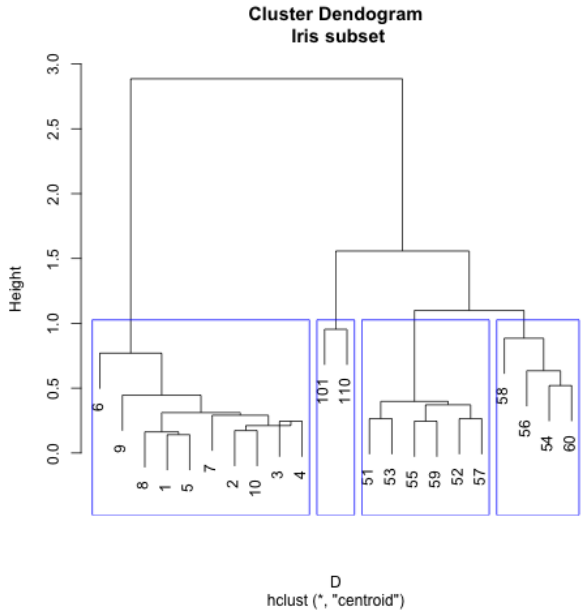


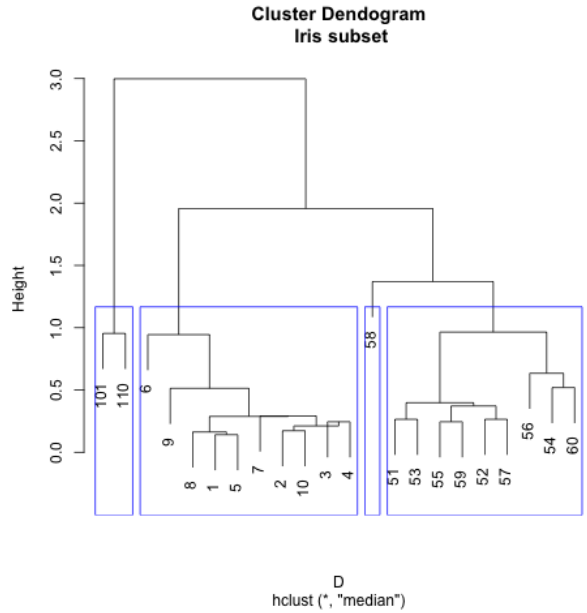


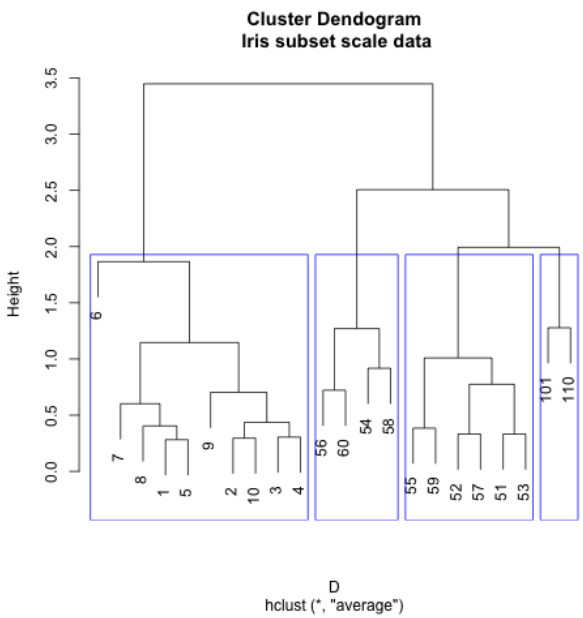


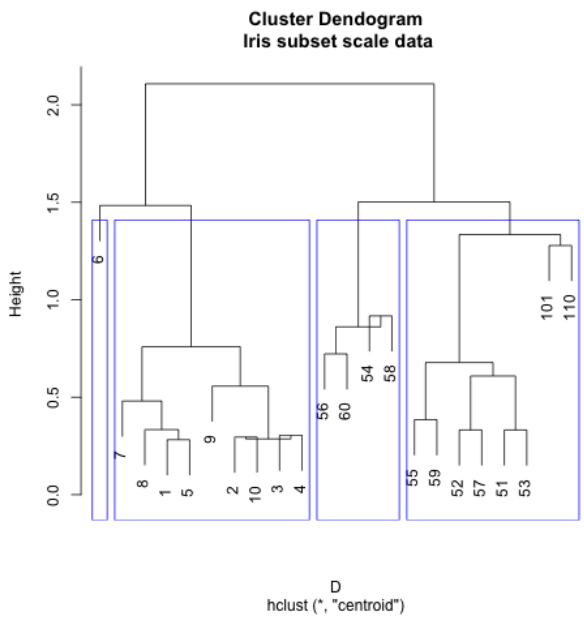
Hierachical Clustering. Impact of Scaling. Illustration

