Big Data & Machine Learning

SÉANCE 3 – CLUSTERING ET GPU

Organisation

Séance 1
- Introduction sur l'évolution du Big Data et du Machine Learning
- Modélisation avec le Framework Spark (technique de Random Forest)
- Modélisation avec XGBoost (Gradient Boosting)

Séance 2
- Le pourquoi du Framework DASK?
- Utilisation de DASK via une régression logistique et un Gradient Boosting
- Introduction à l'explicabilité / interprétabilité des modèles (techniques Shap et Lime)
- Introduction à TPOT (algo évolutionniste génétique de sélection de modèles)

Séance 3
- Implémentation des techniques de clustering (KMeans, KNN, Spectral)
- Comparaison CPU vs GPU (Framework RAPIDS de Nvidia)

Séance 4
- Distinction des différentes méthodes de réseaux de neurones (Perceptron, CNN, RNN)
- Classification de texte, analyse de sentiment, NLP

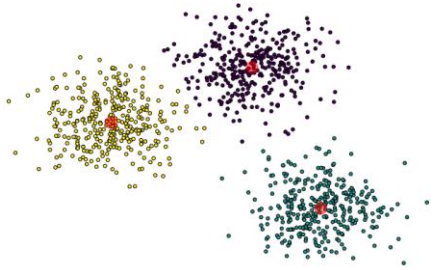
Initialisation

Environnement de travail

Utilisation de Google Colab

- Installation de Mini Conda
- Installation de RAPIDS 0.18
- Installation de Python 3.7

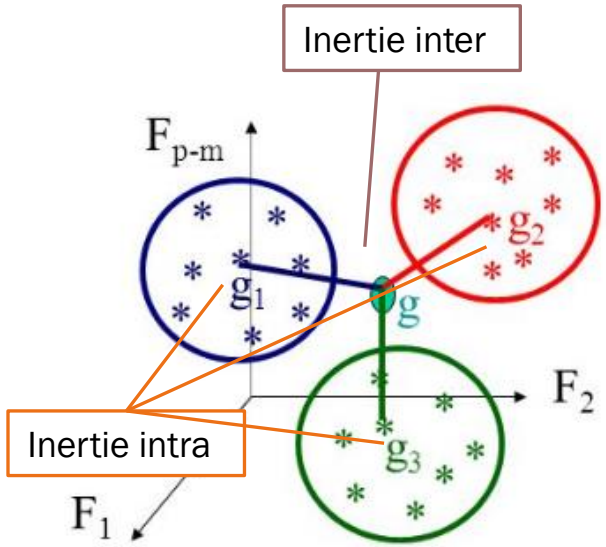
Le clustering



Hormis le KNN (vu un peu plus loin), le clustering se classe dans la famille des algorithmes non supervisés (pas de variable explicative)
Il s'agit alors de regrouper des individus par similarité, afin de créer des groupes d'individus

Les nouveaux entrants seront alors classifiés dans un groupe existant

- A cette étape cela devient du supervisé : prédire à quel groupe appartient un nouvel individu



Minimiser l'inertie intra class, maximiser l'inertie inter class

KMeans

Regrouper des individus entre eux en les classant, via une distance euclidienne (distance géométrique dans un espace multidimensionnel)

