Master IA2S (Intelligence Artificielle, Science des données et Systèmes Cyber-Physiques)

Apprentissage Automatique

Ferhat ATTAL
ferhat.attal@u-pec.fr
Université Paris Est Créteil (UPEC)

Novembre 2021

Cours 4 : Apprentissage non supervisé

Apprentissage non supervisé

- K-means
- Classification hiérarchique ascendante
- Modèle de mélange (Gaussien)
- Modèle de Markov caché

K-means

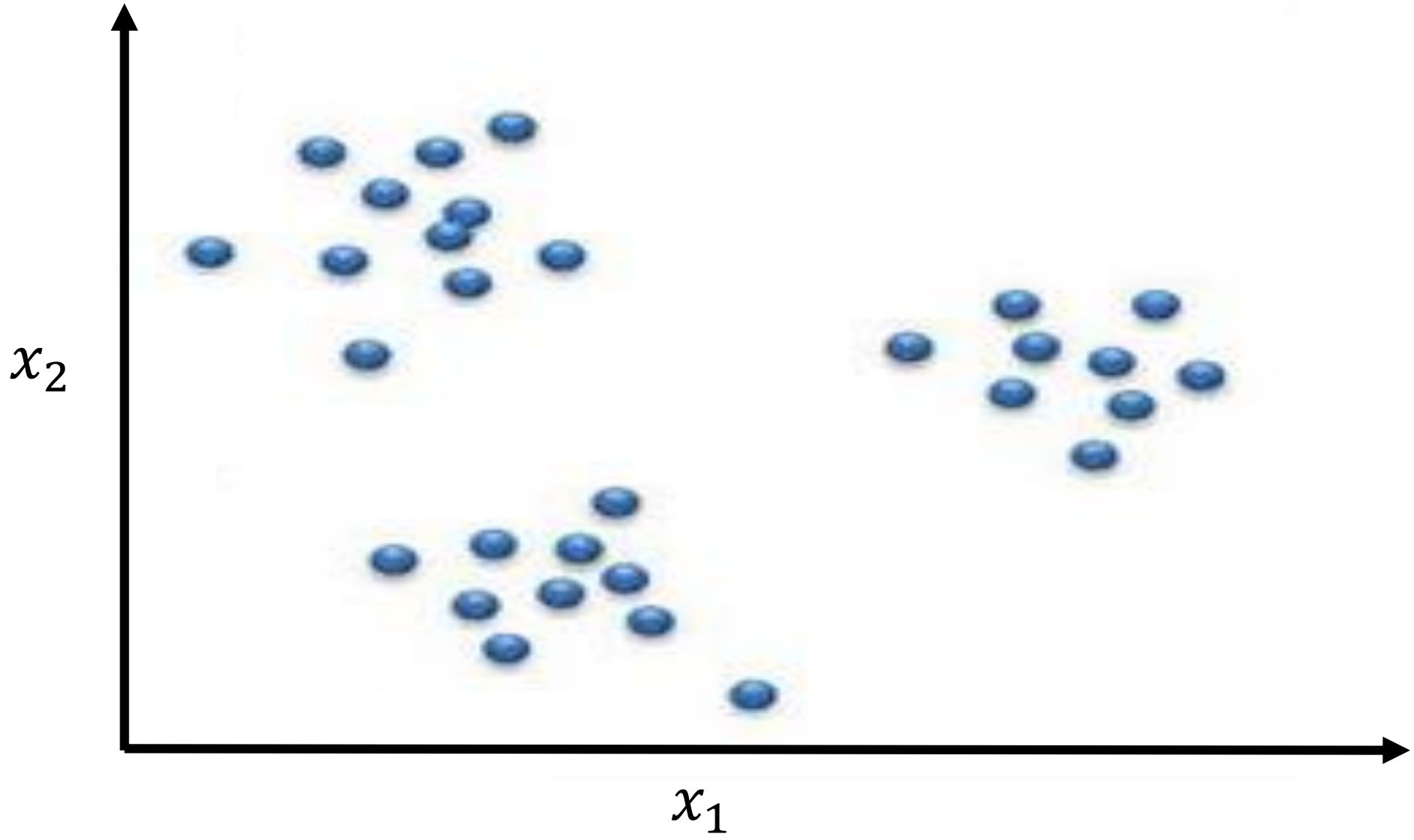
- Généralités
- Principe
- Algorithme
- Exemple

Généralités

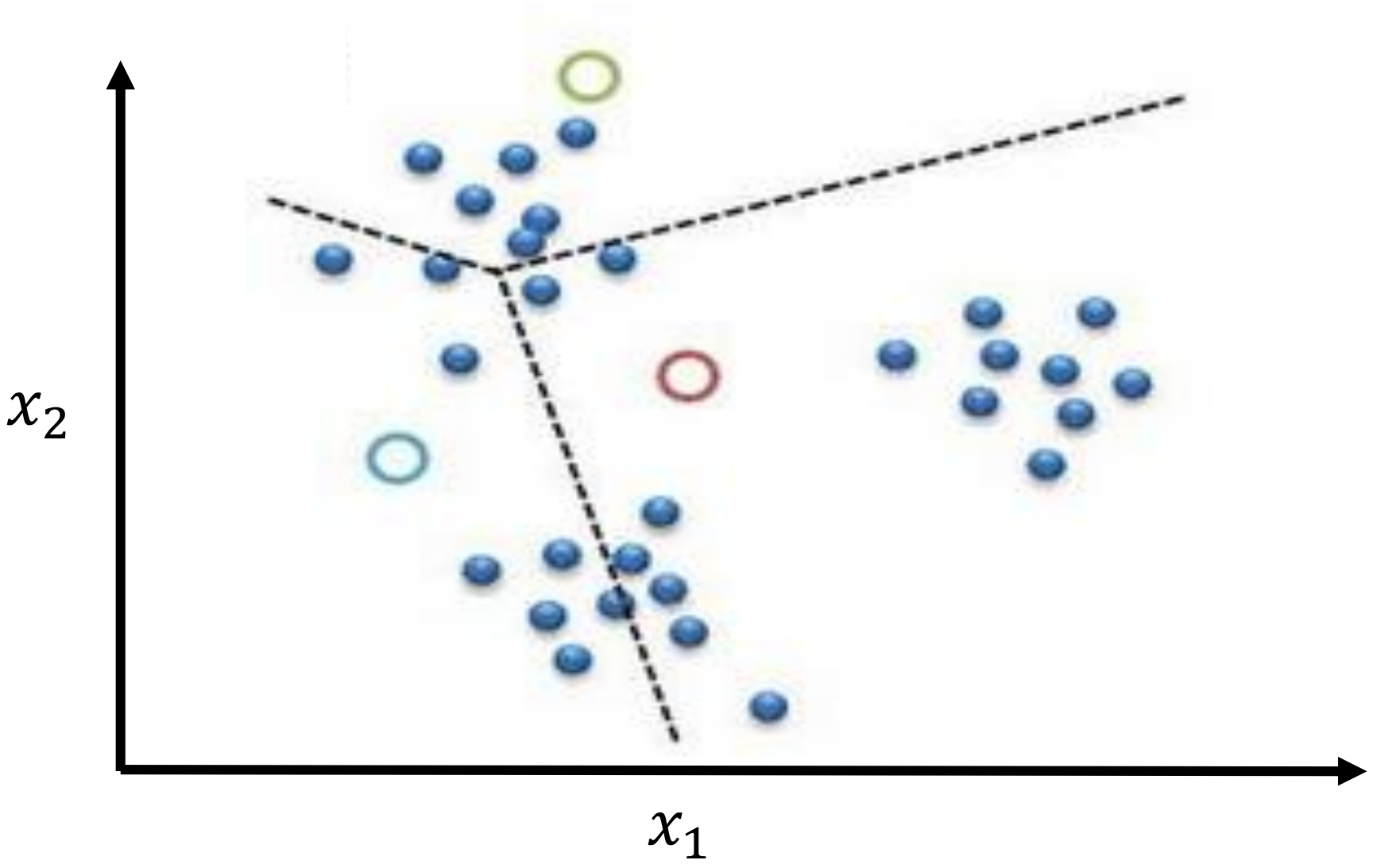
k-means (centres mobiles) est une méthode d'apprentissage non supervisée, utilisée principalement pour le partitionnement de données et la quantification vectorielle. Le k-means vise à partitionner ***n*** observations en ***k*** groupes (classes) dans lesquels chaque observation appartient au groupe dont la moyenne (centres de groupe) est la plus proche.

Principe

Choix du nombre de cluster (classe)

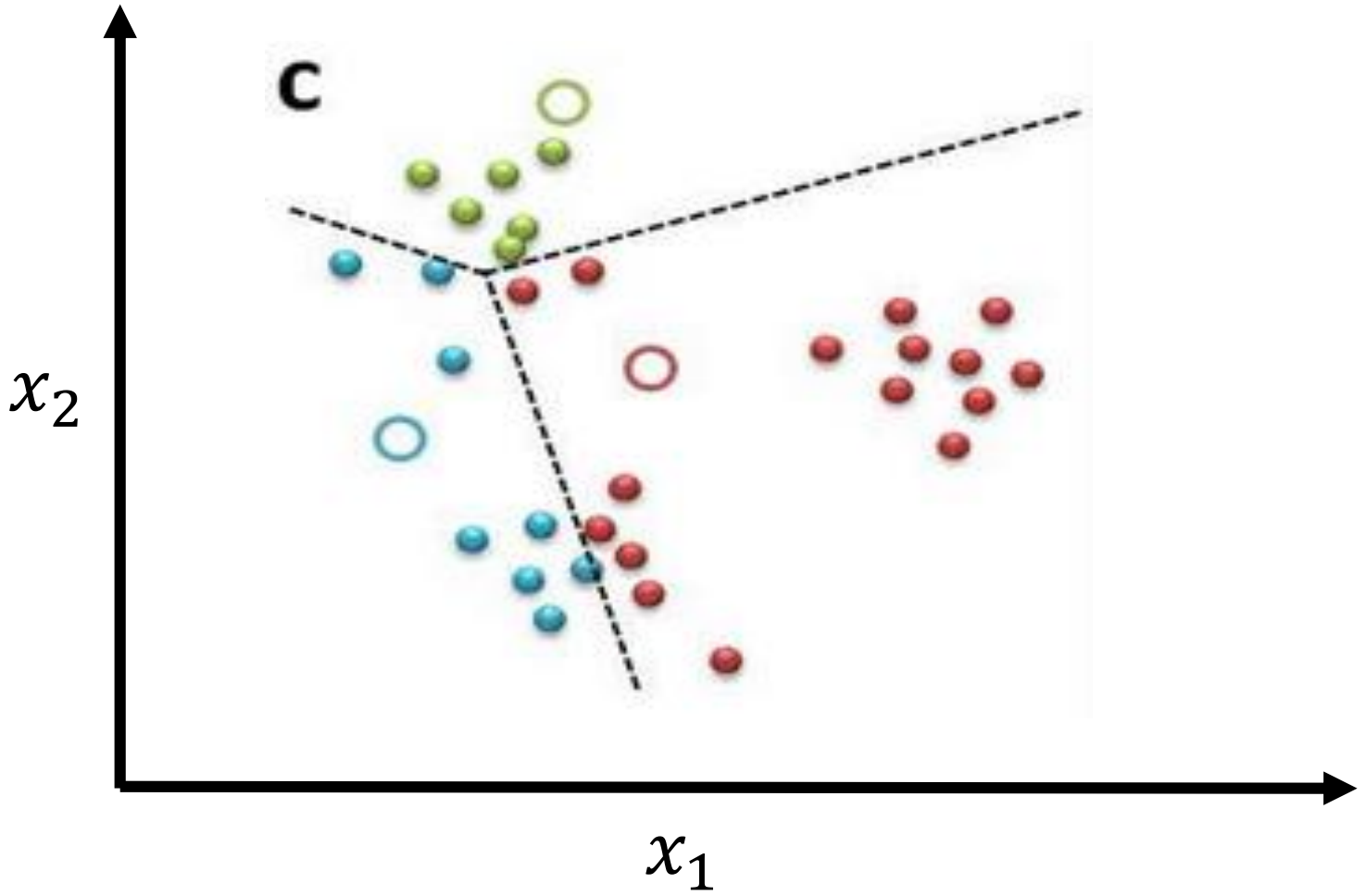


1) Initialisation aléatoire des centres de classes



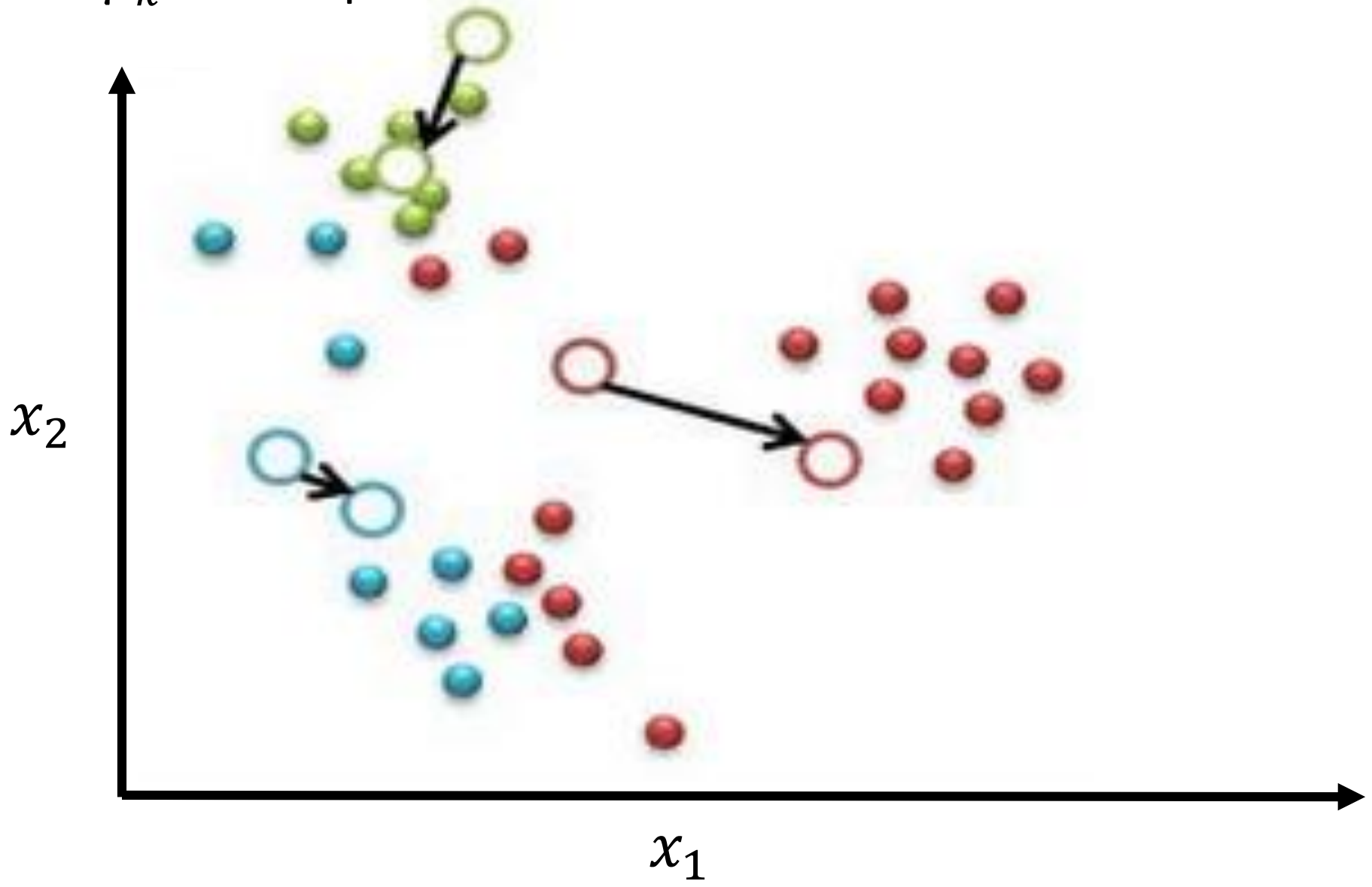
Principe

2) Affectation de chaque observation x au cluster C_k de centre μ_k tel que dist(x, μ_k) est minimale



