

# Reconnaissance de Formes et Biométrie

Sonia Garcia-Salicetti

[sonia.garcia@telecom-sudparis.eu](mailto:sonia.garcia@telecom-sudparis.eu)

# Déroulement du module

1. Introduction, Apprentissage non-supervisé / Clustering (S. Garcia-Salicetti, TSP)
  - K-Means, Classification Hiérarchique Ascendante (CAH), Mélanges de Gaussiennes
2. Classification bayésienne, Réseaux de neurones / TP classification automatique (O. Galarraga, Centre Coubert)
3. Régression linéaire et non linéaire / TP régression (O. Galarraga, Centre Coubert)
4. Introduction à la Reconnaissance de sons (D. Istrate, UTC)
5. Réseaux de convolution et Transfer Learning (O. Galarraga)

# Déroulement du module

6. Introduction à la modélisation de séquences:
  - Application à la Biométrie Signature
  - Application en santé numérique: analyse automatique du mouvement global (S. Garcia-Salicetti, TSP)
7. Machines à Vecteurs de Support et applications santé numérique (J. Boudy, TSP)
8. Vérification du locuteur (D. Istrate)
9. Traitement du langage naturel (O. Galarraga)
10. Apprentissage Profond (Deep Neural Networks) et application à la vidéo (Jean Emmanuel Haugeard, Thalès)

# Déroulement du module

- 5 Intervenants

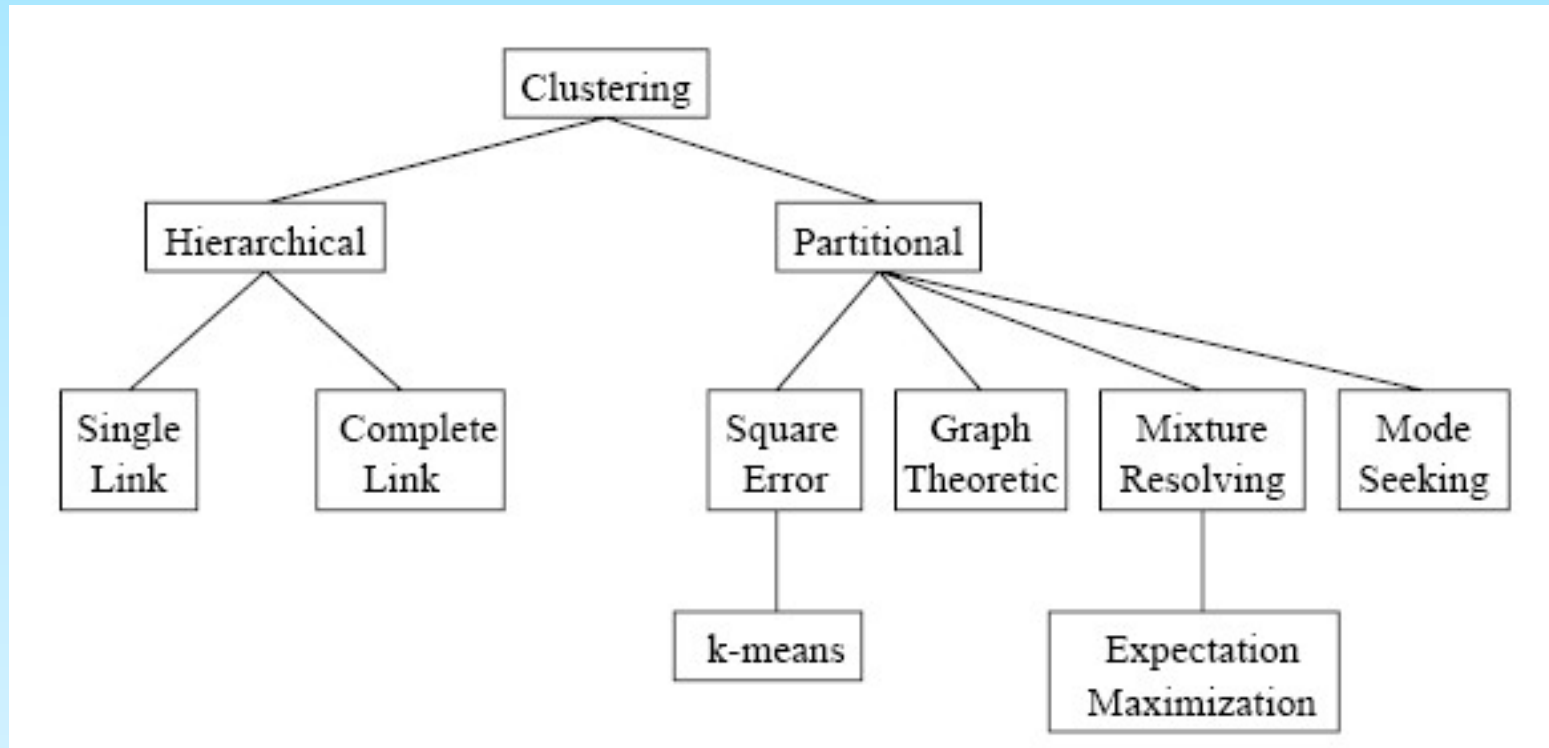
## EVALUATION

### 6 TPs avec compte-rendus:

- Classification supervisée
  - Régression
  - Apprentissage profond
  - Vérification du locuteur
  - Machines à Vecteurs de Support (SVMs)
  - Traitement du langage naturel
- 
- 1 projet: Clustering (présentation en binôme) → 1/2 note
  - Moyenne des compte-rendus TPs → 1/2 note

# Clustering

# Une Classification des algorithmes de Clustering



# K-moyennes

- Idée du K-Moyennes : partitionner l'espace en K groupes ou clusters (chaque cluster est représenté par sa moyenne)

$$J = \sum_{j=1}^K \sum_{i=1}^{n_j} \left( x_i^{(j)} - \mu_j \right)^2$$

$\mu_j$  : Centroid of Cluster  $C_j$

$x_i^{(j)}$  :  $i^{th}$  Pattern belonging to Cluster  $C_j$

$n_j$  : Number of Patterns assigned to Cluster  $C_j$

- Par cluster: on calcule la somme des écarts quadratiques à la moyenne  $m$
- On somme cette quantité sur tous les clusters
  - J= erreur totale effectuée quand on représente les données par les centres des clusters : ERREUR DE QUANTIFICATION
- J est minimisée: erreur de quantification minimale

# K-moyennes

## "k-means"

- Critère à minimiser= erreur quadratique globale  $J$

$$J = \sum_{j=1}^K \sum_{i=1}^{n_j} \left( x_i^{(j)} - \mu_j \right)^2$$

$\mu_j$  : Centroid of Cluster  $C_j$

$x_i^{(j)}$  :  $i^{th}$  Pattern belonging to Cluster  $C_j$

$n_j$  : Number of Patterns assigned to Cluster  $C_j$

- Algorithme des  $k$ -moyennes
  - Le plus simple
  - Le plus utilisé en pratique

1. Initialisation aléatoire de  $K$  Centres (Prototypes) dans l'espace
2. Etape de Clustering : on affecte chaque donnée au centre le plus proche
3. Mise à jour des centres des Clusters
  - Centroid = moyenne de chaque Cluster
4. Répéter 2) et 3) jusqu'à convergence

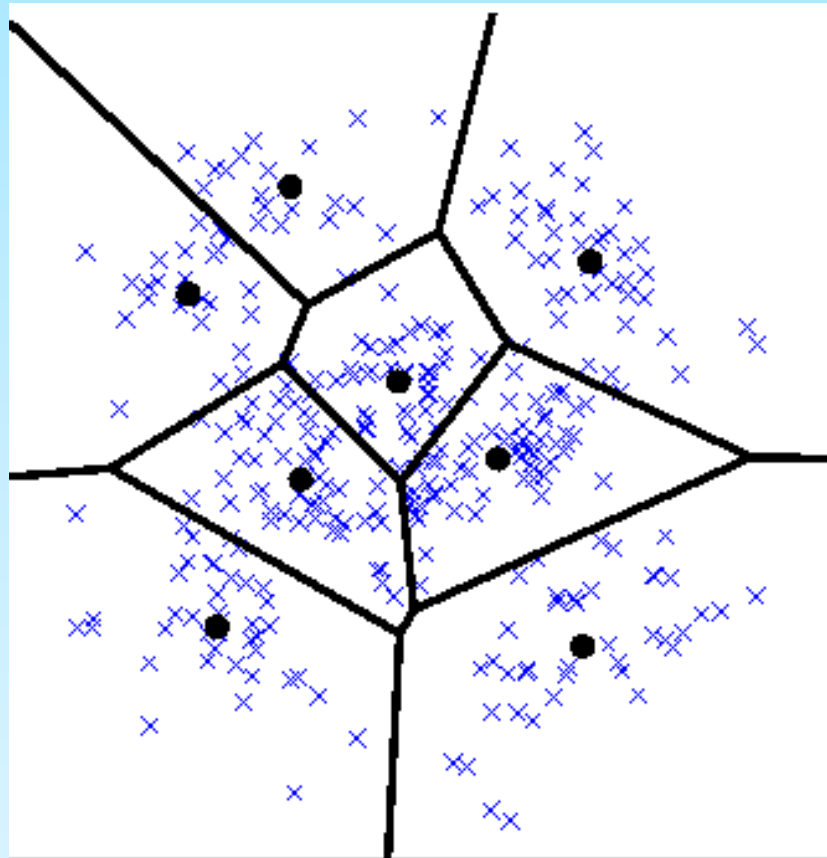
Convergence signifie:

