Compare partitional clustering and hierarchical clustering in classification

This paper presents a comparative analysis of partitional and hierarchical clustering algorithms, two cornerstone techniques in unsupervised machine learning aimed at classifying unlabelled data. Specifically, it contrasts the widely-used k-means algorithm, a model of partitional clustering, with agglomerative hierarchical clustering, detailing their methodologies, applications, and performance metrics. The study focuses on their utility in analyzing complex datasets, with a particular emphasis on a dataset consisting of musical tracks. Using measures such as the Sum of Squared Errors (SSE) and average score errors, the analysis provides insights into each method's efficacy in managing data variability and structure. It also examines the impact of different outlier detection and normalization strategies on the clustering results. The findings highlight the conditions under which each clustering approach excels, offering guidance to practitioners on selecting the most appropriate method for specific data challenges. This paper aims to enhance understanding of these fundamental clustering techniques, fostering informed decisions in their application across diverse research and industry contexts.

1. Introduction

In an era where data proliferates at an unprecedented rate, the ability to extract meaningful patterns and insights from vast datasets has become indispensable. Clustering, a core technique in unsupervised machine learning, serves as a pivotal tool for unravelling the complexities of large datasets without predefined labels. This paper delves into two prevalent clustering techniques: partitional clustering, exemplified by k-means, and agglomerative hierarchical clustering. Our comparative analysis aims to illuminate their distinct methodologies and applicability to various data-driven challenges.

Partitional clustering, particularly k-means, partitions the dataset into a predefined number of clusters by optimizing the centroids of these clusters. On the other hand, agglomerative hierarchical clustering builds nested clusters through a stepwise amalgamation of data points or existing clusters. This paper evaluates these methods using metrics such as the Sum of Squared Errors (SSE) and average score errors, providing a nuanced understanding of their performance and suitability for specific types of problems.

Through empirical analysis, this study also explores the significant factors influencing the selection of clustering methods, particularly in the context of musical data. The ultimate goal is to furnish a comprehensive understanding of these algorithms' operational mechanisms and their effectiveness in real-world applications, guiding users in choosing the appropriate clustering strategy for their specific needs.

This introduction sets the stage for a detailed discussion on the practical applications of these clustering techniques, their strengths and weaknesses, and the scenarios where one may be preferred over the other. By the end of this paper, readers will gain not only foundational knowledge but also practical insights into implementing k-means and hierarchical clustering for efficient data analysis and decision-making.

2. Dataset

The dataset was extracted from Kaggle (https://www.kaggle.com/geomack/spotifyclassification) Each row of the initial dataset represents a song; there are 16 columns. Thirteen of those columns are song attributes, one is the song title, one is the artist, and the column called "target" is the song label. Here are the 13 track attributes: acousticness, danceability, duration ms, energy, instrumentalness, key, liveness, loudness, mode, speechiness, tempo, time signature, valence. In this paper, we have used PCA algorithms to reduce the feature space from 13 dimensions to 2 dimensions to reduce the complexity. The scatter plot Fig.1 shows what the data looks like after dimensionality reduction and visualization.

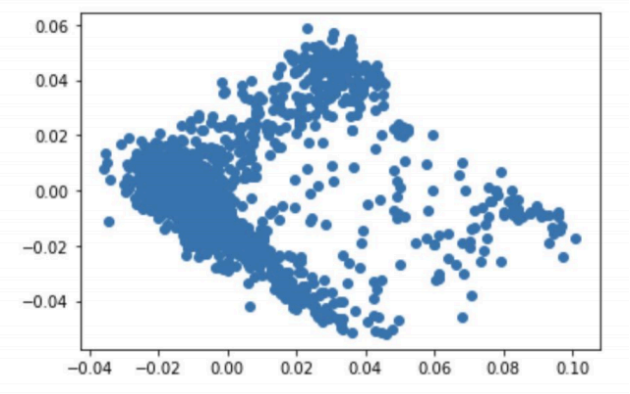

Fig.1 visualized data

3. Method

A. Outliers

A.1.un

K-means clustering and hierarchical clustering have their advantages, but K- K-means are more easily affected by outliers. So we need to minimize the outlier in the dataset. IQR is a mathematical way to take outliers out. standard deviation is one of the mathematical ways to remove outliers