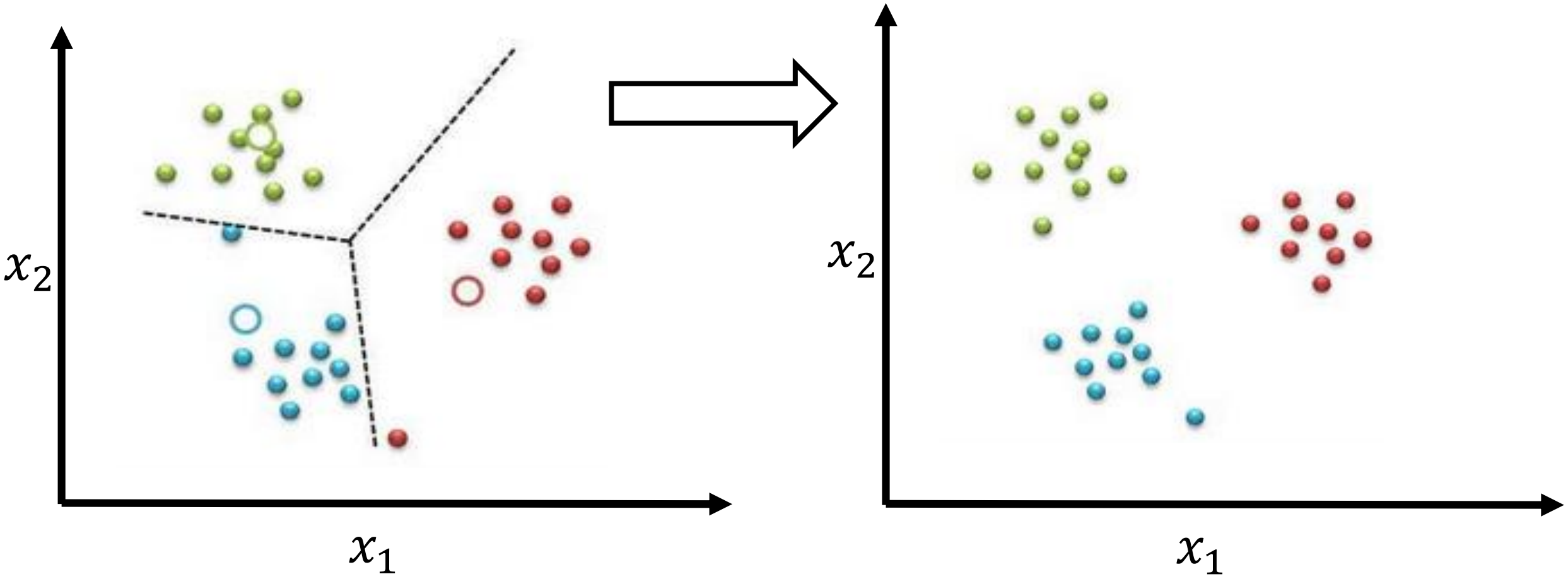
Principe

3) Recalcul des centres μ_k de chaque cluster



Principe

4) Répéter les étapes 2 et 3 jusqu'à convergence



Algorithme

Soit un ensemble d'apprentissage $S = \{x_i\}_{i \leq n}$ où la variable d'entrée x_i appartient à $\mathbb{R}^d$.

L'objectif est d'estimer $Y = \{y_i\}_{i \leq n}$ avec $y_i \in \{c_1, c_2, \ldots, c_k, \ldots, c_K\}$
La fonction objective (mesure de distorsion) à minimiser est :

$$J = \sum_{k=1}^{K} \sum_{i|y_i=c_k} || x_i - \mu_k ||^2$$

$$= \sum_{k=1}^{K} \sum_{i}^{n} y_i^k || x_i - \mu_k ||^2$$

$y_i^k = 1$ *si* $y_i = c_k$ $y_i^k = 0$ sinon

Algorithme

y_i^k et μ_k permettant de minimiser J peuvent être estimés itérativement :

$$y_i^k = \begin{cases} 1 \; si \quad c_k = \underset{j=\{1,2,..,K\}}{\text{argmax}} \; || x_i - \mu_j ||^2 \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad sinon \end{cases}$$

$$\mu_k = \frac{\sum_{i=1}^{n} y_i^k x_i}{\sum_{i=1}^{n} y_i^k}$$

Algorithme

Entrées : $S = \{x_1, \ldots, x_j, \ldots, x_n\}$ un ensemble de données.
$Y = \{c_1, c_2, \ldots, c_k, \ldots, c_K\}$ K : représente le nombre total de cluster.
$J^{(b)}$: la fonction objective à l'itération b.
ε : un seuil (valeur très petite .

b←0 ;
$J^{(b)} = \infty$;
/* *Initialisation*
1 : Choisir les centroides initiaux $\boldsymbol{\mu}$

$$\boldsymbol{\mu}^{(b)} = \{\mu_1^{(b)}, \ldots, \mu_k^{(b)}, \ldots, \mu_K^{(b)}\}$$

2 : Répéter
/**Affecter chaque observation à un cluster en se basant sur la distance Euclidien*
3 : For i=1 to n Do

$$y_{ik}^{(b)} = \begin{cases} 1 & si\ c_k = \underset{j=\{1,2,\ldots,K\}}{argmin} \| x_i - \mu_j^{(b)} \|^2 \\ 0 & sinon\ . \end{cases}$$

4 : End For

/* *mise à jour des* centroides
9 : mettre à jours les centroides $\mu^{(b+1)} = \{\mu_1^{(b+1)}, \ldots, \mu_k^{(b+1)}, \ldots, \mu_K^{(b+1)}\}$

10 : For k=1 to K Do

$$\mu_k^{(b+1)} = \frac{\sum_{i=1}^{n} y_{ik}^{(b)} x_i}{\sum_{j=1}^{n} z_{jl}^{(b)}}$$

11 : End For

$$J^{(b+1)} = \sum_{k=1}^{K} \sum_{i=1}^{n} y_{jk}^{(b)} \| x_j - \mu_k^{(b+1)} \|^2$$

b←b+1;
/* *Test de Convergence*
12 : Until $\frac{|J^{(b)} - J^{(b+1)}|}{J^{(b+1)}} < \varepsilon$
Sorties : les centroides $\mu = \{\mu_1, \ldots, \mu_k\}$, l'ensemble des K clusters $CL_1, \ldots, CL_K$ avec $CL_k = \{x_i, y_{ik} = 1\}$, $\forall\ i \in [1, n], \forall\ k \in [1, K]$.

Exemple

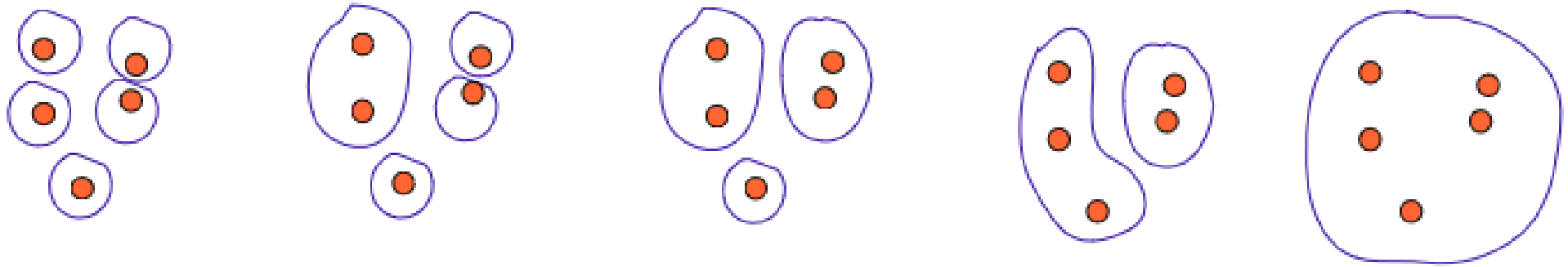
petal.length	petal.width	variety
1.4	0.2	Setosa
1.4	0.2	Setosa
1.3	0.2	Setosa
1.5	0.2	Setosa
1.4	0.2	Setosa
5.8	1.6	Virginica
6.1	1.9	Virginica
6.4	2	Virginica
5.6	2.2	Virginica
5.1	1.5	Virginica

Classification hiérarchique ascendante

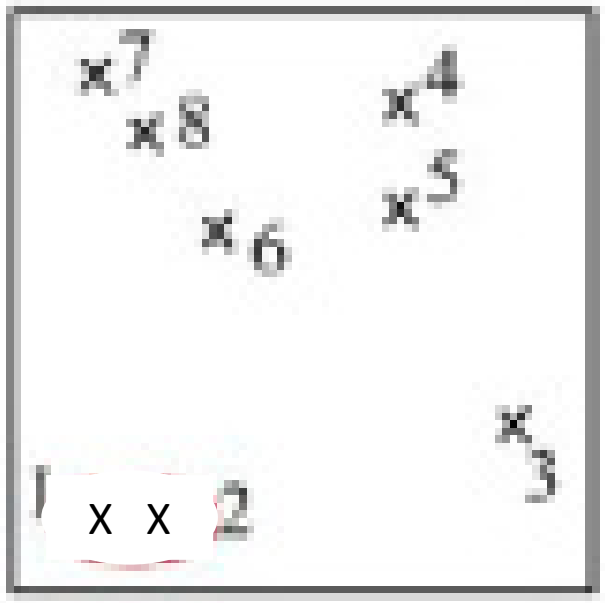
- Généralités
- Principe
- Algorithme
- Exemple

Généralités

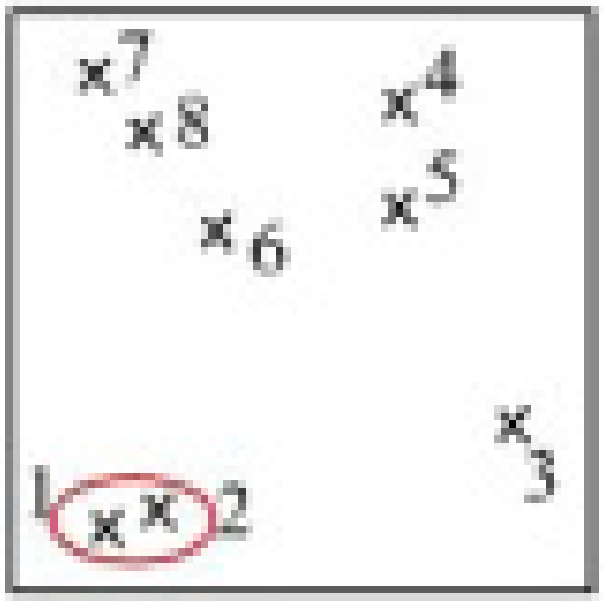
La classification hiérarchique ascendante est une méthode de classification non supervisé. Le principe de cette méthode, repose sur l'agrégation des individus qui ont des similarités au sens d'une mesure qui peut être une de distance, un indice de similarité, etc. La CAH permet de rassembler les individus de manière itérative afin de produire un dendrogramme ou arbre de classification.



