Analysis of K-means and DBSCAN algorithm

Abstract

Cluster analysis is a significant method within unsupervised learning that enables users to uncover concealed insights within collected data. It encompasses a range of clustering algorithms designed to address diverse challenges, such as generating clusters with differing shapes, sizes, and densities, ensuring robustness against noise and outliers, requiring minimal prior domain knowledge, and scaling for large-scale or high-dimensional datasets. However, it is important to note that no single clustering algorithm exists that can universally address all of these challenges, as each algorithm has its own strengths and limitations.[1][2]

In this research work K-means and DBSCAN clustering algorithm are analysed on soil nutrients(i.e. Nitrogen, Phosphorous, Potassium) of a soil dataset and clusters obtained through both the algorithms are validated using silhouette coefficient. K-means provide good results for large data sets and when the clusters are in proper shape and size and DBSCAN is efficient in detecting outliers and separating them from being part of any clusters.

Keywords: Clustering, analysis, K-means, DBSCAN, silhouette coefficient, Soil nutrient

1 Introduction

1.1 Introduction to domain

There are three main branches of machine learning:

Supervised learning: In supervised learning, the machine is trained using labeled data, where the input data is already labeled with the correct output. The goal is to learn a mapping function that can predict the correct output for new input data. Examples of supervised learning algorithms include linear regression, logistic regression, decision trees, and neural networks.

Unsupervised learning: In unsupervised learning, the machine is trained using unlabelled data, where the input data is not labeled with the correct output. The goal is to learn the hidden structure or patterns in the data. Examples of unsupervised learning algorithms include clustering, dimensionality reduction, and anomaly detection.

Reinforcement learning: In reinforcement learning, the machine learns to make decisions by interacting with an environment, receiving rewards or punishments for its actions. The goal is to learn a policy that maximizes the cumulative reward over time. Examples of reinforcement learning algorithms include Q-learning, policy gradients, and actor-critic methods.

Clustering is extensively applied across a wide range of fields, including biology, statistics, pattern recognition, information retrieval, machine learning, psychology, and data mining. The task of cluster analysis presents several challenges, such as determining the optimal number of clusters, handling clusters with diverse shapes, sizes, and densities, ensuring robustness against noise and outliers, and achieving accurate clustering with minimal prior domain knowledge. These challenges become particularly crucial when dealing with large-scale and high-dimensional datasets. Various clustering methods exist, including partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods. Each method addresses specific aspects of clustering; however, no single universal algorithm exists that can effectively handle all the associated challenges comprehensively. [1][10]

1.2 K-means clustering algorithm

Partitioned clustering algorithm attempts to determine k partitions from a collection of n d-dimensional objects (vectors). K-means which is a popular clustering algorithm used to partition a dataset into K clusters, where K is a pre-specified number of clusters. The algorithm works by iteratively assigning each data point to the nearest cluster center, and then updating the cluster centers based on the mean of the data points assigned to each cluster. The process continues until convergence is reached, meaning that the cluster centers no longer change significantly. The algorithm can be sensitive to the initial placement of the cluster centers, and multiple runs with different initializations may be necessary to find a good solution.[2]

1.3 DBSCAN clustering algorithm

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a density-based clustering algorithm that groups together data points that are closely packed together in high-density regions and separates out data points that are in low-density regions.[3] The algorithm requires two parameters: eps (epsilon) and minPts (minimum points), where eps defines the radius of a neighbourhood around a data point, and minPts defines the minimum number of data points required to form a dense region. The algorithm starts by selecting a random data point and finding all nearby points within eps distance. If the number of nearby points is greater than minPts, then a dense region is formed and all connected points are assigned to the same cluster. The algorithm then repeats this process for all unvisited points until all data points have been assigned to a cluster or labeled as noise.[5]

To find out which clustering algorithm should be used which circumstances i.e. size of dataset, nature of cluster we have analysed K-means and DBSCAN clustering algorithm and validated the results obtained from both the algorithms using silhouette coefficient which is discussed in detail in section 4 of analysis and validation.