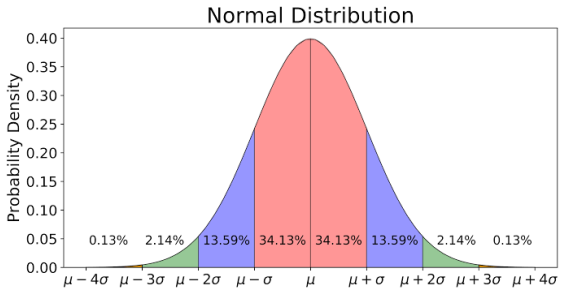

Fig.2 standard deviation percentage[2]

IQR is calculated as the difference between the first and third quartile which is the value from $\mu - \sigma$ to $\mu + \sigma$ in Fig.2. To determine outliers. At first, find the number that is smaller than the first quartile($\mu - \sigma$) minus $1.5 \times IQR$ or the bigger number than the number in the third quartile ($\mu + \sigma$) and add $1.5 \times IQR$. [3]

A.2.Local outlier factor[5]

A density unsupervised-based detection, it finds the local outlier by comparing a data point's local density to its neighbors' local density. neighbour's number represents as number k, it is a hyperparameter that is decided by the practitioner. When the dataset does not have the same density in each label then this method will be useful.

The local outlier density is calculated by the average reachability distance and the local reachability distance. Reachability distance is the distance from the original data point to other data points and the average reachability distance is calculated in:

$$Average\ RD\ = \frac{1}{k}\sum_{k} Max[Max[distance\ to\ its\ neighbor],\ distance\ from\ one\ point\ original\ data.].$$

local reachability density is the inverse of average reachability distance, with higher LRD means the point is far away from the closest cluster.

$$Average\ LRD\ = \frac{1}{Average\ RD}$$

The local outlier factor is the average local reachability distance of other data points in the dataset divided by the local reachability distance of this point.

$$LOF\ _k(x)\ = \frac{\sum_{n=N_k(x)} LRD\ _k(n)}{k \times LRD\ _k(x)}$$

After calculation, it will find the local outlier ratio of each point, and the user will identify the final outlier ratio they want. Usually, people will likely have a local outlier factor bigger than 1 which is the value we will use in discussing this dataset.[6]

B. Normalize

B.1.Feature scaling

The normalization is used to make all the attributes have the same usability in separating clusters, when data have different ratios of value it will mislead the machine to learn.
Feature scaling is one of the most common ways to normalize the data, it turns the maximum value of one attribute into 1 and the minimum value into 0 and finding the rest of the value will turn into the

value between 1-0 by ratio. This normalization is sensitive to outliers which will break the ratio of the whole dataset.
For example, if we want to transform k in attribute x then we can use the formula below:
$x_k = \frac{Value-Min}{Max-Min}$[7]

C. Algorithm

C.1 Partitional clustering

The most popular algorithm in clustering is also called the iterative relocation algorithm. In each cluster, it will find its data point by calculating the distance between points and the centroid. It will minimize the error by resetting the centroid of each cluster. Finding the actual centroid is an NP-hard problem which can not be solved by exhaustive methods.[9]

C.1.1K-means clustering

K-means is the most common partitional clustering algorithm. Each object in K-means will be a vector placed in a n-dimensional space with n attributes contained in a dataset. At the beginning of the process, it will create the K number of centroids with different numbers of data points. In each separating, K- means will start the cycle of evaluating the similarity of points between centroids. This uses the Euclidean distance and puts them in different clusters(1). The whole process will stop when all clusters have a minimum value of dissimilarity.[4]

$$Distance = \sum_{k=1}^{k} \left| \vec{p}_k - \vec{centroid} \right| \quad (1)$$

K is the number of datapoint in a cluster
$\left| \vec{p}_k - \vec{centroid} \right|$ is to find the magnitude of distance between data point and centroid.

C.1.2 K-means++

The improvement of K-means's randomness centroid, it has the same method as K-means but this algorithm will calculate the distance between initial centroids and put the centroid as far as possible. That means, there will only be one centroid chosen by random, after this we have to calculate the distance between different data points and choose the furthest as the next centroid. There will be a better chance for data points to separate into a further cluster, it can provide the actual group separate into multiple clusters. [8]

D.1 Agglomerative hierarchical clustering

Agglomerative hierarchical clustering is using data points within clusters to evaluate the similarity between objects. First, we will calculate the distance between all data points by Euclidean distance as K-means(1), aggregating data points with the least distance together.

