International Journal of Engineering Applied Sciences and Technology, 2021
Vol. 5, Issue 11, ISSN No. 2455-2143, Pages 255-261
Published Online March 2021 in IJEAST (http://www.ijeast.com)

A COMPREHENSIVE SURVEY OF VARIOUS CLUSTERING PARADIGMS

Pranav Kangane, Vivek Joshi, Astha Kacker, Manan Jain
Department of Computer Science and Engineering,
NMIMS Mukesh Patel School of Technology Management & Engineering, Mumbai, India

***Abstract*— Clustering is one of the most significant tasks in data mining. It is highly significant in the entire procedure of knowledge discovery. Clustering is an unsupervised learning task that is utilized to discover hidden patterns in the dataset that can't be classified appropriately and plainly. It groups the dataset in the form of clusters having similar characteristics. There are various clustering paradigms which are applied on the dataset depending on expected cluster characteristics. These algorithms are applied based on the data type of attributes, the algorithm's complexity, and aptness for a specific grouping. Clustering is performed for various purposes. This paper attempts to study different clustering algorithms that are categorized into various clustering paradigms. The algorithms are reviewed and analyzed on various parameters. It also presents a comparison table based on efficiency to handle high dimensional data, handle outliers and noise, scalability, sensitiveness to the input sequence, space and time complexities.**

Keywords— **Clustering Techniques, Density-Based, Hierarchical, Partitional**

I. INTRODUCTION

A large amount of heterogeneous data is collected from various sources which are generally present in unstructured format and thereby increasing the computational time to process the data and analyse it. Data mining is a process that deals with the extraction of usable information from raw unstructured data. Its objective is to learn from records and discover patterns that can be used to develop effective strategies related to various business functions. This will help organizations make better decisions and get closer to their objectives. Data mining has applications in various fields. It can be utilized to predict patterns automatically that are based on trends, making groups based on finding, a prediction that is likely based on results, and depict groups of facts visually which were previously unknown. To help in measuring enormous information in a sensible measure of time, machine learning comes into the picture. The objective of machine learning is to program a computer to utilize data examples or past encounters for handling a given issue. Machine Learning can assist us with understanding the structure of information and fit that information into models that can be perceived and used by people [14]. Machine Learning is categorized into supervised, unsupervised and reinforcement learning.

II. UNSUPERVISED LEARNING

In unsupervised learning, the dataset comprises a set of inputs with no corresponding set of labelled outputs. The unsupervised learning algorithm itself needs to find patterns present in its raw input. In this type of classification, it does not have a training dataset. It has to discover patterns that are hidden in a dataset containing unlabelled outputs. The principle objective is to diminish closeness in data points belonging to similar groups and make every cluster disparate from one another.

A. Issues with Unsupervised Learning

These are the following issues while using unsupervised learning [8]:

- Since the input data is unknown and unlabelled, the results tend to be less accurate. The machine itself has to discover patterns.
- Not certain if the results we get are meaningful or not because of unlabelled classes.
- The labelled classes formed after the classification are not easily interpretable. Thus, the user needs time for interpreting the classes as they are not always informational classes.

B. Why to use Unsupervised Learning

The reasons for using unsupervised learning are mentioned below [8]:

- Unsupervised machine learning discovers all kinds of hidden and unknown patterns in data.
- Splits the dataset automatically into groups based on their similar characteristics.
- It helps in finding features that can be possibly useful for categorization.
- It is useful in detecting unusual points in a dataset that can help in anomaly detection applications such as fraudulent transactions.

- Association mining helps in identifying a set of objects that may occur together in the dataset

III. CLUSTERING

Clustering involves discovering hidden patterns that exist in data which is uncategorized. Clustering algorithms will process the data points and classify them into specific groups or clusters. The number of clusters that need to be identified can be modified by the user. While doing analysis of clusters, the data points are classified into groups based on likeness level and afterwards labels are assigned to the groups.

