***Abstract*:** Clustering is process of grouping data into clusters based on some kind of similarities between the data. Clustering of data is essential in organizing huge amount data generated so that complexity in accessing and fetching of such data is reduced. The selection of a clustering method is of great importance; as it determines the ease at which data is grouped. It varies according to the various metrics of the generated data. This paper deals with various batch and data stream clustering techniques, along with their strengths and weaknesses and where these data stream techniques are applied.

Bharath P Hegde Shreyas M Yashwan R

Department of ISE, BMSIT&M Department of ISE, BMSIT&M Department of ISE,BMSIT&M

Bangalore, Karnataka, India Bangalore, Karnataka, India Bangalore, Karnataka, India

[bharathsasuke@gmail.com](mailto:bharathsasuke@gmail.com) [shreyas1195@gmail.com](mailto:shreyas1195@gmail.com) [yashwan74@gmail.com](mailto:yashwan74@gmail.com)

S Mahalakshmi

Asst.Prof, Dept of ISE, BMSIT&M

Bangalore, Karnataka, India

maha.shanmugam@gmail.com

***1. Introduction***

The data is increasing at rapid speed not only in size but also in variety. With this growing data there comes challenge and difficulties to handle such large amount of data. This huge amount of data generated every day is termed as Big Data. For example, there are 294 billion emails sent every day, over 1 billion Google searches are made every day Facebook generates more than 30 Petabytes of user-generated data which are stored, accessed and analyzed, Twitter generates more than 230 tweets per day. This Big data exhibits different characteristics like volume, variety, variability, value, velocity and complexity due to which it is very difficult to analyze data and obtain information with traditional data mining techniques

***1.1Characteristics of Big Data***

The main characteristics of Big Data often referred to as the 4 Vs. They are

**Volume**: The quantity of generated data is important in this context. The size of the data determines the value and potential of the data under consideration, and whether it can actually be considered big data or not. The name ‘big data’ itself contains a term related to size, and hence the characteristic.

**Variety**: The type of content (For example text, images, audio, video etc.), and an essential fact that data analysts must know. This helps people who are associated with and analyze the data to effectively use the data to their advantage and thus uphold its importance.

**Velocity**: In this context, the speed at which the data is generated and processed to meet the demands and the challenges that lie in the path of growth and development.

**Veracity**: It refers to the quality of data. It also refers to the nosiness of the data. It deals with the reliability of the source and accuracy of the data.

Along with these 4 characteristics Big Data also has the following characteristics

**Variability**: The inconsistency the data can show at times—-which can hamper the process of handling and managing the data effectively.

**Complexity**: Data management can be very complex, especially when large volumes of data come from multiple sources. Data must be linked, connected, and correlated so users can grasp the information the data is supposed to convey.

***2. Clustering algorithm***

As there are so many clustering algorithms this section introduces a categorizing framework that groups the various clustering algorithms found into distinct categories. The different clustering algorithms can be broadly classified as follows:

* **Partitioning based**: The partitioning algorithm divided the data objects into number of partitions, where each partition represents a cluster. These clusters should fulfill following requirements: (1) Each group must contain at least one object, (2) Each object must belong to exactly one group. Examples: K-Mean
* **Hierarchical-based:** Data are organized in a hierarchical manner depending on the medium of proximity. Hierarchical clustering methods can be agglomerative (bottom up) or divisive (top-down). An agglomerative clustering starts with one object for each cluster and recursively merges two or more of the most appropriate clusters. A divisive clustering starts with the dataset as one cluster and recursively splits the most appropriate cluster. BIRCH, CURE, ROCK and Chameleon are some of the well-known algorithms of this category.
* **Density-based**: Here, data objects are separated based on their regions of density, connectivity and boundary. DBSCAN, OPTICS, DBCLASD and DENCLUE are algorithms that use such a method to ﬁlter out noise (outliers) and discover clusters of arbitrary shape.
* **Grid-based**: The space of the data objects is divided into grids. The main advantage of this approach is its fast processing time, because it goes through the dataset once to compute the statistical values for the grids the performance of a grid-based method depends on the size of the grid, which is usually much less than the size of the database Wave-Cluster and STING are typical examples of this category.
* **Model-based**: It is based on the assumption that the data is generated by a mixture of underlying probability distributions. There are two major approaches that are based on the model-based method: statistical and neural network approaches. MCLUST is probably the best-known model-based algorithm. The statistical approach uses probability measures in determining the concepts or clusters. Probabilistic descriptions are typically used to represent each derived concept. The neural network approach uses a set of connected input/output units, where each connection has a weight associated with it. Neural networks have several properties that make them popular for clustering. First, neural networks are inherently parallel and distributed processing architectures. Second, neural networks learn by adjusting their interconnection weights so as to best ﬁt the data.