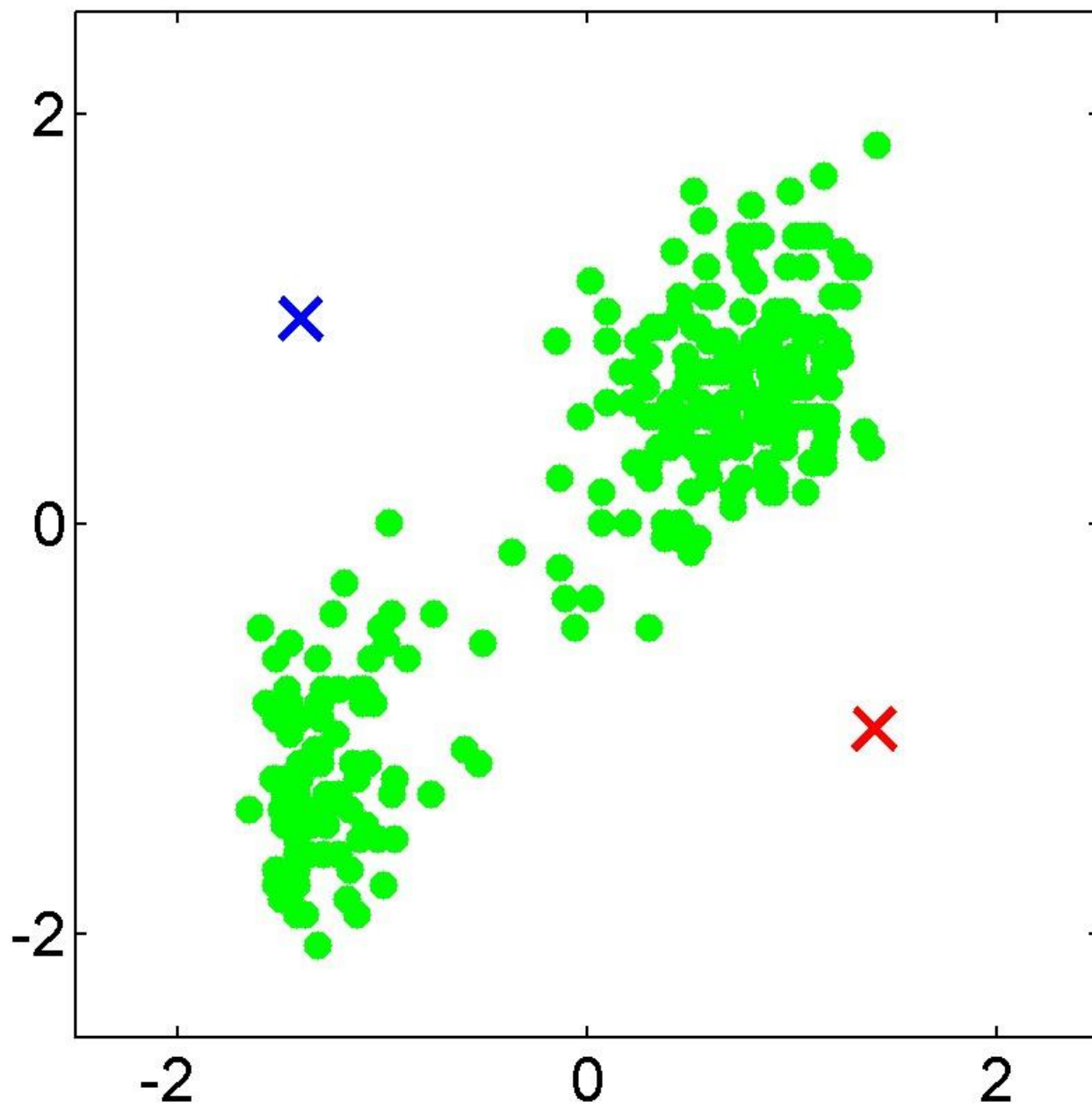
- Plus de nouvelle ré-affectation
- La baisse de l'erreur n'est plus significative
- Garantie d'une minimisation locale
  - **Essayer différentes initialisations**
  - Choisir celle menant à l'erreur minimale