A. *Why to use Clustering Algorithms*

The following are the applications of cluster analysis [8]:

- Cluster analysis is used in data analysis, trends in market research
- In businesses, customers are segmented based on their pattern of purchasing
- Helps in detection of outliers and therefore, can be used in the detection of credit card frauds
- It can be used to detect the accident-prone areas for a particular geographic location
- It is used in information discovery by classifying web documents
- It is useful in image processing
- Helps in giving an insight into data distribution to observe cluster characteristics
- Helps in pattern identification

B. *Requirements of Clustering Algorithms*

The following are the requirements of clustering algorithms [2]:

- The algorithm should be able to handle all kinds of data such as numeric, categorical, interval-based.
- It should be able to detect arbitrary shaped clusters.
- To deal with large datasets, it should be highly scalable.
- It should be able to handle high dimensional data.
- It should be able to handle outliers and erroneous data.
- The clustering result should be interpretable and usable.

IV. CLUSTERING METHODS

The clustering methods can be classified into various categories based on various factors and initial conditions. Different algorithms are used to provide solutions in different fields. All clustering problems cannot be solved using one single algorithm. Thus, different algorithms are used to solve different problems. Fig.1 shows various clustering paradigms and their types. The widely accepted clustering methods are:

1. Clustering based on Partition
2. Clustering based on Density
3. Clustering based on Hierarchy

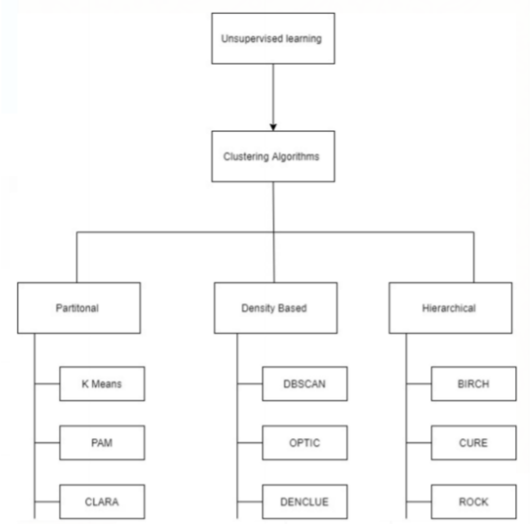

Fig. 1. Different Clustering Paradigms

A. PARTITIONAL CLUSTERING METHODS

This method creates K partitions of the given dataset. The number of clusters or partitions to be formed is specified by the value of K. It at that point begins relocating the data points iteratively from one cluster to another cluster. This is done to improve the partitioning and is carried out until an ideal partition is obtained. The data points are only partitioned on one level and not more than that. The partitioning depends on a specific objective function. The clusters are shaped in such a way that it optimizes an objective partitioning basis, such as dissimilarity function dependent on separation, with the goal that the items inside a cluster are "similar", while the data points of various groups are "dissimilar". The primary downside of this algorithm is that it gives helpless results when data points overlap each other [13]. This generally happens when a data point is near to the center of a different cluster. K-means and PAM are utilized when a constrained number of clusters are to be shaped. For huge datasets, a sampling-based strategy CLARA is utilized.

K-MEANS

The algorithm starts by dividing the dataset into K partitions where the value of K is taken from the user. The main objective is that for k clusters, k centroids need to be characterized. These centroids ought to be put carefully because different areas cause diverse results [7]. So, it is way better to put them as distant as conceivable from each other. After making K clusters, each data point belonging to the dataset is linked to the nearest cluster. The algorithm continues changing and appointing data points to the nearest current cluster till the moment when no new task of assigning data points to the cluster can be made.

Advantages:

- The algorithm is moderately easy to implement
- Scales to a huge dataset.
- The clusters have convex shapes [7].
- It gives the best result when the dataset is distinct or well separated from each other [7].

Drawbacks:

- User has to input the value of K.
- The results largely depend on initial values.
- The algorithm experiences trouble in clustering data points having a varying size and density.
- Instead of ignoring the outliers, there is a possibility that the outliers can form their own cluster.
- The algorithm cannot handle high dimensional data.
- This algorithm will not work for non-linear data [4].
- This algorithm uses a Euclidean distance whose measures can unequally weight underlying factors [4].
- In case of data overlapping, K-Means is unable to solve as there are different clusters formed at the same space [4].