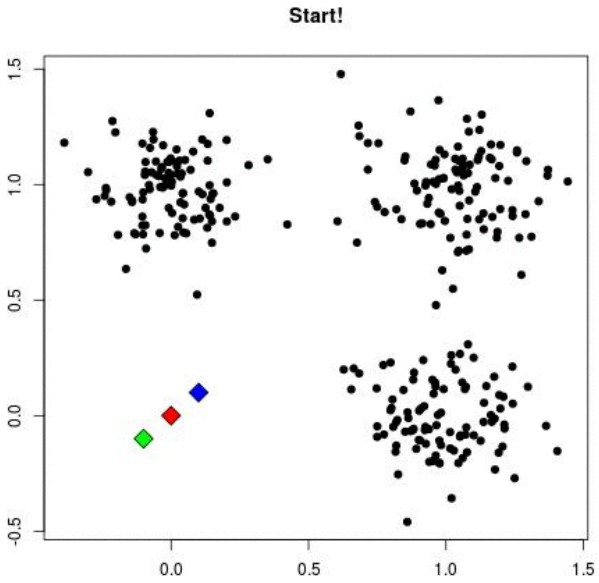
Distance euclidienne

$De(x,y) = \sqrt{\sum(xi - yi)^2}$

Distance Manhattan

$Dm(x,y) = \sum|xi - yi|$

Barycentre / centre de gravité : centre géométrique de l'ensemble des points finis



- **Etape 1** : placer les centroïdes aléatoirement
- **Etape 2** : affecter les points au centroïde le plus proche
- **Etape 3** : bouger les centroïdes sur leur barycentre
- Répétition des étapes 2 et 3

Inconvénients

Le choix des barycentres au départ aléatoires influencera la suite des étapes

Les KMeans ne détectent bien que les clusters aux formes sphériques

KMeans

Déterminer le bon nombre de cluster

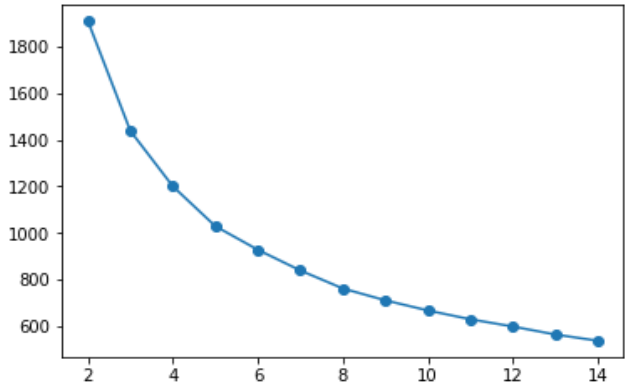
Courbe de Elbow

Plus on augmente le nombre de cluster en X, plus l'inertie intra class en Y diminue

On trace ici la courbe de Elbow :

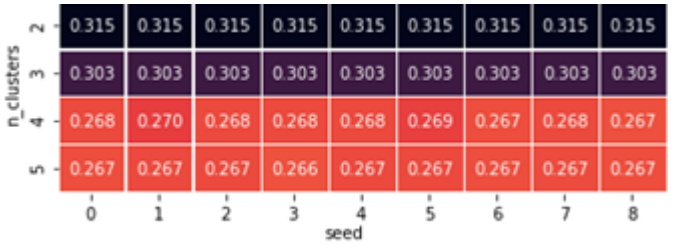
- X : nombre de cluster
- Y : Inertie intra class

Le nombre de cluster optimal va se trouver au niveau du coude (les derniers sauts importants)



Silhouette score

- Score entre [-1 , 1]
- Etudie la distance de séparation entre les clusters, comment chaque point d'un cluster est proche du cluster voisin
- 1 signifie que les clusters sont éloignés, 0 indique que les clusters sont proches et que la décision d'affectation est très proche entre plusieurs clusters, négatif signifie que les observations peuvent être affectées au mauvais cluster
  - Il faut donc tendre vers le score optimal : 1