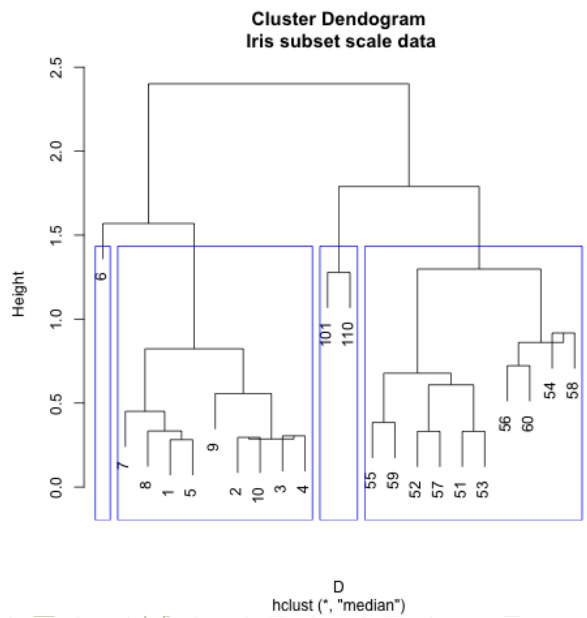




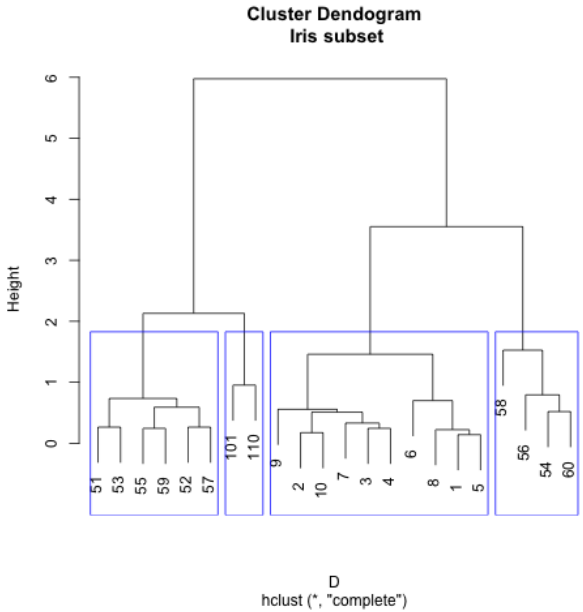


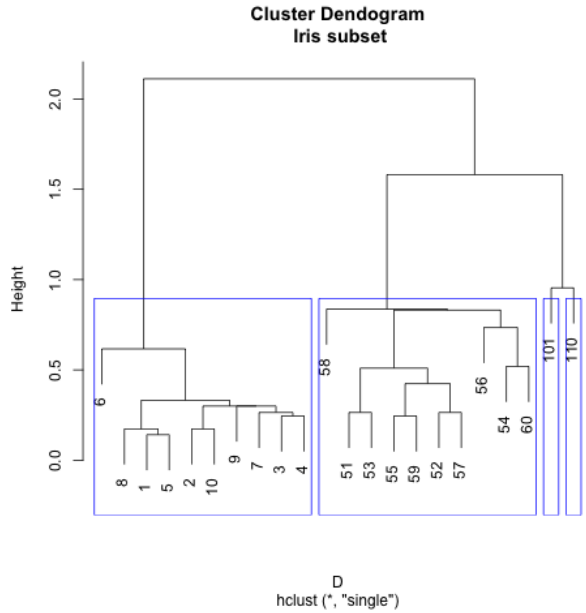


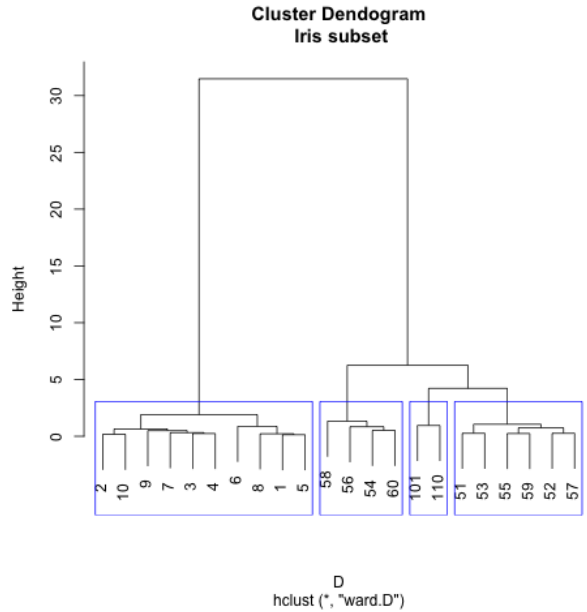


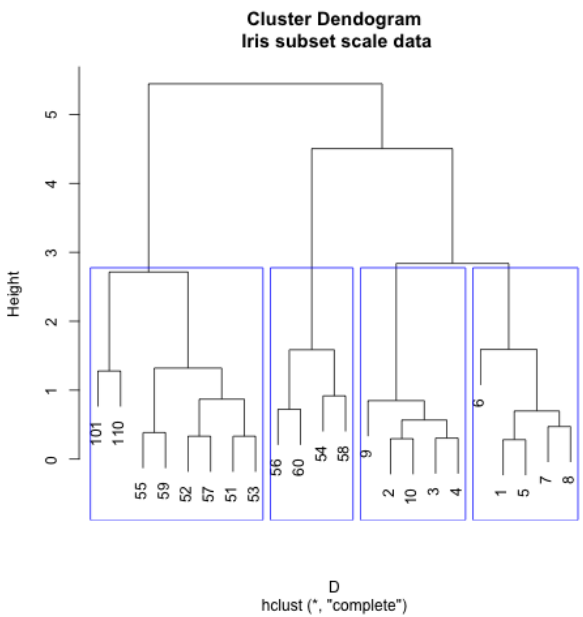


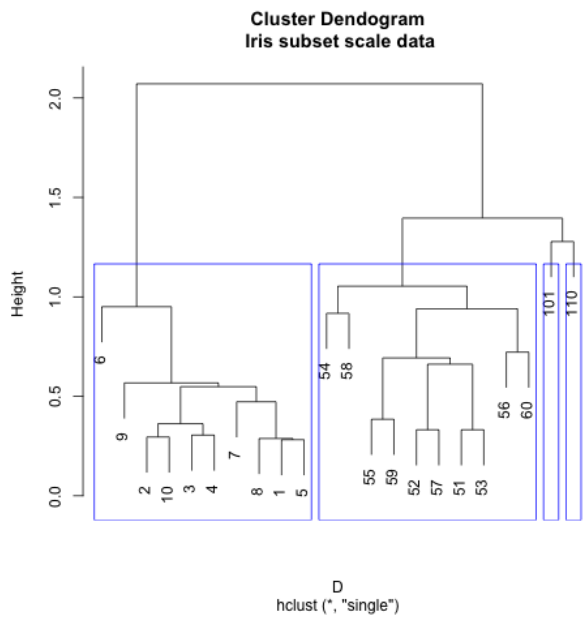
Hierachical Clustering. Impact of Scaling. Illustration

