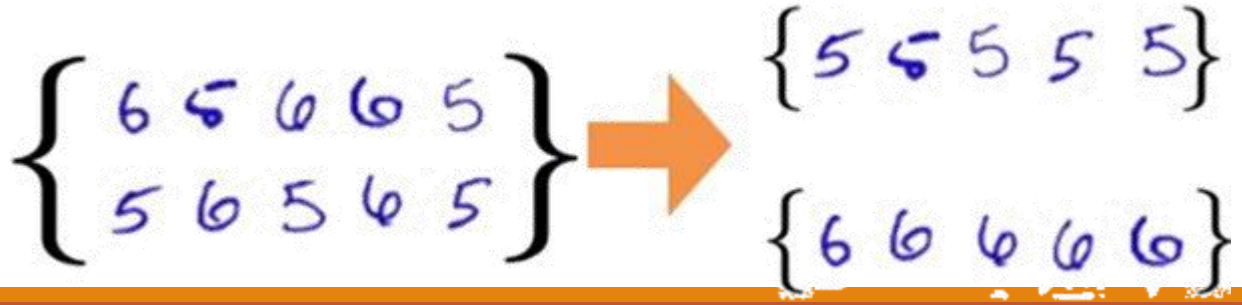
Classification et groupement de données

HADJADJ ABDELHALIM

Centre universitaire de Mila
Master I: Apprentissage automatique

Introduction

•L’apprentissage non supervisé est un **type d’apprentissage automatique**.
•Il est utilisé lorsque les **données d’entrée ne sont pas accompagnées d’étiquettes ou de sorties connues**.
•L’objectif est de **découvrir automatiquement des structures, des regroupements ou des modèles** dans les données.
•Cet apprentissage **ne repose pas sur des exemples pré-classés**, contrairement à l’apprentissage supervisé.
•Il permet d’**explorer et analyser les données** pour en extraire des **informations cachées**.



Apprentissage non supervisé

Dans de nombreuses situations, les données ne sont pas étiquetées (pas de classes connues).

•Le groupement (clustering) permet de :

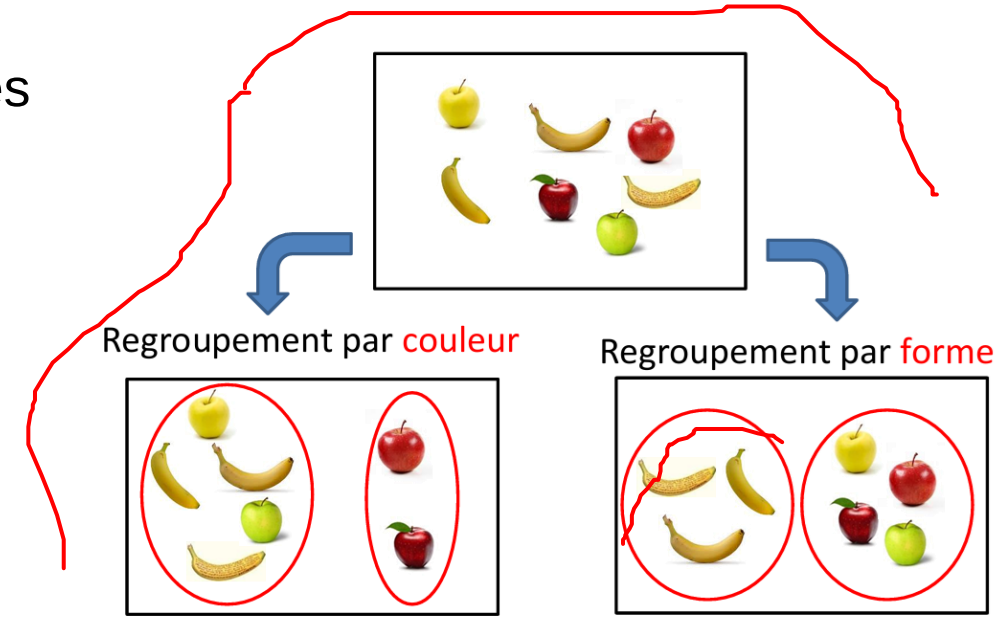
➤ Regrouper automatiquement des objets similaires,

➤ Identifier des structures cachées,

➤ Réduire la complexité en classant les données en groupes homogènes.

•C'est un outil clé pour :

- •Pour le codage et la compression de données.
- •Pour réduire le nombre de données d'apprentissage
- •Pour la classification de données. ,
- •La préparation de données pour d'autres algorithmes.



Algorithme des K-moyennes

Principe

- Un **cluster** (ou groupe) est un ensemble de points (ou objets) proches les uns des autres dans l'espace.
- L'objectif est de **regrouper** les données en **K groupes** tels que :
  - Les **éléments d'un même cluster** soient **similaires** (proches les uns des autres).
  - Les **éléments de différents clusters** soient **différents** (éloignés).

Algorithme des K-moyennes

Pour chaque donnée $x(i)$, on associe une variable indicatrice $r_{ik} \in 0,1$, qui prendra sa valeur comme suit: $r_{ik} = 1\ si\ x^{(i)} \in G^k$ ou $0\ si\ x(i) \notin G^k$

• On peut définir une fonction objective (somme des carrés résiduels: SCR) à minimiser pour réaliser un groupement optimal:

$$SCR = \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \| x^{(i)} - \boldsymbol{\mu}^{(k)} \|^2$$

• Le but est de trouver les r_{ik} et les $\mu_{(k)}$ pour minimiser SCR(somme des carrés résiduels)?

Algorithme des K-moyennes

Étape 1 : Affectation (Assignment step)
Pour chaque donnée $x^{(i)}$, on l'associe au centre le plus proche :

$$r_{ik} = \begin{cases} 1 & \text{si } k = \arg\min_l \|x^{(i)} - \mu^{(l)}\|^2 \\ 0 & \text{sinon} \end{cases}$$

Cela minimise le SCR par rapport à r_{ik}(somme des carrés résiduels).