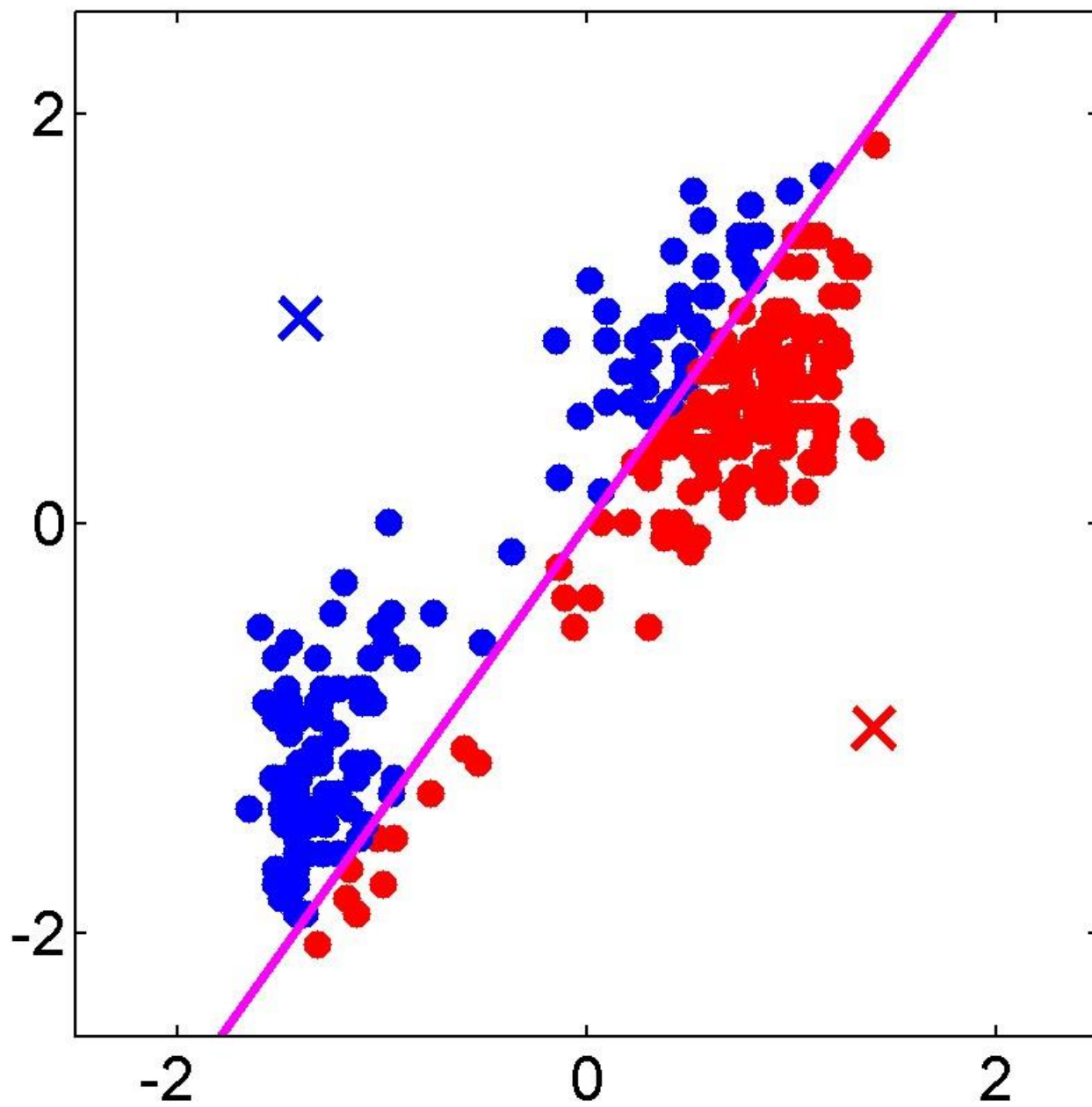


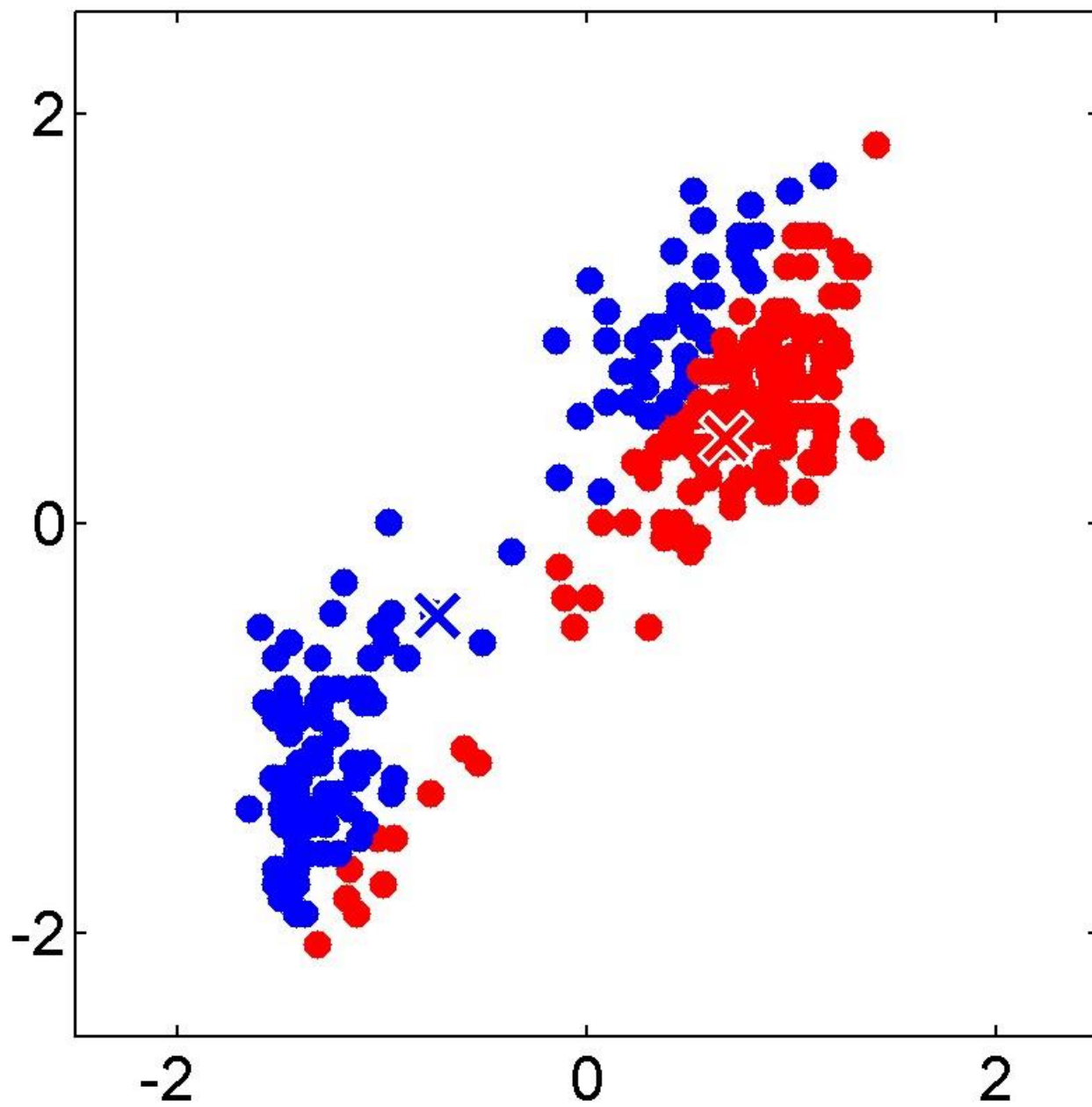
# $K$ -moyennes ( $K=8$ )

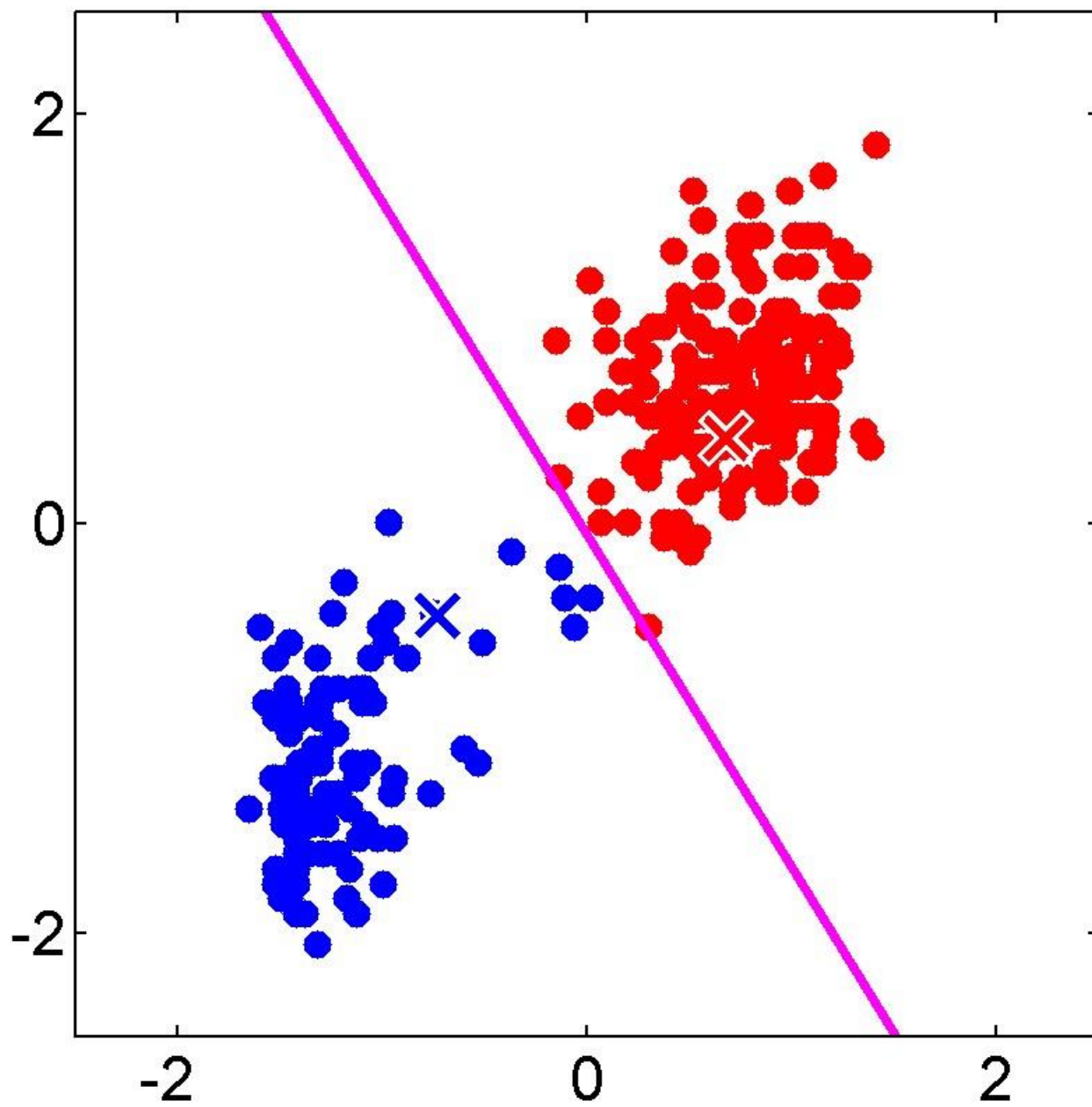
Une seule partition des données est obtenue