PAM

The PAM algorithm takes in an iterative and greedy way [11]. The PAM algorithm takes a data point as a reference point which is located in the center of the cluster. Whereas in the case of K-means, the cluster considers the mean value. It is additionally called the K-Medoids algorithm. Inside a cluster, the medoid and the other data points should have dissimilarity as minimum as possible. The algorithm first takes a set of k medoids, every data point is then assigned to a medoid nearest to it. The next step involves selecting a medoid m and swapping it with a non-medoid data point and then objective function S is computed which corresponds to the sum of dissimilarities of all data points to their nearest medoid. The selected object is then replaced with an unselected object only it further diminishes the objective function. This process is carried out until S can no longer be decreased.

Advantages:

- More robust to outliers than K-means.
- Easy to implement and understand.
- It uses Manhattan distance for calculating the dissimilarity among the nodes, which is more robust than Euclidean distance [4].

Disadvantages:

- Functions admirably with little datasets yet doesn't scale well for huge datasets.
- Quite expensive in terms of computational complexity.
- For enormous estimations of n and k, this computation turns out to be costlier than the KMeans method [11].

CLARA – Clustering Large Applications

The CLARA algorithm arbitrarily picks a small portion of the original dataset. It does not consider the entire dataset but takes the small portion to represent the entire dataset. PAM is utilized for choosing the medoids from the sample [5]. After this initial step, each object not having a place in the beginning sample is distributed to the closest delegate object, and a measure of clustering of the whole dataset is acquired. This measure is contrasted with n different measures acquired from the application of algorithms in n diverse beginning samples [5]. The algorithm then picks the best clusters obtained from various samples.

Advantages:

- Deals with larger datasets.
- More robust to outliers and noise.
- It takes less time to execute as compared to other techniques.

Disadvantages:

- If any of the best sampled medoids is a long way from the best k medoids then it will not be able to find a decent clustering [11].
- The algorithm produces spherical clusters [11].

B. DENSITY-BASED CLUSTERING METHODS

In density-based clustering algorithms, the data points are classified based on their relative position to the other data points and the dense regions that they form together. The density-based algorithms can locate regions(clusters) of arbitrary shapes and exclude those data points that do not belong to any clusters. Density-based algorithms classify objects into clusters based on the areas that are concentrated and separate them by areas that are empty or sparse. The data points that are not assigned to any clustered regions are classified as noise and excluded. Thus, density-based clustering algorithms can deal with outliers in a robust manner while simultaneously forming clusters.

DBSCAN - Density-Based Spatial Clustering of Applications with Noise

The DBSCAN algorithm accepts two parameters, eps and minPts. Eps defines the distance that is used to classify a point as a neighbourhood point, whereas minPts gives us the minimum number of points that are required in a dense region. The algorithm starts by selecting a random point from the dataset. It keeps selecting points until all points have been visited. If a point has more than minPts points within a distance of eps(ε), then this point and all of its neighbours will be part of the same cluster. This calculation is done for the neighbouring points themselves and in this way, recursively, the clusters are expanded. Any point that doesn't have minPts points in a distance of eps(ε) and is not present within the eps(ε) distance of any other point that is part of a cluster, is considered as noise and not assigned to any cluster.

Advantages:

- Can discover arbitrarily shaped clusters, including clusters that are completely surrounded by other clusters [12].
- Robust towards outlier detection and noise.
- Eps and minPts are the only two parameters which are required and are not sensitive to the order in which the data points are present in the dataset.
- DBSCAN figures out the number of clusters on its own, and we do not have to explicitly mention the number of clusters to form like in k-means clustering.

Disadvantages:

- Not partitionable for multiprocessor systems.
- Fails to identify clusters if density varies and if the dataset is too sparse.
- DBSCAN heavily relies on its distance measure for appropriate clustering. It is only as good as the distance measure used. The most common distance measure used is the Euclidean one, which gets increasingly harder to use with high dimensional data.
- DBSCAN does not work well on data with wide variation in densities, for a given epsilon-minPts combination.