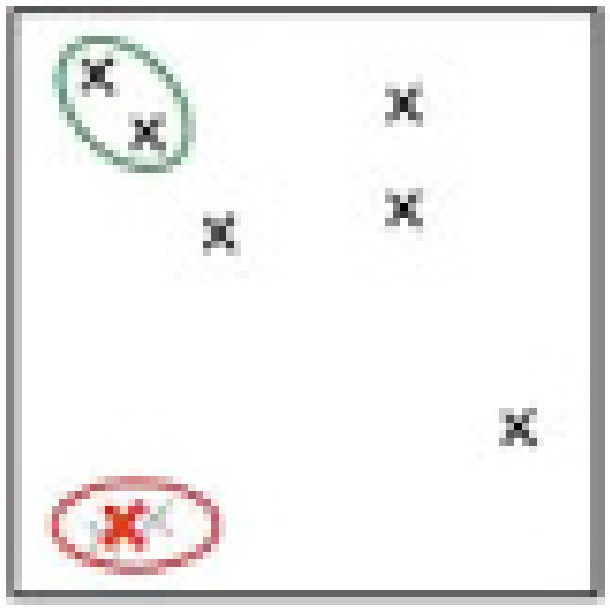
Généralités



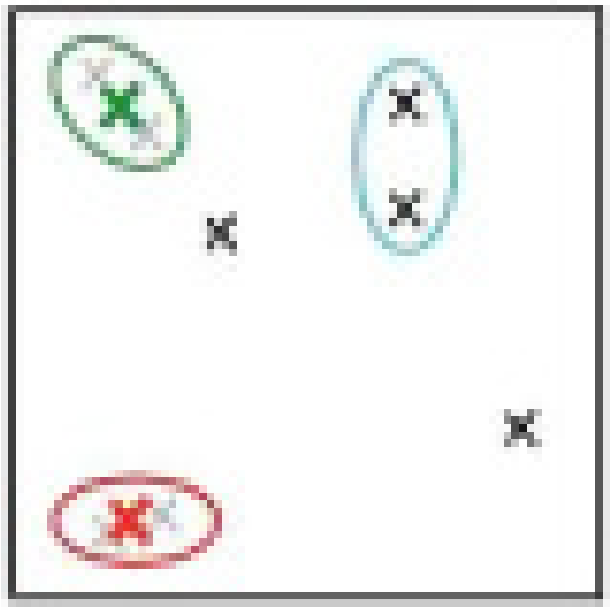
Généralités

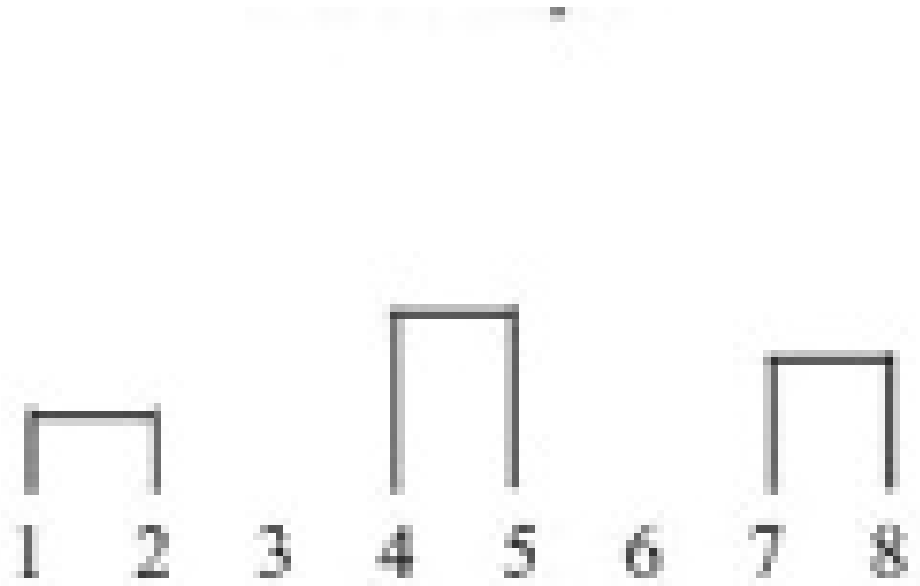


Généralités

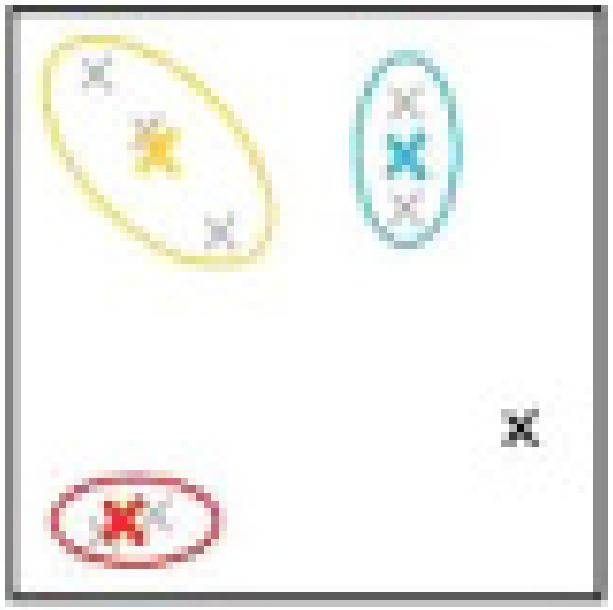


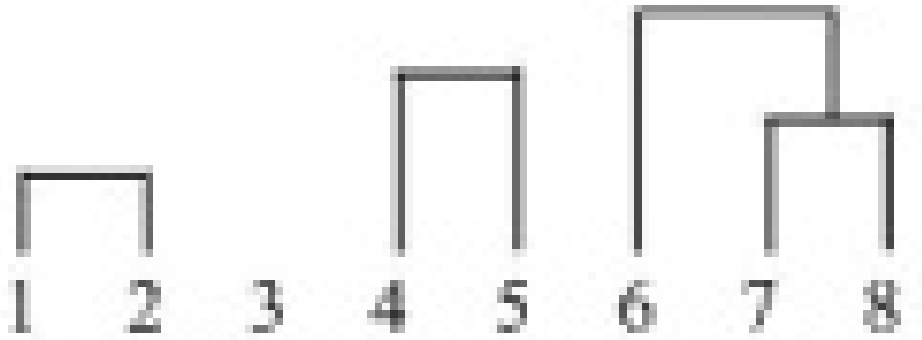
Généralités



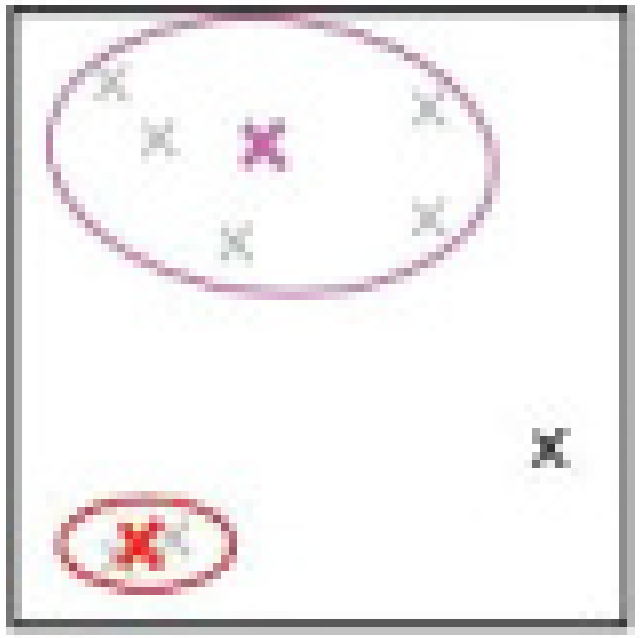


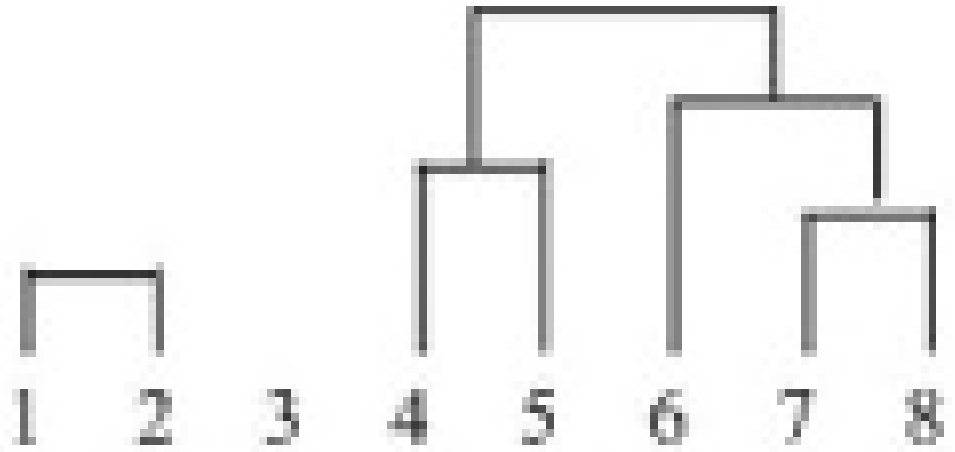
Généralités



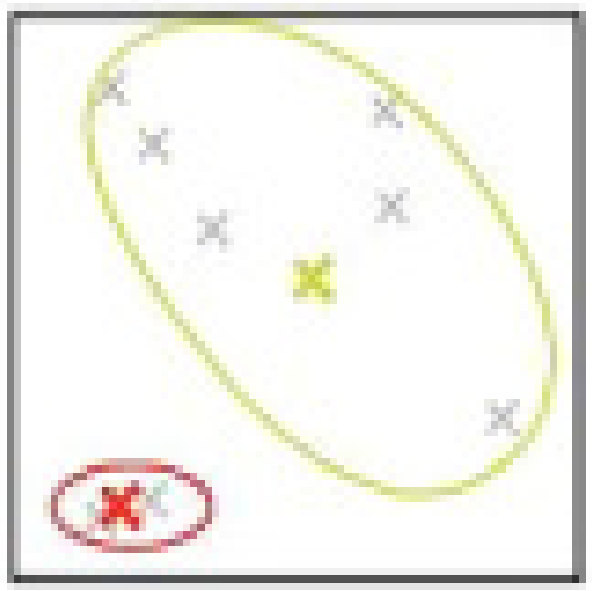


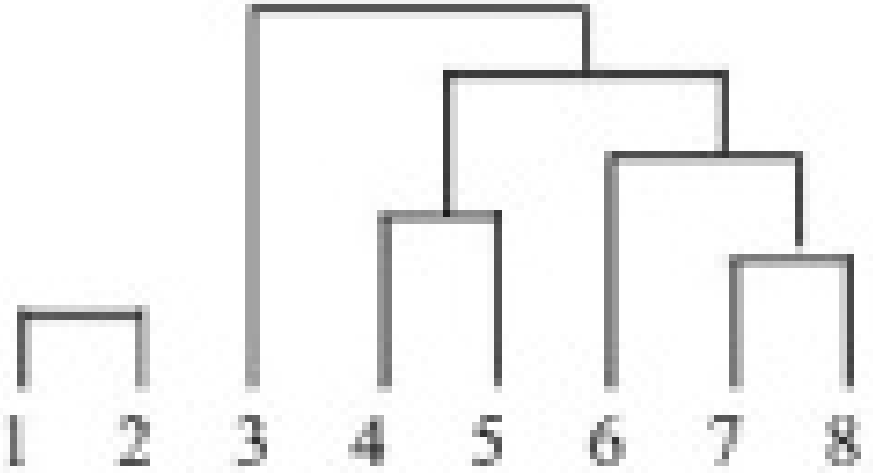
Généralités



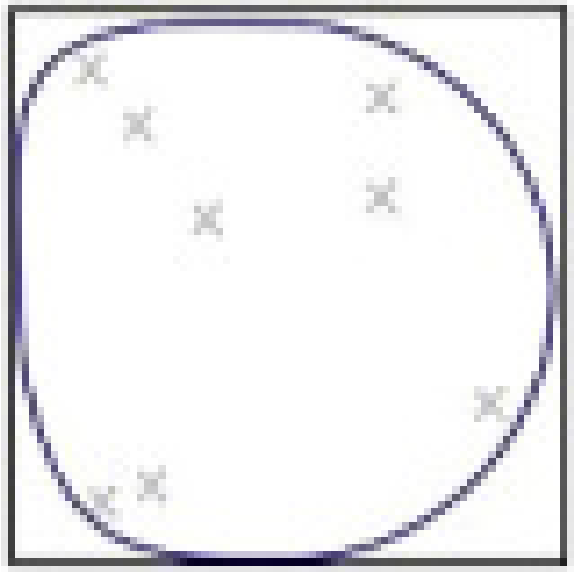


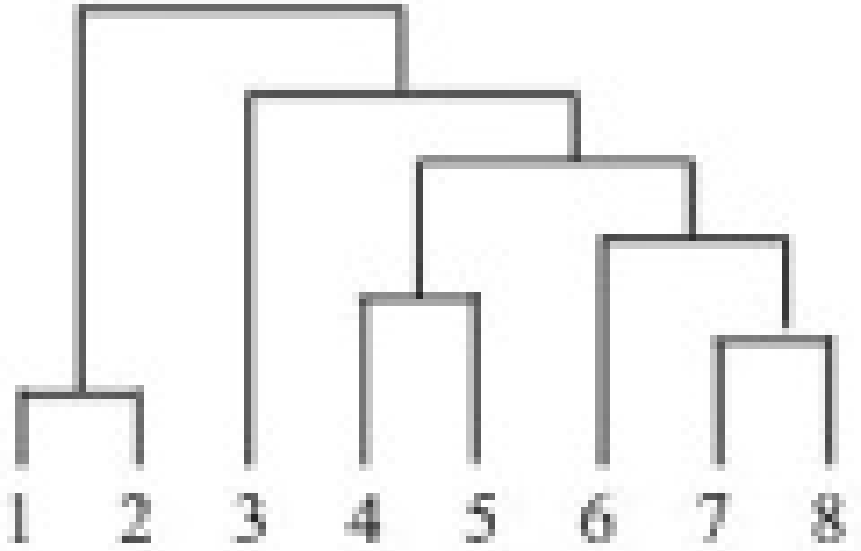
Généralités





Généralités





Ressemblance entre individus :

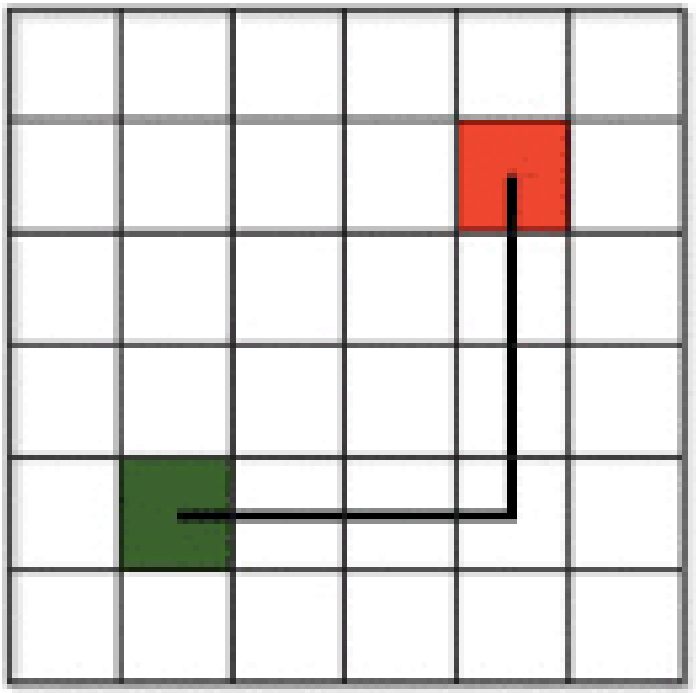
La distance Euclidienne

$$d(x_n, x_i) = \sqrt{\sum_{j=1}^{d} (x_{new}^{j} - x^{j})^2}$$

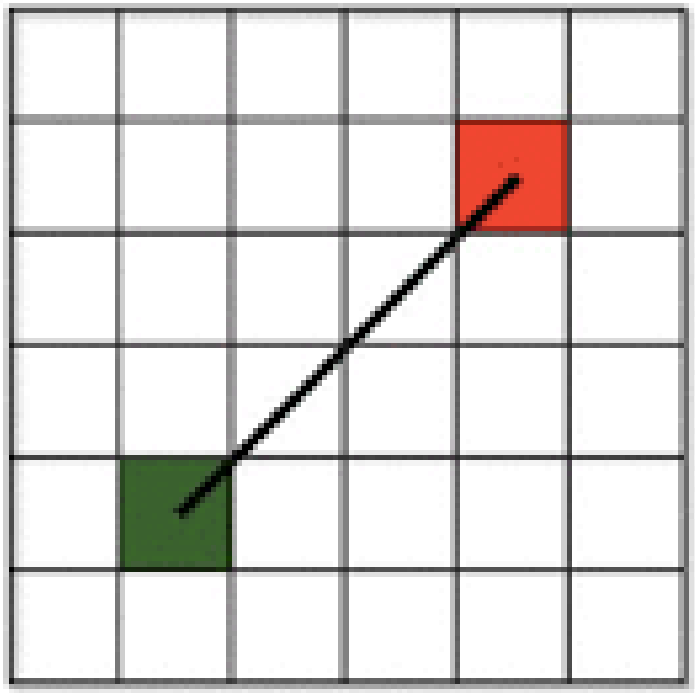
La distance de Manhattan

$$d(x_n, x_i) = \sum_{j=1}^{d} |(x_{new}^{j} - x^{j})|$$

….. etc.