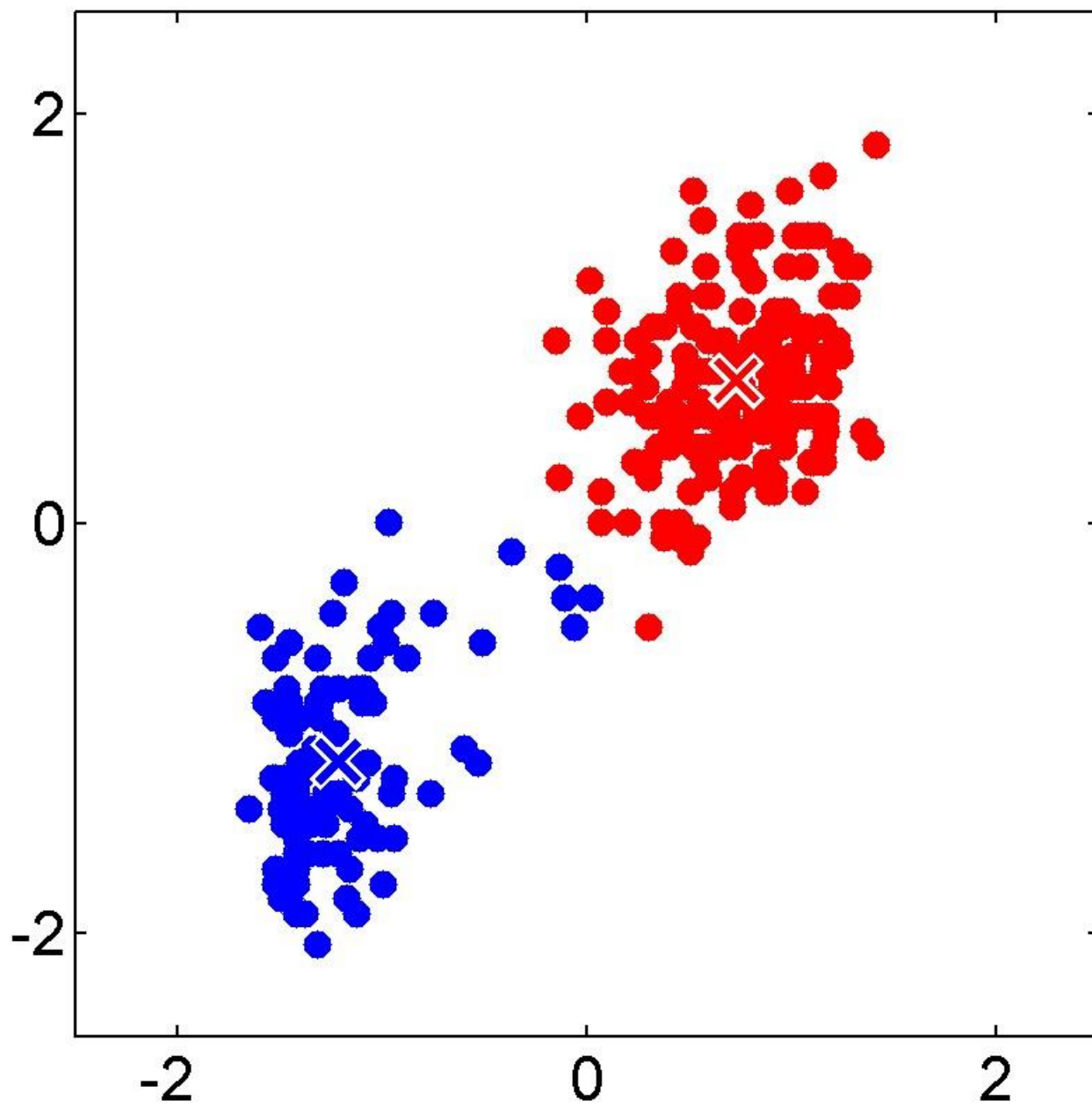


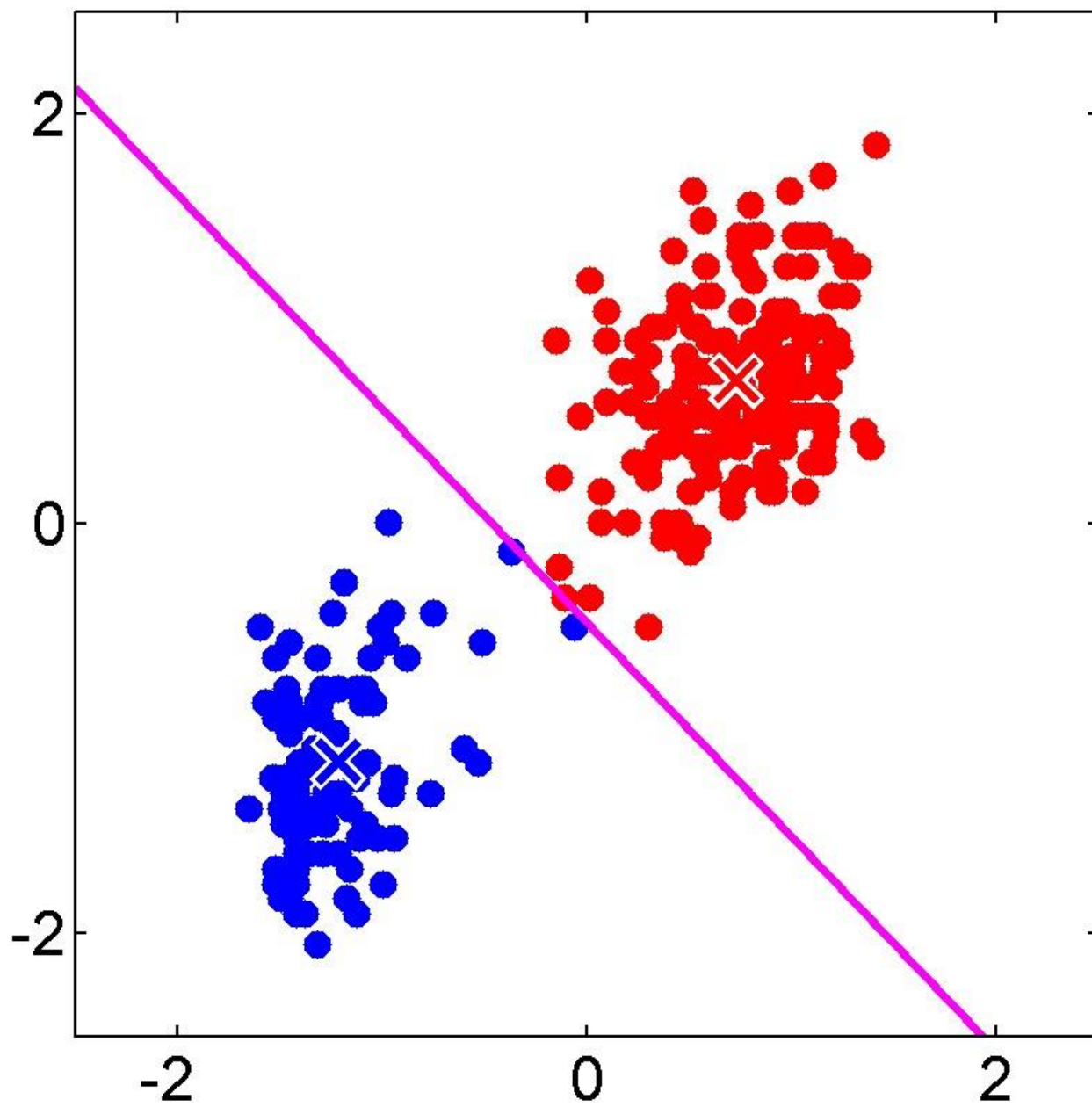


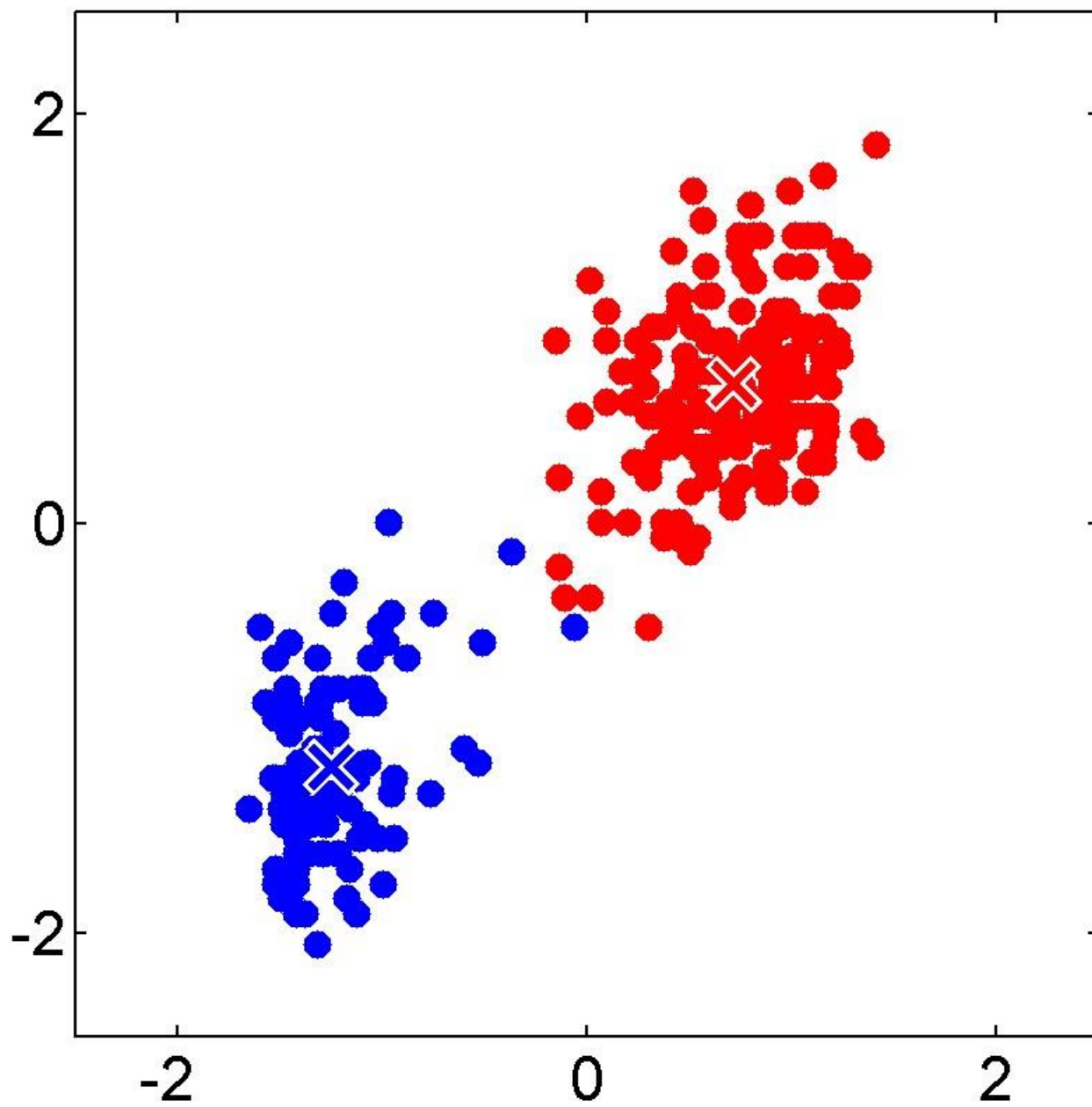




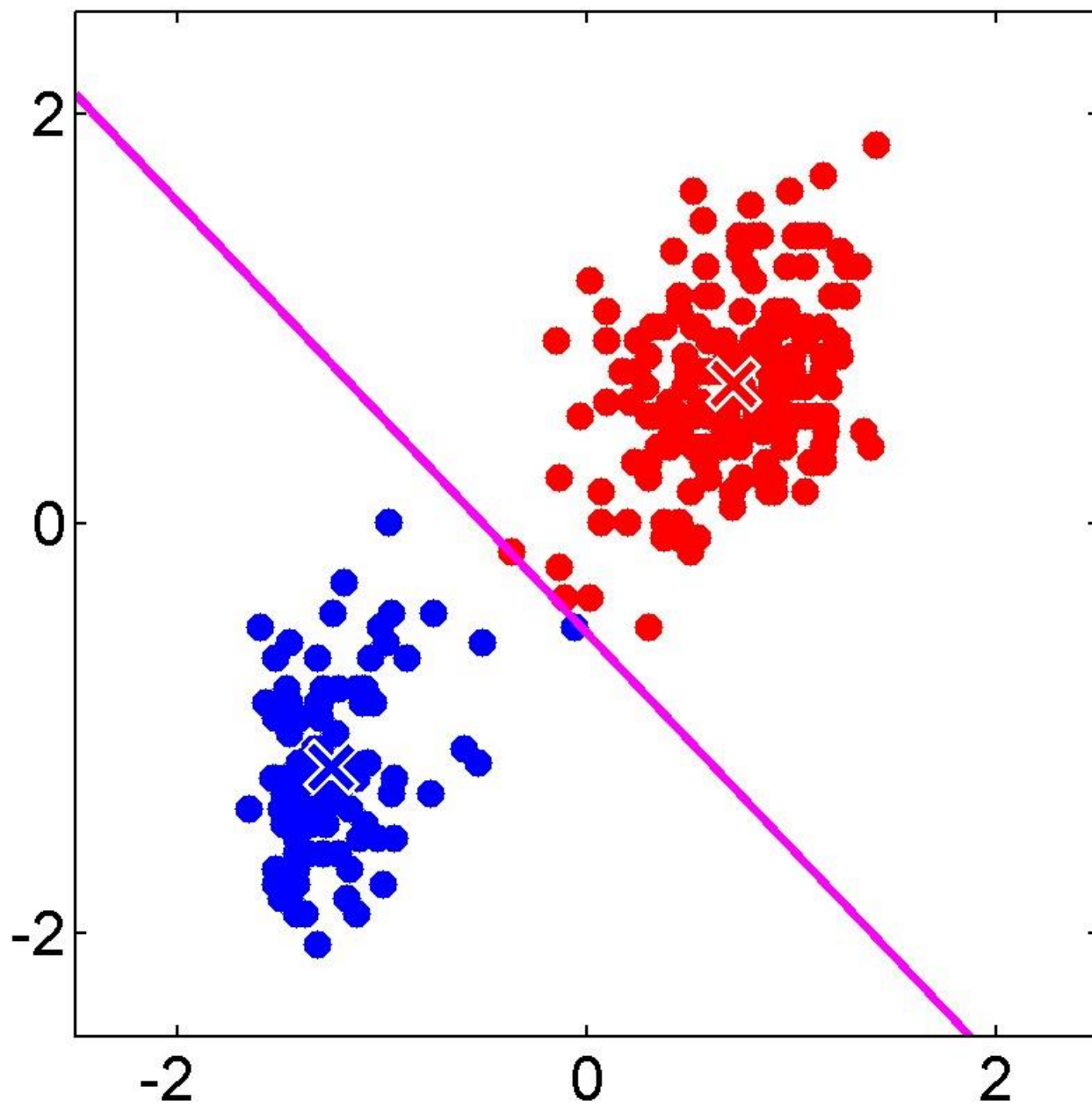


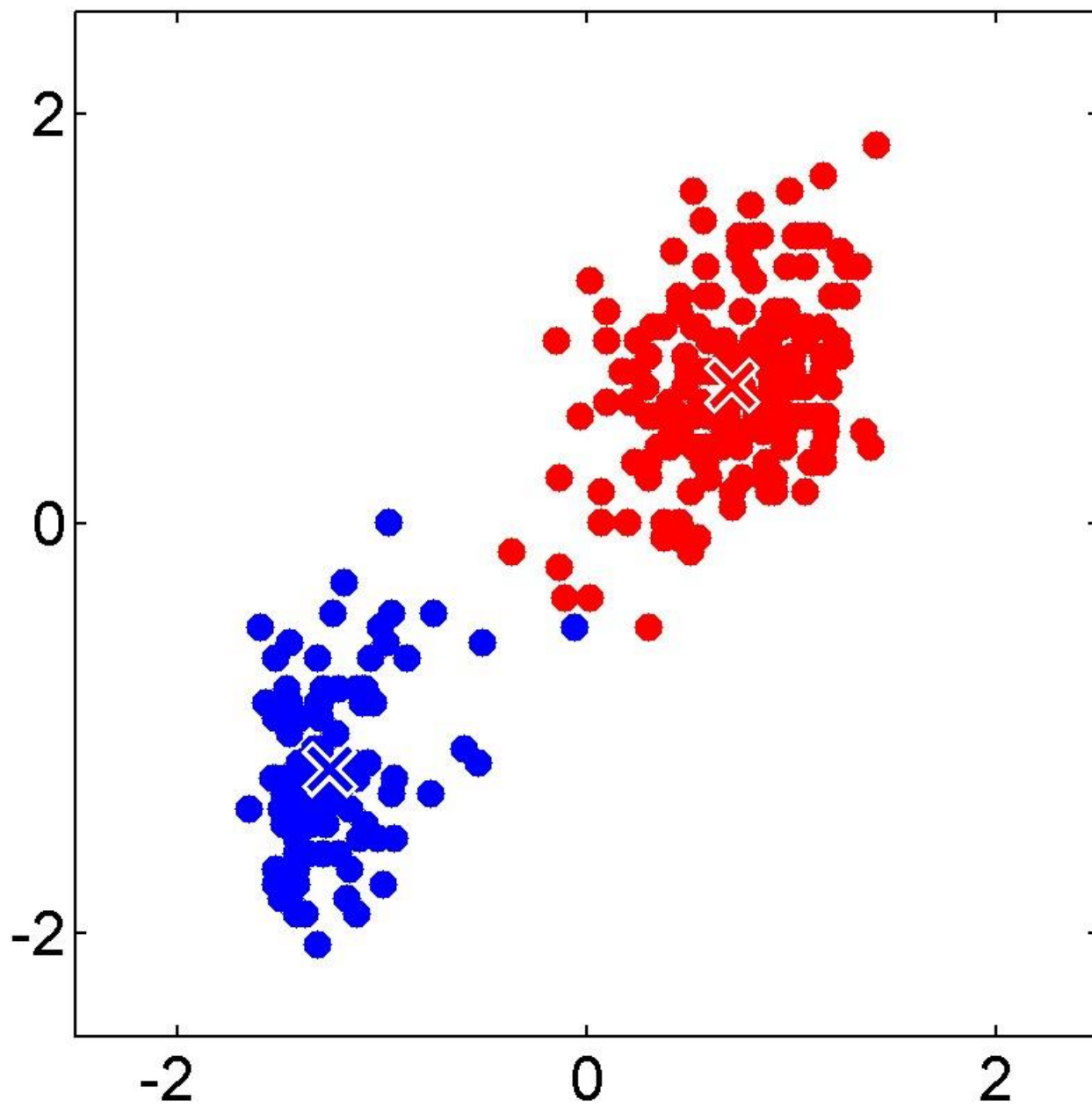




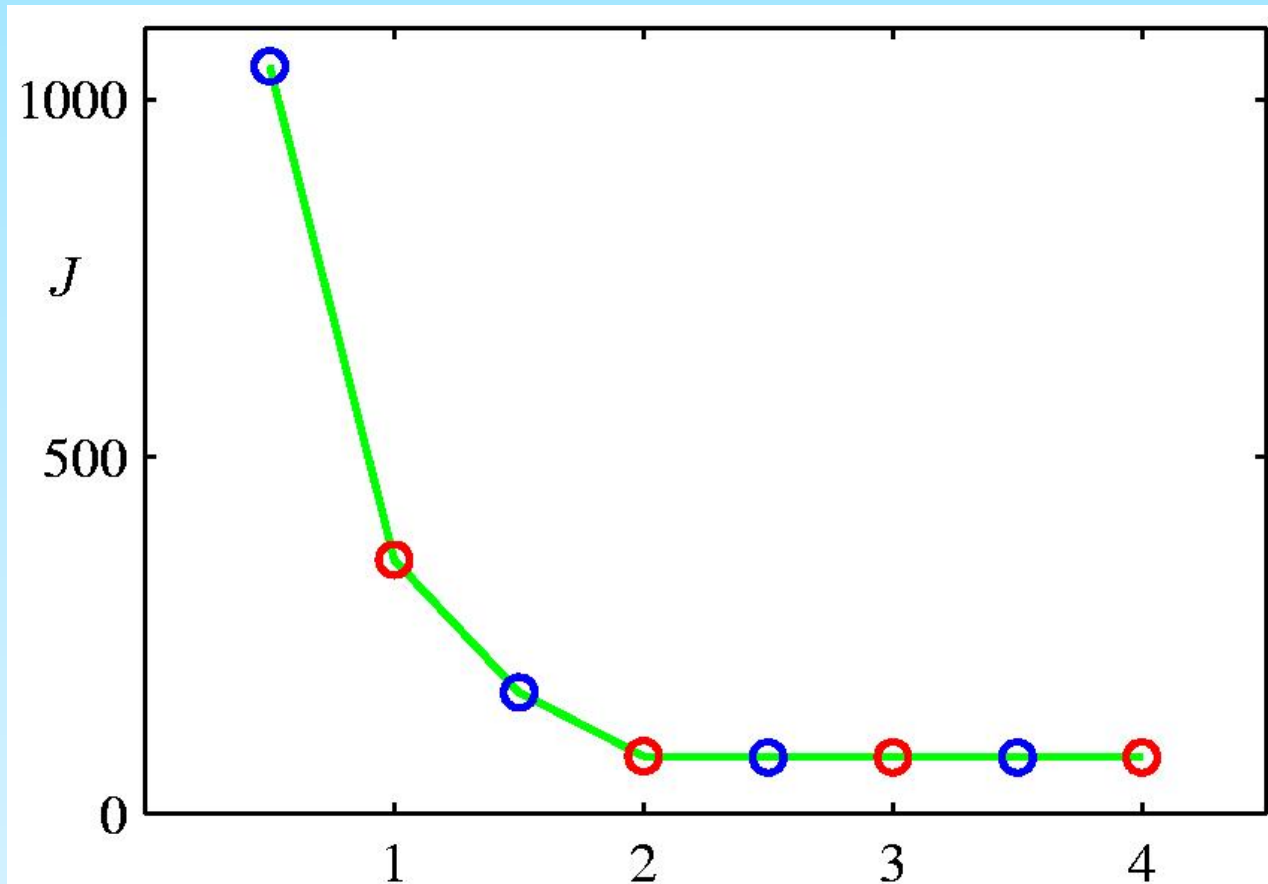








# La Fonction de Coût



$$\frac{(J_i - J_{i-1})}{J_{i-1}} < \varepsilon$$

# K-means : cas des chiffres manuscrits

## Methodologie

1. Visualiser les données

2. Lancer K-Means → connaissance a priori :  $K=10$

3. K-means est sensible à l'initialisation