Algorithme des K-moyennes

Algorithme K-moyennes

Données :

Un ensemble de points : $D = \{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\}$

Chaque point a D dimensions (par exemple : (x, y) dans un plan).

Étapes principales :

1. **Initialisation** : choisir aléatoirement K centres $\mu^{(1)},\mu^{(2)},\ldots,\mu^{(K)}$
2. **Affectation** : pour chaque point $x^{(i)}$, l'assigner au cluster dont le centre est **le plus proche** (distance euclidienne).
3. **Mise à jour** : recalculer chaque centre $\mu^{(k)}$ comme la **moyenne** des points assignés à ce cluster.
4. Répéter les étapes 2 et 3 jusqu'à ce que les affectations ne changent plus (convergence).

Algorithme des K-moyennes

Étape 2 : Mise à jour des centres (Update step)

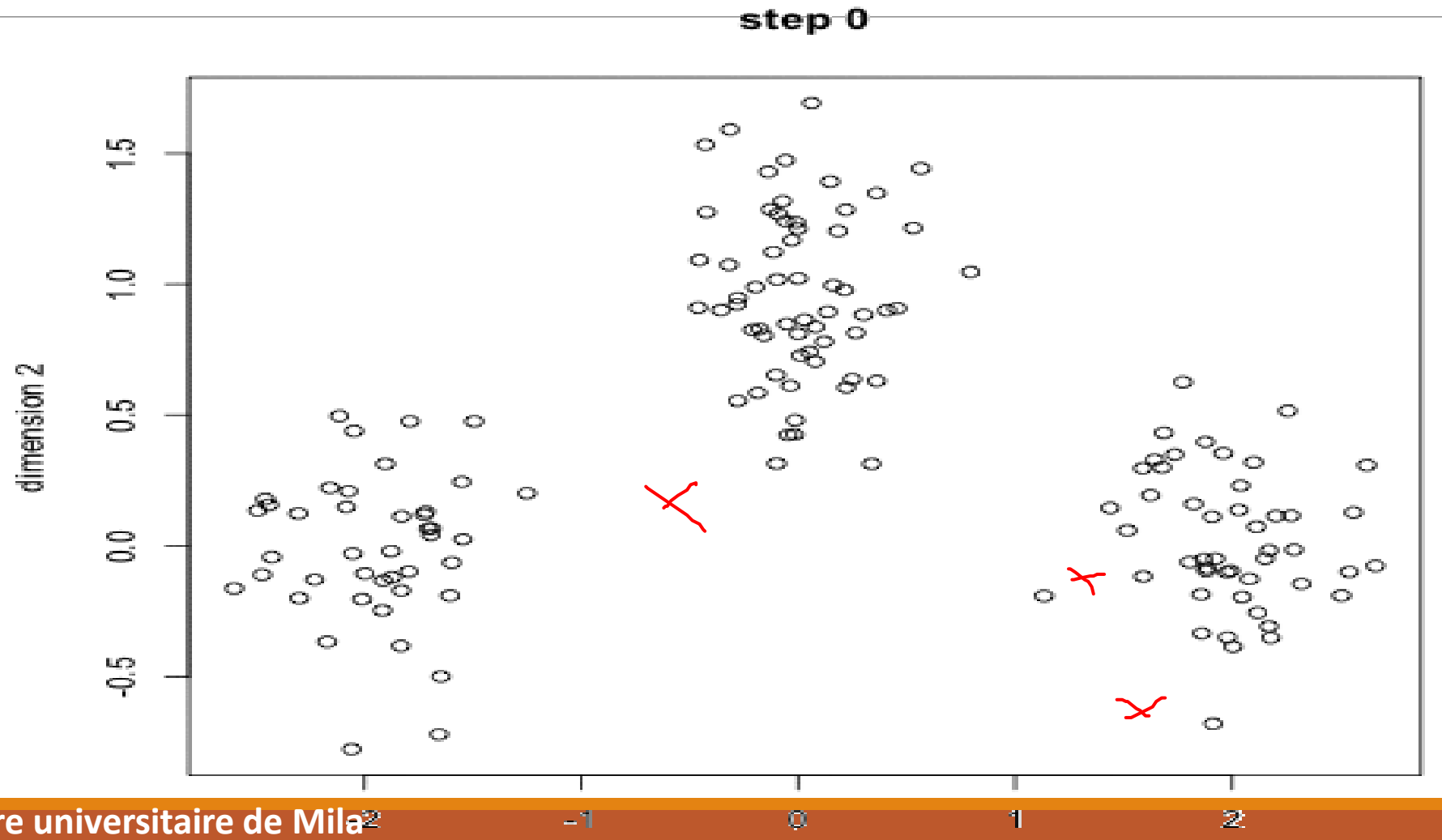
On met à jour chaque centre $\mu^{(k)}$ en faisant la moyenne des données qui lui sont affectées :

$$\mu^{(k)} = \frac{\sum_{i=1}^{N} r_{ik} x^{(i)}}{\sum_{i=1}^{N} r_{ik}}$$

Cela minimise le SCR par rapport à $\mu^{(k)}$

- Les deux étapes (affectation et mise à jour) sont répétées jusqu'à convergence (i.e., plus de changement dans les affectations ou les centres).
- Les centres $\mu^{(k)}$ sont initialisés aléatoirement (ou avec des méthodes comme K-means^{++}).