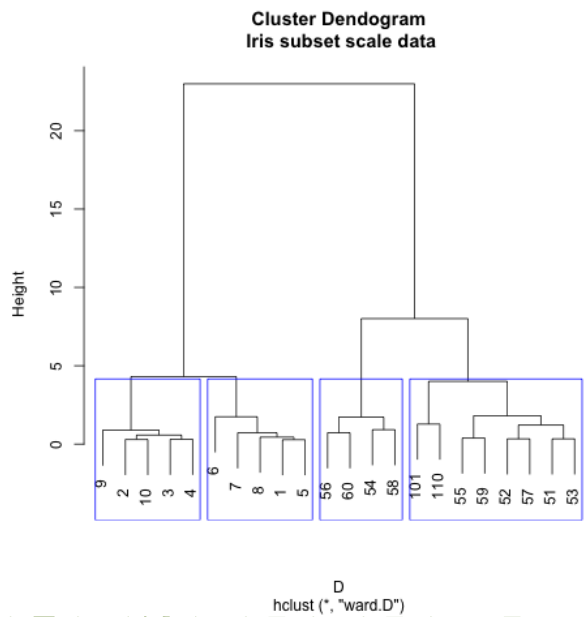






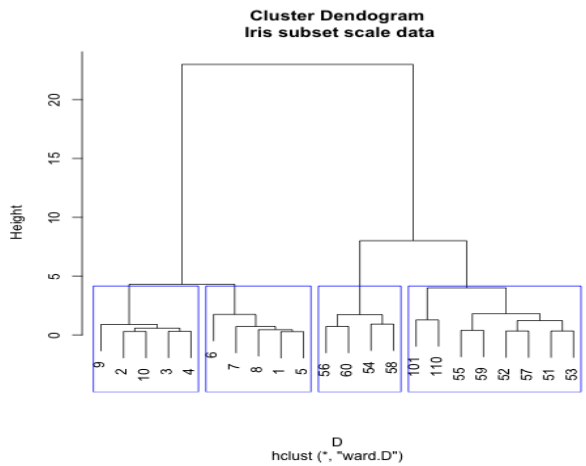






Hierachical Clustering. R instructions.

```
# Tab : dataframe
help(Hclust) ; D=dist(tab) ;
HC=hclust(D,method="ward.D") ;
plot(HC) ;
rect.hclust(HC, k = 4) ;
members <- cutree(HC, k = 4) ;
```



Clustering method #2b -Top-down heuristic

Parameters : Threshold t

Initial step : find the pair (x_i, x_j) having the maximal element in the distance matrix denoted d_M

Division step : if $d_M > t$ consider each of them as a center and affect remaining points to the closest center

Iteration : Iterate until $d_M < t$ within each cluster.

What about theory ?

- Pessimistic result (Kleinberg, 2003)

  There is no clustering function satisfying scale invariance, richness and consistency.

- Optimistic result (von Luxburg, Ben-David, 2005)

  Some stability for clustering can be guaranteed.

Centroid-based clustering

The celebrated K-means (1)

- Use for : Quantitative features
- Squared Euclidean distance
- Barycenter cluster k with n_k elements

$$\bar{x}_k = n_k^{-1} \sum_{C(i)=k} x_i$$

- Note that : for any subset $I \subset \{1, \ldots, n\}$ of individuals

$$\bar{x}_I = \arg\min_m \sum_{i \in I} \|x_i - m\|^2$$

The celebrated K-means (2)

- Optimization criterion

$$W(C) = \sum_{k=1}^{K} \sum_{C(i)=k} \|x_i - \bar{x}_k\|^2$$

- Solution

$$C^* = \arg\min_{C} W(C) = \arg\min_{C, m_1, \ldots, m_K} \sum_{k=1}^{K} \sum_{C(i)=k} \|x_i - m_k\|^2$$

Clustering method #3a - K-means algorithm

Parameter : encoder range K

Initialization : initial encoder $C^{(0)}$, and centers $m_k^{(0)}$

Step 1 : Fix encoder C, compute the centers

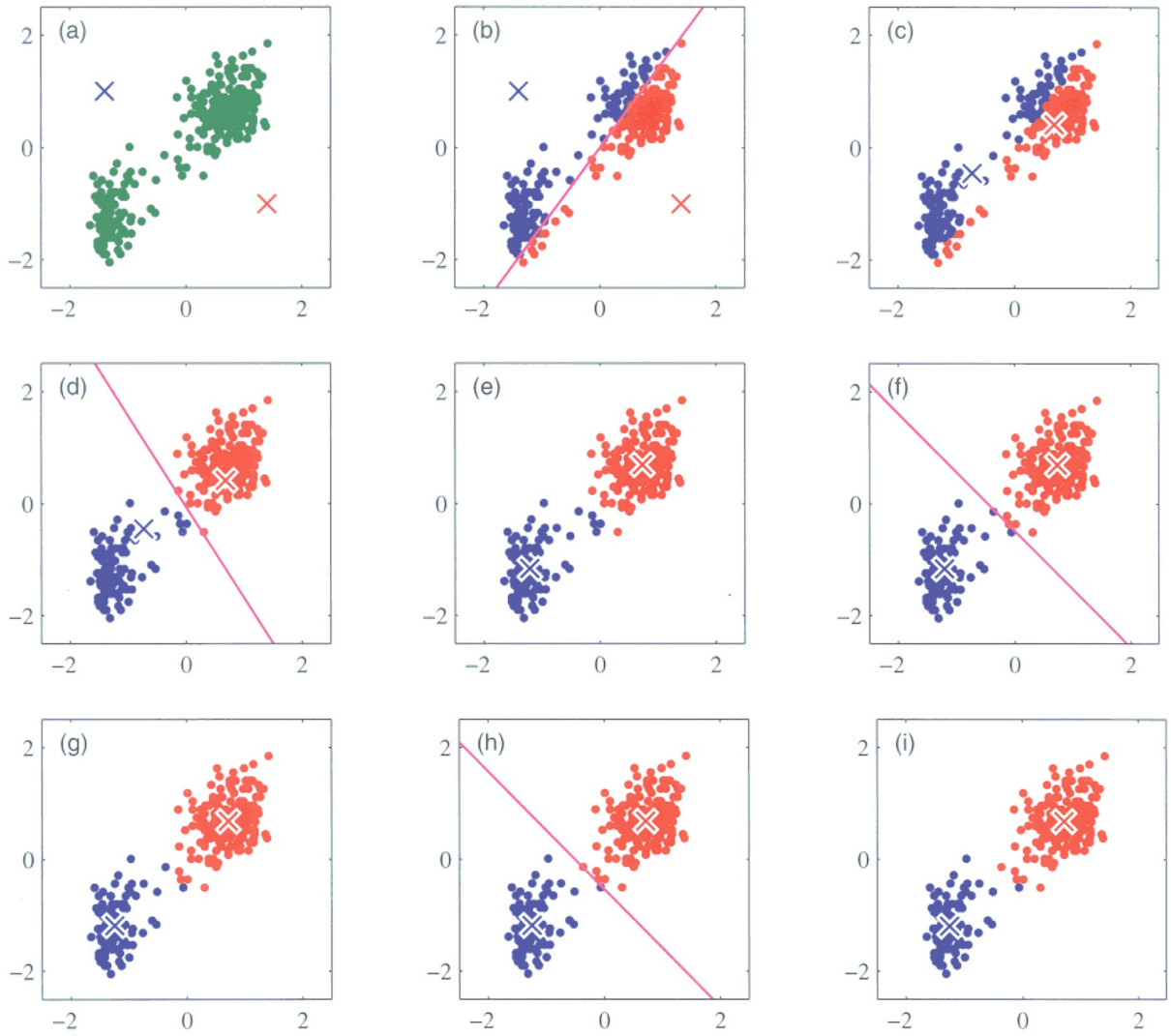
$$m_k = \frac{1}{n_k} \sum_{C(i)=k} x_i$$

