TP3 : Apprentissage non supervisé

L'objectif de ce Tp est d'implémenter les modèles d'apprentissage non supervisé vus en cours à savoir : K-means, HCA, Modèle de mélange gaussien et HMM. Pour l'implémentation, vous avez le libre choix du langage de programmation (Python, Matlab, R, etc.).
Afin de tester, les modèles développés, vous allez utiliser la base de données Iris téléchargeable ici. Iris est un ensemble de données introduit en 1936 par Ronald Aylmer Fisher comme un exemple d'analyse discriminante. Cet ensemble contient 150 exemples de critères observés sur 3 espèces différentes d'iris de Gaspésie. Chaque exemple est composé de quatre attributs (longueur et largeur des sépales en cm, longueur et largeur des pétales en cm) et d'une classe (l'espèce).

1. K-means

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.

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'estimées $Y=\{y_i\}_{i\leq n}$ 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

Etapes de la mise en œuvre du k-means

1) Initialisation aléatoire des centres de classes
2) Affectation de chaque observation x_i au cluster C_k de centre μ_k tel que dist(x_i, μ_k) est minimale

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

3) Recalcule des centres μ_k de chaque cluster

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

4) Répéter 2 et 3 jusqu'à convergence

L'algorithme ci-dessous résume, les différentes étapes de la mise en œuvre du k-means.

Entrées : S = $\{x_1, \dots, x_j, \dots, x_n\}$ un ensemble de données.
$Y = \{c_1, c_2, \dots, c_k, \dots, c_K\}$ 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 $\boldsymbol{\mu}$

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

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

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

4 : End For

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

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

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

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

Travail à réaliser

1.1. Implémenter l'algorithme des k-means ci-dessus.
1.2. Proposer une stratégie d'initialisation pour garantir une meilleure convergence.
1.3. Tester l'algorithme sur la base de données IRIS et évaluer ses performances en termes de taux de bonne classification, rappel, précision et f-mesure.
1.4. Donner la matrice de confusion.
1.5. Analyser les résultats obtenus.
1.6. Utiliser une bibliothèque existante incluant le k-means et tester le sur la base de données IRIS.
1.7. Comparer les résultats les résultats obtenus.

2. Modèle de mélange gaussien

Le modèle de mélange gaussien modélise la densité de x en une combinaison linéaire pondérée des densités de K composantes. La densité du mélange de x_i est donnée par :

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

avec

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

La log-vraisemblance des données observées

$$\mathcal{L}(\theta;\text{x}) = log \prod_{i=1}^{n} p(x_i, \theta)$$
$$= \sum_{i=1}^{n} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(x_i, \mu_k, \Sigma_k)$$

La log-vraisemblance des données complétées :

$$\mathcal{L}(\theta;\text{x,y}) = log \prod_{i=1}^{n} p(x_i, y_i|\theta)$$
$$= \sum_{i=1}^{n} \sum_{k=1}^{K} y_{ik} \log(\pi_k \mathcal{N}(x_i, \mu_k, \Sigma_k)$$

L'espérance de la log-vraisemblance des données complétées :

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

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 apostériori :

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

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

L'algorithme ci-dessous résume, les différentes étapes de la mise en œuvre du modèle de mélange gaussien.

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

Travail à réaliser

2.1. Implémenter l'algorithme du GMM ci-dessus.
2.2. Proposer une stratégie d'initialisation pour garantir une meilleure convergence.
2.3. Tester l'algorithme sur la base de données IRIS et évaluer ses performances en termes de taux de bonne classification, rappel, précision et f-mesure.
2.4. Donner la matrice de confusion.
2.5. Analyser les résultats obtenus.

2.6. Utiliser une bibliothèque existante incluant le GMM et tester le sur la base de données IRIS.
2.7. Comparer les résultats les résultats obtenus.