Algorithme du K-moyennes (exemple 2)



Mélange de distributions Gaussiennes (MDG)

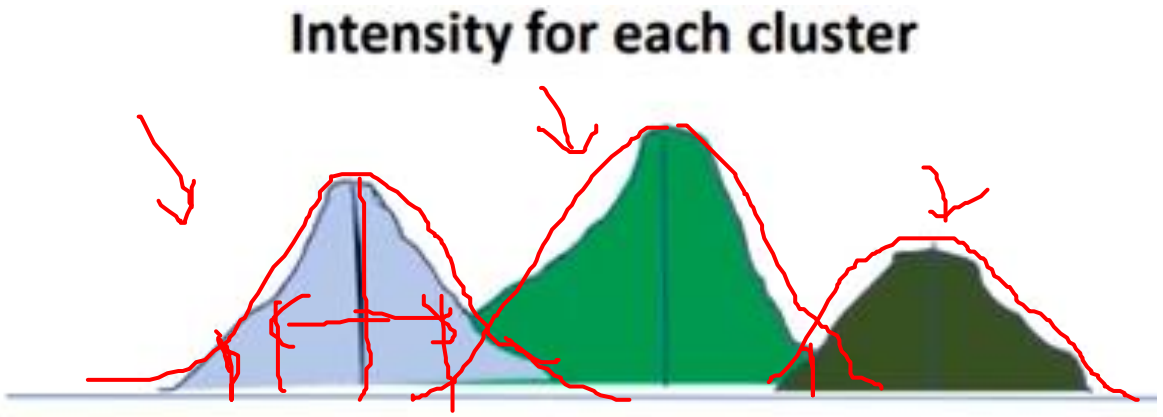
L'un des principaux inconvénients de l'algorithme de clustering K-Means est son utilisation simple de la moyenne du cluster comme centre.

cette méthode n'est pas la meilleure façon de procéder au clustering de certaines données.

- Dans la figure sur le côté gauche, il apparaît assez clairement qu'il y a deux cercles avec des rayons différents. L'algorithme de clustering K-Means ne peut pas gérer ce type de données, car les valeurs moyennes des deux groupes sont très proches l'une de l'autre.
- Sur le côté droit, l'algorithme n'a pas réussi à identifier les deux groupes, car les deux groupes ne sont pas centrés autour du centre ou du milieu, et cet échec se produit en raison de l'utilisation de la moyenne du groupe

Les Mélanges de Distributions Gaussiennes

Les **Mélanges de Distributions Gaussiennes (MDG)** sont une **extension probabiliste** du K-moyennes qui permet de **modéliser des clusters plus réalistes** (forme, orientation, incertitude



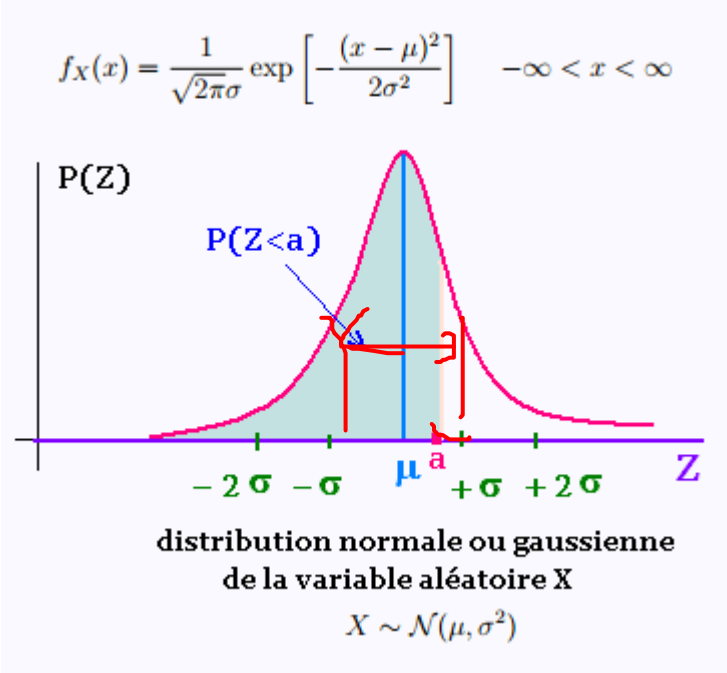
Rappel sur la loi Gaussienne

Soit D variables aléatoires $\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_D$ et soit $\mu_1, \mu_2, \ldots, \mu_D$ leurs moyennes, respectivement.

On définit la matrice de covariance $\boldsymbol{\Sigma}$ de dimension $D \times D$ dont les entrées sont définies par: $\boldsymbol{\Sigma}_{i,j} = cov(\mathrm{x}_i, \mathrm{x}_j)$.

On définit la loi Gaussienne multivariée par: $\mathrm{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ où $\mathrm{x} = (\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_D)^T$ et $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_D)^T$:

$$f(\mathrm{x} = u) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(u - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(u - \boldsymbol{\mu})\right)$$