- **Organization of Paper**

The dataset description is provided in Section 2, detailing the characteristics and properties of the dataset used in this research. Section 3 discusses the performance evaluation measures employed in this study. Moving on to Section 4, it delves into the analysis and validation of the K-means and DBSCAN clustering algorithms, providing a comprehensive exploration of their effectiveness. Section 5 showcases the experimental setup, outlining the procedures and configurations utilized during the experiments. As for Section 6 and 7, they present concise conclusions drawn from the research work, followed by a discussion of potential future enhancements that can be explored in this area of study.

2 Dataset description

In this research work soil fertility dataset has been used.

Dataset[9] contains 7 attributes having nutrients value i.e. nitrogen, phosphorous and potassium (these are 3 features used for clustering analysis) and its comparative content i.e. low, medium and high

Table 1

N	P	K	temperature	humidity	ph	Rainfall	label
90	42	43	20.879744	82.002744	6.502985	202.935536	rice

85	58	41	21.770462	80.319644	7.038096	226.655537	rice
60	55	44	23.004459	82.320763	7.840207	263.964248	rice
74	35	40	26.491096	80.158363	6.980401	242.864034	rice
78	42	42	20.130175	81.604873	7.628473	262.717340	rice

3 Measures used for performance evaluation

1) Elbow Method:

The elbow method is a technique used to determine the optimal number of clusters, denoted as 'K', in a K-means clustering algorithm. It helps in finding a balance between the model's complexity and its ability to effectively capture the underlying patterns in the data.

To apply the elbow method, we calculate the Within-Cluster Sum of Squares (WCSS), which represents the sum of the squared distances between each data point and its assigned cluster centroid. The WCSS is computed for different values of K, which represent the number of clusters.

By plotting the WCSS values on the y-axis against the corresponding K values on the x-axis, we create an elbow graph. The graph typically shows a decreasing trend in WCSS as K increases, as more clusters can better fit the data. However, at a certain point, adding more clusters does not significantly reduce the WCSS.

The elbow point on the graph is where we observe a noticeable bend or "elbow" shape. This point indicates the optimal value of K, where further increasing K does not lead to a substantial improvement in WCSS. It represents a good trade-off between capturing the data's variability and avoiding overfitting by using excessive clusters.[8]

2) Silhouette score:

Silhouette score is a metric used to evaluate the quality of clustering results. It measures how well each data point in a cluster is separated from other clusters. The silhouette score ranges from -1 to 1, where a score of 1 indicates that the data point is very similar to other data points in its own cluster and very dissimilar to data points in other clusters, while a score of -1 indicates the opposite.[7]

Silhouette Score = b-a/max(b,a) Where a = intra cluster distance and b = inter cluster distance

Intra cluster distance is the average distance between the data point and all other data points in the same cluster and Inter cluster distance is the average distance between the data point and all other data points in the nearest neighbouring cluster.

The overall silhouette score for a clustering result is the average of the silhouette scores for all data points in the dataset. A higher silhouette score indicates a better clustering result, where the clusters are well separated and the data points within each cluster are similar to each other. A lower silhouette score indicates that the clustering is not well-defined, with data points that are not clearly separated from other clusters.[6][7]

3) 3D Scatter Plot:

A 3D scatter plot is a useful visualization tool for visualizing clusters created using a clustering algorithm with 3 features. In a 3D scatter plot, each data point is represented as a point in a 3D

space, where each axis corresponds to one of the features. To create a 3D scatter plot to visualize clusters first apply a clustering algorithm to the data and assign each data point to a cluster. Then, a different color is used for each cluster to distinguish the points belonging to different clusters. The resulting 3D scatter plot will show how the data points are distributed in the 3D space, with each cluster represented by a distinct color or symbol. We can also rotate the plot to view it from different angles, which can help us gain a better understanding of the distribution of the data points and the separation between the clusters.