Manhattan Distance

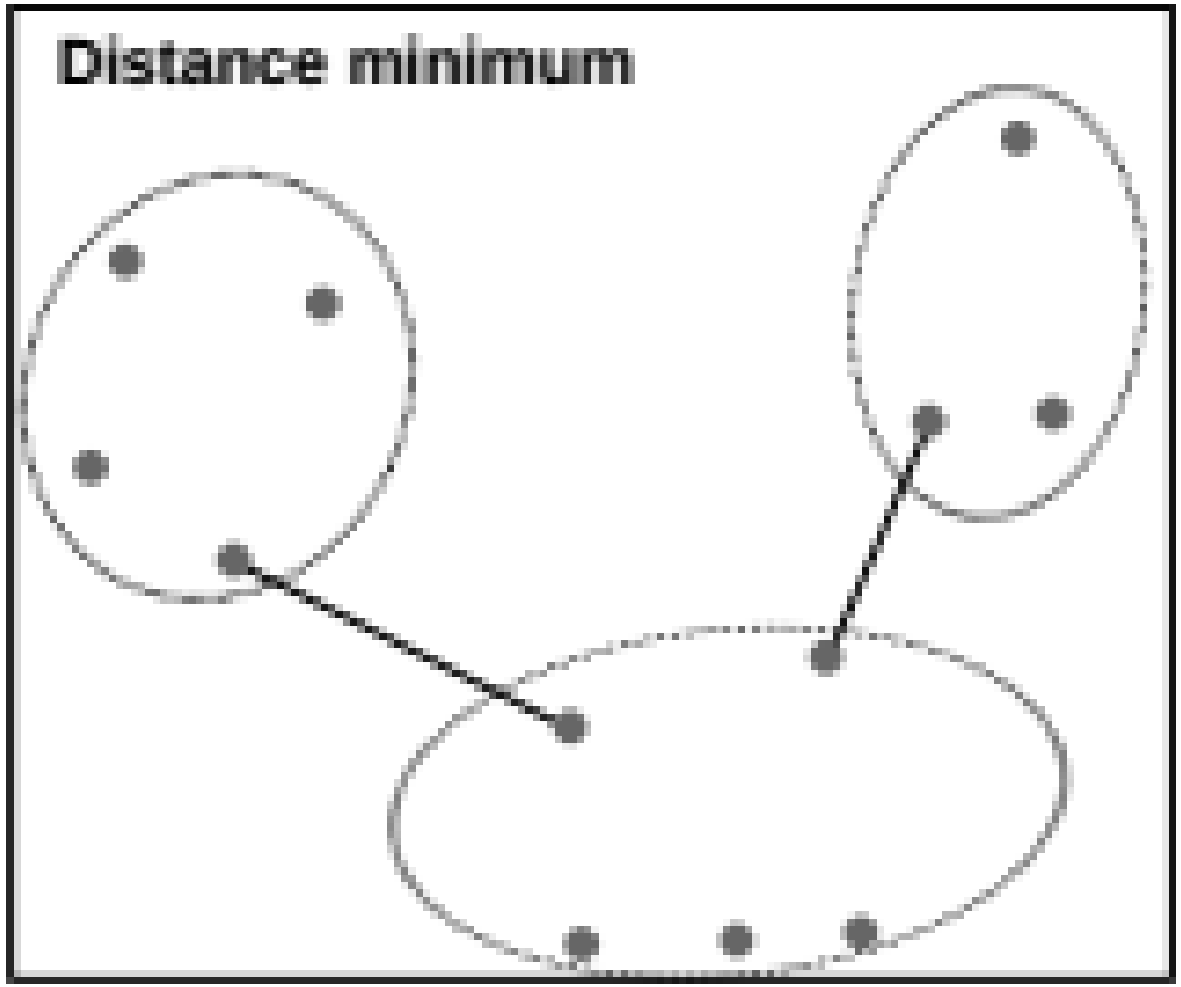


Euclidean Distance

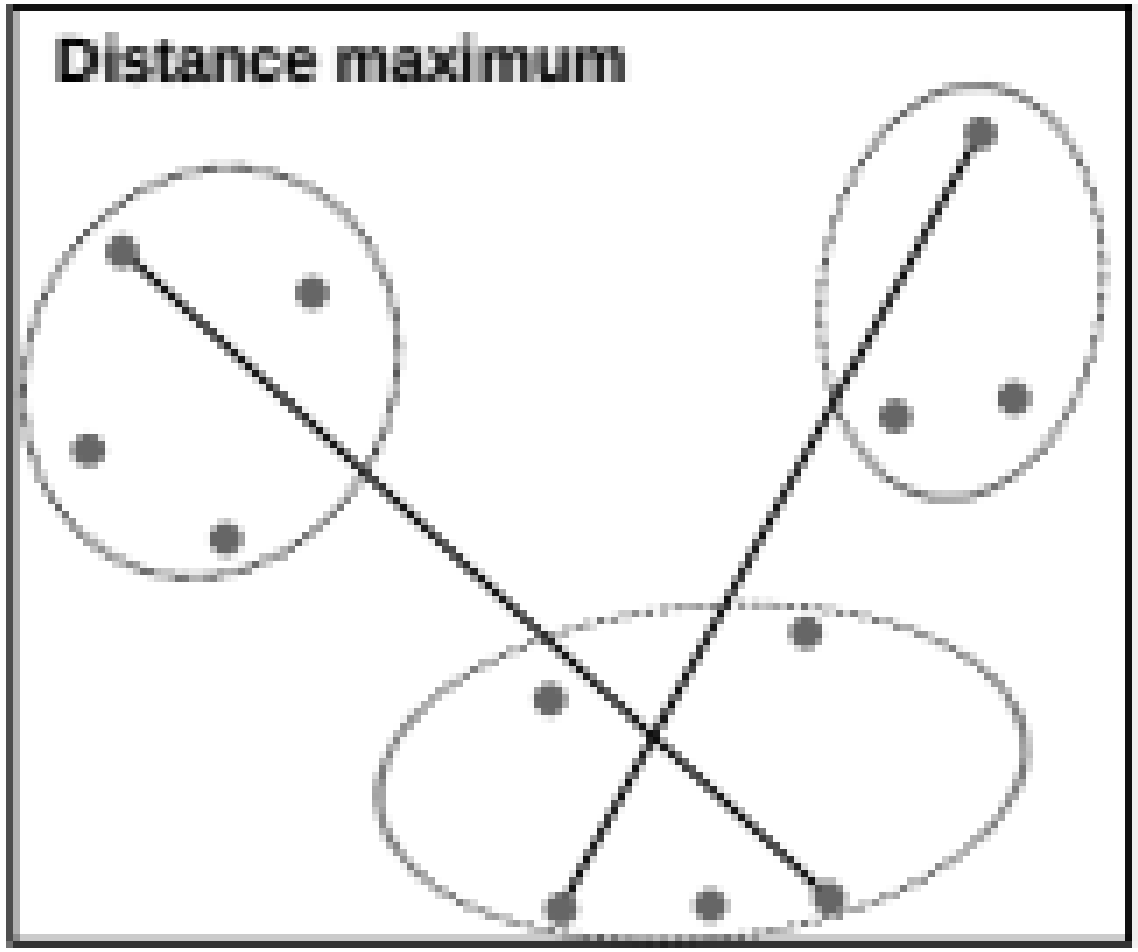
Ressemblance entre groupes d’individus :

- Saut minimum ou lien simple (distance minimum)
- Lien complet (distance maximum)
- Distance entre centres de gravité
- Distance moyenne
- Critère de Ward

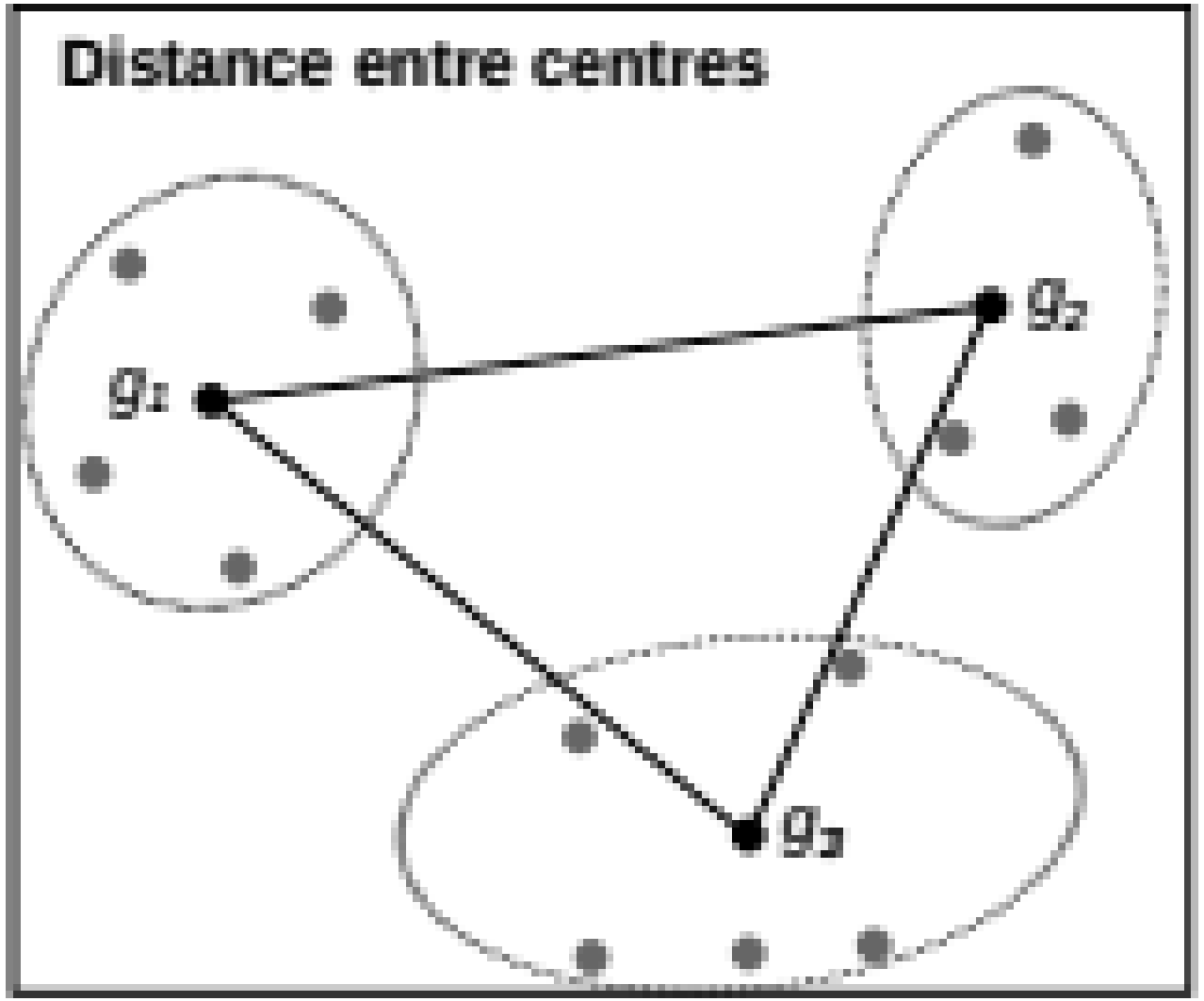
Ressemblance entre groupes d'individus :



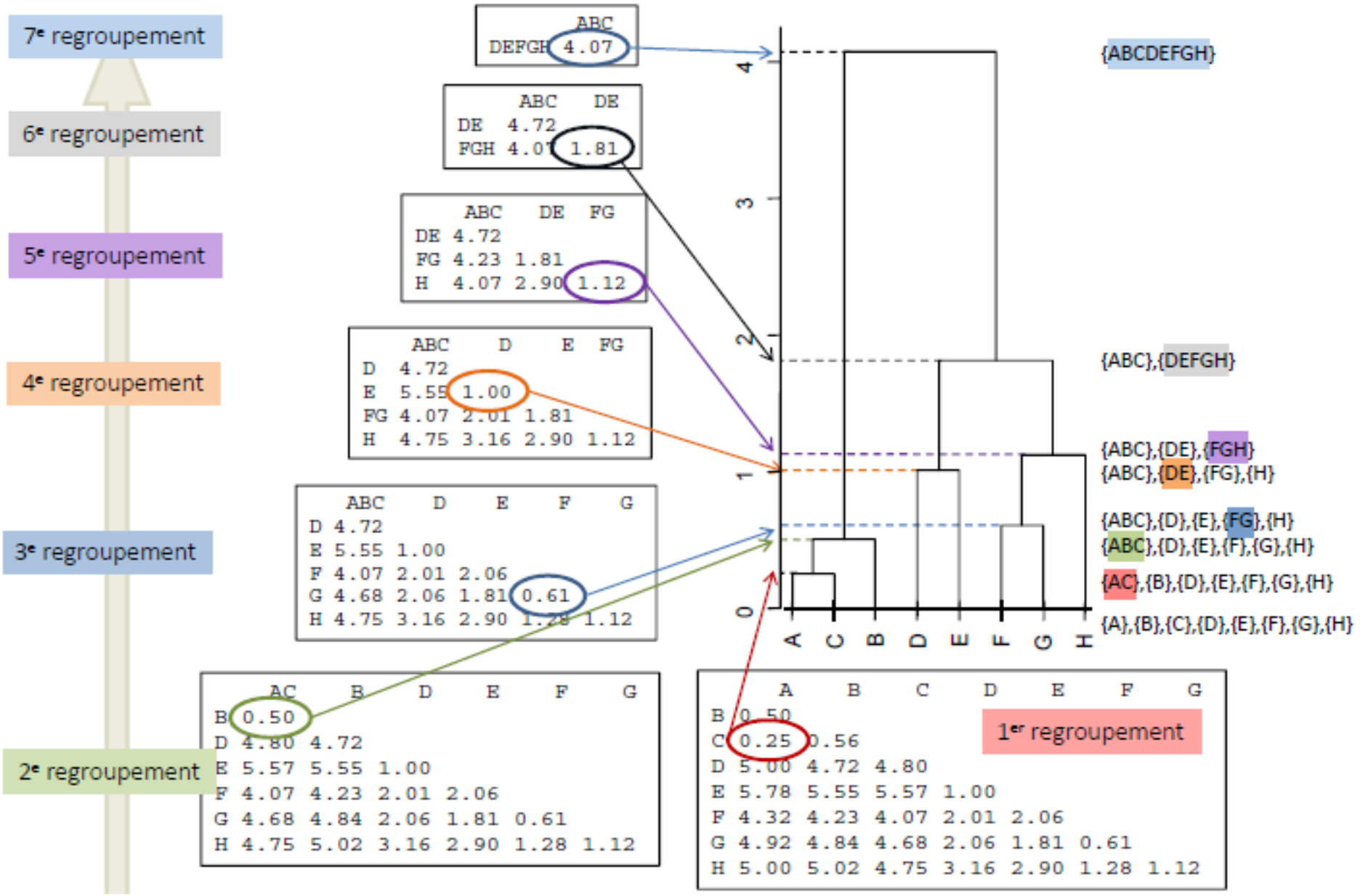
Ressemblance entre groupes d'individus :