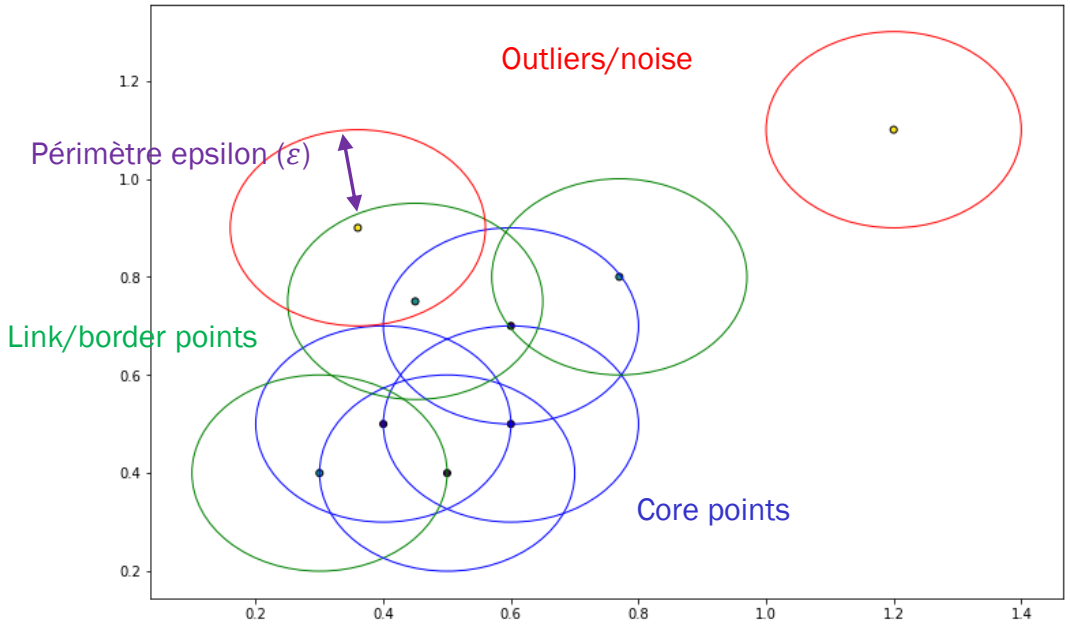
Formule : (b - a) / max(a, b)

- Distance moyenne intra cluster (a)
- Distance moyenne du cluster le plus proche (b)

Spectral clustering
DBSCAN

La méthode **DBSCAN** fait partie de la famille des spectral clustering

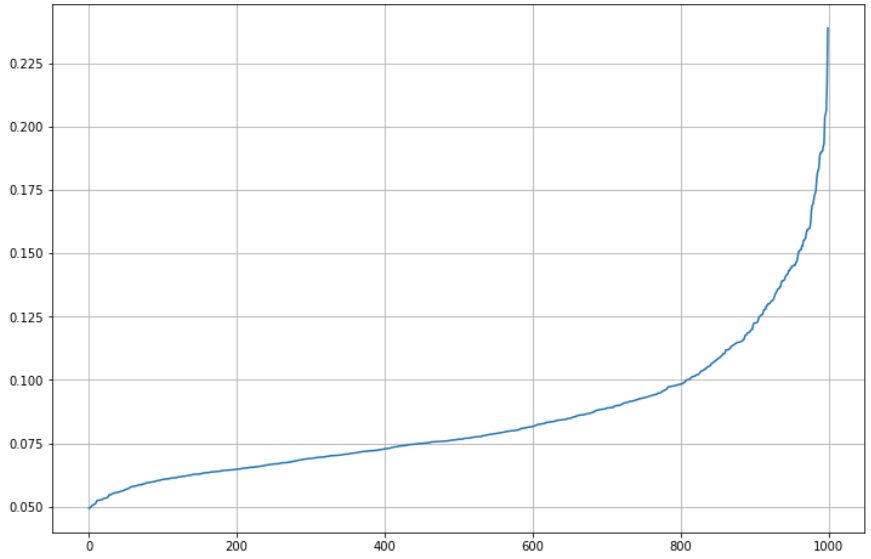
- Elle relie ensemble les sommets dans un périmètre epsilon
- Ensuite pour être considéré comme un point core, le point doit posséder un minimum de voisins dans son périmètre epsilon, afin de pouvoir relier d'autres points
- Les points isolés peuvent donc ne pas être reliés à un cluster



