Proceedings of the 4th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/79/20241631

Analysis of clustering algorithms in Iris and breast cancer datasets

Jiasheng Chen[1,6,*], Changyou Jin[2,7], Hongyu Wang[3,8], Zixuan Huang[4,9], Jingxing Liang[5,10]

[1]Information Technology, Shanghai Ocean University, Shanghai, 201306, China
[2]University of California, San Diego, San Diego, 92092, United States
[3]Sauder School of Business, University of British Columbia, Vancouver, V6T 1Z4, Canada
[4]Northwood High School, Irvine, CA, 92620, United States
[5]Milliken Mills High School, Markham, ON L3R 9S5, Canada

[6]2662930998@qq.com
[7]Changyoukin@gmail.com
[8]whyaes812@gmail.com
[9]emilyh9858@gmail.com
[10]Liangjx4@gmail.com
*corresponding author

Abstract. In the contemporary era of data-driven processes, addressing the challenge of processing vast volumes of data has become a pressing concern. With the rapid advancement of computer science and information technology, data processing efficiency has significantly improved. Within this expansive domain, three prominent clustering techniques—namely, K-Means clustering, spectral clustering, and Density-based spatial clustering of applications with noise (DBSCAN)—have assumed pivotal roles due to their versatility and effectiveness. This essay embarks on a systematic examination of these three methods, deconstructing their fundamental principles and navigating through their practical applications.

Keywords: clustering, K-Means, spectral clustering, DBSCAN.

1. Introduction

In today's data-driven age, how to process huge amounts of data has become a severe problem. With the rapid growth of computer science and information technology, the efficiency of data processing has increased significantly. Within the expansive realm, three standout clustering methods - k-means clustering (K-Means), spectral clustering and Density-based spatial clustering of applications with noise (DBSCAN) have assumed pivotal roles due to their versatility and efficiency. This essay embarks on a journey to test these three methods, dissecting their fundamental principles and navigating through their practical applications.

In Section 2.1, K-Means, n observations are divided into k clusters using a vector quantization technique first used in signal processing. Each observation is assigned to the cluster that is closest to it,

DOI: 10.54254/2755-2721/79/20241631

acting as a prototype for the cluster. In Section 2.2, spectral clustering, a method that uses the data's similarity matrix's spectrum to reduce the number of dimensions before clustering the data into smaller groups. Lastly, in Section 2.3, DBSCAN, a density-based clustering non-parametric algorithm that can group together points that are closely packed together, marking them as noise that lie alone in low-density regions. In Section 3.1, an overview of the datasets is provided. Then, in Section 3.2, the Metrics: Purity and Normalized mutual information, a common measurement of mutual dependence between 2 random variables is introduced, which can test our clustering performance. In Section 3.3.1 and Section 3.3.2, there are detailed experiment results of three clustering methods in two datasets, respectively.

The comparative analysis will uncover the distinct efficiency and performance of K-Means, spectral clustering, and DBSCAN in two different datasets, which can shed light on scenarios where each method excels.

The exploration commences in Section 2.1 with an examination of K-Means, a method rooted in vector quantization, originally emerging from signal processing. Its primary objective is to partition a set of n observations into clusters, where each observation is assigned to the cluster with the nearest distance, effectively serving as a prototype for the cluster. Section 2.2 discusses spectral clustering, a method that leverages the spectral characteristics of the similarity matrix of the data to execute dimensionality reduction before clustering in a lower-dimensional space. Finally, Section 2.3 introduces DBSCAN, which is proficient in grouping points that are closely packed together while marking isolated points in low-density regions as noise.

Section 3.1 provides an overview of the datasets, and in Section 3.2, the metrics used for evaluation are introduced: Purity and Normalized mutual information. These metrics represent a common means of assessing the mutual dependence between two random variables, enabling the evaluation of clustering performance.