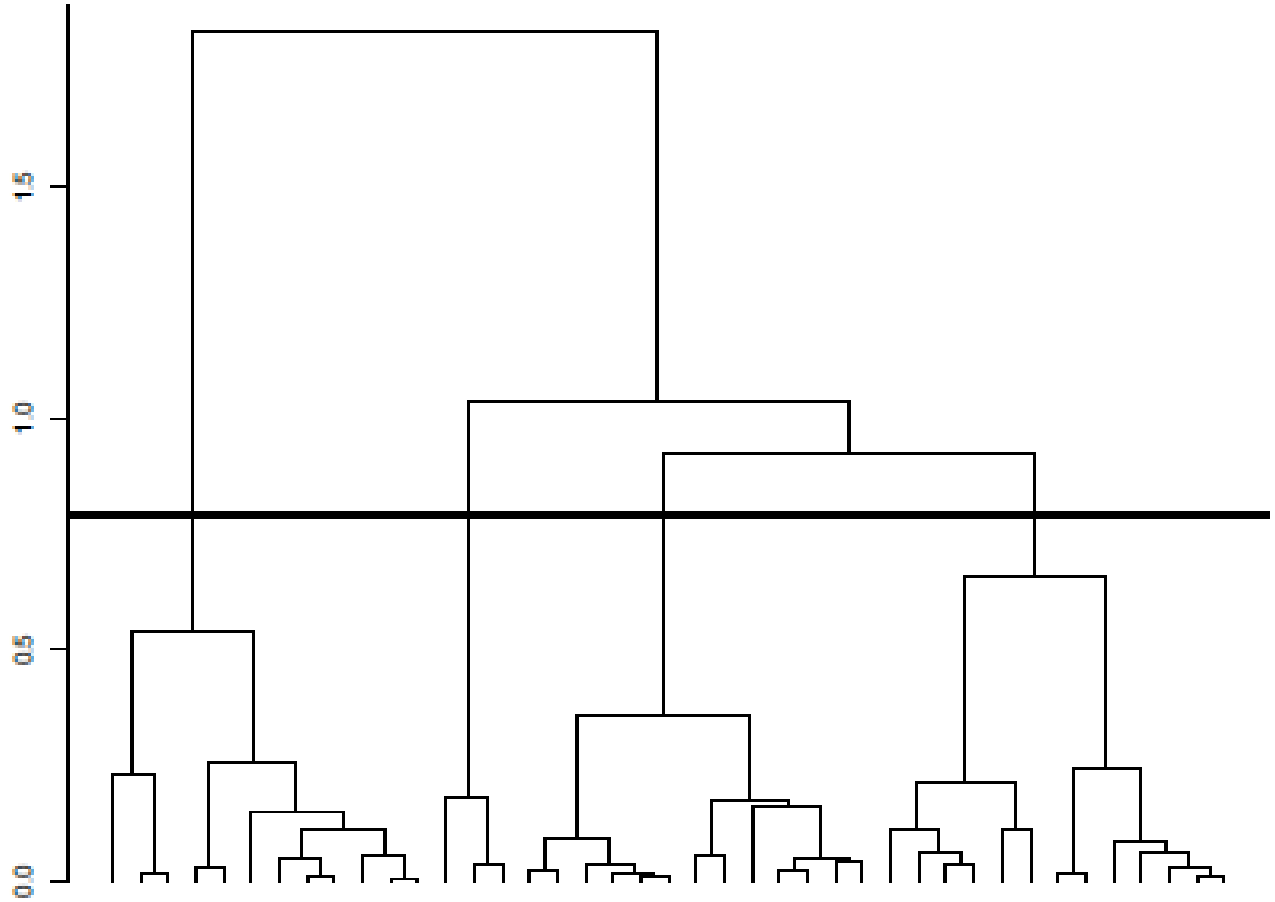
Ressemblance entre groupes d'individus :



Exemple (saut minimum)



Détermination du nombre de cluster



Choisir un seuil de partitionnement

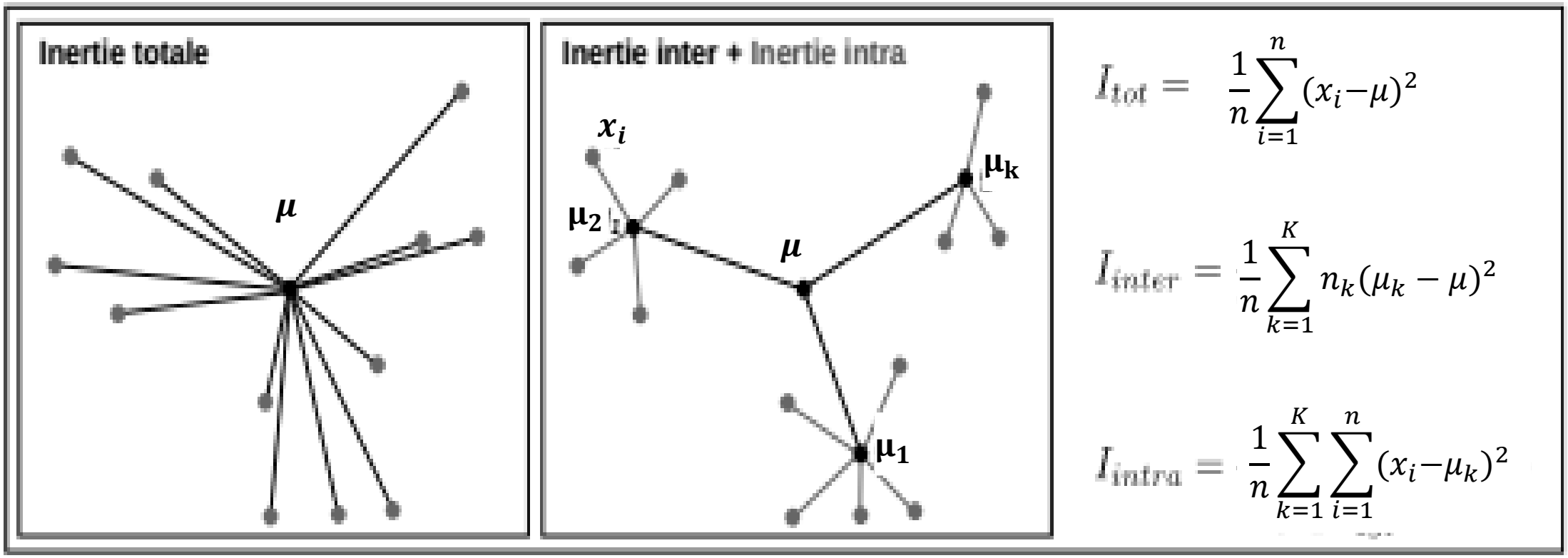
Qualité d’une partition

Choisir un seuil de partitionnement qui permet de :

- Minimiser la variabilité intra-classe
  - La distance entre les individus de la même classe est petite

- Maximiser la variabilité inter-classes
  - La distance entre les individus de classes différentes est grande

Qualité d'une partition



Relation de Huygens:

Inertie Totale = Inertie inter-classes + inertie intra-classe

L'inertie est un indicateur de dispersion.

Qualité d'une partition

$$\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^2 = \frac{1}{n}\sum_{k=1}^{K} n_k(\mu_k-\mu)^2 + \frac{1}{n}\sum_{k=1}^{K}\sum_{i=1}^{n}(x_i-\mu_k)^2$$

Dispersion des centres de groupes autour du centre global (à maximiser)

Dispersion à l'intérieur des groupes (à minimiser)

Qualité d'une partition

La qualité d'une partition est mesurée par le rapport (R) de Inertie inter Inertie totale

$$0 \leq \frac{\text{Inertie inter}}{\text{Inertie totale}} \leq 1$$

- $\frac{\text{Inertie inter}}{\text{Inertie totale}} = 1 \Rightarrow \text{patitionnement parfait}$

- $\frac{\text{Inertie inter}}{\text{Inertie totale}} = 0 \Rightarrow \text{patitionnement impossible}$

Lien de Ward

La perte d'informations lors de la fusion de deux groupes c_k et c_l d'une partition Y est quantifiée par:

$$\delta(c_k, c_l) = \mathrm{I}\,(c_k \cup c_l) - \mathrm{I}\,(c_k) - \mathrm{I}\,(c_l)$$

$$\mathrm{I}\,(c_k) = \sum_{i=1|x_i \in c_k} (x_i - \mu_k)^2$$

La quantité $\delta(c_k, c_l)$ est connue sous le nom de lien de Ward est égale à la variation de l'inertie intra-classes après la fusion de deux groupes. Elle correspond également au carré de la distance entre les centres de gravité :

$$\delta(c_k, c_l) = \frac{n_k n_l}{n_k + n_l} (\mu_k - \mu_l)^2$$

Lien de Ward

En partant de la partition triviale P = $\{\{x_1\}, \{x_2\}, \ldots, \{x_i\}, \ldots, \{x_n\}\}$ avec n singletons, l'algorithme HAC crée une séquence de partitions en fusionnant successivement les deux clusters qui conduisent à la plus petite perte d’inertie, jusqu'à ce que tous les individus aient été fusionnés en un seul cluster

Exemple

petal.length	petal.width	variety
1.4	0.2	Setosa
1.4	0.2	Setosa
1.3	0.2	Setosa
1.5	0.2	Setosa
1.4	0.2	Setosa
5.8	1.6	Virginica
6.1	1.9	Virginica
6.4	2	Virginica
5.6	2.2	Virginica
5.1	1.5	Virginica

Modèle de mélange fini

- Généralités
- Estimation des paramètres du modèle de mélange
- L’algorithme Espérance Maximisation (EM)
- Le modèle de mélange gaussien
- Estimation des paramètres du modèle de mélange gaussien
- Algorithme
- Exemple

Généralités

Le modèle de mélanges finis a été largement développé, appliqué et utilisé dans la classification, le clustering, l'estimation de densités et la reconnaissance de formes. En classification automatique l'utilisation du modèle de mélanges finis revient à supposer que les individus à classifier sont issus d'un modèle de mélanges dont chaque composante représente une classe.

Généralités

Soit un échantillon $\boldsymbol{x} = \{x_1, x_2, \ldots, x_i, \ldots, x_n\}$ de n réalisations indépendantes / $x_i \in \mathbb{R}^d$. Soit $y = \{y_1, y_2, \ldots, y_i, \ldots, y_n\}$ un ensemble de données latentes non observées où y_i est une variable aléatoire discrète (binomiale ou multinomiale) qui prend ses valeurs dans l'ensemble fini $Y = \{1, \ldots, K\}$.

Le modèle de mélanges finis modélise la densité de $\boldsymbol{x}$ en une combinaison linéaire pondérée des densités de K composantes.

Dans un cadre général, la densité du mélange de $\boldsymbol{x}$ est donnée par :

$$f(x) = \sum_{k=1}^{K} p(x, y = k) = \sum_{k=1}^{K} p(y = k)\, p(x|y = k)$$

$$= \sum_{k=1}^{K} \pi_k f_k(x)$$

Généralités

- K : représente le nombre de composantes
- $\pi_k = p(y = k)$ représente la probabilité qu'un point de données sélectionné de façon aléatoire est généré par la composante k
- $\sum_{k=1}^{K} \pi_k = 1$
- $f_k(x)$ représente la densité de probabilité des composantes.

Généralités

On suppose souvent que les composantes $f_k(x)$ du mélange sont paramétrées par α_k . Ainsi, la densité du mélange de x peut s'écrire sous la forme

$$f(x,\theta) = \sum_{k=1}^{K} \pi_k f_k(x,\alpha_k)$$

$\theta = \{\pi_1, \pi_2, \ldots, \pi_k, \ldots, \pi_K, \alpha_1, \alpha_2, \ldots, \alpha_k, \ldots, \alpha_K\}$est le vecteur des paramètres du modèle du mélange qui appartient à l'espace des paramètres Θ

Généralités

- Modèle de mélange de lois de Weibull
- Modèle de mélange de lois multinomiales
- **Modèle de mélange gaussien**

Estimation des paramètres du modèle de mélange

L'estimation des paramètres ($\boldsymbol{\theta}$) du modèle de mélange, se fait généralement par la maximisation de la vraisemblance (ou le log-vraisemblance pour des raisons de facilité de calcul) qui peut être expriméé par :

$$\mathcal{L}(\theta;\mathrm{x}) = log \prod_{i=1}^{n} p(x_i, \theta)$$

$$= \sum_{i=1}^{n} \log \sum_{k=1}^{K} \pi_k f_k(x, \alpha_k)$$

- Le log-vraisemblance est une fonction non linéaire (log de la somme)
- En général, la maximisation du log-vraisemblance ne possède pas de solution analytique

Estimation des paramètres du modèle de mélange

La maximiser du log-vraisemblance peut se faire localement en utilisant des procédures itératives :

- La méthode décente du gradient
- La méthode de Newton Raphson
- **L’algorithme Espérance Maximisation (EM)**

L’algorithme Espérance Maximisation (EM)

- L’algorithme EM est largement appliqué au calcul itératif pour l’estimations du maximum de vraisemblance dans le cadre des modèles de données latentes.

- L'algorithme EM simplifie considérablement le problème d’estimation des paramètres des modèles de mélanges finis par le maximum de vraisemblance.

- L’algorithme EM alterne itérativement entre les deux étapes:

Etape E : Cette étape consiste à calculer l’espérance du log-vraisemblance des données complétées.

Étape M (Maximisation) : Cette étape consiste à maximiser l’espérance du log-vraisemblance des données complétées.

L’algorithme Espérance Maximisation (EM)

$y = \{y_1, y_2, \ldots, y_i, \ldots, y_n\}$ un ensemble de données latentes non observées où y_i est une variable aléatoire discrète (binomiale ou multinomiale) qui prend ses valeurs dans l’ensemble fini $Y = \{1, \ldots, K\}$.
Le log-vraisemblance des données complétées :

$$\mathcal{L}(\theta; x, y) = log \prod_{i=1}^{n} p(x_i, y_i|\theta)$$

$$= \sum_{i=1}^{n} log \prod_{k=1}^{K} (\pi_k f_k(x_i, \alpha_k))^{y_{ik}}$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{K} y_{ik}\ log(\pi_k f_k(x_i, \alpha_k)$$

y_{ik} est une variable binaire qui vaut 1 si $y_i = k, 0$ sinon

L'algorithme Espérance Maximisation (EM)

Étant donné $\theta^{(0)} = \{\pi_1^{(0)}, \pi_2^{(0)}, \ldots, \pi_k^{(0)}, \ldots, \pi_K^{(0)}, \alpha_1^{(0)}, \alpha_2^{(0)}, \ldots, \alpha_k^{(0)}, \ldots, \alpha_K^{(0)}\}$

L'algorithme EM alterne itérativement entre les deux étapes :

Etape E : Cette étape consiste à calculer l'espérance du log-vraisemblance des données complétées, en tenant compte des observations $\boldsymbol{x}$ et de la valeur courant de $\theta^{(c)}$.

$$Q(\theta, \theta^{(c)}) = E(\mathcal{L}(\theta; x, y) | x, \theta^{(c)})$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{K} E\left(y_{ik}\, log(\pi_k f_k(x_i, \alpha_k) | x, \theta^{(c)})\right)$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{K} E\left(y_{ik} | x, \theta^{(c)}\right) log(\pi_k f_k(x_i, \alpha_k))$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{K} p\left(y_i = k | x, \theta^{(c)}\right) log(\pi_k f_k(x_i, \alpha_k))$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{K} \tau_{ik} log(\pi_k f_k(x_i, \alpha_k))$$

L’algorithme Espérance Maximisation (EM)

$$\tau_{ik} = \mathrm{p}\left(y_i = k|\mathrm{x},\theta^{(c)}\right) = \frac{\pi_k^{(c)} f_k\left(x_i,\alpha_k^{(c)}\right)}{\sum_{g=1}^{K} \pi_g^{(c)} f_g\left(x_i,\alpha_g^{(c)}\right)}$$

Étape M (Maximisation) : Cette étape consiste à maximiser l’espérance du log-vraisemblance des données complétées, en tenant compte des observations $\boldsymbol{x}$ et de la valeur courant de $\theta^{(c)}$

$$\theta^{(c+1)} = \underset{\theta \in \Theta}{\operatorname{argmax}}\, Q(\theta, \theta^{(c)})$$

Avec

$$Q(\theta,\theta^{(c)}) = \underbrace{\sum_{i=1}^{n}\sum_{k=1}^{K} \tau_{ik} log(\pi_k)}_{Q_\pi(\theta,\pi_k^{(c)})} + \underbrace{\sum_{i=1}^{n}\sum_{k=1}^{K} \tau_{ik} \log(f_k(x_i,\alpha_k))}_{Q_\alpha(\theta,\alpha_k^{(c)})}$$