DBSCAN

Déterminer le bon nombre de cluster

- Déterminer les K plus proches voisins de chaque point
- Réaliser la moyenne des distances des K voisins pour chaque point
- Trier par ordre croissant les distances
- Tracer la courbe de Elbow
- Le coude, permet de déterminer le périmètre epsilon



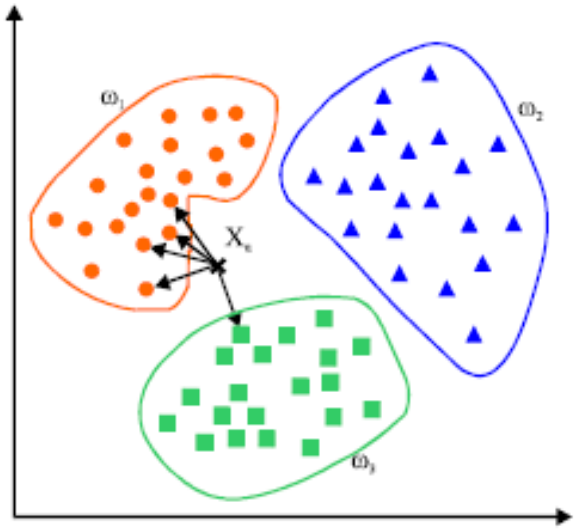
KNN

K Nearest Neighbors

Méthode supervisé sans entrainement, qui va permettre de prédire une valeur Y, en fonction de la proximité des valeurs X, de K observations

- Pour une régression on prendra la moyenne ou médiane des K observations
- Pour une classification, on prendra le mode / modale, c'est-à-dire la modalité la plus fréquente

L'individu est classifié en fonction de l'appartenance de ses K plus proches voisins



Inconvénients

Le jeu d'entrainement doit tenir entièrement en mémoire

CAH

Clustering hiérarchique ascendant

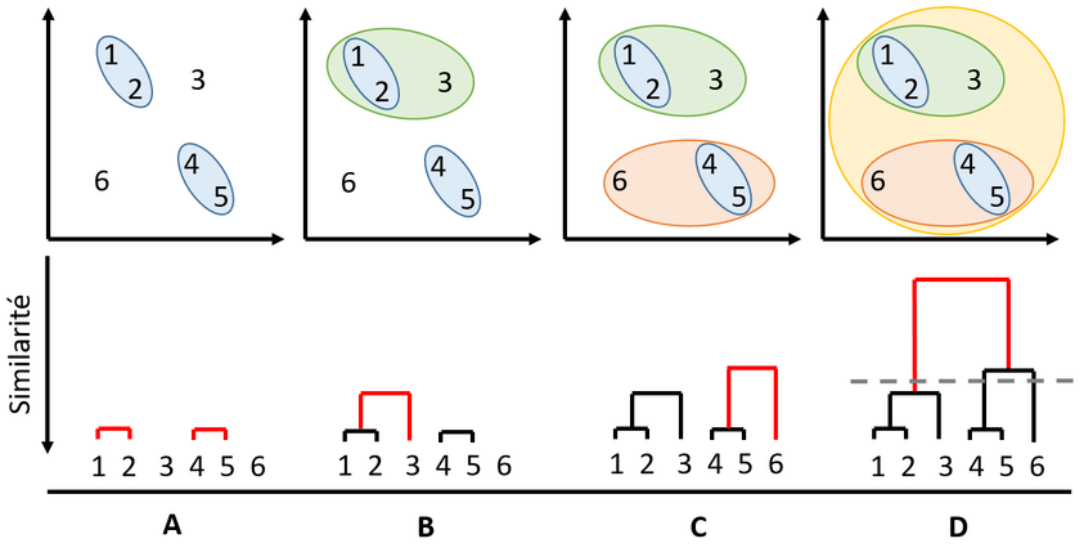
Ce type de clustering va construire incrémentalement des clusters, en regroupant progressivement les individus puis les groupes les plus similaires, jusqu'à n'avoir qu'un cluster global