In Section 3.3.1 and Section 3.3.2, the detailed experimental results of the three clustering methods on two distinct datasets are presented. These results serve as the foundation for our comparative analysis, revealing the distinctive efficiency and performance characteristics of K-Means, spectral clustering, and DBSCAN in the context of these datasets. This analysis sheds light on scenarios where each of these methods excels.

2. Methodology

2.1. K-Means

K-Means is a partitioning algorithm that divides data points into numerous non-overlapping areas, which unveils hidden patterns in selected datasets. The primary objective of applying K-Means is to investigate whether there are clusters hidden in the data sample. Although the randomization of centroids results in unexpected convergence, K-Means is still considered a powerful data mining tool [10].

Conventionally, the data will be divided into k clusters. Grouping a set of data points into k clusters, where each data point belongs to the cluster with the nearest mean, is the main objective of K-Means clustering. In other words, it aims to find natural groupings in data in a way that makes data points in the same cluster more comparable to one another than to data points outside of it.

There are three basic elements in the K-Means algorithm. The first one is Centroid. At the core of K-Means is the concept of a centroid, which is the mean or average position of all the data points in a cluster. Each cluster has its centroid, and the proximity of data points to the centroids determines the cluster assignments. The second one is a number of clusters k. represents the number of clusters that you want to form in the dataset. The choice of k is a crucial decision in K-Means clustering, as it directly affects the quality and interpretability of the resulting clusters. The third one is Distance Metric. K-Means uses a distance metric (usually Euclidean distance) to measure the similarity or dissimilarity between data points. Each data point's separation from the cluster centroids is computed.

Choosing k data points at random to serve as the initial centroids is the first step. These initial centroids can greatly impact the final clustering results, so different initialization methods have been proposed.

DOI: 10.54254/2755-2721/79/20241631

Determine the distance between each of the k centroids for each data point, then allocate the data point to the cluster that is connected to the closest centroid. The coordinate of point x is (x_1, y_1), while that of point y is (x_2, y_2)

The distance formula is based on the Euclidean formula:

$$\text{Distance} = \sqrt{((x_1 - x_2)^2 + \left((y_1 - y_2)\right)^2)}$$

Recalculate the centroids of each cluster by taking the mean of all data points assigned to that cluster.

Then, it comes to the second round. The processes of calculating the distance to each of the k centroids and recalculating the centroids of each cluster are repeated iteratively until one of the stopping conditions is met. Common stopping conditions include a maximum number of iterations, minimal change in cluster assignments, or achieving a predefined convergence threshold.

Once all points are assigned to clusters, the pattern of distribution is recorded, and next round is implemented until the change in pattern is less than an assumed critical value.

2.2. Spectral Clustering

The process of using the spectral clustering algorithm involves first constructing an affinity matrix that describes the similarities between data points from a sample dataset, computing the matrix's eigenvalues and eigenvectors, and then choosing the relevant eigenvectors to cluster the various data points [5]. Due to its straightforward implementation and promising results in numerous graph-based clustering applications, spectral clustering has grown in popularity. It can be solved quickly using conventional linear algebra software, and it frequently performs better than more established techniques like the k-means algorithm [1].

The initial phase in the spectral clustering algorithm necessitates the establishment of an affinity matrix, which encapsulates the similarities among the data points in a given dataset. This matrix is typically computed employing the Gaussian (RBF) kernel or by determining the k-nearest neighbors. In the k-nearest neighbors approach, an edge is directed from vertex u to vertex v only if v resides among the k-nearest neighbors of u. This not only results in a weighted graph but also a directed one due to the non-mutuality of nearest-neighbor relationships. One of two approaches could be used to turn this into an undirected graph: either direct an edge reciprocally while whether the vertex is one of the other's k-nearest neighbors or only direct an edge reciprocally if both vertices are among the other's k-nearest neighbors [4].