OPTICS - Ordering Points to Identify Cluster Structure

The OPTICS algorithm forms reachability plots instead of the actual clusters themselves. These reachability plots can then be used to find out the clusters. We will first define what are core distances and reachability distances. Core distance is the minimum distance that is required for a point to have minPts other points in its neighbourhood. Reachability distance of one object from another object is the minimum distance between them, given that the second object is a core object [1]. The core distance for each point in the dataset is calculated. Then we calculate the reachability distance for every point, updating only those distances which are an improvement over their previous values. The successive data points chosen are the ones that have the nearest reachability distance. In this manner, the data points that are situated closer together are ordered near to each other in the output. In addition, the algorithm also stores another distance that tells us the density that is required for a cluster in order to classify the two points as part of the same cluster. We can then extract the clusters from the reachability plots by forming valleys using the local minima and maxima in the reachability plot.

Advantages:

- Can discover arbitrarily shaped clusters, including clusters that are completely surrounded by other clusters [12].
- The algorithm works well with datasets having a wide variation in densities. Since it forms a reachability plot and not the clusters themselves, there is no need for a density parameter.
- The output order of the data points can be used to find out clusters of various densities, using the valleys found in the reachability plot. The deeper the valley, the denser the cluster represented by it.

Disadvantages:

- In order to handle data with a wide variation in density, the algorithm only produces a reachability plot, while the actual clusters are not produced, but rather have to be extracted from the plot.
- It cannot handle high dimensional data.
- The time complexity is O(n^2), which is very high [12].

DENCLUE

The DENCLUE calculation is an uncommon instance of Kernel Density Estimation (KDE). It is a non-parametric calculation. The technique is based on the possibility that the impact of every information point can be officially displayed utilizing a numerical capacity called an influence function, which shows the effect of that point on the other data points in its neighbourhood density of the entire dataset can be modelled by taking the aggregate of the influence function of each and every point in the dataset. Clusters can then be determined by using density attractors, which are the nothing but the local maxima of the aggregated density function. The Denclue algorithm works in two stages, a pre clustering step and a clustering step. The pre-clustering step calculates the influence functions of all the data points and then aggregates them to form the general density function of the dataset. The clustering step identifies clusters by using the density attractors, which are the local maxima of the general density function obtained in the previous step [6]. All the data points in the dataset are assigned to a cluster with the help of density attractors using a hill climbing algorithm to find those local maxima. The Gaussian function is one of the more popular density functions used.

Advantages:

- The sensitivity of density (owing to the radius parameter ε) is removed [12].
- Clusters of arbitrary shapes can be represented in a compact manner using mathematical functions.
- It has good clustering properties for data sets with large amounts of noise.
- It is significantly faster than other density-based clustering algorithms like DBSCAN.

Disadvantages:

- The quality of clusters produced significantly depends on the noise threshold and the density parameter provided to the algorithm. Thus, this algorithm is very sensitive on the parameters provided to it for the production of good results.
- Instead of the actual clusters it only produces a cluster ordering.

C. HIERARCHICAL CLUSTERING METHODS

This category involves constructing a hierarchical relationship among the data in order to cluster. It generates a sequence of nested partitions which can be visualized in the form of a tree or hierarchy of clusters known as cluster dendrogram. In the beginning, each data point stands for an individual cluster then merges with the most neighbouring cluster to form a new one until only one cluster is left. The hierarchical clustering can be divided into two categories on the basis of the approach. Agglomerative clustering follows a bottom-up approach where all data points start as individual clusters and merge with most neighbouring one until only one cluster is left. Divisive clustering works in a top-down manner. It starts with one cluster having all points and then recursively splits until all points are in a separate cluster. BIRCH, CURE, ROCK follows an agglomerative approach whereas DIANA and MONA follow a divisive approach. In hierarchical algorithms, previously taken steps, whether merging or splitting, is irreversible even if they are erroneous. These algorithms are sensitive to outliers and noise. However, these algorithms are suitable for data sets with arbitrary shape and type, it can easily detect the hierarchical relationship among clusters.

