CM3 : Classification NON Supervisée

Le Clustering :

Technique d'apprentissage auto, permet de regrouper chaînes de données, distance ou similarité, populaire.
Le clustering est utilisé lorsqu'il devient difficile de récolter les données que l'algorithme va traité
Avantages:

- Disponibilité totale
- Répartition des charges
- Fonctionnalité calcul parallèles

Si un ordinateur subit une erreur son travail sera redirigé vers les autres ordinateurs s'il y en a, cela peut aussi être un serveur.
Il y a 3 type méthode:

- Hiérarchique : les méthodes sont distinct et disposent d'une matrice a distance
- Centroïde : utilise k-moyen/k-means, se fais en 1 fois
- Densité : basé sur la densité

Clustering partitionne en sous groupe

K moyen (K means):

On veut mettre N observations dans différents catégories k. le K-means est un algorithme qui tente de diviser les données en fonction de leur similarité sans connaître à l'avance les étiquettes ou les catégories.

Affinity Propagation :

1. Chaque élément repère dans son voisinage un élément qui lui ressemble suffisamment, et augmente son affinité pour cet objet.
2. Deuxième étapes c'est de propager cette affinité

a. Chaque élément "c" repère pour qui il a la plus grande affinité, noté m
b. Il ajoute à ses propres affinités celles de m
c. Cette étape est répétée un certain nombre de fois, ou bien jusqu'à ce que le nombre d'éléments passe en dessous d'un certain seuil - ou encore quand cette étape n'apporte plus aucun changement.

3. Il y a alors trois cas :

a. L'élément considéré possède une affinité maximale pour un autre élément : il lui ressemble ;
b. L'élément considéré possède une affinité maximale pour lui-même : il est « exemplaire » (exemplar)
c. L'élément considéré possède une affinité nulle : il est « isolé »

Hierarchical clustering :

Il s'agit d'une méthode de clustering qui construit une hiérarchie de clusters.

On distingue deux approches principales :

- **Agglomératif (ascendant)** : chaque point de données commence comme son propre cluster, puis les clusters les plus proches sont fusionnés jusqu'à ce qu'il ne reste qu'un seul cluster.
- **Divisif (descendant)** : on commence avec un seul cluster contenant toutes les données, puis il est divisé successivement en sous-clusters.

Le résultat est souvent visualisé sous forme de **dendrogramme**.

HDBSCAN :

HDBSCAN (**Hierarchical Density-Based Spatial Clustering of Applications with Noise**) est une extension améliorée de l'algorithme **DBSCAN** (Density-Based Spatial Clustering of Applications with Noise). Il est conçu pour résoudre certaines des limitations de DBSCAN, notamment en ce qui concerne la détection de clusters de densités variées et l'automatisation du choix des paramètres.

L'algorithme combine une approche **hiérarchique** et une approche **basée sur la densité** pour trouver des clusters de manière plus flexible. Il repose sur une analyse de la densité des points de données, et il crée une hiérarchie de clusters à partir de laquelle des clusters stables sont extraits automatiquement.

OPTICS :

OPTICS ne cherche pas directement à former des clusters. Il produit plutôt un **ordre des points** en fonction de leur structure de densité dans l'espace des données. Ensuite, cet ordre peut être utilisé pour visualiser les regroupements ou extraire des clusters avec différents paramètres de densité.

BIRCH :

BIRCH (**Balanced Iterative Reducing and Clustering using Hierarchies**) est un algorithme de **clustering hiérarchique** conçu pour être particulièrement efficace avec de **grands ensembles de données**. Son objectif principal est d'être **rapide** et **évolutif**, même sur des jeux de données massifs, tout en utilisant une quantité de mémoire limitée.

Contrairement à des algorithmes comme **K-means** ou **DBSCAN**, BIRCH n'exige pas que l'ensemble des données soit chargé en mémoire à la fois. Il traite les données de manière incrémentielle, ce qui en fait une solution idéale pour des données à grande échelle.