Rappel sur la loi Gaussienne

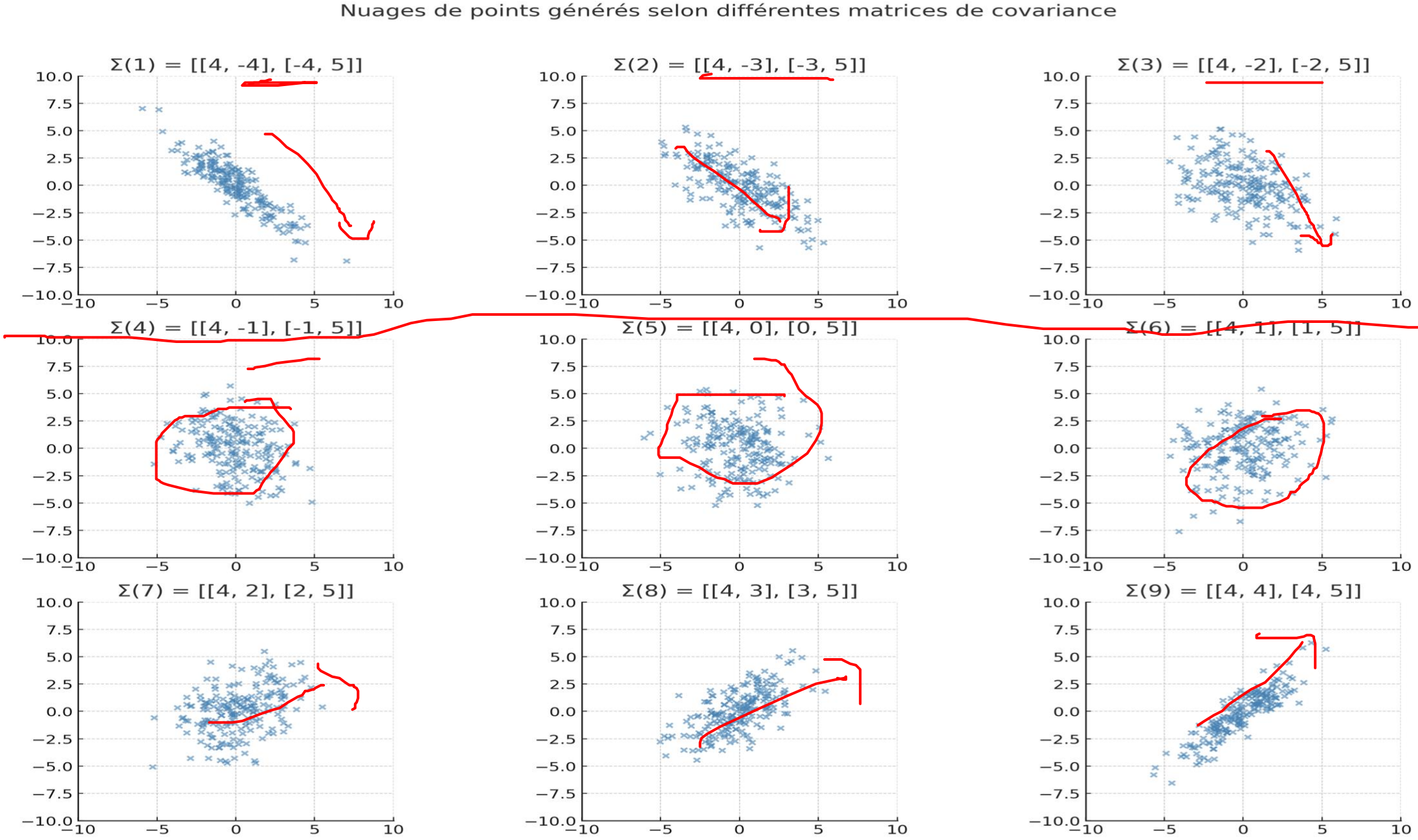
- Par exemple, pour $D = 2$, on peut définir les distribution Gaussiennes suivantes $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Pour chaque distribution on a $\boldsymbol{\mu} = (0,0)^T$ et les matrices de covariances sont données comme suit:

$$\boldsymbol{\Sigma}^{(1)} = \begin{bmatrix} 4 & -4 \\ -4 & 5 \end{bmatrix} \quad \boldsymbol{\Sigma}^{(2)} = \begin{bmatrix} 4 & -3 \\ -3 & 5 \end{bmatrix} \quad \boldsymbol{\Sigma}^{(3)} = \begin{bmatrix} 4 & -2 \\ -2 & 5 \end{bmatrix}$$

$$\boldsymbol{\Sigma}^{(4)} = \begin{bmatrix} 4 & -1 \\ -1 & 5 \end{bmatrix} \quad \boldsymbol{\Sigma}^{(5)} = \begin{bmatrix} 4 & 0 \\ 0 & 5 \end{bmatrix} \quad \boldsymbol{\Sigma}^{(6)} = \begin{bmatrix} 4 & 1 \\ 1 & 5 \end{bmatrix}$$

$$\boldsymbol{\Sigma}^{(7)} = \begin{bmatrix} 4 & 2 \\ 2 & 5 \end{bmatrix} \quad \boldsymbol{\Sigma}^{(8)} = \begin{bmatrix} 4 & 3 \\ 3 & 5 \end{bmatrix} \quad \boldsymbol{\Sigma}^{(9)} = \begin{bmatrix} 4 & 4 \\ 4 & 5 \end{bmatrix}$$

- 200 données ont été générées pour chaque Gaussienne.


Nuages de points générés selon différentes matrices de covariance
Σ(1) = [[4, -4], [-4, 5]]
Σ(2) = [[4, -3], [-3, 5]]
Σ(3) = [[4, -2], [-2, 5]]
Σ(4) = [[4, -1], [-1, 5]]
Σ(5) = [[4, 0], [0, 5]]
Σ(6) = [[4, 1], [1, 5]]
Σ(7) = [[4, 2], [2, 5]]
Σ(8) = [[4, 3], [3, 5]]
Σ(9) = [[4, 4], [4, 5]]

Mélanges de distributions Gaussiennes

- On suppose que les données de $\mathcal{D}$ sont constituées de K groupes: $G_1, \ldots, G_K$. Chaque groupe G_k a une distribution Gaussienne de moyenne $\boldsymbol{\mu}^{(k)}$ et covariance $\boldsymbol{\Sigma}^{(k)}$.

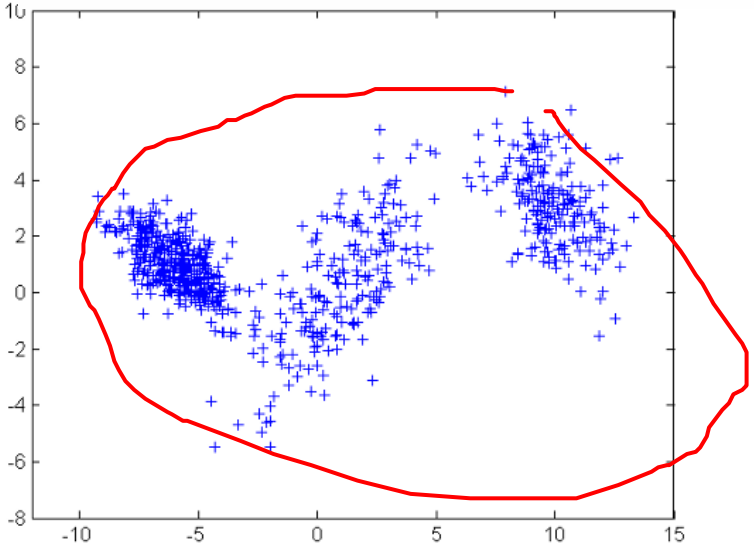
- Soit une donnée $x^{(i)} \in \mathcal{D}$. Par marginalisation, on définit la probabilité $p(x^{(i)})$ de cette donnée comme suit:

$$p(x^{(i)}) = \sum_{k=1}^{K} p(x^{(i)}|G_k)p(G_k) \qquad \text{où} \qquad \sum_{k=1}^{K} p(G_k) = 1$$

Où $p(x|G_k) = \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ et $p(G_k)$ est la probabilité du groupe G_k.