Upon constructing the affinity matrix, the next stride involves computing the Laplacian matrix. Initially, the Degree matrix (a diagonal matrix representing the degree of each node) is constructed. Subsequently, the unnormalized graph Laplacian matrix is calculated utilizing the equation $L = D - W$, where W symbolizes the weighted adjacency matrix, and D is the Degree Matrix. Further, two normalized graph Laplacians, $L_{sym} = D^{-\frac{1}{2}}LD^{-\frac{1}{2}}$ and $L_{rw} = D^{-1}L$, where L_{sym} is the symmetric normalized graph and L_{rw} is the random walk normalized graph, can also be derived [9].

The ensuing step revolves around the computation of the eigenvectors of the Laplacian matrix. In the case of unnormalized spectral clustering, the eigenvectors $\phi_1, \dots, \phi_k$, corresponding to the smallest eigenvalues λ_l (where $1 \leq l \leq n$) are utilized. The matrix $\phi \in R^{n \times k}$, composed of the first k eigenvectors of L_{sym}, undertakes a nonlinear transformation of the original data points from R^n to R^k. In essence, standardize the rows to norm 1 and create the matrix $T \in R^{n \times k}$ from ϕ. The vector $y_i \in R^k$ represents the i-th row of T for $i = 1, \dots, n$. Upon obtaining the eigenvectors, each data point is represented in a reduced-dimensional space. The data points are then divided into clusters $C_1, \dots, C_k$ using a conventional clustering technique, like k-means, in the following phase [7].

2.3. DBSCAN

DBSCAN(density-based spatial clustering of applications with noise [6], is among these algorithms for clustering. It can be applied to group data points together in locations with a high sample count by clustering them according to density. This makes it particularly helpful for clustering in noisy

DOI: 10.54254/2755-2721/79/20241631

environments because DBSCAN can identify noisy points in the dataset that can be removed if desired. Obviously, DBSCAN is widely used in various applications, including but not limited to: Image segmentation, Anomaly detection. It is also necessary to notice that DBSCAN operates based on the following principles: Firstly, DBSCAN identifies clusters based on the density of data points. It starts with an arbitrary data point and expands the cluster by connecting it to its dense neighbors. Until no more core points can be added to the cluster, this procedure keeps going. Secondly, DBSCAN is robust to noise and can effectively handle outliers. Noise points are not assigned to any cluster, allowing the algorithm to isolate and ignore them. Thirdly, unlike k-Means, which assumes spherical clusters, DBSCAN can discover clusters of various shapes, making it suitable for complex datasets.

Three categories are used by DBSCAN to classify data points: core points, boundary points, and noise points. A core point is a data point that has at least a specified number of neighboring data points within a defined distance ε (epsilon). Clusters revolve around these central points. When data points are within ε of a core point but do not have enough core points in their ε-neighborhood to be classified as core points, they are referred to as border points. Though not at the center of the cluster, they are a part of it. Data points that fall into neither the core nor border categories are known as noise points. They do not belong to any cluster and are considered outliers.

The first step is initializing the algorithm with the dataset, ε (epsilon), and minPts parameters. Epsilon (ε) is a user-defined parameter that determines the maximum separation between two data points for them to be considered neighbors. It influences the density of clusters. MinPts is another user-defined parameter that specifies the bare minimum of data points that must be located within ε of a core point in order for it to be classified as a cluster. Authors use a collection of n data points, typically denoted as $D = \{p1, p2, \ldots, pn\}$ [2] as dataset, while the epsilon and MinPts are positive values. Then randomly select a data point X that has not been visited.

The second step is calculating the group of data points $N(p)$ that fall within the $\varepsilon - neighborhood$ of X, where it can define the $\varepsilon - neighborhood$ of N as:

$$N_\varepsilon(x) = \{y \in X : dist(x, y) \leq \varepsilon\}$$

If the selected point is a core point (has at least MinPts neighbors within ε distance), which means $NP \geq MinPts$, then create a new cluster and expand it by adding all reachable core points to the cluster. Density-reachability, also known as directly density-reachable, is a concept used in DBSCAN to determine whether one data point can be reached directly from another data point within a specified distance ε (epsilon) and with a minimum number of neighboring data points, referred to as MinPts. Moreover, Density-connectivity, also known as density-reachable, is a more generalized concept in DBSCAN that extends the idea of density-reachability. It allows data points to be connected in a chain-like manner through other data points, even if they are not directly reachable, where it can define x and y as:

$$x, y \in X;\ x \in X_c\ , y \in N_\varepsilon(x)$$

The third step is to mark all visited points as part of the cluster, then repeat step 1 and step 2 until all data points have been visited. Any unvisited points are considered noise, where it can define the noise as:

$$x \in X_{nc};\ \exists y \in X;\ y \in N_\varepsilon(x) \bigcap X_c$$

3. Experiments

3.1. Dataset

In this study, two datasets are used separately, namely, the Iris Flower Dataset and Breast Cancer Wisconsin Dataset.

The first dataset is the Iris Flower Dataset [3], a seminal repository in the field of machine learning and medical research, offering a resourceful measurement of 150 iris flowers from three different species.

DOI: 10.54254/2755-2721/79/20241631

The dataset provides a comprehensive insight into the factors influencing different characteristics of flowers.

The second dataset is the Wholesale Customers dataset [8], sourced from the UCI Machine Learning Repository, a comprehensive collection of information pertaining to clients of a wholesale distributor. This dataset offers valuable insights into the annual spending habits of various clients across different product categories, measured in monetary units (m.u.). Each entry in the dataset represents a client's consumption of that product.

The data within this dataset is essential for businesses in the wholesale and retail sectors, as it enables them to gain a deeper understanding of their clients' preferences and requirements. By studying the spending patterns of wholesale customers, businesses can make informed decisions regarding product selection, inventory management, and pricing strategies to enhance customer satisfaction and overall profitability.

Moreover, it is worth noting that this dataset was last donated on March 30, 2014, indicating that it reflects historical spending data up to that date. Analyzing this dataset can provide valuable insights into past customer behavior, which can be used to inform future business strategies and decisions.

3.2. Metrics

Normalized mutual information is a measurement of mutual dependence between 2 random variables. The process of calculating normalized mutual information is to calculate the mutual information then normalize it by using the entropy of the 2 random variables. To calculate the mutual information, the following formula will be used:

$$MI(A,B) = D_{KL}(P(A,B)\big|\big|P(A)\otimes P(B)\big),$$

which A,B is for 2 random variables; MI is for mutual information; D_{KL} is Kullback–Leibler divergence, which is an indicator used to measure the similarity of probability distributions; $P(A,B)$ is the joint distribution of A and B, $P(A)$ and $P(B)$ are marginal distributions of A and B. To normalize it, the following formulas will be used:

$$H(A) = -\Sigma P(A)\cdot log_2 P(A)$$

$$NMI(A,B) = \frac{MI(A,B)}{H(A)\cdot H(B)}$$

which $H(A)$ is entropy of A and NMI is normalized mutual information.

Purity is a metric used to evaluate clustering quality, which measures whether the data points within each cluster in a clustering result belong to the same category. For each cluster, find the most frequent category in that cluster. Assign all data points to this most frequent category. Calculate the accuracy of all clusters and take the average. The value of the average is the purity.

3.3. Experiments based on dataset

3.3.1. Iris data. The first dataset is Iris dataset [3], in this experiment, actually, the first four columns in this dataset are taken to draw the data distribution map, which can demonstrate the features of the iris clearly. After loading the iris dataset, scatter plot is drawn illustrating the relationship between each data sample's sepal length and width. Additionally, there are points that have been color-coded based on flower clustering. Figure 1 is the clustering results by three distinct methods in iris dataset. Picture A is the original data distribution map. Picture B is clustering prediction after DBSCAN which we set epsilon as 0.4 and min_samples as 4. Picture C is clustering prediction after K-Means and picture D is the clustering prediction after spectral clustering.