→ Besoin de faire plusieurs initialisations: lancer 5 fois K-means

- (i) Garder le meilleur cas: erreur de quantification minimale

- (ii) Visualiser la convergence du K-means

## Run K-Means 5 times

iteration                      quantization error

1	1	480	52683.1
2	1	120	50474.1
3	1	35	50041.6
4	1	23	49821.5
5	1	15	49679.5
6	1	8	49626.3
7	1	5	49587.2
8	1	5	49553.5
9	1	3	49532.1
10	1	3	49517.9
11	1	2	49508.2
12	1	2	49500.8
13	1	2	49491.1

1	1	480	53130.4
2	1	135	50661.4
3	1	52	49788.4
4	1	30	49424.6
5	1	11	49346.7
6	1	6	49293.3
7	1	2	49277.2
8	1	3	49258.4
9	1	2	49246.6
10	1	2	49233.8
11	1	2	49225.5
12	1	2	49216.2
13	1	1	49210.8

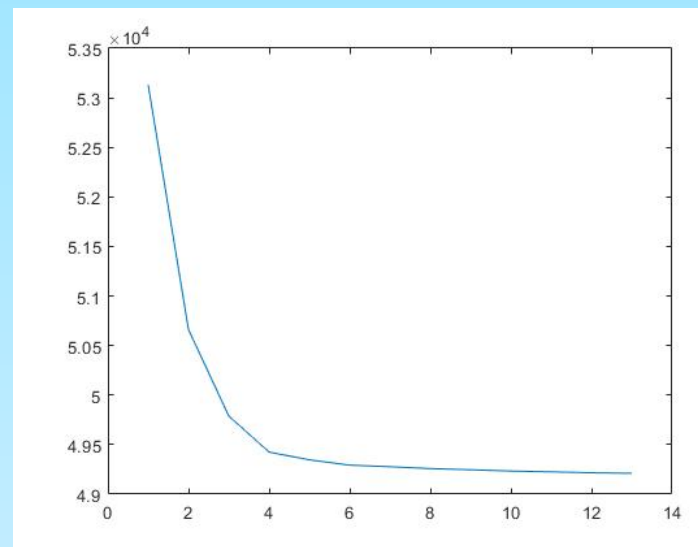
iteration

quantization error

1	1	480	53313.5
2	1	122	50481.5
3	1	37	49973.9
4	1	20	49759.4
5	1	11	49641.4
6	1	5	49608.3
7	1	7	49553.6
8	1	1	49548
9	1	1	49545.3
10	1	3	49531
11	1	5	49474.9

1	1	480	53407.5
2	1	118	51110.7
3	1	43	50556.7
4	1	28	50243.4
5	1	18	50035.9
6	1	5	49990.8
7	1	4	49966.8
8	1	3	49950.4
9	1	1	49945.2