Mélanges de distributions Gaussiennes

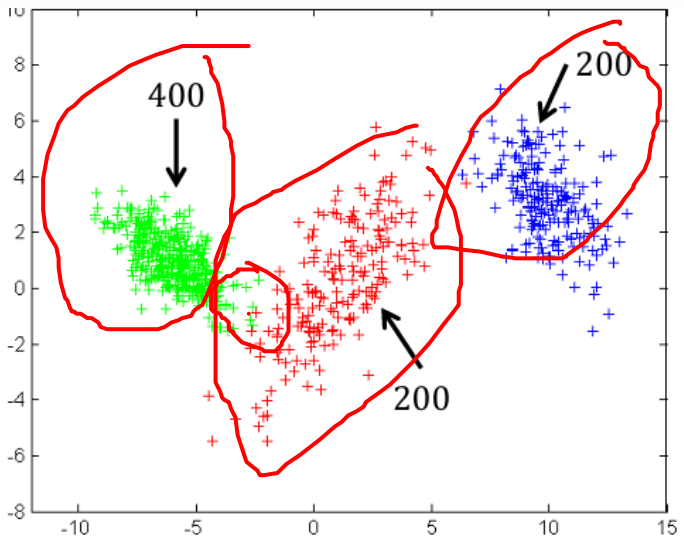
- Par exemple, soit l'ensemble $\mathcal{D}$ de $N = 800$ données de dimension $D = 2$ qui forme un mélange de $K = 3$ groupes:



$\boldsymbol{\mu}^{(1)} = \begin{bmatrix} -6 \\ 1 \end{bmatrix}$

$\boldsymbol{\Sigma}^{(1)} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}$

$p(G_1) = 1/2$



$\boldsymbol{\mu}^{(2)} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

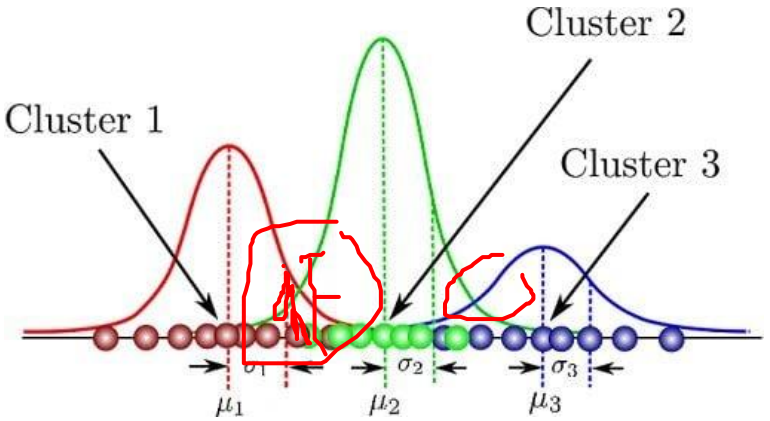
$\boldsymbol{\Sigma}^{(2)} = \begin{bmatrix} 4 & 3 \\ 3 & 5 \end{bmatrix}$

$p(G_2) = 1/4$

$\boldsymbol{\mu}^{(3)} = \begin{bmatrix} 10 \\ 3 \end{bmatrix}$

$\boldsymbol{\Sigma}^{(3)} = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$

$p(G_3) = 1/4$



Estimation des paramètres d'un MDG

Cas 1: Les données sont déjà assignées aux groupes.

- Pour K groupes: $G_1, \ldots, G_K$, on doit estimer 3 paramètres pour chaque groupe G_k: $\boldsymbol{\mu}^{(k)}, \boldsymbol{\Sigma}^{(k)}, p(G_k)$.

- On dénote par $\boldsymbol{\varphi}$ l'ensemble de tous les paramètres à estimer:

$$\boldsymbol{\varphi} = \{G_k: \boldsymbol{\mu}^{(k)}, \boldsymbol{\Sigma}^{(k)}, p(G_k),\ k = 1, \ldots, K\}$$

- Pour chaque donnée $x^{(i)}$, on associe une variable indicatrice:

$$r_{ik} = \begin{cases} 1 & si \quad x^{(i)} \in G_k \\ 0 & si \quad x^{(i)} \notin G_k \end{cases},\ k \in \{1, \ldots, K\}.$$

Mélanges de distributions Gaussiennes

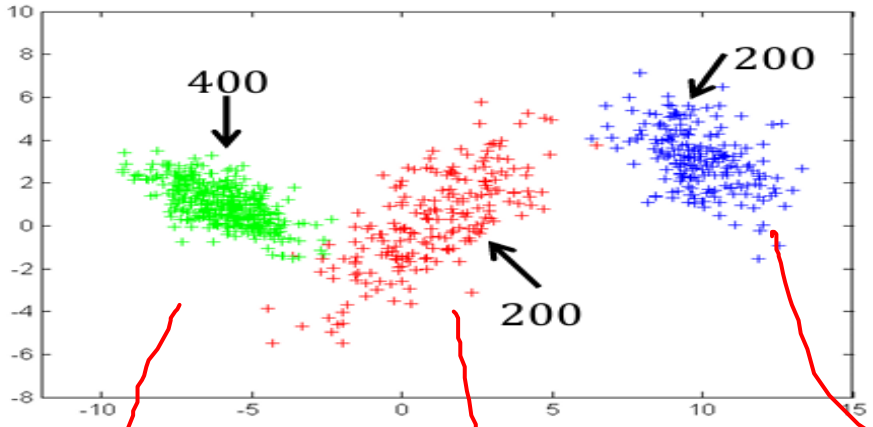
- Il est facile de démontrer que:

$$p(G_k) = \frac{\sum_{i=1}^{N} r_{ik}}{N}$$

$$\hat{\boldsymbol{\mu}}^{(k)} = \frac{\sum_{i=1}^{N} r_{ik} x^{(i)}}{\sum_{i=1}^{N} r_{ik}}$$

$$\hat{\boldsymbol{\Sigma}}^{(k)} = \frac{\sum_{i=1}^{N} r_{ik} \left(x^{(i)} - \mu^{(k)}\right)\left(x^{(i)} - \mu^{(k)}\right)^{T}}{\sum_{i=1}^{N} r_{ik}}$$

Estimation des paramètres d'un MDG



- L'estimation des paramètres par Matlab avec $\mathcal{D}$ va donner :

$$\widehat{\boldsymbol{\mu}}^{(1)} = \begin{bmatrix} -5{,}9 \\ 0{,}96 \end{bmatrix} \qquad \widehat{\boldsymbol{\Sigma}}^{(1)} = \begin{bmatrix} 1{,}72 & -0{,}85 \\ -0{,}85 & 0{,}97 \end{bmatrix} \qquad p(G_1) = 1/2$$

$$\widehat{\boldsymbol{\mu}}^{(2)} = \begin{bmatrix} 1{,}08 \\ 0{,}03 \end{bmatrix} \qquad \widehat{\boldsymbol{\Sigma}}^{(2)} = \begin{bmatrix} 3{,}63 & 2{,}23 \\ 2{,}23 & 4{,}06 \end{bmatrix} \qquad p(G_2) = 1/4$$

$$\widehat{\boldsymbol{\mu}}^{(3)} = \begin{bmatrix} 10{,}1 \\ 2{,}99 \end{bmatrix} \qquad \widehat{\boldsymbol{\Sigma}}^{(3)} = \begin{bmatrix} 1{,}91 & -1{,}06 \\ -1{,}06 & 1{,}97 \end{bmatrix} \qquad p(G_3) = 1/4$$

Estimation des paramètres d'un MDG

Cas 2: Les donnée ne sont pas déjà assignées aux groupes.

- On utilise le maximum de vraisemblance:

$$\ell(\boldsymbol{\varphi}) = \log\left[\prod_{i=1}^{N} p(x^{(i)}|\boldsymbol{\varphi})\right]$$

$$= \sum_{i=1}^{N} \log\left[\sum_{k=1}^{K} p(x^{(i)}|G_k)p(G_k)\right]$$

- On ne peut pas trouver les paramètres $\boldsymbol{\varphi}$ de manière exacte.

Algorithme Espérance-Maximisation

☞ **Étape 1 (Espérance):**

Pour chaque données $x^{(i)}$, et pour chaque groupe G_k, on calcule la probabilité a posteriori $p(G_k|x^{(i)})$, comme suit:

$$p(G_k|x^{(i)}) = \frac{p(x^{(i)}|G_k)p(G_k)}{\sum_{j=1}^{K} p(x^{(i)}|G_j)p(G_j)}$$

- On dénotera cette probabilité $p(G_k|x^{(i)})$ par: t_{ik}.

Algorithme Espérance-Maximisation

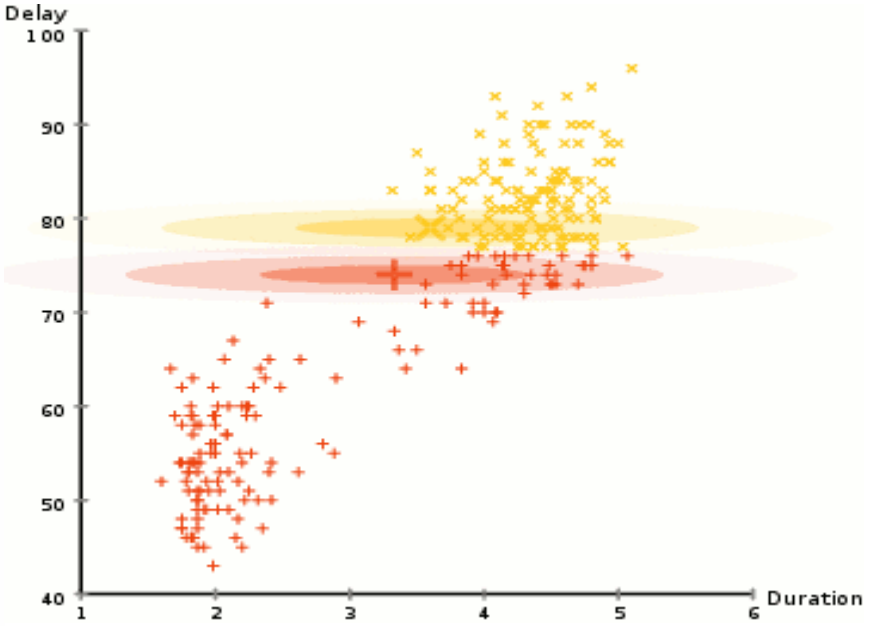
☞ **Étape 2 (Maximisation):**

Pour chaque groupe G_k, on estime ses paramètres G_k: $\mu^{(k)}$, $\boldsymbol{\Sigma}^{(k)}$, $p(G_k)$ comme suit:

$$p(G_k) = \frac{\sum_{i=1}^{N} t_{ik}}{N}$$

$$\widehat{\boldsymbol{\mu}}^{(k)} = \frac{\sum_{i=1}^{N} t_{ik} x^{(i)}}{\sum_{i=1}^{N} t_{ik}}$$

$$\widehat{\boldsymbol{\Sigma}}^{(k)} = \frac{\sum_{i=1}^{N} t_{ik} \left(x^{(i)} - \widehat{\boldsymbol{\mu}}^{(k)}\right)\left(x^{(i)} - \widehat{\boldsymbol{\mu}}^{(k)}\right)^{T}}{\sum_{i=1}^{N} t_{ik}}$$



On répète les étapes 1 et 2 jusqu'à la convergence de l'algorithme EM.