Step 2 : Fix the centers $m_1, \ldots, m_K$, assign with new encoder

$$C(i) = \underset{1 \leq k \leq K}{\arg\min} \|x_i - m_k\|^2$$

Iteration : repeat Steps 1 & 2 until C does not change anymore.

K-means algorithm - How it works



A variation : K-medoids

- Use for : any type of features
- Arbitrary distance
- Centers $\{m_1, \ldots, m_K\}$ belong to the data set $\{x_1, \ldots, x_n\}$

Clustering method #3b - K-medoids algorithm

Parameter : encoder range K

Initialization : initial encoder $C^{(0)}$, and centers $m_k^{(0)}$

Step 1 : Fix encoder C, compute the centers $m_k = x_{i_k^*}$ with

$$i_k^* = \underset{C(i)=k}{\arg\min} \sum_{C(j)=k} D(x_i, x_j)$$

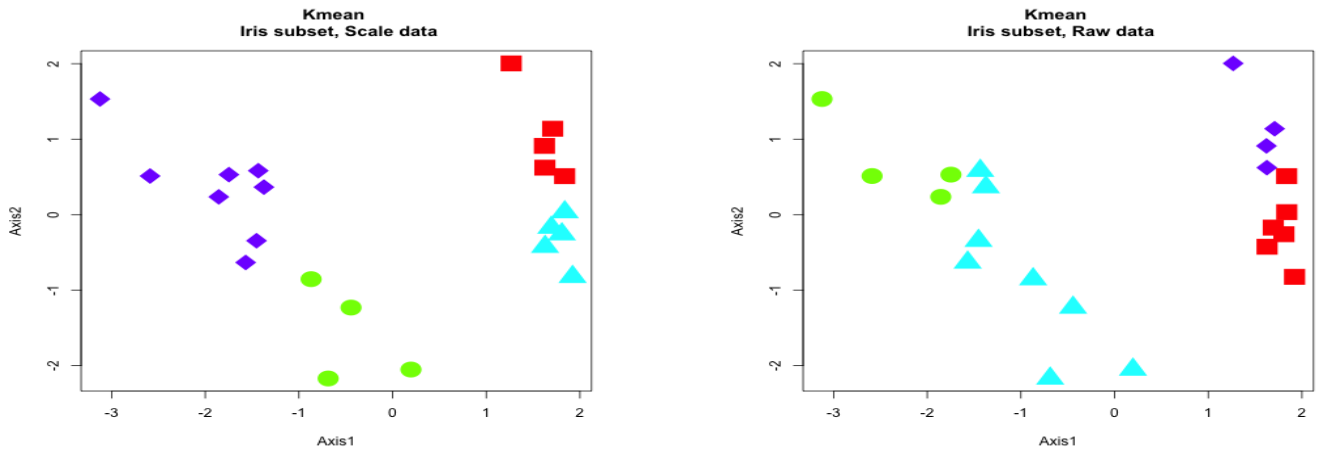
Step 2 : Fix the centers $m_1, \ldots, m_K$, assign with new encoder

$$C(i) = \underset{1 \leq k \leq K}{\arg\min}\, D(x_i, m_k)$$

Iteration : repeat Steps 1 & 2 until C does not change anymore.

Kmeans Clustering. Illustration

Use of PCA to project the observations, and to represent the clusters.



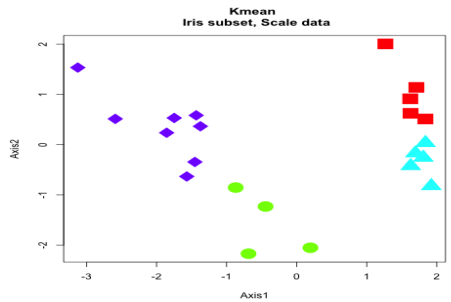
Subset (#30) of Iris Data Set

Impact of Scaling data

(left :scaled vs right :raw)

Kmeans Clustering. R instructions.

```
# tab : dataframe
pca=dudi.pca(tab,scannf=FALSE,nf=2) ;
K=4 ;
mycol=rainbow(K) ; mypch=c(1,3,4,8)
#Scale data
res=kmeans(tab,centers=4) ;
plot(pca$li,col=mycol[res$cluster],pch=mypch[res$cluster])
```



Calibrating the number of clusters

Calibrating the number of clusters (1)

- Calinski & Harabasz (1974)

$$\text{maximize} \quad F_{CH}(K) = \frac{B(C_K)/(K-1)}{W(C_K)/(n-K)} \ , \quad \forall K > 1 \ .$$