D.1.1 Group link

By using the group link, after merging data points we set the distance as the average value(2) among data points. Complete links are too sensitive to outliers and single links can not identify spherical clusters well, so people will use the compromise between these two methods, which is group link.

$$d(C_i, C_j): \sum_{a \epsilon C_i, b \epsilon C_j} \frac{d(a,b)}{|C_i||C_j|} \quad (2)[5]$$

D.1.2 Single link

Using a single link we only take the minimum value (3) after we merge the closest set. A single link is a good way of finding the cluster when all data points are mixed, but it can be easily misled when that is a shape cluster.

$d(C_i, C_j): min(a \epsilon C_i, b \epsilon C_j)\ d(a, b)$ (3)[5]

D.1.3 Complete link

a complete link is opposite to a single link, it takes the maximum value (4) of the set we merge. but it is sensitive to outliers, when we merge the point the whole distance matrix will change, if the point is an outlier then the whole process can be ruined by this little step.

$d(C_i, C_j): max(a \epsilon C_i, b \epsilon C_j)\ d(a, b)$ (4)[5]

E.1 Mean squared error

Mean squared error in machine learning is a way to evaluate unsupervised learning in finding the square distance between the actual and estimated values. In clustering, people find the squared error by the square distance of the centroid and other points in this cluster. (5)

$$MSE = \frac{1}{k}\sum_{n=1}^{k} (\vec{p} - \vec{centroid})^2 \text{(5)}$$

In this test, we will also find the average of each cluster's mean squared error so we can evaluate each method with a different number of clusters.

E.2 Silhouette score

Silhouette score is one of the ways to evaluate how data has grouped the cluster. The score depends on two parts: separation and cohesion of the cluster.

$$Silhouette\ score = \frac{b-a}{Max\{a,b\}}$$

a and b are the data points in two different clusters which have a minimum distance to their centroid, which evaluates the distance of two of the clusters so we can confirm how well we separate the group. Divided by the maximum distance of a and b, so we can find the ratio of its pros and cons. Sometimes people will also use:

$$if\ a > b,\ Silhouette\ score = \frac{b}{a} - 1$$

$$if\ a = b,\ Silhouette\ score = 0$$

$$if\ a < b,\ Silhouette\ score = 1 - \frac{a}{b}$$ [10]

When the silhouette scores bigger than 0 it means it separates well, when the silhouette scores equal to 0 that means those clusters have touched each other at the side, when the silhouette scores smaller than 0 that means those two clusters have some data point that mix.[10]

4. Result

Outlier detection:

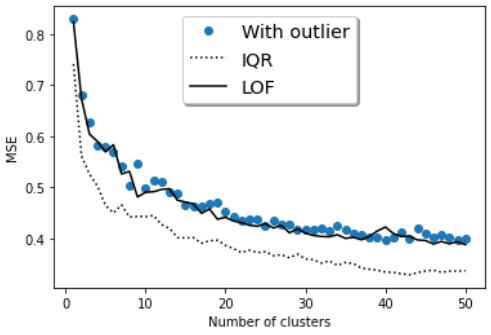

Fig.3 Outlier detection comparison with hierarchical clustering

In this graph, we use agglomerative hierarchical clustering's single link as a method to check the error that is created by different outlier detection. In this data, LOF is unusable compared with IQR, although it decreases the squared error sometimes it is not as efficient as IQR. IQR works well in this process, with the deletion of outliers the average squared error has decreased a lot as shown in Fig.3.

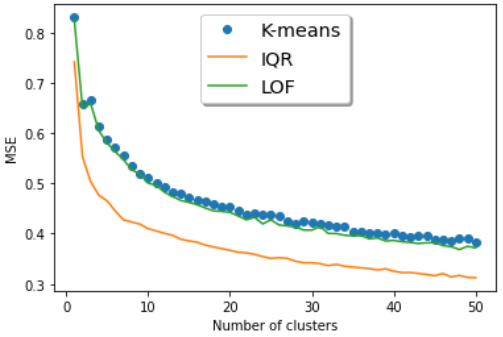

Fig.4 Outlier detection comparison with partitional clustering

Fig.4 shows the K-means average squared error using the Outlier detection. In conclusion, LOF can't help a lot in this dataset, its uncertain value of the ratio is not able to be calculated, which means although we deleted the data point that had a standard evaluation LOF value(which is 1.), the data would not change a lot. IQR has minimized the error of clustering. IQR is very useful in partitional clustering since it will delete the point that is far away from another point which makes the centroid in the cluster more accurate.