Regroupement hiérarchique de données

Le regroupement hiérarchique de données (ou clustering hiérarchique) est une méthode d'analyse de données qui vise à regrouper des objets similaires en formant une hiérarchie d'ensembles emboîtés (sous forme d'un arbre appelé dendrogramme

Regroupement hiérarchique de données

Le regroupement hiérarchique se base uniquement sur l'**analyse de similarité entre les données** pour former des groupes.

- Il cherche à identifier des **groupes d'instances** telles que les éléments d'un même groupe soient **plus similaires entre eux** que ceux appartenant à des groupes différents.
- Cette similarité est généralement mesurée à l'aide d'une **fonction de distance**, la plus couramment utilisée étant la **distance euclidienne**.

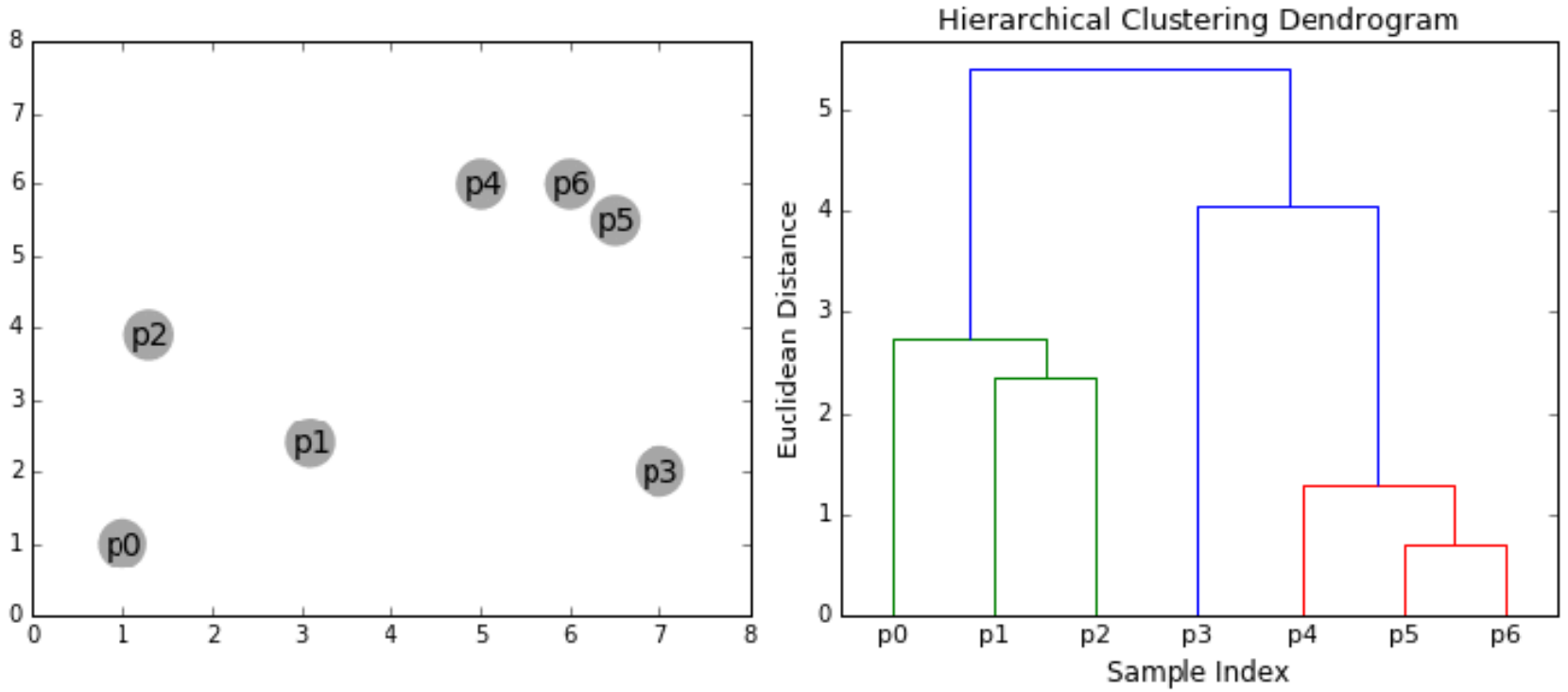
Principe du groupement hiérarchique

On considère une **séquence de partitionnements** de N données $x^{(1)},x^{(2)},\ldots,x^{(N)}$ en groupes :

- Au départ, chaque donnée constitue un groupe distinct : on a donc **N groupes**.
- Au second niveau, les deux données ou groupes les plus proches sont fusionnés, ce qui donne **N−1 groupes**.
- Au niveau suivant, un autre regroupement est effectué, menant à **N−2 groupes**, Ainsi de suite, jusqu'à ce que toutes les données soient regroupées en **un seul cluster** : le **N-ième partitionnement contient 1 seul groupe**.

Principe du groupement hiérarchique

La représentation naturelle d'un groupement hiérarchique ressemble à un arbre et est appelé un dendrogramme.



Principe du groupement hiérarchique

La construction de regroupements hiérarchiques peut suivre deux approches principales :

- **Approche agglomérative (ascendante)** :
  Elle débute avec N groupes, où chaque donnée forme un groupe distinct.
  Les groupes sont ensuite **fusionnés progressivement** jusqu'à obtenir un seul groupe global.
- **Approche divisive (descendante)** :
  Elle commence par un **seul groupe contenant toutes les données**, qui est ensuite **divisé successivement** en sous-groupes plus petits.

Parmi ces deux méthodes, l'approche agglomérative est généralement **plus simple à mettre en œuvre**.