- Hartigan (1975)

$$\text{take smallest } K \geq 1 \text{ such that} \quad F_H(K) \leq 10 \ ,$$

where

$$F_H(K) = \left(\frac{W(C_K)}{W(C_{K+1})} - 1 \right) / (n - K - 1) \ .$$

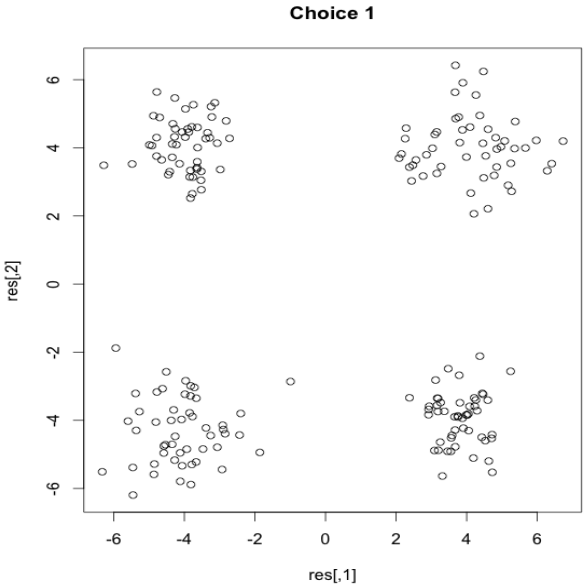
Calibrating the number of clusters (2)

- Krzanowski & Lai (1985)

$$\text{maximize} \quad F_{KL}(K) = \left| \frac{\Delta(K)}{\Delta(K+1)} \right|$$

where $\Delta(K) = (K-1)^{2/d} W(C_{K-1}) - K^{2/d} W(C_K)$ and d is the dimension of input data.

Illustration. Calibrating the number of clusters



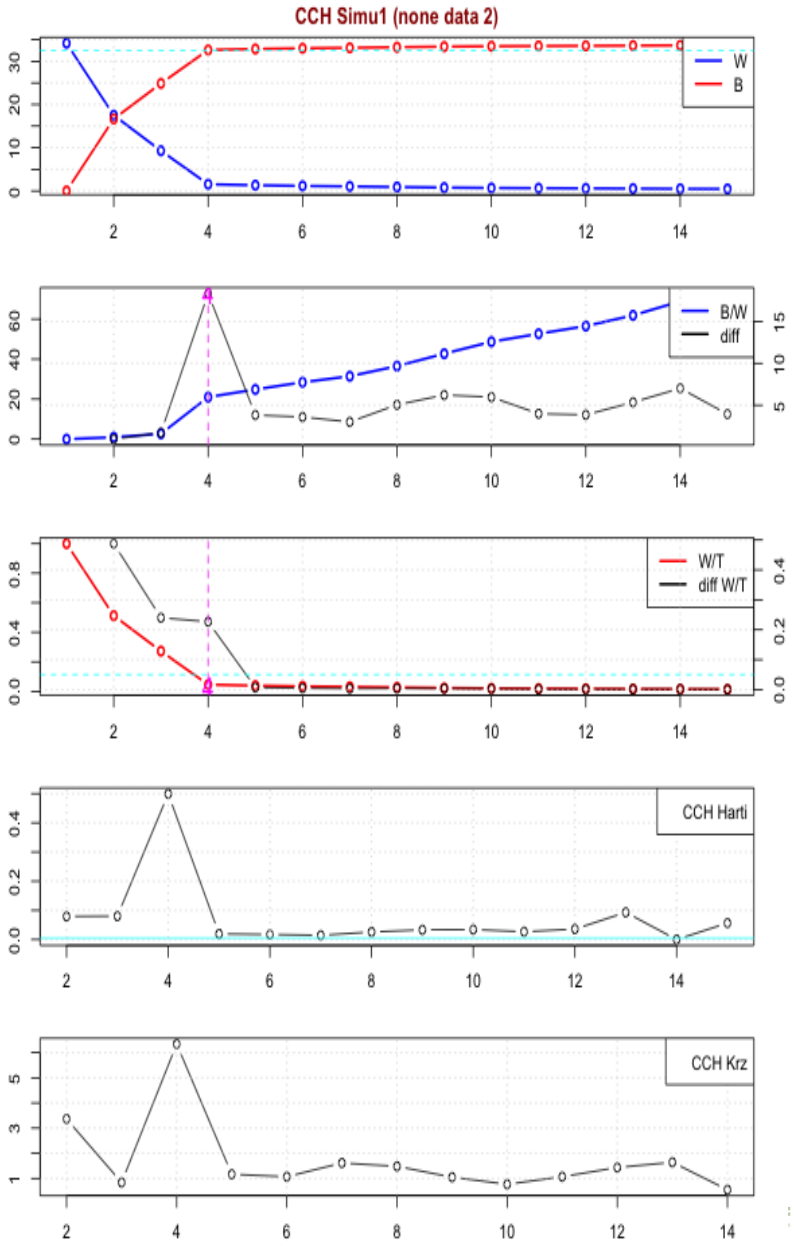
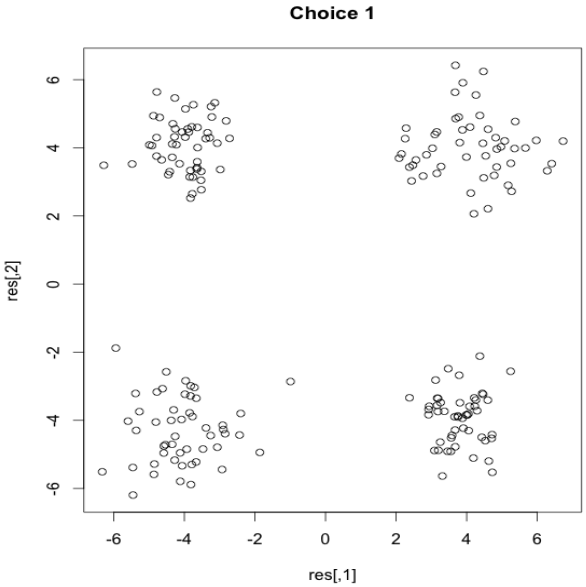
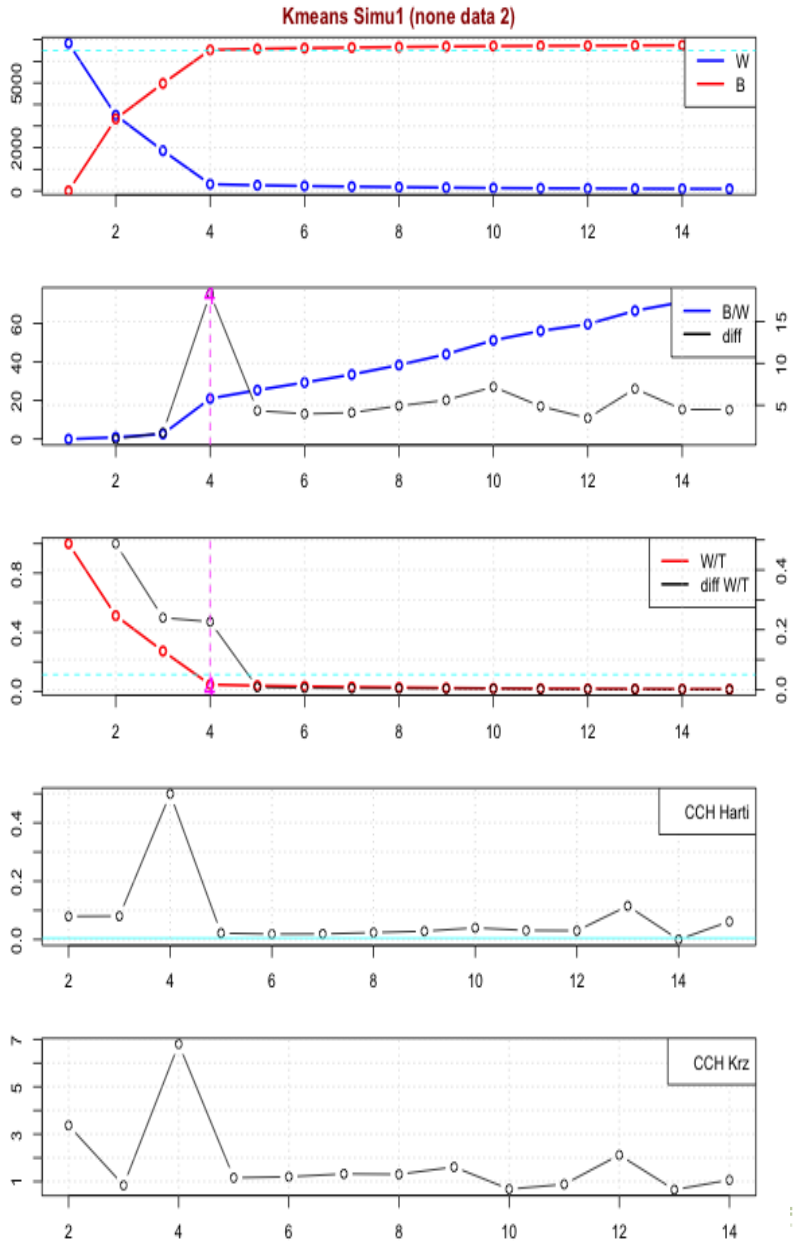