1	1	480	51828.5
2	1	85	50395.8
3	1	39	49923
4	1	23	49631.3
5	1	5	49577.4
6	1	3	49553.1
7	1	2	49536.4
8	1	3	49509.6
9	1	1	49501.1

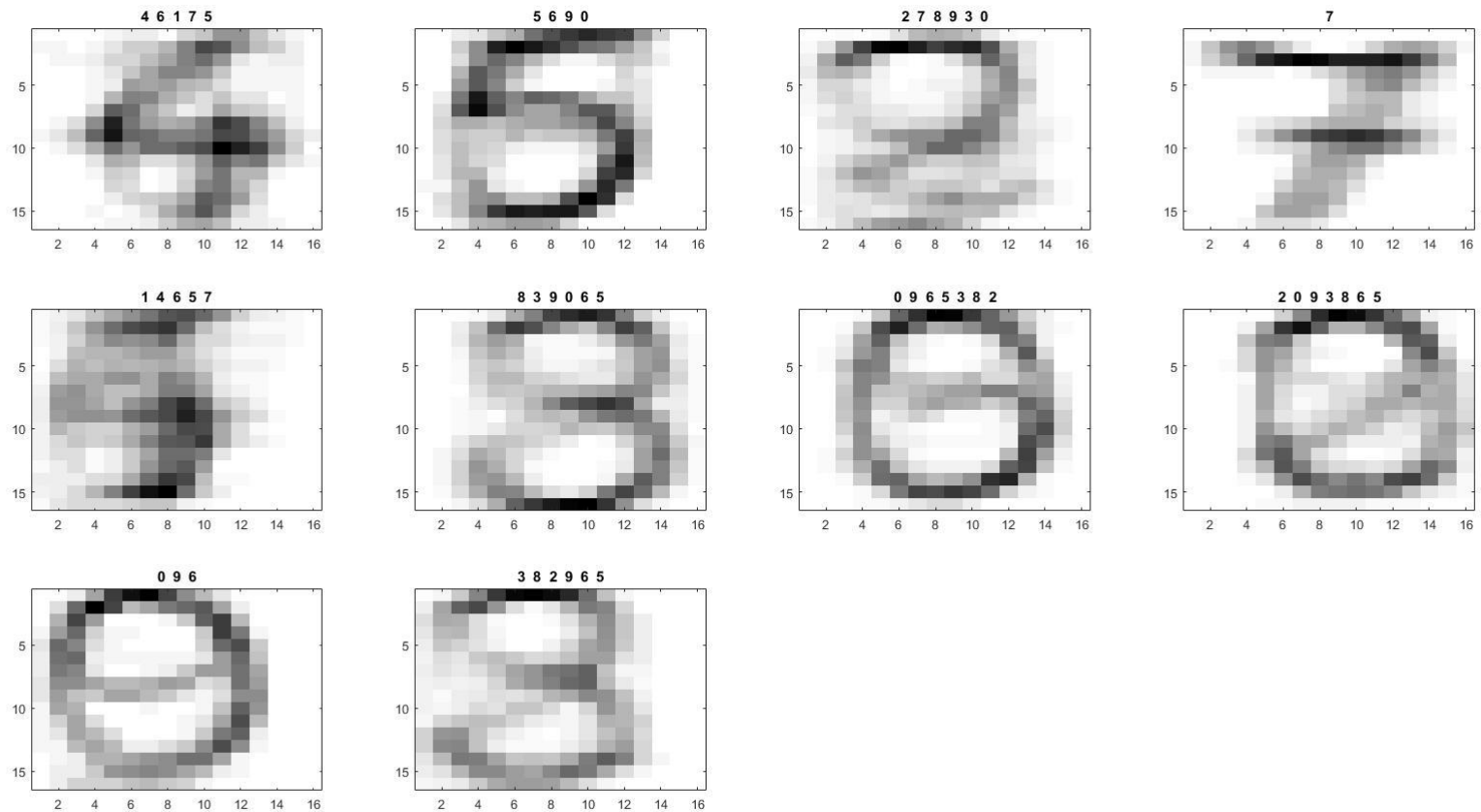


**Garder le meilleur cas:**

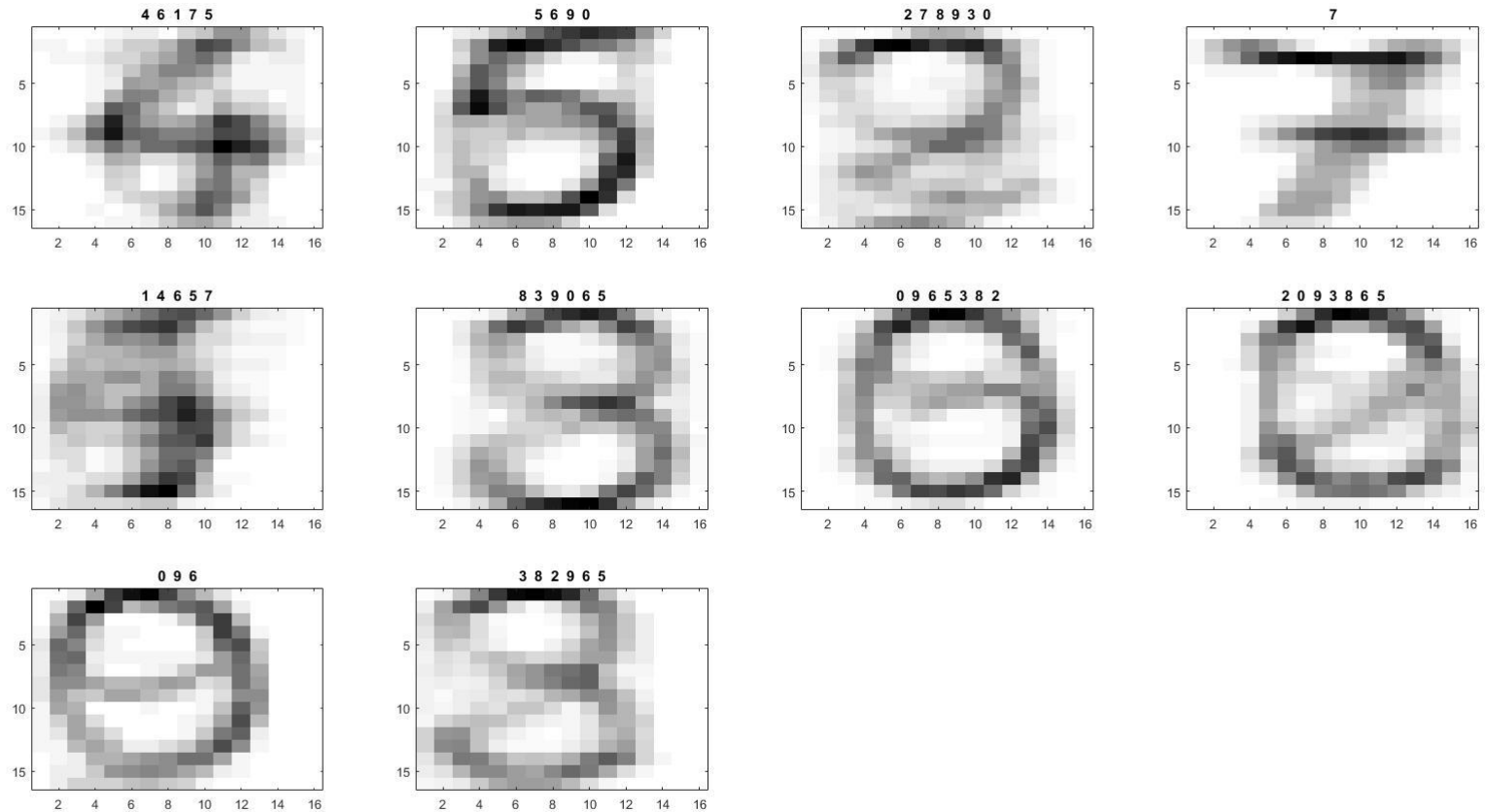
Erreur minimale= 49210.8

# Clustering

## Visualisation des centres obtenus



**Les 10 prototypes ou centres avec labels de chaque cluster dessus** 22



- 1 cluster à 1 seule classe: Cluster 4 (que le chiffre 7)
- 1 cluster à 3 classes: Cluster 9 (mélange de 0,9,6)
- 1 cluster à 4 classes: Cluster 2 (mélange de 5,6,9,0)
- 7 clusters à plus de 5 classes