Principe du groupement hiérarchique

Algorithme:

Entrées: $K, \{x^{(1)}, x^{(2)}, \dots, x^{(N)}\}$,

Sortie: K groupes.

$G_i \leftarrow \{x^{(i)}\}, i = 1, \dots, N;$

$c \leftarrow N;$

Tant que $(c \neq K)$

Trouver les groupes G_j et G_h les plus similaires;

Fusionner G_j et G_h;

$c \leftarrow c - 1;$

Fin

Principe du groupement hiérarchique

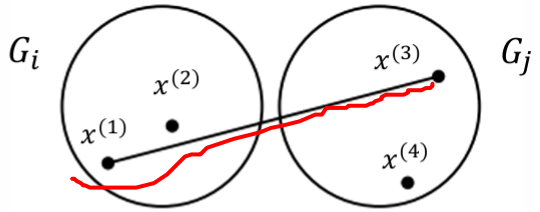
- Notons que le groupement est imbriqué, c.-à-d. si 2 exemples de données appartiennent au même groupe dans un niveau, il resteront dans un même groupe dans les niveaux supérieurs.

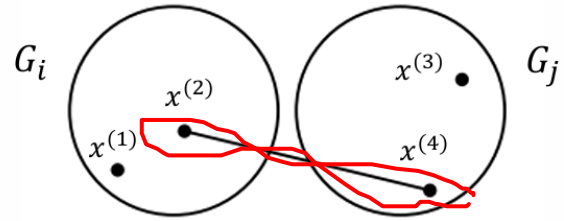
- Cela implique aussi un désavantage: si une erreur de groupement existe dans un niveau inferieur, elle se propagera dans le reste des niveaux supérieurs.

- À un niveau k, il existe $N - k + 1$ groupes. Pour choisir la paire de groupes à fusionner, il faudra faire $\binom{N-k+1}{2}$ calculs de similarité. Le nombre total de calculs exigés pour avoir K groupes est: $\sum_{k=1}^{K} \binom{N-k+1}{2}$

Mesures de similarité

- Groupement par points similaires (single Link) mesure la similarité entre deux groupes par la distance entre les points les plus proches appartenant aux deux groupes.
- Groupement par points dissimilaires (complete Link) mesure la similarité entre deux groupes par la distance entre les points les plus éloignés appartenant aux deux groupes.





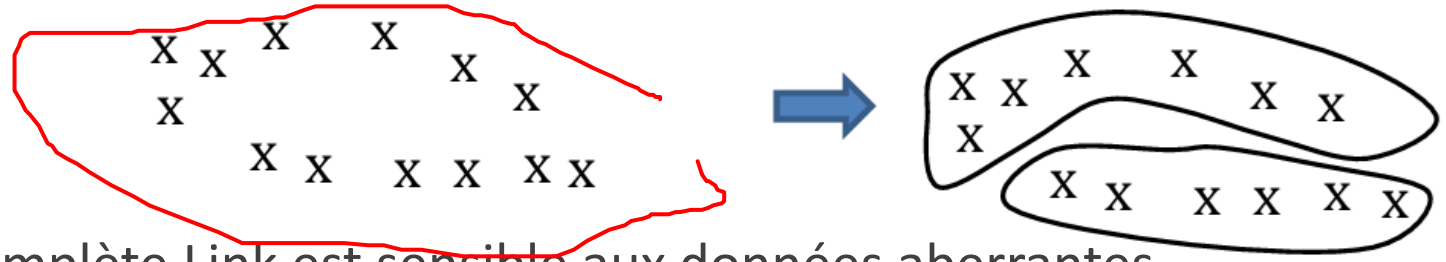
$$dist_{max}(G_i, G_j) = max\|x^{(i)} - x^{(j)}\|;$$

$$dist_{min}(G_i, G_j) = min\|x^{(i)} - x^{(j)}\|;$$

Mesures de similarité

Inconvénients:

1. La mesure single Link peut produire des groupes épars.



2. La mesure complète Link est sensible aux données aberrantes

3. D'autres mesures de similarité existes aussi

Agglomération et partitions

Une partition désigne une **découpe de l'ensemble des données** en un certain **nombre de groupes (clusters)**

Nombre de groupes

- Dans les algorithmes de groupement présentés, nous avons supposé que le nombre de groupes est connu d'avance.

- Cependant, dans plusieurs situations, on ignore la structure des données et le nombre de groupes est inconnu d'avance.

- L'approche la plus commune est construire une fonction objective $\rho(K)$et de répéter le groupement pour K = 1, 2,3,…,etc.

- La valeur optimale de K serait alors celle qui donnera la meilleure valeur de la fonction objective.

Nombre de groupes pour K-Moyennes

$$\rho(K) = SCR + \lambda K$$

- La fonction $\rho(K)$ est pénalisée par un grand nombre de groupes K et va s'éloigner de 0.

- Le paramètre λ permet de contrôler le degré de pénalisation des grand nombre de groupes de sorte que.

  ☞ Si λ est grand $\Rightarrow$ on encourage des K petits.

  ☞ Si λ est petit $\Rightarrow$ on encourage des K grands.

- Le choix du K optimal sera établi par: $K^* = argmin_K\ \rho(K)$

exemple

K	SCR(K)	$\lambda = 10$	$\rho(K) = \mathrm{SCR} + \lambda K$
1	2000	10	$2000 + 10 \times 1 = 2010$
2	1200	10	$1200 + 10 \times 2 = 1220$
3	900	10	$900 + 10 \times 3 = 930$
4	800	10	$800 + 10 \times 4 = 840$
5	780	10	$780 + 10 \times 5 = 830$
6	770	10	$770 + 10 \times 6 = 830$
7	765	10	$765 + 10 \times 7 = 835$

$\rho(K)$ est minimum pour $K = 5$ ou $K = 6$.

on choisira $K^* = 5$ (ou 6).

J. Wu. (2012). Advances in K-means clustering: a data mining thinking. Springer Science & Business Media.

Fin de chapitre