Illustration. Calibrating the number of clusters





Calibrating the number of clusters (3)

- Rousseeuw (1987) - Silhouette statistic

$$F_S(K) = \sum_{i=1}^{n} \left(\frac{b(i) - a(i)}{\max(a(i), b(i))} \right) ,$$

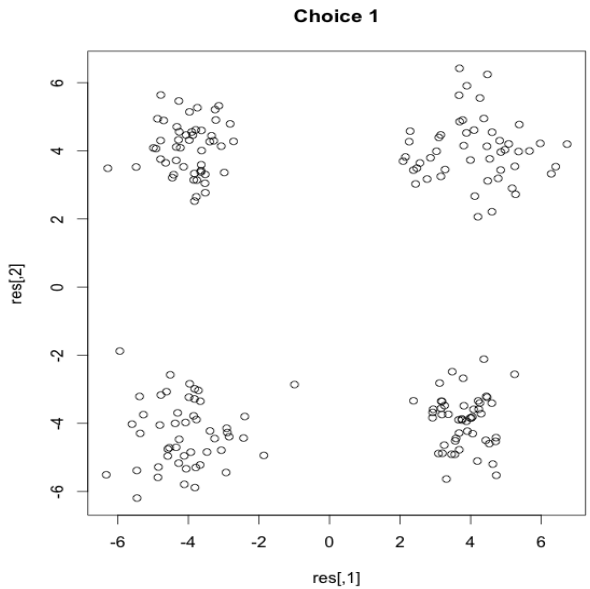
where for $i \in C_k$

$$a(i) = \frac{1}{n_k} \sum_{C(j)=k} \|x_j - x_i\|_2^2 ,$$

and, if $\ell = \ell(i)$ is the next nearest cluster of the point x_i :