BIRCH

The BIRCH stands for Balanced Iterative Reducing and Clustering Using Hierarchies. Here data space is not uniformly occupied and hence not every data point is equally important for clustering purposes [16]. It is mainly used when a small number of I/O operations are needed [11]. It incrementally constructs a CF (Clustering feature) tree, which is a hierarchical data structure for multiphase clustering. The clustering features are stored in a height balanced tree called a CF tree for hierarchical clustering. A cluster of data points is represented by three components. N - number of datapoints, LS-linear sum of points and SS - the sum of the squared of the points. In a CF Tree structure, each non-leaf node has utmost B entries. Each leaf node has at most L CF entries which satisfy threshold T, a maximum diameter of the radius. The maximum size of a node is given by P (page size in bytes). The phase-1 begins with loading data into memory. It scans the DB and loads data into memory by building a CF tree. The tree is rebuilt from the leaf node if memory is exhausted. The phase-2 consists of condensing the data where we resize the data set by building a smaller CF tree. Outliers are removed. In phase-3 clustering algorithms such as K-means are applied on CF entries to form good clusters. The CF trees having data points of the same value might get assigned to different leaf entries. In such a case, the last phase solves this problem by refining the clusters.

Advantages:

- With a single scan, it is able to determine good clusters which are subsequently improved with few additional scans [11].
- Works with very large data sets [11].
- It has the capacity to fix what has been done in the past step [2].
- BIRCH is exceptionally compelling in taking care of enormous datasets by making memory requirements and time explicit and it is likewise the best for discovering noise [16].

Disadvantages:

- The clusters formed are generally spherical due to radius and diameter measures.
- Arbitrary shaped clusters cannot be identified.
- Is equipped for taking care of just numeric data [14].
- It is Sensitive to the sequence in which the data is recorded [16].

CURE

The CURE (Clustering Using Representatives) is an agglomerative hierarchical technique used for large scale clustering. A balance between the centroid and other data points is created by the algorithm. The algorithm uses a collection of representative points to represent the clusters and not by their centroids. The algorithm is divided into two phases. The initialization phase starts with taking a small data sample and clustering it in main memory. CURE is intended to deal with arbitrary shaped clusters. It then chooses a small set of points as representative points from each cluster. These data points need to be selected as far as possible from each other using K-Means. The representative points are then moved between its location and cluster's centroid by a fixed fraction of distance. This progression requires a Euclidean space since otherwise, there won't be any notion of a line between two points. The completion phase involves converging clusters if the clusters have a pair of representative points adequately close to each cluster. The user can define the distance for "close". The merging step is continued until there are no more clusters which are adequately close to each other.

Advantages:

- With a single scan, it is able to determine good clusters which are subsequently improved with few additional scans [11].
- It is developed for identifying more complex cluster shapes [16].
- Even in the presence of outliers and noise, the algorithm is robust.
- It is suitable for managing huge datasets.

Disadvantages:

- Random sampling can influence prerequisites of memory and thus, making it a costlier choice for databases.
- Cannot handle different densities [16].

ROCK

ROCK stands for Robust clustering using links. It is an agglomerative hierarchical technique. For estimating the vicinity between a pair of data points with categorical type of attributes, the algorithm presents the idea of "links". The algorithm makes use of "goodness measure" for deciding how comparable the clusters are. This algorithm starts by drawing a random sample from the database. After this link is applied to the samples which means it iteratively merges two clusters that maximise the goodness function and stops merging when clusters are present in necessary numbers or when no more links are present between the clusters. Clusters involving only the sampled points are utilized to allot the remaining data points on disk to the appropriate clusters.

Advantages:

- Handles the data with categorical and Boolean attributes [16].
- Handles large datasets and it reduces complexity [16].

Disadvantages:

- Clusters having various shapes and sizes can't define the clusters accurately.

V. PERFORMANCE EVALUATION OF VARIOUS ALGORITHMS

Table 1: Partitional Algorithms

Category	K MEANS	PAM	CLARA
Large Scale Dataset	Yes	No	Yes
High Dimensional Dataset	No	No	No
Sensitive to Noise & Outliers	Highly	Little	Little
Scalability	Middle	Low	High
Sensitive to Input Sequence	Highly	Moderate	Moderate
Type of Data	Numerical	Numeric	Numerical

