esiea
ECOLE D'INGENIEURS
DU MONDE NUMERIQUE

4A : DATAMINING ET APPLICATIONS

TP2 (durée : 3h00)
Clustering : algorithme des k-moyennes

1. Algorithme

L'algorithme k-means calcule une partition d'un ensemble $\mathbf{S}$ d'observations en k sous-ensembles. Il est constitué d'une étape d'initialisation (consistant à choisir arbitrairement une partition de départ, ou, de façon équivalente, les barycentres des sous-ensembles de cette partition) et de deux étapes répétées jusqu'à la convergence de la méthode (la partition ne change plus) :

— **Affectation** : chaque observation est associée à la partition du barycentre le plus proche

$$S_i^{(t)} = \left\{ \mathbf{x}_j : \left\| \mathbf{x}_j - \mathbf{m}_i^{(t)} \right\| \leq \left\| \mathbf{x}_j - \mathbf{m}_{i^*}^{(t)} \right\| \ \forall \ i^* = 1, \ldots, k \right\}$$

— **Mise à jour** : recalculer le barycentre de chaque partition

$$\mathbf{m}_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{\mathbf{x}_j \in S_i^{(t)}} \mathbf{x}_j$$

Cet algorithme trouve un minimum local de la fonction de coût suivante

$$\sum_{i=1}^{k} \sum_{\mathbf{x}_j \in S_i} \|\mathbf{x}_j - \boldsymbol{m}_i\|^2$$

Cette fonction représente l'erreur faite en approchant chaque point d'une classe par le barycentre de cette classe.

2. Evaluation

1) Utiliser (sur la *base1*) le kmeans de sklearn avec différentes valeurs de *k*. Pour chaque valeur de *k* :

- Observer la distribution des données autour des barycentres ;
- Noter la valeur du coût (somme des variances intra-clusters) final.

2) Pour chaque valeur de *k* :

- Réaliser 5 initialisations ;
- Afficher le coût moyen (et en bonus la variance) et utiliser la méthode du coude (ou le critère de Calinski/Harabasz) pour déterminer la valeur optimale de *k*.
- Vérifier la stabilité de l'algorithme (convergence vers les mêmes centroïdes).

3) Répéter les expérimentions précédentes sur *base3*.

3. Application

On souhaite reconnaître des chiffres manuscrits. Télécharger la base *Digits* et la séparer en deux bases d'apprentissage (70%) et de test (fonction sklearn *train_test_split*).

Clustering :

Déterminer *k=10* clusters et mesurer leur pureté *p*. Pour le cluster *i* :

p_i = *#exemples de la classe majoritaire du cluster/ #exemples dans le cluster*

Estimer la moyenne et la variance de *p*. Que peut-on conclure ?

Effectuer plusieurs initialisations et répéter les expériences précédentes. Conclure.

Classification :

Utiliser l'algorithme des k-moyennes pour déterminer, **dans chaque classe**, *k'* centroïdes (*k'* = 1, 2, …).

Utiliser l'algorithme du plus-proche-voisin pour classer la base de test en utilisant comme base d'apprentissage les *10k'* centroïdes précédents.

Effectuer, pour chaque valeur de *k'*, plusieurs initialisations et estimer la moyenne et la variance des taux de reconnaissance obtenus. Rapporter ces résultats dans un tableau/graphe. Conclure.

4. BONUS

Codez vous-même l'algorithme du kmeans et vérifier que ses résultats concordent avec le kmeans de sklearn.