# Indice de Validité : Silhouette

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}$$

- Mesure de la qualité du Clustering:  
rôle des indices de validité
- Indice de validité **interne**: n'utilise pas les étiquettes de classe
- Il existe des indices externes



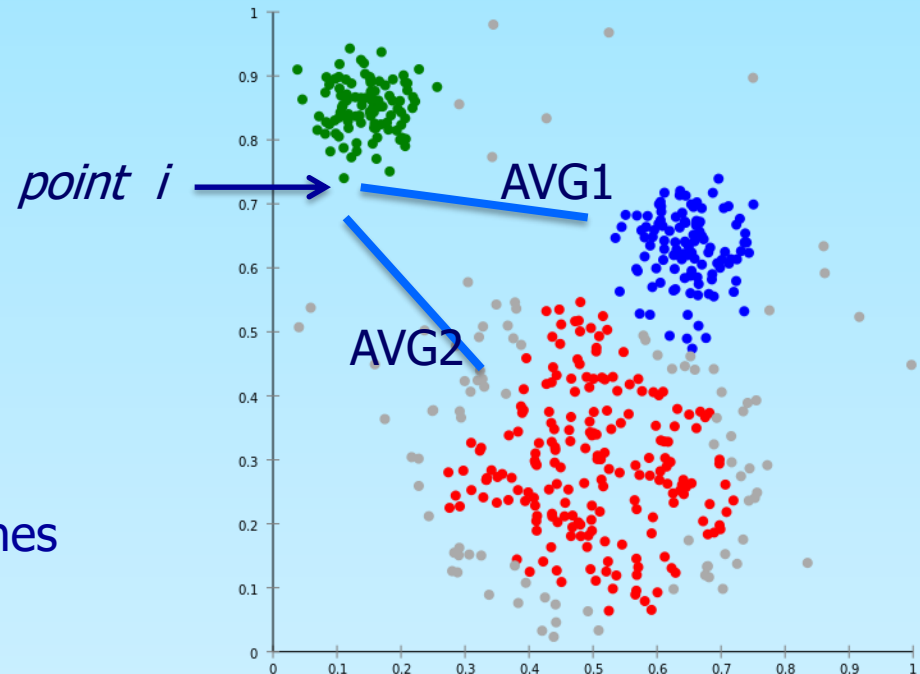
# Indice de Validité : Silhouette

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}$$

Pour le point  $i$  :

-  $a(i)$  : distance moyenne entre  $i$  et les points du cluster vert

-  $b(i)$  : *minimum* des distances moyennes entre  $i$  et les points de chaque autre cluster :  $\min(\text{AVG1}, \text{AVG2})$



**Plus  $a(i) \rightarrow 0$ , plus le cluster du point  $i$  sera mieux séparé des autres**

**$\rightarrow$  dans ce cas :  $b(i) > a(i)$  donc  $s(i) \rightarrow 1$**

**On moyenne les  $s(i)$  : indice de la Silhouette (dans  $[-1,1]$ )**

# Indice de Validité externe: Entropie

$$\eta(C_k) = - \sum_{i=1}^{N_A} \frac{p(A_i|C_k) \log_2(p(A_i|C_k))}{\log_2(N_A)}$$

$$E[\eta] = \sum_{k=1}^{N_C} \frac{|C_k|}{\left| \bigcup_{j=1}^{N_C} C_j \right|} \eta(C_k)$$


$A_i$  : class  $i$  : 0,1,2,...,9

$N_A = 10$  classes

*Left : Entropy per cluster normalized by  $\log_2(10)$*

*= max entropy when  $P(A_i|C_k) = \frac{1}{10}$*

*Right : Weighted average entropy of the partition*


$$\begin{aligned} H_{\max} &= -10(0.1 \log_2(0.1)) \\ &= -\log_2(1/10) \\ &= -(\log_2(1) - \log_2(10)) \\ &= \log_2(10) \end{aligned}$$

- Calcul par cluster d'abord puis calcul de l'indice d'entropie globale du Clustering obtenu: une valeur dans [0,1]
- Si 1 cluster a une seule classe: son entropie est 0
- Plus l'entropie globale est faible, meilleur est le Clustering

# Bibliography

- [1] Jain A.K. et al. "Data Clustering: A Review", *Pattern Recognition Letters* 31(8), pp. 651-666, 2010.
- [2] Kaufman L. and P. J. Rouseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. Hoboken, NJ: John Wiley & Sons, Inc., 1990.
- [3] Rouseeuw, P. J. "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis", *Journal of Computational and Applied Mathematics*. Vol. 20, No. 1, 1987, pp. 53–65.
- [4] Calinski, T., and J. Harabasz. "A dendrite method for cluster analysis." *Communications in Statistics*. Vol. 3, No. 1, 1974, pp. 1–27.
- [5] Davies, D. L., and D. W. Bouldin. "A Cluster Separation Measure." *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. PAMI-1, No. 2, 1979, pp. 224–227.
- [6] Maria Halkidi, Yannis Batistakis, Michalis Vazirgiannis, "On Clustering Validation Techniques", *Journal of Intelligent Information Systems*, 17:2/3, 107-145, 2001.
- [7] R.O. Duda, P. E. Hart, D.G. Stork, *Pattern Classification*, Second Edition, John Wiley, 2001.
- [8] McLachlan, G., and D. Peel, *Finite Mixture Models*, Hoboken, NJ: John Wiley & Sons, Inc., 2000.

# Travail Pratique (1)

## Classer des chiffres manuscrits en exploitant l'algorithme des K-moyennes

Optical Recognition of Handwritten Digits Data Set (*UC Irvine ML Repository*)

<https://archive.ics.uci.edu/ml/datasets/optical+recognition+of+handwritten+digits>

### A propos de la base:

Des codes de prétraitement mis à disposition par NIST ont été utilisés pour extraire des bitmaps normalisés de chiffres manuscrits d'un formulaire

Sur un total de **43 personnes**:

- **30 personnes** ont contribué à l'ensemble d'apprentissage (BA)
- **les 13 personnes restantes** à l'ensemble de test (BT)
- **Format original des données:**     **1 chiffre = 1 bitmap 32x32**
- **Extraction de caractéristiques:**
  - **chaque chiffre (bitmap 32x32) a été divisé en blocs sans recouvrement**
  - **chaque bloc est de taille 4x4**
  - **le nombre de pixels est compté dans chaque bloc** (= valeur entre 0 et 16; permet une compression des entrées et un lissage des faibles distorsions)
- **1 chiffre est devenu 1 matrice de taille 8x8**
- **chaque élément est un entier dans l'intervalle 0..16 (Nb pixels par bloc)**

# Travail Pratique (2)

## I. La base d'apprentissage (BA) à utiliser: **optdigits.tra**

- BA: 3823 chiffres manuscrits
  - X= matrice des entrées de la BA:
    - Chaque ligne = 1 chiffre → X : 3823 lignes
    - Chaque chiffre est représenté par  $8 \times 8 = 64$  valeurs + étiquette = 65 colonnes
- X: matrice 3823x65

## II. La base de test (BT) à utiliser pour classer : **optdigits.tes**

- 1797 chiffres manuscrits

# Travail Pratique (3)

- L'implémentation: outil de votre choix (R, Matlab, Python, etc. )
- Rédiger un rapport décrivant: techniques, résultats et analyses

## DEMARCHE A SUIVRE (APPRENTISSAGE)

### **I. Apprentissage**

1. **Faire un K-moyennes avec  $K=10$  sur la base d'apprentissage (BA) :**  
**optdigits.tra**
2. **Par cluster: faire un histogramme du nombre de chiffres de chaque classe**  
→ Analyser si les clusters ont un sens (classe la plus représentée, ressemblance avec d'autres classes...)
3. **Mesurer la qualité du Clustering avec l'indice de la Silhouette**  
→ Est-ce un bon Clustering?
4. **Faire varier K entre 10 et 20 clusters et calculer pour chaque K l'indice de la Silhouette** → Pour quelle valeur de K obtenez-vous un meilleur Clustering?

# Travail Pratique (4)

**DEMARCHE A SUIVRE (TEST): considérer le meilleur Clustering obtenu**

**II. Test: Classification à partir du Clustering obtenu sur `optdigits.tra`**

- 1. Par cluster: faire un vote à la majorité pour attribuer un label à chaque cluster** (la classe la plus représentée dans chaque cluster)
- 2. Pour chaque élément de la BT (Base de Test) : `optdigits.tes`**
  - Chercher le Cluster (Centre) le plus proche
  - Attribuer à cet élément de la BT le label associé au Cluster le plus proche
  - Calculer la matrice de confusions (matrice 10x10) et la performance globale: analyser les confusions