$$b(i) = \frac{1}{n_\ell} \sum_{C(j)=\ell} \|x_j - x_i\|_2^2 .$$

Illustration. Calibrating the number of clusters



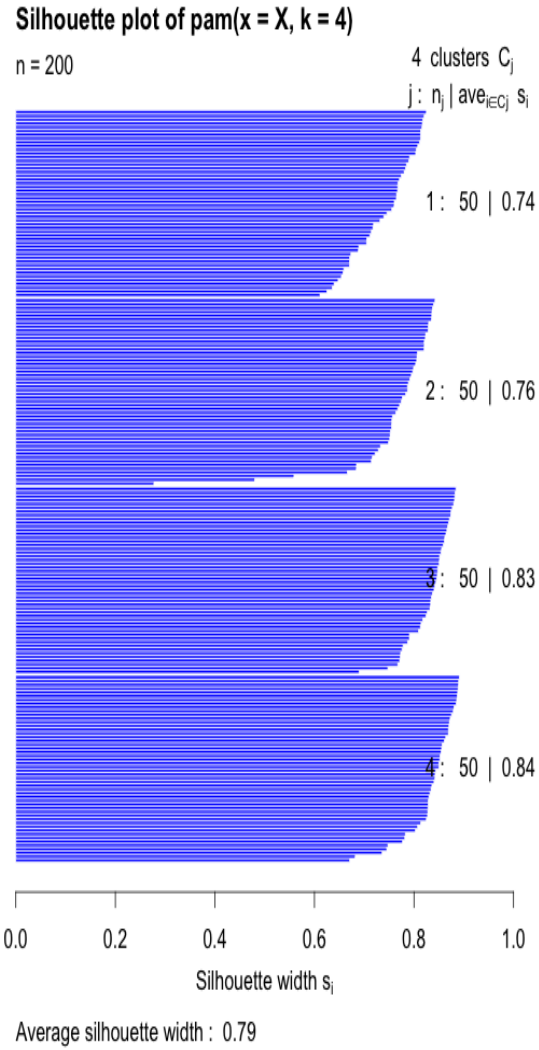
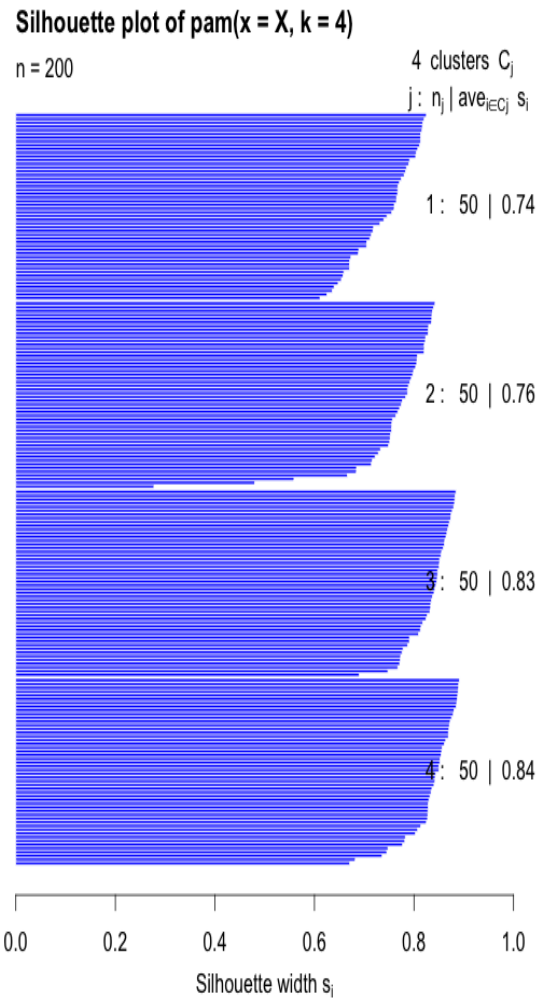
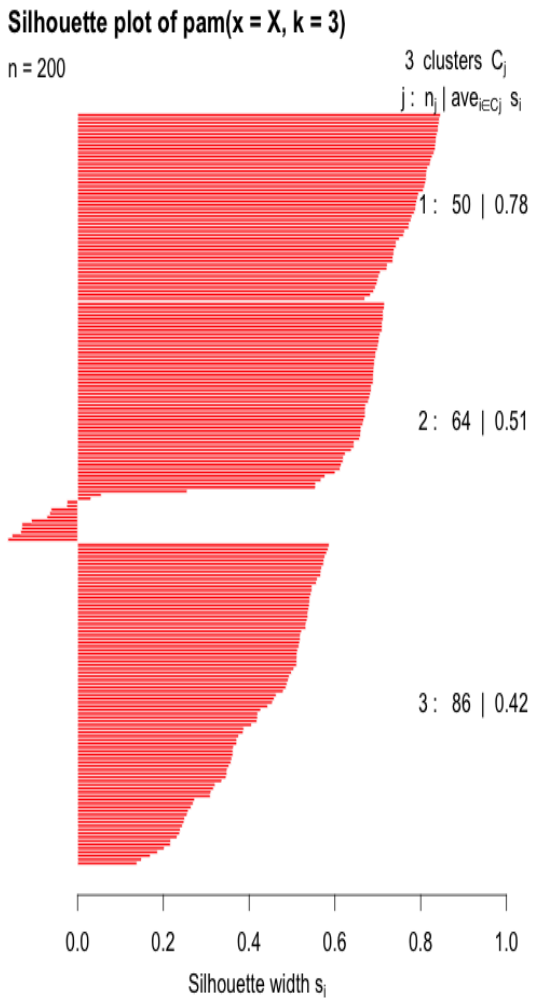
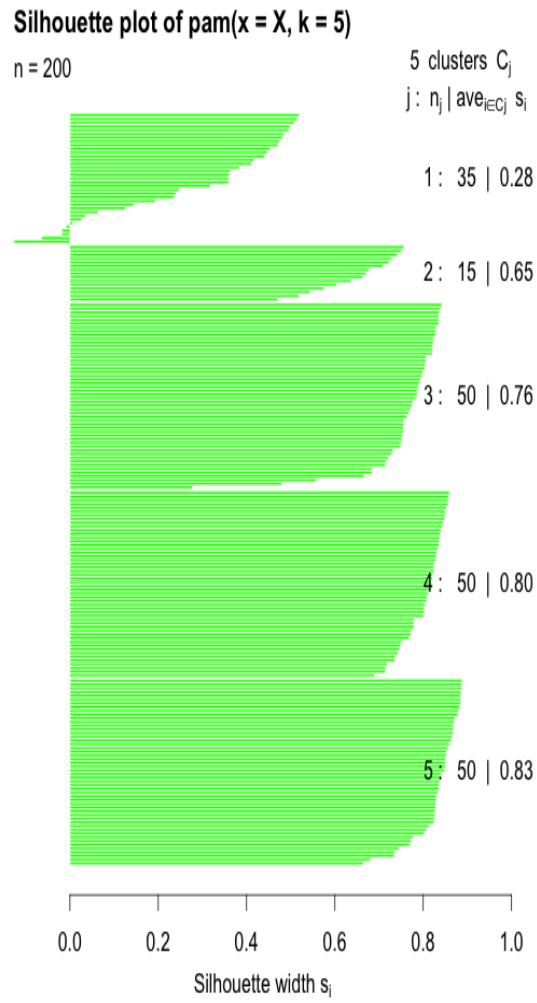