L’algorithme Espérance Maximisation (EM)

La maximisation de la fonction $Q\left(\theta, \theta^{(c)}\right)$ peut se faire séparément par la maximisation de Q_π par rapport à π et Q_α par rapport à α

- La fonction Q_π est maximisée par rapport à π sous contraintes $\sum_{k=1}^{K} \pi_k = 1$

On obtient:

$$\pi_k = \frac{\sum_{k=1}^{K} \tau_{ik}}{n}$$

- La maximisation de fonction Q_α par rapport α dépond de la forme de la densité $f_k(x_i, \alpha_k)$

Le modèle de mélange gaussien

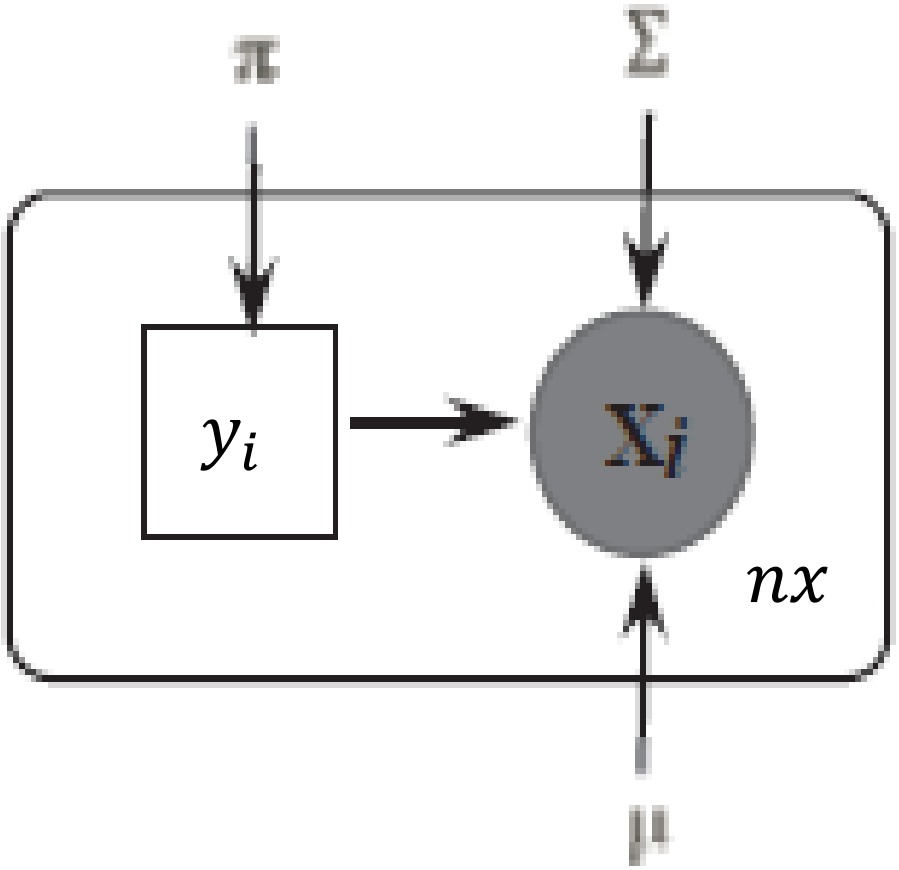
Dans le cas du modèle de mélange gaussien, la forme de la loi conditionnelle utilisée quand les observations sont à valeur dans $\mathbb{R}^d$ est la loi normale multivariée donnée par :

$$\mathcal{N}(x_i, \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{\frac{d}{2}}\sqrt{det(\Sigma_k)}} e^{-\frac{1}{2}(x_i-\mu_k)^T\Sigma_k^{-1}(x_i-\mu_k)}$$

$$f(x_i, \theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x_i, \mu_k, \Sigma_k)$$

$\theta = \{\pi_1 \ ,\ldots,\pi_k \ ,\ldots,\pi_K \ , \mu_1, \ldots, \mu_k \ , \ldots, \mu_K \ , \Sigma_1 \ , \ldots \Sigma_k \ , \ldots, \Sigma_K\}$

Le modèle de mélange gaussien



Modèle graphique de génération des données d'un modèle de mélange gaussien.

Estimation des paramètres du modèle de mélange gaussien

Le log-vraisemblance des données observées

$$\mathcal{L}(\theta;\mathrm{x}) = log \prod_{i=1}^{n} p(x_i, \theta)$$

$$= \sum_{i=1}^{n} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(x_i, \mu_k, \Sigma_k)$$

Le log-vraisemblance des données complétées :

$$\mathcal{L}(\theta;\mathrm{x},\mathrm{y}) = log \prod_{i=1}^{n} p(x_i, y_i|\theta)$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{K} y_{ik} \log(\pi_k \mathcal{N}(x_i, \mu_k, \Sigma_k))$$

Estimation des paramètres du modèle de mélange gaussien

L’espérance du log-vraisemblance des données complétées :

$$Q(\theta, \theta^{(c)}) = E(\mathcal{L}(\theta;x,y)|x,\theta^{(c)})$$
$$= \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_{ik} \log(\pi_k \mathcal{N}(x_i, \mu_k, \Sigma_k))$$

Estimation des paramètres du modèle de mélange gaussien

Étant donné $\theta^{(0)} = \{\pi_1^{(0)},\ldots,\pi_k^{(0)},\ldots,\pi_K^{(0)}, \mu_1^{(0)},\ldots,\mu_k^{(0)},\ldots,\mu_K^{(0)}, \Sigma_1^{(0)},\ldots,\Sigma_k^{(0)},\ldots,\Sigma_K^{(0)}\}$

Dans **l'étape E** on calcule les probabilités à postériori:

$$\tau_{ik}^{(c+1)} = \frac{\pi_k^{(c)} \mathcal{N}(x_i, \mu_k^{(c)} \Sigma_k^{(c)})}{\sum_{g=1}^{K} \pi_g^{(c)} \mathcal{N}(x_i, \mu_g^{(c)}, \Sigma_g^{(c)})}$$

Dans **l'étape M** on maximise la fonction Q:

$$\pi_k^{(c+1)} = \frac{\sum_{i=1}^{n} \tau_{ik}^{(c+1)}}{n}$$

$$\mu_k^{(c+1)} = \frac{\sum_{i=1}^{n} \tau_{ik}^{(c+1)} x_i}{n_k^{(c)}}$$

$$\Sigma_k^{(c+1)} = \frac{\sum_{i=1}^{n} \tau_{ik}^{(c+1)} (x_i - \mu_k^{(c+1)})(x_i - \mu_k^{(c+1)})^T}{n_k^{(c)}}$$

Les deux étapes **E** et **M** sont répétées jusqu'à convergence de l'algorithme **EM**

Algorithm .

Inputs : l'ensemble de donnée X et le nombre de composant K

Initialisation :
$q \leftarrow 0$;
fixer un ϵ ;
$\theta^{(0)} = (\pi_k^{(0)}, \ldots, \pi_k^{(K)}, \alpha_1^{(0)}, \ldots, \alpha_k^{(0)})$, avec $\alpha_k^{(0)} = (\mu_k^{(0)}, \Sigma_k^{(0)})$;
while $(L^{q+1} - L^q) > \epsilon$ **do**
    Étape E
    for k=1,. . . ,K **do**
        calculer $\tau_{ik}^{(q+1)}$
    end for
    Etape M
    calculer $\pi_k^{(q+1)}$
    calculer $\mu_k^{(q+1)}$
    calculer $\Sigma_k^{(q+1)}$
    $q \leftarrow q + 1$;
end while

Outputs : $\widehat{\theta} = \theta^{(q)}$
$\widehat{\tau_{ik}} = \tau_{ik}^{(q)}$.

Exemple

petal.length	petal.width	variety
1.4	0.2	Setosa
1.4	0.2	Setosa
1.3	0.2	Setosa
1.5	0.2	Setosa
1.4	0.2	Setosa
5.8	1.6	Virginica
6.1	1.9	Virginica
6.4	2	Virginica
5.6	2.2	Virginica
5.1	1.5	Virginica

Modèles de Markov Cachés

- Généralité
- Chaine de Markov observable
- Modèles de Markov Cachés
- Que peut-on faire avec un HMM

    Problème de reconnaissance : *Algorithme forward/backward*

    Problème d'analyse : *Algorithme de VITERBI*

    Problème d'apprentissage : *Algorithme EM*

- Différentes topologie des HMMs

    Le modèle de Markov ergodique

    Le modèle de Markov Bakis

    Le modèle de Markov gauche-droite

Généralité

Pour décrire l'évolution d'une variable aléatoire (V.A) au cours du temps on utilise une famille de V.A indexée par le temps *'t'* on parle alors d'un processus stochastique.

Les chaines de Markov est un type de processus stochastique qui vérifie :

1) $p(z_1 = e_1, z_2 = e_2 \,,\ldots\, z_n = e_n) \geq 0$
2) $p(z_{n+1} = e_{n+1} | z_1 = e_1, z_2 = e_2 \,,\ldots\, z_n = e_n) = p(z_{n+1} = e_{n+1} | z_n = e_n)$

Tel que

- z_n est une suite de V.A dans un ensemble fini ou dénombrable E.
- e_n : est une suite d'éléments dans E.
- n : entier naturel.

Généralité

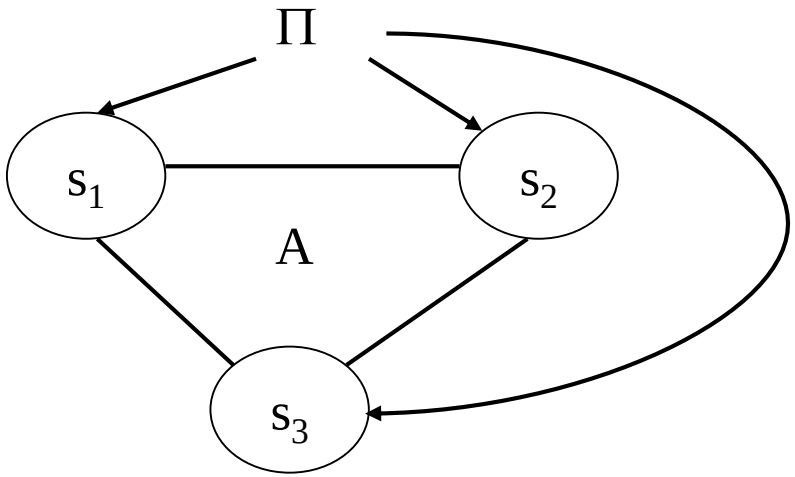
Ordre de la chaine de Markov

Définition : La suite $(z_1, z_2, \ldots z_n)$ est une chaîne de Markov d'ordre $k \geq 1$ si quelque soit $p \geq k$,

$$p(z_p = e_p | z_1 = e_1, z_2 = e_2, \ldots z_{p-1} = e_{p-1}) = p(z_{n+1} = e_{n+1} | z_{p-k} = e_{p-k}, \ldots z_{p-1} = e_{p-1})$$

Modèles de Markov observables

La chaine de Markov est dite observable si elle est définie par :



- Les états de la chaine on parle de l'alphabet S = (s_1, s_2,... s_K)

K étant le nombre d'états

- Les transitions de passer d'un état s_k à l'état s_l qui sont définies par les probabilités $p(s_l|s_k)$ qui sont regroupées dans une matrice A appelé matrice de transition définie par :

$$\mathrm{A} = \{A_{kl} = p(s_l\,|s_k)/\ k, l \in [1, K]^2\ \}$$

$$\text{avec} \sum_{k=1}^{K} A_{kl} = 1$$

- Les probabilités initiales qui représentent les probabilités de départ (les probabilités de commencer par un état s_k défini par $\mathrm{p}(s_k)$), elles sont regroupées dans un vecteur appelé vecteur de probabilité initial noté π défini par :

$$\pi = \{\pi_k = \mathrm{p}(s_k) / \; k \in [1, K]\}$$

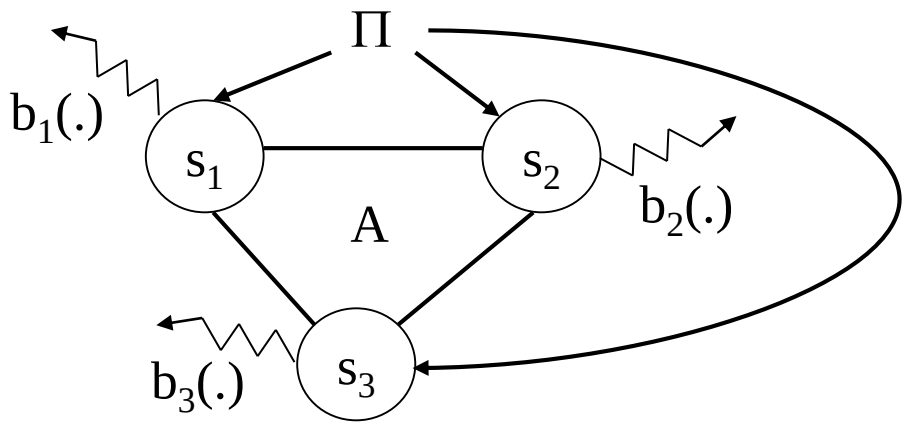
$$\text{avec} \sum_{k=1}^{K} \pi_k = 1$$

Ainsi le model de Markov observable λ est défini par une matrice de transition A et un vecteur d'initialisation π.

$$\lambda = \{\pi, A\}$$

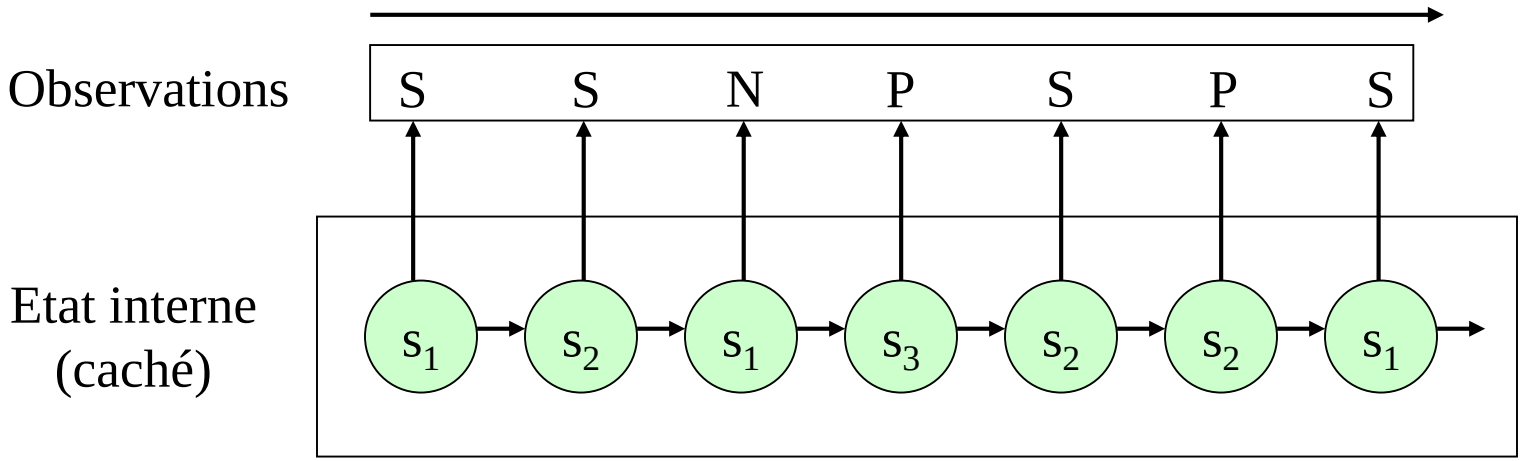
Modèles de Markov Cachés

Le modèle des chaînes de Markov cachées (hidden Markov model HMM) a été introduit par Baum et al dans les années 1960-1970. Depuis 1975, les HMM sont utilisés dans de nombreuses applications, principalement dans le domaine de la parole. Son application s'est diversifié ces derniers temps dans plusieurs domaine comme la reconnaissance de : texte manuscrit, séquences ADN, l'activité humaine …



Modèles de Markov Cachés

La séquence observée est l'évidence d'une chaîne de Markov sous-jacente cachée



L'émission d'un état observé n'est pas déterministe ! Chaque état caché émet, de manière aléatoire, un parmi K symboles d'un alphabet.

