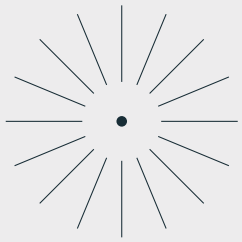
Rapport Fouille de données

Comparaison des Méthodes de Partitionnement et Hiérarchiques de Clustering



Dr M.camara

par
Salif Biaye
Mouhamadou Tidiane Seck

Comparaison des Méthodes de Partitionnement et Hiérarchiques de Clustering

July 14, 2025

Abstract

Ce rapport répond aux exigences du projet USL2 en comparant les méthodes de partitionnement et hiérarchiques de clustering, en mettant l'accent sur leurs différences et leurs applications pratiques avec la bibliothèque Python Scikit-learn. Nous proposons la segmentation des clients pour K-means et le clustering de documents pour le clustering agglomératif, avec des exemples de code et des visualisations des résultats, incluant les étiquettes des clusters.

1 Introduction

Le clustering est une technique d'apprentissage non supervisé utilisée pour regrouper des points de données similaires en fonction de leurs caractéristiques. Ce rapport répond aux exigences du projet USL2 en expliquant les différences entre les méthodes de partitionnement (comme K-means) et les méthodes hiérarchiques (comme le clustering agglomératif) de clustering, et en proposant une application pratique pour chaque catégorie en utilisant la bibliothèque Scikit-learn en Python, avec des résultats affichés sous forme de visualisations et d'étiquettes.

2 Méthodes de Partitionnement

Les méthodes de partitionnement divisent les données en un nombre fixe de clusters non chevauchants. L'algorithme le plus courant est K-means.

2.1 Algorithme K-means

K-means partitionne les données en K clusters en suivant ces étapes :

1. Initialisation aléatoire des K centroïdes.
2. Assignation des points aux clusters en fonction de la distance euclidienne.
3. Mise à jour des centroïdes en calculant la moyenne des points dans chaque cluster.
4. Répétition jusqu'à convergence.

K-means est rapide et efficace, mais il nécessite de spécifier le nombre de clusters à l'avance et suppose des clusters sphériques de taille similaire.

2.2 Application : Segmentation des Clients

Une application courante des méthodes de partitionnement est la **segmentation des clients** en marketing. Cette approche regroupe les clients en fonction de leurs comportements d'achat, tels que la fréquence des achats ou le montant dépensé, pour concevoir des campagnes marketing ciblées.

2.3 Exemple de Code avec Scikit-learn

Voici un exemple d'implémentation de K-means avec Scikit-learn, utilisant des données simulées :

```
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import numpy as np

# Simulation de donn es clients
np.random.seed(42)
client_data = np.random.rand(150, 2) * 100
scaler = StandardScaler()
client_data_scaled = scaler.fit_transform(client_data)

# Application de K-means
kmeans = KMeans(n_clusters=3, random_state=42)
labels = kmeans.fit_predict(client_data_scaled)
print("  tiquettes   des clusters K-means :", labels)
```

Dans cet exemple, les données sont standardisées, et K-means regroupe les clients en trois segments.

2.4 Résultats Affichés

Les résultats de la segmentation des clients incluent les étiquettes des clusters (par exemple, '[0 1 0 2 ...]'), indiquant à quel groupe chaque client appartient. Un graphique de dispersion en 2D (après réduction de dimensionnalité avec PCA) peut également être généré pour visualiser les clusters.

3 Méthodes Hiérarchiques

Les méthodes hiérarchiques construisent une hiérarchie de clusters, soit en fusionnant des clusters plus petits (approche agglomérative), soit en divisant des clusters plus grands (approche divisive). Nous nous concentrons ici sur le clustering agglomératif.

3.1 Clustering Agglomératif

Le clustering agglomératif commence avec chaque point de données comme un cluster individuel et fusionne progressivement les paires de clusters les plus proches en fonction d'une mesure de distance. Le résultat est une structure arborescente, souvent représentée par un dendrogramme.

3.2 Application : Clustering de Documents

Une application typique des méthodes hiérarchiques est le **clustering de documents**. Cette approche regroupe des documents similaires en fonction de leur contenu, facilitant l'organisation de bases de données textuelles.

3.3 Exemple de Code avec Scikit-learn

Voici un exemple d'implémentation du clustering agglomératif avec Scikit-learn, utilisant le dataset Iris :

```
from sklearn.cluster import AgglomerativeClustering
from sklearn.datasets import load_iris
from sklearn.preprocessing import StandardScaler

# Chargement et pr paration des donn es
iris = load_iris()
X = iris.data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Application du clustering agglom ratif
agglo = AgglomerativeClustering(n_clusters=3)
```

```
labels = agglo.fit_predict(X_scaled)
print("  tiquettes   des clusters agglom ratifs :", labels)
```

3.4 Résultats Affichés

Les résultats du clustering de documents incluent les étiquettes des clusters (par exemple, '[0 2 0 2 0 0 1 ...]'), indiquant l'appartenance de chaque document à un cluster. Un graphique de dispersion en 2D (après PCA) et un dendrogramme sont générés pour visualiser la structure des clusters, comme montré dans le projet Python associé.

4 Comparaison des Méthodes

Les méthodes de partitionnement et hiérarchiques diffèrent sur plusieurs aspects, comme résumé dans le tableau suivant :

Caractéristique	**Clustering Hiérarchique**	**Clustering de Partitionnement**
Ordre de Clustering	Structure arborescente	Clusters non chevauchants
Fiabilité	Moins fiable	Plus fiable
Vitesse	Plus lent	Plus rapide
Gestion des Erreurs	Sensible aux erreurs	Plus robuste
Lisibilité	Dendrogramme, facile à interpréter	Clusters moins intuitifs
Stabilité	Relativement instable	Relativement stable

Table 1: Comparaison des méthodes de clustering hiérarchique et de partitionnement

5 Conclusion

Les méthodes de partitionnement, comme K-means, sont adaptées à la segmentation des clients avec des résultats affichés sous forme d'étiquettes et de graphiques de dispersion. Les méthodes hiérarchiques, comme le clustering agglomératif, conviennent au clustering de documents, avec des étiquettes et des visualisations comme des dendrogrammes. Ces résultats sont implémentés avec Scikit-learn, comme démontré dans les exemples.

6 Références

References

[1] Scikit-learn: Machine Learning in Python, https://scikit-learn.org/stable/

[2] Différence entre Clustering Hiérarchique et Non Hiérarchique, https://www.geeksforgeeks.org/difference-between-hierarchical-and-non-hierarchical-clustering/

[3] Customer Segmentation Using K-Means Clustering, https://www.kdnuggets.com/2019/11/customer-segmentation-using-k-means-clustering.html

[4] Hierarchical Clustering with Python and Scikit-learn, https://stackabuse.com/hierarchical-clustering-with-python-and-scikit-learn/