Table 2: Density-Based Algorithms

Category	DBSCAN	OPTIC	DENCLUE
Large Scale Dataset	Yes	Yes	Yes
High Dimensional Dataset	No	No	Yes
Sensitive to Noise & Outliers	Little	Little	Little
Scalability	Middle	Middle	-
Sensitive to Input Sequence	Moderate	Little	-
Type of Data	Numerical	Numerical	Numerical

Table 3: Hierarchical Algorithms

Category	BIRCH	ROCK	CURE
Large Scale Dataset	Yes	No	Yes
High Dimensional Dataset	No	Yes	Yes
Sensitive to Noise & Outliers	Little	Little	Little
Scalability	High	Middle	High

Sensitive to Input Sequence	Moderate	Moderate	Moderate
Type of Data	Numerical	Categorical	Numerical

Table 4: Complexity Table

Algorithm	Time
K-MEANS	O(idkn)
PAM	O(ik (n-k)2)
CLARA	O(ksˆ2+k(n-k))
SBSCAN	O(n^2)
OPTICS	O(n^2)
DENCLUE	O(n logn)
BIRCH	O(d*e*p)
CURE	O(n^2logn)
ROCK	O(nˆ2*logn)

VI. CONCLUSION

Cluster analysis is of paramount importance for discovering hidden patterns in the dataset. In the paper, we compared and analysed various popular clustering algorithms. In addition, we likewise examined the various classifications where these algorithms can be characterized (partitional, hierarchical, density based). Every paradigm is equipped for dealing with novel prerequisites of the user application. The underlying technique for every algorithm along with their advantages and disadvantages have been discussed.

VII. REFERENCES

1. Mihael, A., Breunig, M., Kriegel, P., & Sander, J. (1999). OPTICS: ordering points to identify the clustering structure. *ACM Sigmod Record*, 49-60.

2. Bhuyan, R., & Borah, S. (2013). A Survey of Some Density Based Clustering Techniques. *10.13140/2.1.4554.6887*.

3. Bindra, K., & Misra, A. (2017). A Detailed Study of Clustering Algorithms. *752-757. 10.1109/CTCEEC.2017.8454973*.

4. Chitra, D. (2017). A Comparative Study of Various. *International Journal of Computer Science and Mobile Computing.*

5. Firdaus, S., & Uddin, A. (2018). A Survey on Clustering Algorithms and Complexity Analysis.

6. Thilagavathi, G., Srivaishnavi, S. (2013), A Survey on Efficient Hierarchical Algorithm used in Clustering, *INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT)* Volume 02, Issue 09 (September 2013).

7. Gupta, T., & Panda, S. (2019). A Comparison of K-Means Clustering Algorithm and CLARA Clustering Algorithm on Iris Dataset. *7. 4766-4768. 10.14419/ijet.v7i4.21472.*

8. Han, J., & Kamber, M., & Pei, J. (2012). 10 - Cluster Analysis: Basic Concepts and Methods.

9. Rehioui, H., Idrissi, A., Abourezq, M., & Zegrari, F. (2016). DENCLUE-IM: A New Approach for Big Data Clustering. *Procedia Computer Science*, 83, 560–567. doi:10.1016/j.procs.2016.04.265

10. AnithaElavarasi, S., & Akilandeswari, J. (2011). A Survey On Partition Clustering Algorithms. *International Journal of Enterprise Computing and Business Systems*.

11. Popat S., et al, (2012).). A Survey On Density Clustering Algorithms. / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (1) , 2014, 805-812

12. Colin, S. (2019, Jan 2). Clustering Using optics. Retrieved from *towardsdatascience.com*.

13. Velmurugan T., & Santhanam, T., (2011). A Survey of Partition based Clustering Algorithms in Data Mining: An Experimental Approach. *Information Technology Journal*. 10. 10.3923/itj.2011.478.484.

14. Xehen. (2019, Sep 9). *GUIDE ON UNSUPERVISED LEARNING*. Retrieved from Medium.

15. Dongkuan, X., & Yingjie T. (2015). A Comprehensive Survey of Clustering Algorithms. *Annals of Data Science. 2. 10.1007/s40745-015-0040-1.*

16. Rani, Y., & Rohil, H. (2013). A Study of Hierarchical Clustering Algorithms. *International Journal of Information and Computation Technology*.