Partitional clustering:

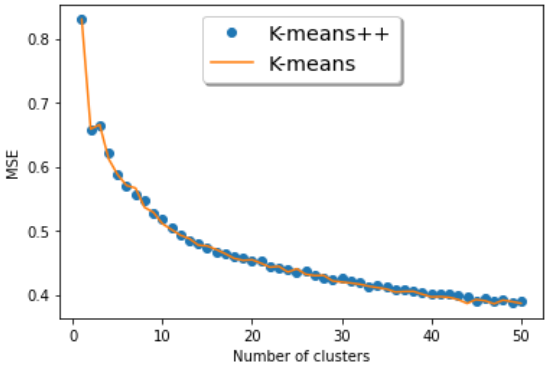

Fig.5 Error comparison between two different partitional clustering

Fig.5 shows the error comparison between K-means++ and K-means, they have some output of cluster, and after we tested over ten times the results are still the same which means this dataset has less probability of choosing a wrong centroid, or because the data points are very close to each other which cause the final centroid that decided by K-means and K-means++ are similar.

Agglomerative hierarchical clustering:

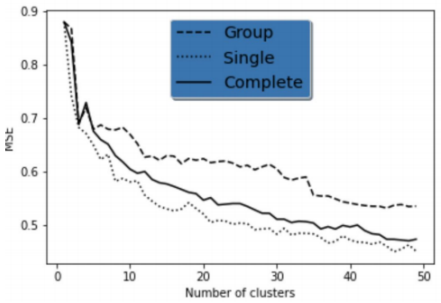

Fig.6 Error comparison between different links of agglomerative hierarchical clustering

In Fig.6, it compares different links in agglomerative clustering. As a result, a single link will be a more useful method in this dataset, since all data points are mixed (which is shown in Fig.1) and a single link is good at separating this kind of data.

Agglomerative hierarchical clustering and Partitional clustering in MSE:

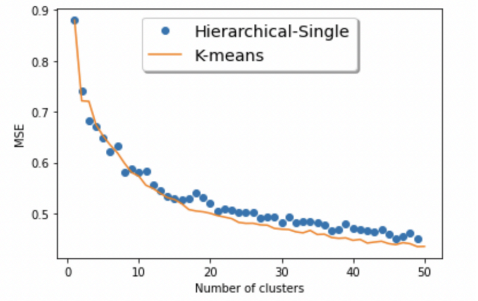

Fig.7 Average squared error between Agglomerative hierarchical clustering and Partitional clustering(With IQR outlier detection)

Average squared error in Fig.7, we can see hierarchical clustering and partitional clustering have similar values in the first ten numbers of clusters but when the number of clusters increases, partitional clustering will perform better. Which dataset's shape is also affected by this part, which means partitional clustering may have better accuracy than single linkage clustering in mixing data.

Agglomerative hierarchical clustering and Partitional clustering in silhouette score:

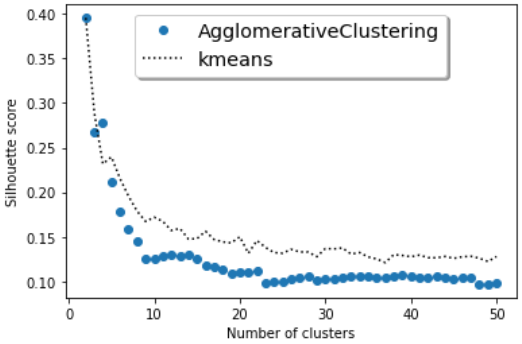

Fig.8 Silhouette score between agglomerative hierarchical clustering and Partitional clustering(With IQR outlier detection)

Fig.8 shows silhouette scores of different clustering algorithms, K-means and single linkage clustering. Agglomerative clustering in the first three numbers of clusters has a better score than K-means but K-means separate better after.

The graph has more difference than the average squared error since it finds both separation and cohesion of the cluster. That means, K-means clustering is not only good at cohesion which shows as average squared error but it might also be good at separating and that is why it got more silhouette scores than agglomerative clustering when the number of clusters is bigger than three.
At the third point in Fig.8, agglomerative clustering and K-means it rebound the silhouette score which I think will be a better number of clusters to use because it has a better score than another number of clusters which means it has the best separate and cohesion than other number (except one, but that is not clustering).