4. Analysis and Evaluation

4.1 K-means

To apply K-means algorithm and obtain clusters it is required to have optimal value of 'K'. So Elbow method is applied and below graph is obtained

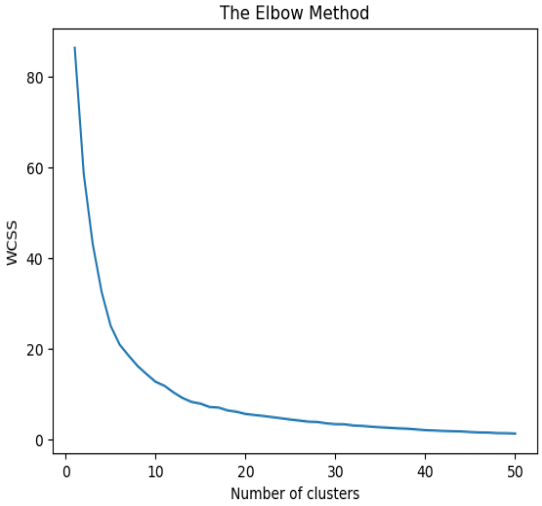

Fig 1 Graph of elbow method

According to graph K value is roughly 7 or 8. To find the accurate value of K and to evaluate the value observed from elbow method silhouette score is calculated for K equal to 5 to 14. From the graph it is clear that K value cannot be more than 15 and less than 5.

Table 1

No of clusters	Silhouette_Score
5	0.384627
6	0.401305
7	0.417535
8	0.436840
9	0.408780
10	0.429608
11	0.441726
12	0.388546
13	0.435551
14	0.404723

Among all the above values for K = 8 silhouette score is highest which proves that result obtained through elbow method is correct. So K-means model for K=8 is considered for clustering and all the 8 clusters generated by K-means algorithm are visualized using 3D scatter plot. By analysing the below 3D graph we observed that points are very much scattered and there are some outliers which should not be considered for clustering but the algorithm has considered all the outliers for clustering which affect the coordinates of centroid of particular cluster and its quality.(Colouring dots indicating different clusters obtained and black dots inside the clusters indicates centroid of that particular cluster)

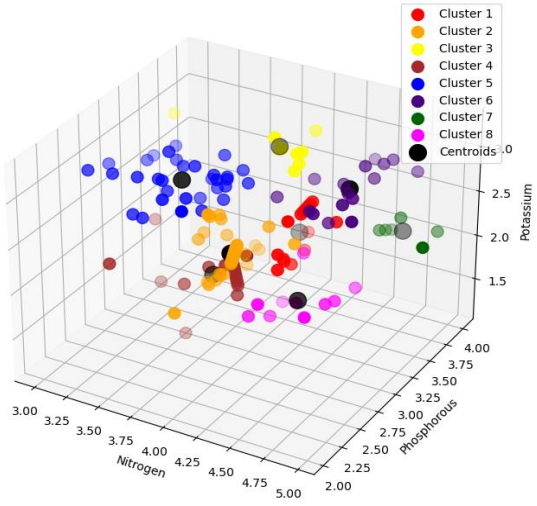

Fig 2 clusters of K-means algorithm

Analysis:

- As the dataset we have used is very small which impact on the performance and results of the k-means clustering algorithm. The k-means algorithm is particularly sensitive to the initial positioning of cluster centroids. In the case of small datasets, where the number of data points is limited, the initial placement of centroids plays a critical role in determining the final clustering outcome. If the initial placement is suboptimal, it can adversely affect the quality of the resulting clusters.
- K-means is sensitive to outliers since it minimizes the within-cluster sum of squares. Outliers can significantly affect the centroid calculation and distort the cluster boundaries. Preprocessing steps or outlier handling techniques may be required before applying k-means.
- K-means algorithm works better when the clusters have normal shape. It does not work well when the clusters have arbitrary shape.

4.2 DBSCAN

There are 2 parameters required to find out for DBSCAN algorithm which are epsilon and minpoints. There is no predefined way to find out minPts value but in general minPts should be greater than or equal to the dimensionality of the data set. As our data is having 3 variables , the

dimension of dataset is 3. For dimension > 2 we can have minPts = 2 * dimension of data.[4]. So we have considered minPts as 6.

Now to find out optimum value of epsilon a technique is used which involves calculating the average distance between each point and its k nearest neighbours, where the value of k corresponds to the MinPts parameter we have chosen. The resulting average k-distances are then arranged in ascending order and plotted on a k-distance graph. By examining the graph the optimal value for the epsilon (ε) parameter can be determine at the point where the graph exhibits the maximum curvature. This point of maximum curvature serves as a guide for selecting an appropriate value of ε for the DBSCAN algorithm.[4] The above technique is implemented using NearestNeighbours of Scikit learn and calculated the average distance between each point and its n_neighbors and obtained the k-distance elbow plot. Ideal value of eps is point of maximum curvature.