DOI: 10.54254/2755-2721/79/20241631

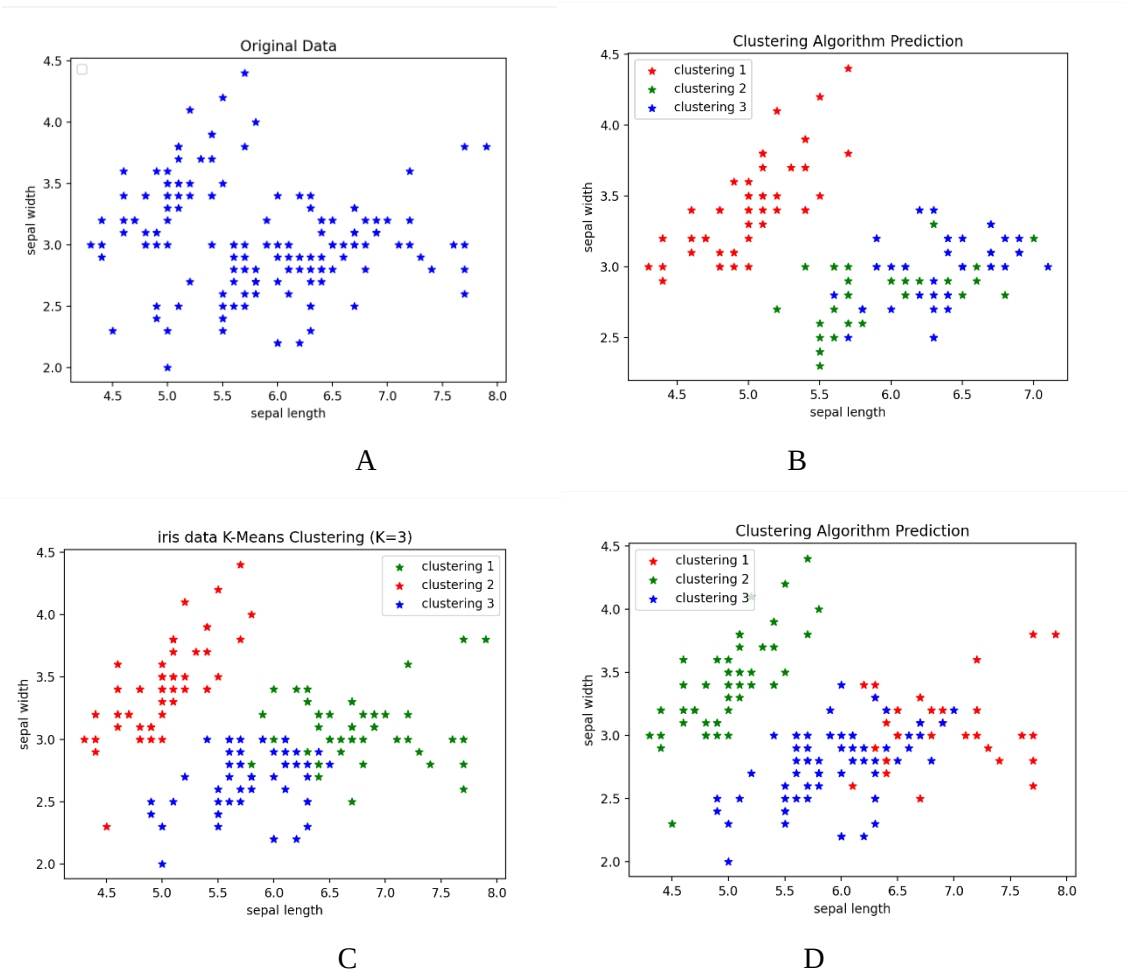

Figure 1. Clustering results in Iris Dataset

The purity and NMI between the clustering prediction of DBSCAN and the original data are 0.68 and 0.69, the purity and NMI between the clustering prediction of K-Means and the original data are 0.89 and 0.76 and the Spectral clustering are 0.9 and 0.79, which means Spectral clustering does a better prediction on this dataset. Table 1 shows the result of the purity and NMI among three methods:

Table 1. Purity and NMI on the Iris Dataset for three distinct clustering algorithms

	DBSCAN	K-Means	Spectral clustering
Purity	0.68	0.89	0.9
Normalized mutual information (NMI)	0.69	0.76	0.79

3.3.2. Breast Cancer Wisconsin (Diagnostic). The second dataset is the Breast Cancer Wisconsin [8]. The main objective is to identify potential relationships among all the variables and features in the dataset. Figure 2 is the clustering results by three distinct methods in Breast Cancer Wisconsin dataset. Picture E is the original data distribution map. Picture F is clustering prediction after DBSCAN which we set epsilon as 40 and min_samples as 5. Picture G is clustering prediction after K-Means and picture H is the clustering prediction after spectral clustering.