After we separated into three clusters we found out the radar chart below shows the centroid's characteristic of each cluster:

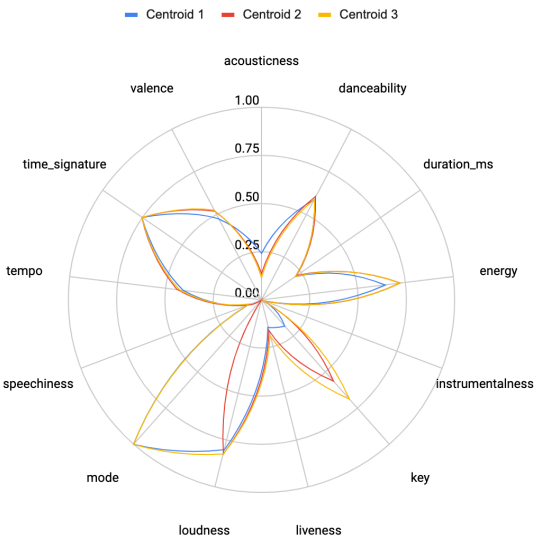

Fig.9 Radar chart of three centroid

The biggest differences are mode, key, energy and acousticness. Centroid 2 has no mode and lower key than Centroid 3. Centroid 1 has the lowest key and the highest acousticness. Centroid 3 has the highest energy and key.

All in all, after those tests, we can find that partitional clustering might have better usability in separate music types, and IQR can help more in finding the music data point's outlier. In music, the attribute: mode, key, energy and acousticness may help separate the data point better.

5. Future investigation

Clustering is good at isolating and helps identify user groups and music types. Each of these can be used to assign customer interests and help find future recommendation types. Some recommendation systems also compare customer interests with product similarities. According to what we have done in the experiments Fig.9 we can find that mode, key, energy and acousticness are the four factors that may be indicated as the main factors of analysis of the type of music.

Reference

[1]Lakshmanan, S. (2020, May 29). *How, when, and why should you normalize / standardize / rescale your data?* Towards AI - The Best of Tech, Science, and Engineering,https://towardsai.net/p/data-science/how-when-and-why-should-you-normalize-standardize-rescale-your-data-3f083def38ff

[2]Chaudhary, S. (2020, August 26). *Why "1.5" in iqr method of Outlier Detection?* Medium,https://towardsdatascience.com/why-1-5-in-iqr-method-of-outlier-detection-5d07fdc82097

[3]Taylor, C. (2019, May 22). *How do we determine outliers in statistics?* ThoughtCo. https://www.thoughtco.com/what-is-an-outlier-3126227

[4]Ansari, Z., Azeem, M. F., Ahmed, W., & Babu, A. V. (2011, June). Quantitative Evaluation of Performance and Validity Indices for Clustering the Web Navigational Sessions. Retrieved September 10, 2021, https://arxiv.org/pdf/1507.03340.pdf

[5]Wenig, P. (2019, August 1). *Local outlier factor for anomaly detection*. Medium. Retrieved September 21, 2021, https://towardsdatascience.com/local-outlier-factor-for-anomaly-detection-cc0c770d2ebe

[6]Wenig, P. (2019, August 1). *Local outlier factor for anomaly detection*. Medium. Retrieved September 21, 2021, https://towardsdatascience.com/local-outlier-factor-for-anomaly-detection-cc0c770d2ebe

***[7]Normalization*. Codecademy. (n.d.). Retrieved September 21, 2021, https://www.codecademy.com/articles/normalization**

[8]Ali, M. (2021, September 11). *Implementing k-means clustering WITH K-Means++ initialization in Python.* Medium. Retrieved September 22, 2021, https://medium.com/geekculture/implementing-k-means-clustering-with-k-means-initialization-in-python-7ca5a859d63a

[9]Ayramo, S., & Karkkainen, T. (2006, January). Introduction to partitioning-based clustering methods with a robust example. Retrieved September 22, 2021, http://users.jyu.fi/~samiayr/pdf/introtoclustering_report.pdf

[10]*聚类评估算法-轮廓系数（Silhouette coefficient***）. 聚类评估算法-轮廓系数（Silhouette Coefficient）_wangxiaopeng0329的博客-CSDN博客. (2016, December 9). Retrieved September 22, 2021, https://blog.csdn.net/wangxiaopeng0329/article/details/**53542606