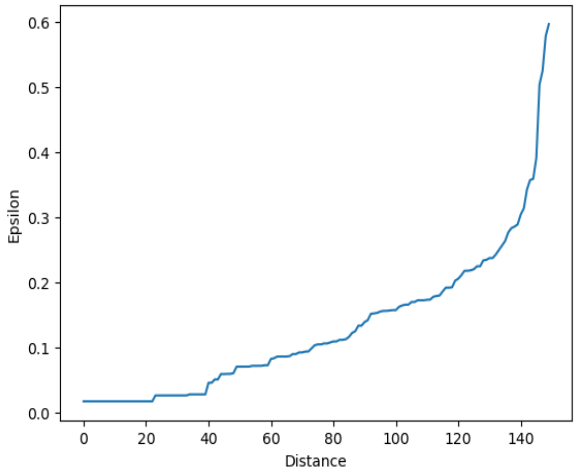

Fig 3 K distance graph

According to graph at eps = 0.3 curvature is maximum. To evaluate the value of minPts and epsilon we have taken a small range of values of epsilon and minPts and find out the silhouette score

Table 2

Epsilon	MinPts	No of clusters	Silhouette_Score
0.15	3	11	0.052748
0.20	3	13	0.192641
0.25	3	8	0.163395
0.30	3	6	0.259036
0.35	3	3	0.204943
0.15	4	7	0.015147
0.20	4	10	0.151316
0.25	4	6	0.221395
0.30	4	7	0.214950
0.35	4	3	0.208808
0.15	5	6	-0.001323
0.20	5	7	0.093662

0.25	5	6	0.159940
0.30	5	6	0.258532
0.35	5	3	0.209821
0.15	6	3	0.005443
0.20	6	5	0.044160
0.25	6	6	0.067092
0.30	6	5	0.147643
0.35	6	3	0.193302

The above table shows the silhouette score observed for eps ranging from 0.15 to 0.35 with minPts ranging from 3 to 7 (range of minPts value is also taken into consideration while finding silhouette score as the value which is taken in K-distance graph is just as a thumb rule.) From the above table for Epsilon = 0.30 and minPts = 6 the highest value of silhouette score is observed. Which means that the result observed from K-distance graph and silhouette score is exactly same and accurate. So we have considered our DBSCAN model with eps=0.3 and minPts = 6

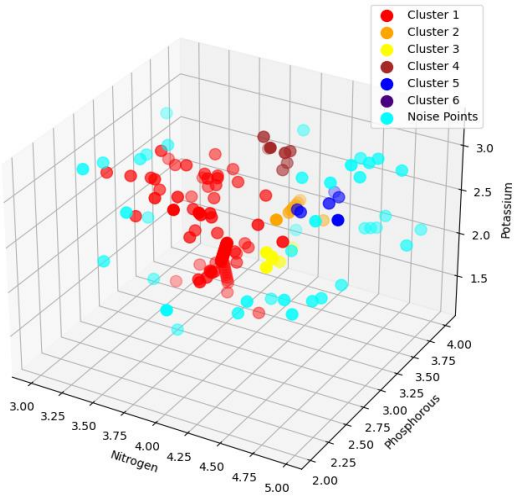

Fig 3 clusters of DBSCAN algorithm

The above is a 3D scatter graph of clusters observed from DBSCAN algorithm and with comparing this graph with the previous K-means' 3D graph, DBSCAN algorithm is separating the noise points(outliers) from the clusters. (Noise points are the points which are not eligible to be recognized as border points or core points as they do not have minimum no of neighbours (minPts) in the radius = eps.) The large no of noisy points in the graph indicates that points are scattered.