Modèles de Markov Cachés

Soit $Z = (z_{1,}\ z_2\ , \dots ,\ z_n, \dots, z_N\ )$, les états cachés du modèle de Markov, $X = (x_{1,}\ x_2\ , \dots, x_n\ , \dots, x_N\ )$ les observations émises par les états cachés, avec $z_n\ \in \{1,2, \dots, k\ \dots, K\}$.

- N représente la taille de la séquence observée,

- K représente le nombre d'états cachés.

Un modèle de Markov caché est défini par le triplet $\lambda = \{\pi\ ,\ A\ ,B\ \}$

Modèles de Markov Cachés

Avec,

- **A** : Représente la matrice de transition définie par :

$$\boldsymbol{A} = \{A_{lk} = p(z_n = k/z_{n-1} = l)/\ k, l \in [1\ K]^2\}$$

$$\sum_{k=1}^{K} A_{lk} = 1$$

- **π** : Représente le vecteur de probabilité initiale défini par

$$\boldsymbol{\pi} = \{\pi_k = p(z_k)/\ \ k \in [1\ K]\}$$

$$\sum_{k=1}^{K} \pi_k = 1$$

- ***B*** : représente la matrice d'observation définie par:

$$\boldsymbol{B} = \{b_k(x_n) = \mathrm{P}(x_n/z_n = k)/\ n \in [1\ N], k \in [1\ K]\}$$

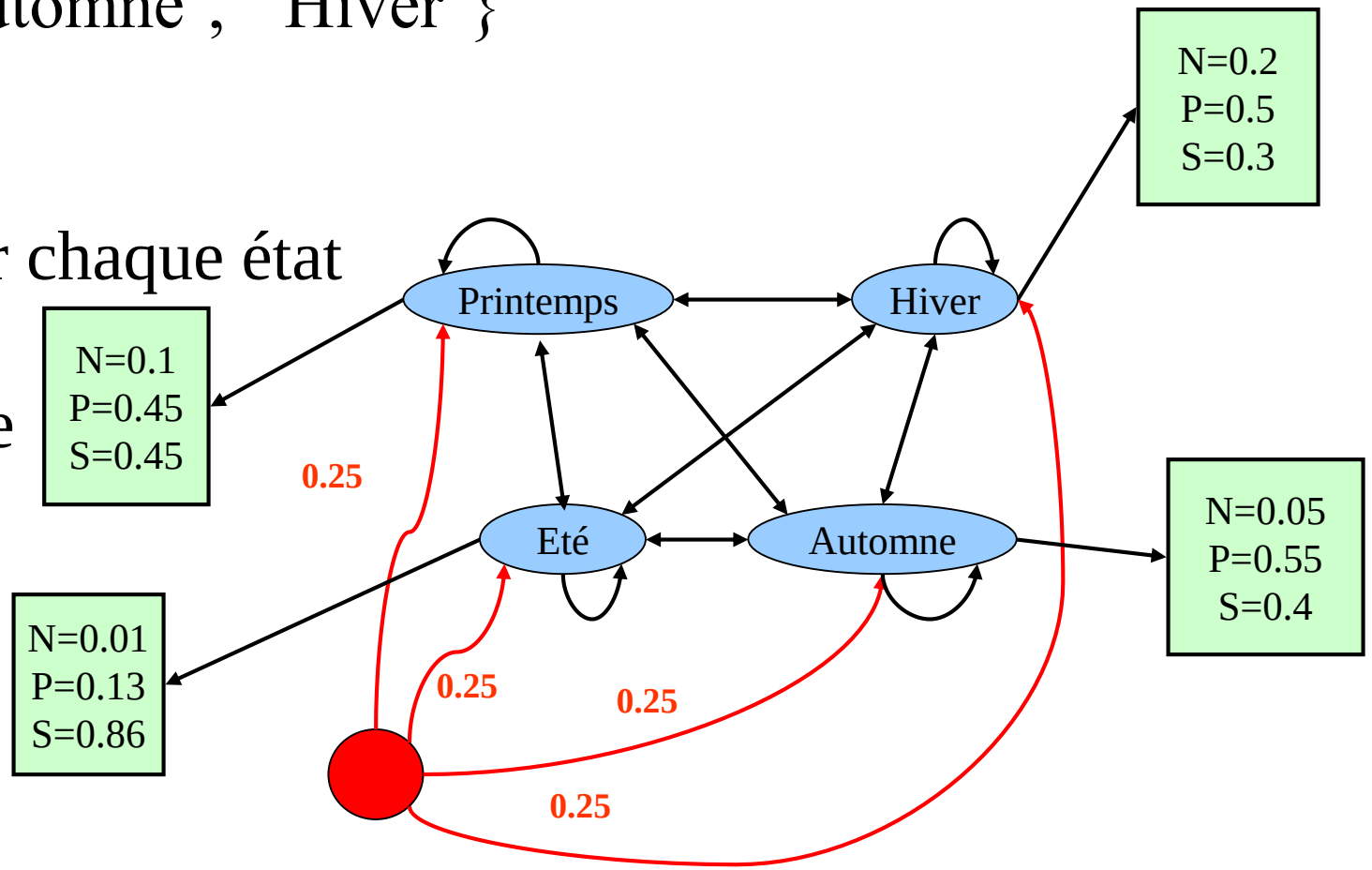
$$\sum_{k=1}^{K} b_k(x_n) = 1$$

Exemple de HMM

États :

- S={‘Printemps’, ‘Été ’,‘Automne’, ‘Hiver’}
- A={a_{ij}}

- Symboles observables émis par chaque état
  - Ω={‘N’, ‘P ’, ‘S’}
  - B={$b_j(.)$} : loi multinomiale



Que peut-on faire avec un MMC

Dans la création du model de Markov caché, il existe trois problèmes à résoudre

- **Problème de reconnaissance:** Étant donné un HMM λ et une séquence d'observation X, quelle est la probabilité P (X| λ) que X soit générée par le modèle.

- **Problème d'analyse:** Le problème posé ici est de trouver la séquence d'état Z = (z_1, z_2,... z_n) qui maximise la probabilité P (X| λ), en d'autre terme, étant donné un HMM λ et une séquence d'observation X quelle est la séquence d'état Z qui maximise la probabilité d'avoir généré X.

- **Problème d'apprentissage :** Étant donné une séquence d'observation X, comment estimer le paramètre du modèle λ = {π , A ,B } afin de maximiser la probabilité P(X| λ).

Problème de reconnaissance

Étant donné un HMM et une séquence d'observation X, quelle est la probabilité $p(X|\lambda)$ que X soit générée par le modèle

$$p(X|\lambda) = \sum_{z} p(X, Z|\lambda)$$

Problème de reconnaissance

Algorithme forward

$$\alpha_{nk} = p(x_{1,}, x_2 \ , \dots, x_n \ , z_n = k|\lambda)$$

$$= p\big(x_{1,}, x_2 \ , \dots, x_n \ \big| z_n = k; \lambda\big) p(z_n|\lambda)$$

$$= p\big(x_{1,}, x_2 \ , \dots, x_{n-1} \ \big| z_n = k; \lambda\big) p(\ x_n \ | z_n = k; \lambda) p(z_n = k|\lambda)$$

$$= p(x_{1,}, x_2 \ , \dots, x_{n-1} \ , z_n = k|\lambda) p(\ x_n \ | z_n = k; \lambda)$$

$$= \sum_{l=1}^{K} p(x_{1,}, x_2 \ , \dots, x_{n-1} \ , z_{n-1} = l, \ z_n = k|\lambda) \ p(\ x_n \ | z_n = k; \lambda)$$

$$= \sum_{l=1}^{K} p(x_{1,}, x_2 \ , \dots, x_{n-1} \ , z_n = k | z_{n-1} = l; \lambda) p(z_{n-1} = l) \ p(\ x_n \ | z_n = k; \lambda)$$

$$= \sum_{l=1}^{K} p(x_{1,}, x_2 \ , \dots, x_{n-1} | z_{n-1} = l; \lambda) p(z_n = k | z_{n-1} = l; \lambda) p(z_{n-1} = l) \ p(\ x_n \ | z_n = k; \lambda)$$

$$= \sum_{l=1}^{K} p(x_{1,}, x_2 \ , \dots, x_{n-1} \ , z_{n-1} = l|\lambda) p(z_n = k | z_{n-1} = l; \lambda) p \ (\ x_n \ | z_n = k; \lambda)$$

$$= \sum_{l=1}^{K} [\alpha_{(n-1)l} A_{lk}] p \ (\ x_n \ | z_n = k; \lambda)$$

Algorithme forward

Initialisation

$$\alpha_{1k} = \pi_k p(x_1 | z_1 = k; \lambda)$$

Etape récursive

$$\alpha_{(n+1)k} = \sum_{l=1}^{K} [\alpha_{nl} A_{lk}] p\,(\, x_n \,| z_n = k; \lambda)$$

Pour tout 1<k< K, et 1<n<N-1

Calcul de la probabilité totale

$$p(X|\lambda) = \sum_{k=1}^{K} \alpha_{Nk}$$

Algorithme backward

$$\beta_{nk} = p(x_{n+1}, \dots, x_N \,|z_n = k; \lambda);$$
$$= \sum_{l=1}^{K} p(x_{n+1}, \dots, x_N, z_{n+1} = l | z_n = k; \lambda)$$
$$= \sum_{l=1}^{K} p(x_{n+1}, \dots, x_N \,|z_{n+1} = l, z_n = k; \lambda)\, p(z_{n+1} = l | z_n = k; \lambda)$$
$$= \sum_{l=1}^{K} p(x_{n+2}, \dots, x_N \,|z_{n+1} = l, z_n = k; \lambda)\, p(z_{n+1} = l | z_n = k; \lambda) p(x_{n+1} \,|z_{n+1} = l; \lambda)$$
$$= \sum_{l=1}^{K} p(x_{n+2}, \dots, x_N \,|z_{n+1} = l; \lambda)\, p(z_{n+1} = l | z_n = k; \lambda) p(x_{n+1} \,|z_{n+1} = l; \lambda)$$
$$= \sum_{l=1}^{K} [\beta_{(n+1)l} A_{kl}] p(x_{n+1} \,|z_{n+1} = l; \lambda)$$

Algorithme backward

Initialisation

$$\beta_{Nk} = 1$$

Etape récursive

$$\beta_{nk} = \sum_{l=1}^{K} [\beta_{(n+1)l} A_{kl}] p\,(\, x_{n+1} \,|z_{n+1} = l; \lambda)$$

Pour tout 1<k< K, et n = N-1, N-2 , … ,1

Calcul de la probabilité totale

$$p(X|\lambda) = \sum_{k=1}^{K} \pi_k \, p(x_1|z_1 = k; \lambda)\beta_{1k}$$

Problème d'analyse

Le problème posé ici est de trouver la séquence d'état Z = (z_1, z_2,... z_n) qui maximise la probabilité P (Z|X, λ), en d'autre terme.

$$Z = \underset{z}{\operatorname{argmax}}\, p(x_1, x_2, \dots, x_n, z_1, z_2, \dots, z_n; \lambda)$$

$$= \underset{z}{\operatorname{argmax}}\, p(z_1;\lambda)p(x_1|z_1;\lambda)\prod_{n=2}^{N} p(x_n|z_n;\lambda)p(z_n|z_{n-1};\lambda)$$

$$= \underset{z}{\operatorname{argmax}}\, \log\big(p(z_1;\lambda)\big) + \log\big(p(x_1|z_1;\lambda)\big) + \sum_{n=2}^{N} log\big(p(x_n|z_n;\lambda)\big) + \log(p(z_n|z_{n-1};\lambda))$$

$$= \underset{z}{\operatorname{arg}min}\, -\log\big(p(z_1;\lambda)\big) - \log\big(p(x_1|z_1;\lambda)\big) - \sum_{n=2}^{N} log\big(p(x_n|z_n;\lambda)\big) - \log(p(z_n|z_{n-1};\lambda))$$

Problème d'analyse

Pseudo code de l'algorithme de VITERBI.