DOI: 10.54254/2755-2721/79/20241631

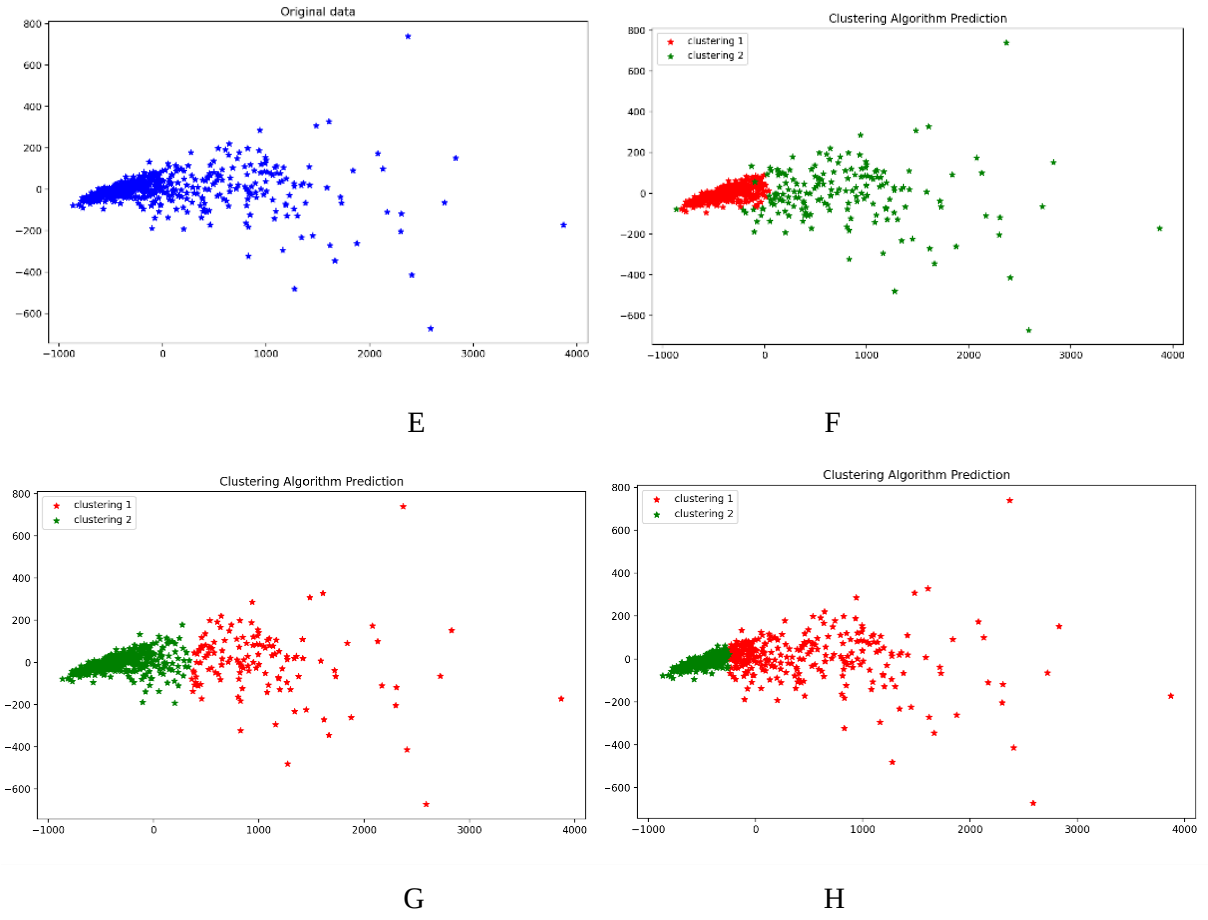

Figure 2. Clustering results in Breast Cancer Dataset

The purity and NMI between the clustering prediction of DBSCAN and the original data are 0.89 and 0.57, the purity and NMI between the clustering prediction of K-Means and the original data are 9 and 0.46 and the Spectral clustering are 0.83 and 0.42, which means DBSCAN does a better prediction on this dataset. Table 2 shows the result of the purity and NMI among three methods:

Table 2. Purity and NMI for three different clustering methods on the Breast Cancer Dataset

	DBSCAN	K-Means	Spectral clustering
Purity	0.89	0.9	0.83
Normalized mutual information (NMI)	0.57	0.46	0.42

4. Conclusion

In conclusion, the ever-expanding realm of computer science and information technology has ushered in an era of unprecedented data accumulation. As the challenges of data collection have evolved into the complexities of data processing, clustering methods have become indispensable tools for knowledge discovery and data mining, especially in the context of big data analysis. In this paper, there is a comprehensive exploration of three prominent clustering methods: K-Means clustering, spectral clustering, and DBSCAN, each with its own unique approach and strengths.

DOI: 10.54254/2755-2721/79/20241631

Through practical application in two diverse datasets, namely the Iris flower and Wholesale Customer datasets, the intricacies of these clustering techniques are revealed. Our journey has provided a deeper understanding of their fundamental principles, performance, and versatility.

The findings of our comparative analysis have shed light on the distinct strengths and limitations of each method in different data scenarios. K-Means, spectral clustering, and DBSCAN have proven to be valuable tools for addressing clustering challenges arising from large datasets. It is evident that the particular needs and features of the dataset at hand should influence the clustering algorithm selection.