Analysis:

- The DBSCAN algorithm is proficient in detecting clusters with irregular shapes, as it does not impose any assumptions on the shape or size of clusters. It excels in identifying clusters with intricate structures, including elongated, overlapping, or non-convex shapes.
- DBSCAN is capable of effectively handling datasets that consist of both noise and outliers. It is able to differentiate between noise points, which do not belong to any cluster, and the actual clusters. Noise points are identified as outliers, while the denser regions are assigned to their respective clusters.

- DBSCAN requires the selection of two main parameters: epsilon (ε), which defines the neighborhood radius, and the minimum number of points (MinPts) required to form a dense region. The clustering results can vary depending on these parameter values, and selecting appropriate values can be challenging, especially in datasets with varying densities or complex structures.
- DBSCAN algorithm takes more time computation than K-means as DBSCAN examines the neighborhood of each point to determine clusters and k-means often converges in a small number of iterations, making it efficient in many scenarios.

5 Experimental Setup

- The system on which research work is done having the following configuration: OS-Microsoft Windows 10, Version: 10.0.19045, System Type: x64-based PC, Processor: Intel(R) Core (TM) i5-10210U CPU @ 1.60GHz, 2112 MHz, 4 Core(s), 8 Logical Processor(s), Installed Physical Memory (RAM): 16.0 GB, Total Virtual Memory: 19.5 GB

6 Conclusion

The main focus of this paper is to analyse K-means and DBSCAN clustering algorithm and evaluate the cluster obtained using silhouette coefficient. The objective of clustering is that soil expertise can recommend crops to farmers by analysing the clusters which have been obtained through K-means and DBSCAN algorithm for soil nutrients.

K-Means best suited for numerical data, and when the number of clusters is known or can be estimated. It works well when the clusters are well separated and points in a cluster are not scattered too much (so higher the value of silhouette coefficient will be) and when the variance of the clusters is similar.

Density-based clustering algorithms, such as DBSCAN, are useful for identifying clusters of arbitrary shape in noisy data. Density-based clustering works well when the data has varying densities and when the clusters are not well-separated. It can also detect outliers and anomalies and is useful in applications such as anomaly detection.

7 Future Enhancement

In this work only 2 clustering algorithms are considered for analysis and validation but in future other clustering algorithms like hierarchical clustering, Optics clustering, fuzzy clustering or spectral clustering to compare their performance with current algorithms which have been used.

8 References

1. Hetal Bhavsar and Anjali Jivani, “An approach towards the Shared Nearest Neighbor (SNN) Clustering Algorithm”, in the national conference SPCTS’07, DAII-CT, Gandhinagar, India. September(2007)

2. Jiawei Hen and Micheline kember, “Data Mining Concepts and Technique”, Elsevier (2001).

3. Martin Ester. Hans-Peter Kriegel, Jorg Sandar, Xiaowei Xu," A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," KDD 96, Portland, OR, pp. 226-231 (1996) .

4. DBSCAN Parameter Estimation, Tara Mullin. 2020. Medium : https://medium.com/@tarammullin/dbscan-parameter-estimation-ff8330e3a3bd

5. DBSCAN Overview, Examples and Evaluation, Tara Mullin, 2020. Medium : https://medium.com/@tarammullin/dbscan-2788cfce9389

6. Dinh, D. T., Fujinami, T., & Huynh, V. N. (2019). Estimating the optimal number of clusters in categorical data clustering by silhouette coefficient. In *Knowledge and Systems Sciences: 20th International Symposium, KSS 2019, Da Nang, Vietnam, November 29–December 1, 2019, Proceedings 20* (pp. 1-17). Springer Singapore.

7. Silhouette Coefficient Validating Technique, Ashutosh Bhardwaj, 2020. Medium: https://towardsdatascience.com/silhouette-coefficient-validating-clustering-techniques-e976bb81d10c

8. Khyati Mahendru , 2019. How to determine the optimal K for K-means ? Medium. https://medium.com/analytics-vidhya/how-to-determine-the-optimal-k-for-k-means-708505d204eb

9. Dataset , Kaggle : https://www.kaggle.com/datasets/meghaljambhale/soildataset

10. Xu, D., Tian, Y. A Comprehensive Survey of Clustering Algorithms. *Ann. Data. Sci.* **2**, 165–193 (2015). https://doi.org/10.1007/s40745-015-0040-1