Entrées : Matrice des données X et un modèle de Markov $\lambda = \{\pi,\ A,\ B\}$

Initialisation : $z_1 = k$, pour tout $k = 1, \dots, K$

$S_1(z_1 = k) = -\log(\pi_1) - \log p(x_1|z_1 ; \lambda)$

Etape récursive : pour tout $k = 1, \dots, K$ et pour tout $n = 2, \dots, N$

On calcule

$$S_n(z_n = k) = -\log p(x_n|z_n; \lambda) + \min_{z_{n-1}} [S_{n-1}(z_{n-1}) - \log p(z_n = k|z_{n-1}; \lambda)]$$

Soit

$$z^*_{n-1}(z_n) = \operatorname*{argmin}_{z_{n-1}} [S_{n-1}(z_{n-1}) - \log p(z_n = k|z_{n-1}; \lambda)]$$

Stockage du meilleur état précédent via le calcul de : $\min_{z_n} [S_n(z_n)]$ et $\hat{z_n} = \operatorname*{argmin}_{z_n} [S_n(z_n)]$

Détermination du meilleur chemin : tout $n = N - 1, \dots, 1$

$$\hat{z_n} = z^*_n(\hat{z_{n+1}})$$

Problème d'apprentissage

Algorithme de Baum-Welch

Soit $\lambda^{(0)} = \{\pi^{(0)}, A^{(0)}, B^{(0)}\}$ le model initial qui peut prendre des valeurs arbitraires comme on peut l'initialiser avec les connaissances apriori que nous avant déjà sur le problème. Ici on cherche donc les paramètres du model $\lambda^{(q)} = \{\pi^{(q)}, A^{(q)}, B^{(q)}\}$ qui maximise P(X), tel que $p(X|\lambda^{(q-1)}) \leq p(X|\lambda^{(q)})$

On sait que la distribution de la séquence d'état caché $Z = (z_1, z_2, \dots, z_N)$ dans le cas d'une chaîne de Markov cachée d'ordre 1 peut s'écrire de la façon suivante:

$$p(Z;\lambda) = p(z_1;\lambda) \prod_{n=2}^{N} p(z_n|z_{n-1};\lambda)$$

La distribution conditionnelle de la séquence d'observations $X = (x_1, x_2, \dots, x_n, \dots, x_N)$ est donnée par:

$$p(X|Z;\lambda) = p(x_1|z_1;\lambda) \prod_{n=2}^{N} p(x_n|z_n;\lambda)$$

Problème d'apprentissage

Ainsi, nous obtenons la distribution conjointe suivante :

$$p(X, Z; \lambda) = p(Z; \lambda)p(X|Z; \lambda)$$

$$= p(z_1; \lambda)p(x_1|z_1; \lambda) \prod_{n=2}^{N} p(x_n|z_n; \lambda)p(z_n|z_{n-1}; \lambda)$$

L'estimation de ces paramètres peut se faire par la maximisation du log-vraisemblance donné par l'équation suivante:

$$L(X; \lambda) = p(X; \lambda)$$

$$= log \sum_{z} p(X, Z; \lambda)$$

$$= \log \sum_{z} p(z_1; \lambda)p(x_1|z_1; \lambda) \prod_{n=2}^{N} p(x_n|z_n; \lambda)p(z_n|z_{n-1}; \lambda)$$

Problème d'apprentissage

La maximisation de cet log-vraisemblance peut s'avérer difficile analytiquement, d'où l'utilisation de l'algorithme Baum Welch qui est une version de l'algorithme EM appliqué aux HMMs.

$$p(X,Z;\lambda) = \prod_{k=1}^{K} p(z_1 = k;\lambda)^{z_{1k}} \prod_{n=2}^{N}\prod_{k=1}^{K}\prod_{l=1}^{K} p(z_n = k|z_{n-1} = l;\lambda)^{z_{n-1,l}z_{n,k}} \prod_{n=1}^{N}\prod_{k=1}^{K} p(x_n|z_n = k;\lambda)^{z_{nk}}$$
$$= \prod_{k=1}^{K} \pi_k{}^{z_{1k}} \prod_{n=2}^{N}\prod_{k=1}^{K}\prod_{l=1}^{K} A_{kl}{}^{z_{n-1,l}z_{n,k}} \prod_{n=1}^{N}\prod_{k=1}^{K} p(x_n|z_n = k;\lambda)^{z_{nk}}$$

où z_{nk} est une variable binaire qui vaut 1 lorsque x_n est générée par le k*ième* état du modèle λ, et qui vaut 0 sinon. Ainsi le log de vraisemblance des données complétées peut s'écrire comme suit :

$$L(X,Z;\lambda) = \sum_{l=1}^{K} z_{1k}\, log(\pi_k) + \sum_{n=2}^{N}\sum_{k=1}^{K}\sum_{l=1}^{K} z_{n-1,l}z_{n,k} log(A_{kl}) + \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk} log(p(x_n|z_n = k;\lambda)$$

Problème d'apprentissage

L'algorithme EM commence avec un paramètre initial λ^0 et répète les étapes E et M suivantes jusqu'à la convergence.

L'étape E: Elle consiste à calculer l'espérance du log-vraisemblance des données complétées (la fonction auxiliaire Q) :

$$Q\left(\lambda, \lambda^{(q)}\right) = \mathbb{E}\left[L(X, Z; \lambda) \middle| X, \lambda^{(q)}\right]$$

$$= \sum_{k=1}^{K} \mathbb{E}\left[z_{1k} \middle| X, \lambda^{(q)}\right] log(\pi_k) + \sum_{n=2}^{N} \sum_{k=1}^{K} \sum_{l=1}^{K} \mathbb{E}\left[z_{n-1,l} z_{n,k} \middle| X, \lambda^{(q)}\right] log(A_{kl}) + \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{E}\left[z_{nk} \middle| X, \lambda^{(q)}\right] z_{nk} log(p(x_n | z_n = k; \lambda))$$

$$= \sum_{k=1}^{K} p\left[z_1 = k \middle| X, \lambda^{(q)}\right] log(\pi_k) + \sum_{n=2}^{N} \sum_{k=1}^{K} \sum_{l=1}^{K} p\left[z_{n-1} = l, z_n = k \middle| X, \lambda^{(q)}\right] log(A_{kl})$$

$$+ \sum_{n=1}^{N} \sum_{k=1}^{K} p\left[z_n = k \middle| X, \lambda^{(q)}\right] log(p(x_n | z_n = k; \lambda))$$

$$= \sum_{k=1}^{K} \tau_{1k}^{(q)} log(\pi_k) + \sum_{n=2}^{N} \sum_{k=1}^{K} \sum_{l=1}^{K} \xi_{nlk}^{(q)} log(A_{kl}) + \sum_{n=1}^{N} \sum_{k=1}^{K} \tau_{nk}^{(q)} log(p(x_n | z_n = k; \lambda))$$

Problème d'apprentissage

- $\tau_{nk}^{(q)}$ est la probabilité a posteriori de l'état *k* à l'instant *n* étant donnée la séquence d'observation X

- $\xi_{nlk}^{(q)}$ est la probabilité jointe de l'état *k* à l'instant *n* et l'état *l* à l'instant *n-1* étant donnée la séquence X

$\tau_{nk}^{(q)}$ et $\xi_{nlk}^{(q)}$ sont données par :

$$\tau_{nk}^{(q)} = \frac{\alpha_{nk}^{(q)} \beta_{nk}^{(q)}}{\sum_{k=1}^{K} \alpha_{nk}^{(q)} \beta_{nk}^{(q)}}$$

$$\xi_{nlk}^{(q)} = \frac{\alpha_{n-1,l}^{(q)} p(x_n \mid z_n = k; \lambda^{(q)}) \beta_{nk}^{(q)} a_{lk}^{(q)}}{\sum_{k=1}^{K} \sum_{l=1}^{K} \alpha_{n-1,l}^{(q)} p(x_n \mid z_n = k; \lambda^{(q)}) \beta_{nk}^{(q)} a_{lk}^{(q)}}$$

Problème d'apprentissage

L'étape M : Elle permet de mettre à jour les paramètres du modèle par le calcul de $\lambda^{(q+1)}$ en maximisant la fonction Q par rapport à λ. On obtient les formules de mise à jour suivantes:

$$\pi_k^{(q+1)} = \tau_{1k}^{(q)}$$

$$A_{lk}^{(q+1)} = \frac{\sum_{n=2}^{N} \xi_{nlk}^{(q)}}{\sum_{n=2}^{N} \tau_{nk}^{(q)}}$$

La maximisation de fonction Q par rapport aux paramètres de $p(x_n|z_n = k; \lambda)$ dépond de la forme de la densité

Problème d'apprentissage

Dans le cas d'un HMM continue :

$$p(x_n|z_n = k;\lambda) = \mathcal{N}(x_n, \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{\frac{d}{2}}\sqrt{\det(\Sigma_k)}} e^{-\frac{1}{2}(x_n-\mu_k)^T\Sigma_k^{-1}(x_n-\mu_k)}$$

Le log-vraisemblance:

$$L(X, Z;\lambda) = \sum_{l=1}^{K} z_{1k}\, log(\pi_k) + \sum_{n=2}^{N}\sum_{k=1}^{K}\sum_{l=1}^{K} z_{n-1,l} z_{n,k} log(A_{kl}) + \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk} log(p(x_n|z_n = k;\lambda)$$

La fonction auxiliaire Q :

$$Q\left(\lambda, \lambda^{(q)}\right) = \mathbb{E}\left[L(X, Z;\lambda)\middle|X, \lambda^{(q)}\right]$$

$$= \sum_{l=1}^{K} \tau_{nk}^{(q)}\, log(\pi_k) + \sum_{n=2}^{N}\sum_{k=1}^{K}\sum_{l=1}^{K} \xi_{nlk}^{(q)} log(A_{kl}) + \sum_{n=1}^{N}\sum_{k=1}^{K} \tau_{nk}^{(q)} log(\mathcal{N}(x_n, \mu_k, \sum_k))$$

L'étape E

- $\tau_{nk}^{(q)}$ est la probabilité a posteriori de l'état *k* à l'instant *n* étant donné la séquence d'observation X

- $\xi_{nlk}^{(q)}$ est la probabilité jointe de l'état *k* à l'instant *n* et l'état *l* à l'instant *n-1* étant donné la séquence X

$\tau_{nk}^{(q)}$ et $\xi_{nlk}^{(q)}$ sont données par :

$$\tau_{nk}^{(q)} = \frac{\alpha_{nk}^{(q)} \beta_{nk}^{(q)}}{\sum_{k=1}^{K} \alpha_{nk}^{(q)} \beta_{nk}^{(q)}}$$

$$\xi_{nlk}^{(q)} = \frac{\alpha_{n-1,l}^{(q)} p(x_n \mid z_n = k; \lambda^{(q)}) \beta_{nk}^{(q)} a_{lk}^{(q)}}{\sum_{k=1}^{K} \sum_{l=1}^{K} \alpha_{n-1,l}^{(q)} p(x_n \mid z_n = k; \lambda^{(q)}) \beta_{nk}^{(q)} a_{lk}^{(q)}}$$

Problème d'apprentissage

L’étape M : On obtient les formules de mise à jour suivantes:

$$\pi_k^{(q+1)} = \tau_{1k}^{(q)}$$

$$A_{lk}^{(q+1)} = \frac{\sum_{n=2}^{N} \xi_{nlk}^{(q)}}{\sum_{n=2}^{N} \tau_{nk}^{(q)}}$$

$$\mu_k^{(q+1)} = \frac{1}{\sum_{n=2}^{N} \tau_{nk}^{(q)}} \sum_{n=1}^{N} \tau_{nk}^{(q)} x_n$$

$$\sum_k^{(q+1)} = \frac{1}{\sum_{n=2}^{N} \tau_{nk}^{(q)}} \sum_{n=1}^{N} \tau_{nk}^{(q)} (x_n - \mu_k^{(q+1)})(x_n - \mu_k^{(q+1)})^T$$

Pseudocode de l'algorithme EM pour les modèles HMM

Données : Matrice des données : X

Initialisation $\lambda^{(0)} = \{\pi^{(0)}, A^{(0)}, \mu^{(0)}, \sum^{(0)}\}$

tant que test de convergence faire

 # Etape E

calcul des probabilités a posteriori

pour tous les k ∈ [1, K] faire

$$\tau_{nk}^{(q)} \qquad n \in [1, N]$$
$$\xi_{nlk}^{(q)} \qquad n \in [1, N]$$

 # Etape M

maximisation de la fonction auxiliaire

pour tous les k ∈ [1, K] faire

$$\pi_k^{(q+1)} = \tau_{1k}^{(q)}$$

$$a_{lk}^{(q+1)} = \frac{\sum_{n=2}^{N} \xi_{nlk}^{(q)}}{\sum_{n=2}^{N} \tau_{nk}^{(q)}}$$

$$\mu_k^{(q+1)} = \frac{1}{\sum_{n=2}^{N} \tau_{nk}^{(q)}} \sum_{n=1}^{N} \tau_{nk}^{(q)} x_n$$

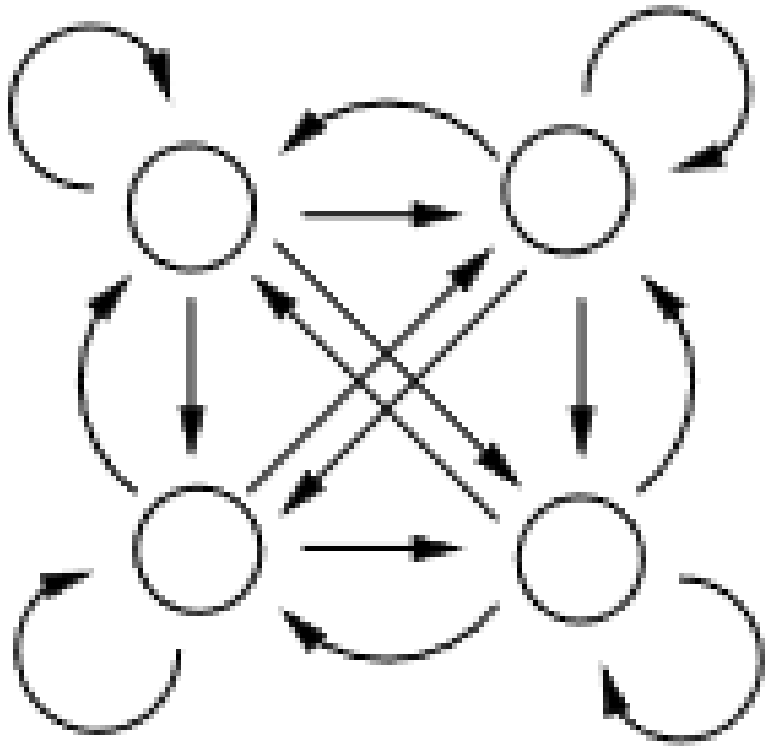
$$\sum_k^{(q+1)} = \frac{1}{\sum_{n=2}^{N} \tau_{nk}^{(q)}} \sum_{n=1}^{N} \tau_{nk}^{(q)} (x_n - \mu_k^{(q+1)})(x_n - \mu_k^{(q+1)})^T$$

q = q + 1

Résultat : Paramètres estimés : $\lambda = \{\pi, A, \mu, \sum\}$

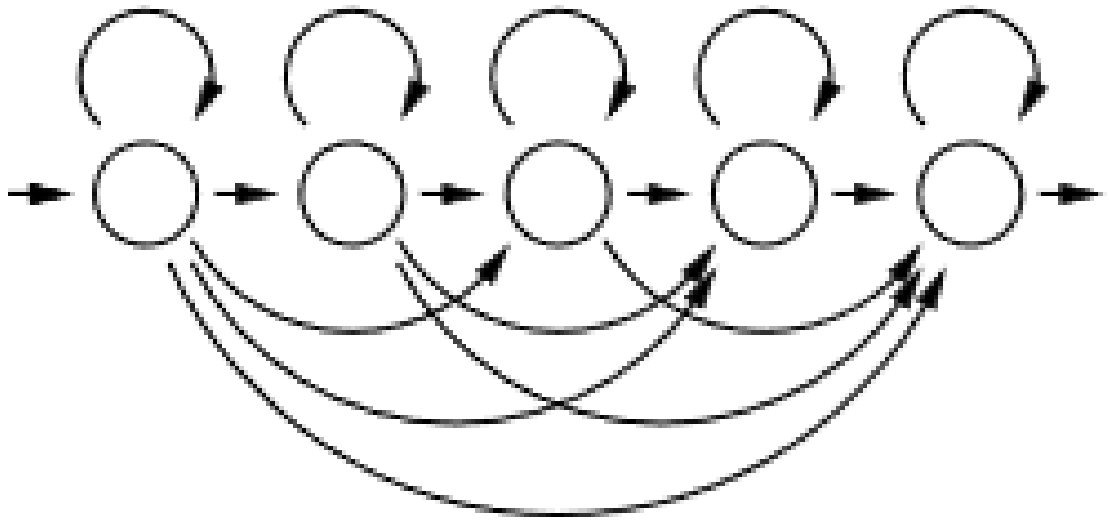
Différentes topologie des HMMs

Le modèle de Markov ergodique : la topologie ergodique est une topologie dont les états peuvent être atteint à partir de n'importe quel état ce qui implique que les coefficients de sa matrice de transition A sont strictement positif ($A_{kl} > 0 \quad \forall k$, l $\in$[1 N]^2).



Différentes topologie des HMMs

Topologie linéaire



Différentes topologie des HMMs

Topologie Bakis