References

[1] Aoullay, A. (2018, June 3). Spectral clustering for beginners. Medium. https://towardsdatascience.com/spectral-clustering-for-beginners-d08b7d25b4d8

[2] Avory Bryant, Krzysztof Cios," RNN-DBSCAN: A Density-Based Clustering Algorithm Using Reverse Nearest Neighbor Density Estimates", IEEE Transactions on Knowledge and Data Engineering, 2018, 30(6): 1109-1121.

[3] Fisher,R. A.. (1988). Iris. UCI Machine Learning Repository. https://doi.org/10.24432/C56C76.

[4] Gandhi, V. (2019, August 31). Spectral clustering - detailed explanation. Kaggle. https://www.kaggle.com/code/vipulgandhi/spectral-clustering-detailed-explanation

[5] Li, Lingli. (2016). A review of spectral clustering algorithms and their applications. Software Guide, 15(7), 54-56. https://doi.org/10.11907/rjdk.161229

[6] Ruitong Zhang, Hao Peng, Yingtong Dou," Automating DBSCAN via Deep Reinforcement Learning", by CIKM2022, 9 Aug 2022.

[7] von Luxburg, U. (2007). A tutorial on spectral clustering. Statistics and Computing, 17(4), 395-416. https://doi.org/10.1007/s11222-007-9033-z

[8] Wolberg,William, Mangasarian,Olvi, Street,Nick, and Street,W.. (1995). Breast Cancer Wisconsin (Diagnostic). UCI Machine Learning Repository. https://doi.org/10.24432/C5DW2B.

[9] Shen, T. (2021). The mathematics behind spectral clustering and the equivalence to PCA. https://doi.org/10.48550/arxiv.2103.00733

[10] S. Na, L. Xumin and G. Yong, "Research on k-means Clustering Algorithm: An Improved k-means Clustering Algorithm," 2010 Third International Symposium on Intelligent Information Technology and Security Informatics, Jian, China, 2010, pp. 63-67, doi: 10.1109/IITSI.2010.74.