Illustration. Calibrating the number of clusters

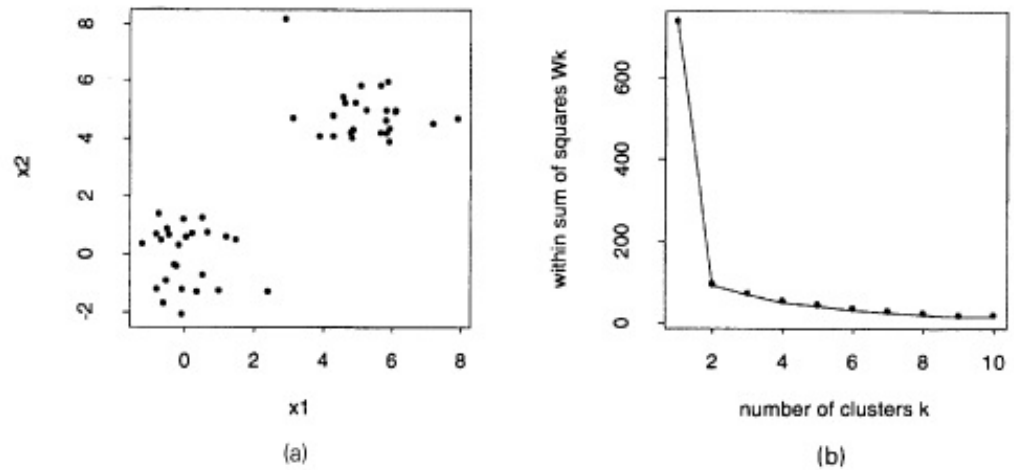






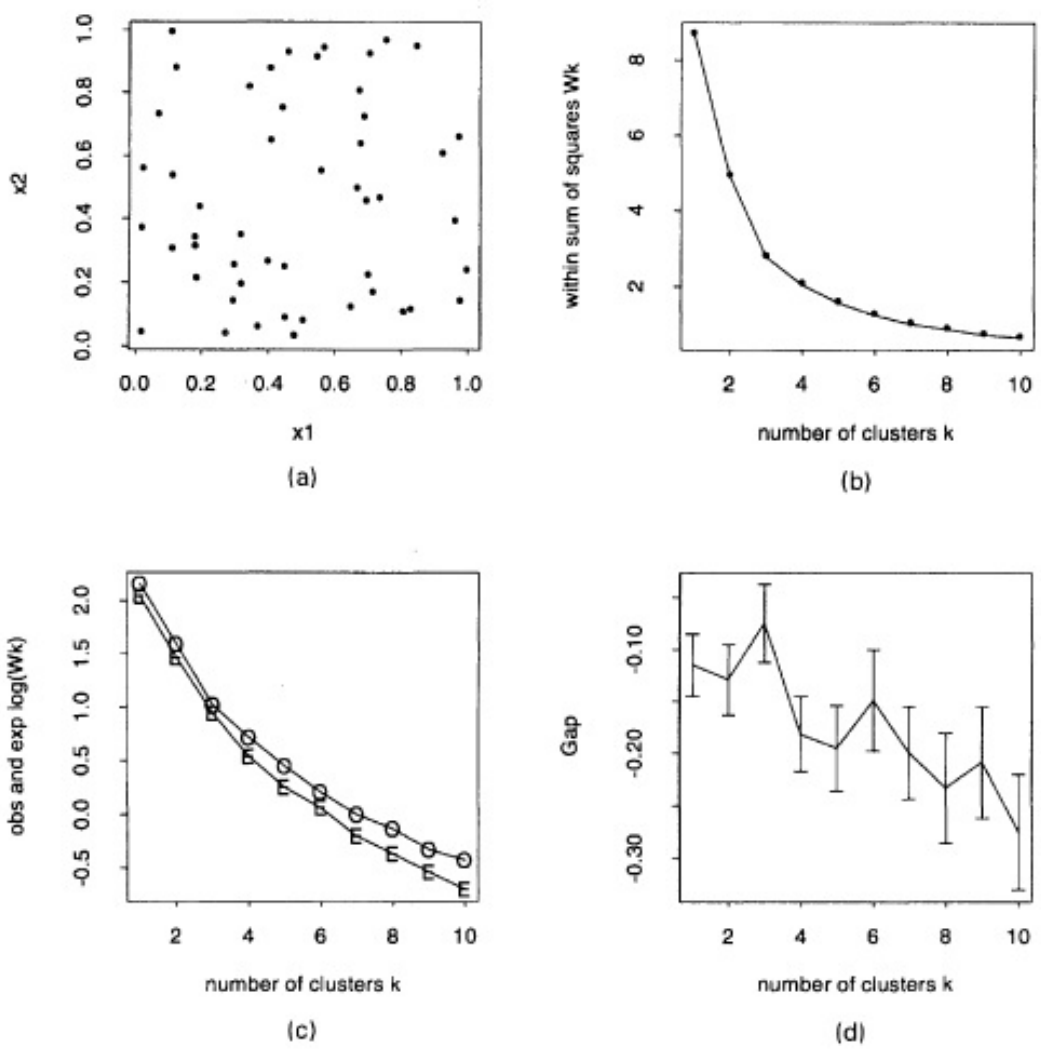
Calibrating the number of clusters (4)

- Tibshirani, Walther, Hastie (2001) - Gap statistic



Calibrating the number of clusters (4)

- Tibshirani, Walther, Hastie (2001) - Gap statistic



Theoretical analysis

- An information-theoretic formulation
- References on Vector Quantization
- Work of Pollard (1981, 1982), but also Linder (2001)
- Proofs of strong consistency of K-means clustering

K-means clustering vs EM Algorithm

- **Distortion measure :**

$$J = \sum_{i=1}^{n} \sum_{k=1}^{k} r_{ik} ||x_i - \mu_k||^2$$

$r_{ik} = 1$ if $k = \arg\min_j ||x_i - \mu_j||^2$ (=0 otherwise)

- **K-means algorithm :**
  1. E-Step : r_{nk} computation
  2. M-step : μ_k computation

  REPEAT 1, 2 UNTIL convergence