Les agglomérations se font 2 à 2

Hauteur des branches : niveau de similarité (inertie)
Perte d'inertie interclass, augmentation de l'inertie intraclass

Les sauts d'inertie permettront d'identifier les seuils de classification, pour obtenir le nombre de classe à retenir

2 approches : top-down ou bottom-up



Dendrogramme

GPU

Nvidia

RAPIDS

RAPIDS

- cuPY (numpy)
- cuDF (Pandas)
- cuML (scikit-learn)
- cuDNN (interface pour le Deep Learning)
- xgboost

Accélération de l'entrainement des modèles (efficace sur les calculs matriciels, meilleure parallélisation)
Peut également être utilisé pour de l'inférence (détection d'objets dans les images)

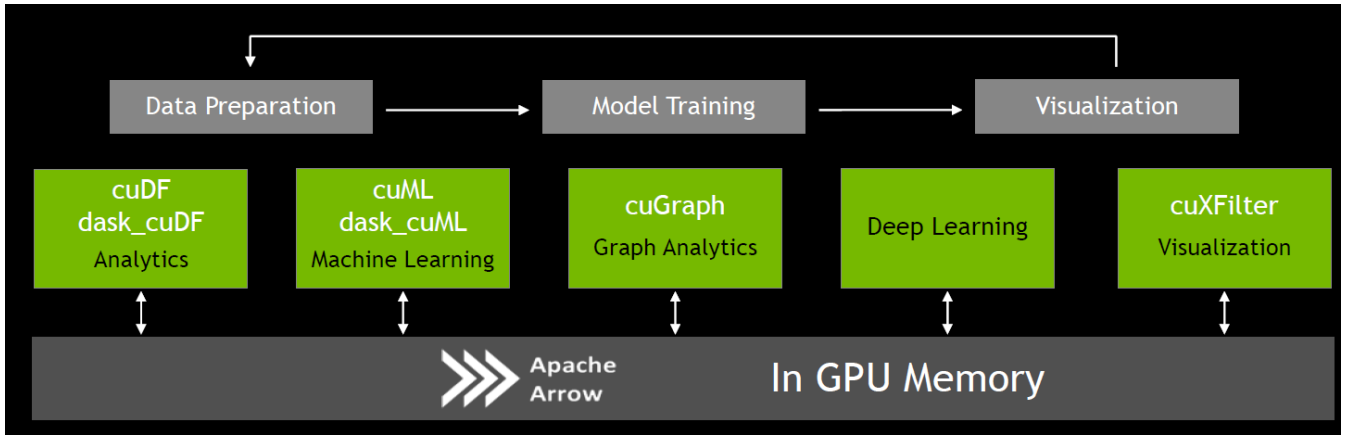
Très utilisé pour les algos de Deep Learning (classification d'images, NLP…)
Et également sur les algos de Machine Learning (Clustering, ACP, XGBoost…), où les gains de temps peuvent être de 4, 5, 13 fois moins…

NVIDIA

Attention à ne pas réaliser trop d'allers-retours entre la mémoire CPU et GPU (sérialisation coûteuse)

RAPIDS

Etapes de transformation



Framework RAPIDS de NVidia, permet de réaliser les différentes étapes de préparation de la donnée, jusqu'à la modélisation

Compatibilité des différentes couches : https://rapids.ai/start.html
(OpenShift) > RedHat > GCC > Python > CUDA > RAPIDS

Exercice

Clustering

Clustering KMeans et DBSCAN

- Savoir différencier les méthodes avec des datasets simples

- Lien du notebook

Clustering KMeans sur GPU

- Comparer les temps d'entrainement entre CPU et GPU
- Réaliser un KMeans sur un dataset réel
- Savoir déterminer la valeur des hyperparamètres

- Lien du notebook
- Lien du dataset

Possibilité d'utiliser BlazingSQL à la place de Google Colab (GPU disponible et librairies déjà installées)