**3. Batch Clustering Algorithms**

**3.1.K-Means**

Prototype-based clustering techniques create a one-level partitioning of the data objects. There are a number of such techniques, but two of the most prominent are K-means and K-medoid. K-means deﬁnes a prototype in terms of a centroid, which is usually the mean of a group of points, and is typically applied to objects in a continuous n-dimensional space. K-medoid deﬁnes a prototype in terms of a medoid, which is the most representative point for a group of points, and can be applied to a wide range of data since it requires only a proximity measure for a pair of objects. While a centroid almost never corresponds to an actual data point, a medoid, by its deﬁnition, must be an actual data point. In this section, we will focus solely on K-means, which is one of the oldest and most widely used clustering algorithms.

**K-means Algorithm**

The K-means clustering technique is simple, and we begin with a description of the basic algorithm. We ﬁrst choose K initial centroids, where K is a user speciﬁed parameter, namely, the number of clusters desired. Each point is then assigned to the closest centroid, and each collection of points assigned to a centroid is a cluster. The centroid of each cluster is then updated based on the points assigned to the cluster. We repeat the assignment and update steps until no point changes clusters, or equivalently, until the centroids remain the same.

The algorithm is as follows:

1. Initialize the center of the clusters

2. Attribute the closest cluster to each data point

3. Set the position of each cluster to the mean of all data points belonging to that cluster

4. Repeat steps 2-3 until convergence

**Advantages**

The wide popularity of k-means algorithm is well deserved. It is simple, straightforward, and is based on the firm foundation of analysis of variances.

**Disadvantages**

1. The result strongly depends on the initial guess of centroids (or assignments).
2. It is not obvious what is a good k to use.
3. The process is sensitive with respect to outliers(Noise).

**3.2. DBSCAN**

Density-based clustering locates regions of high density that are separated from one another by regions of low density. DBSCAN is a simple and eﬀective density-based clustering algorithm that illustrates a number of important concepts that are important for any density-based clustering approach. In this algorithm density is estimated for a particular point in the data set by counting the number of points within a speciﬁed radius, Eps, of that point. This includes the point itself.

Classiﬁcation of Points According to Center-Based Density

The center-based approach to density allows us to classify a point as being (1) in the interior of a dense region (a core point), (2) on the edge of a dense region (a border point), or (3) in a sparsely occupied region (a noise or background point). The following text provides a more precise description. **Core points**: These points are in the interior of a density-based cluster. A point is a core point if the number of points within a given neighborhood around the point as determined by the distance function and a user speciﬁed distance parameter, Eps, exceeds a certain threshold, MinPts, which is also a user-speciﬁed parameter.  **Border points**: A border point is not a core point, but falls within the neighborhood of a core point. A border point can fall within the neighborhoods of several core points. **Noise points**: A noise point is any point that is neither a core point nor a border point.

The DBSCAN Algorithm

Given the previous deﬁnitions of core points, border points, and noise points, the DBSCAN algorithm can be informally described as follows. Any two core points that are close enough within a distance Eps of one another are put in the same cluster. Likewise, any border point that is close enough to a core point is put in the same cluster as the core point. (Ties may need to be resolved if a border point is close to core points from diﬀerent clusters.) Noise points are discarded. The formal details are given in Algorithm. This algorithm uses the same concepts and ﬁnds the same clusters as the original DBSCAN, but is optimized for simplicity, not eﬃciency.

**DBSCAN algorithm**.

1. Label all points as core, border, or noise points.
2. Eliminate noise points.
3. Put an edge between all core points that are within Eps of each other.
4. Make each group of connected core points into a separate cluster.
5. Assign each border point to one of the clusters of its associated core points.

**Advantages**

1. DBSCAN does not require one to specify the number of clusters in the data a priori, as opposed to k-means.
2. DBSCAN can find arbitrarily shaped clusters.
3. DBSCAN is immune to noise.

**Disadvantages**

1. DBSCAN is not entirely deterministic: border points that are reachable from more than one cluster can be part of either cluster, depending on the order the data is processed. Fortunately, this situation does not arise often, and has little impact on the clustering result: both on core points and noise points.
2. DBSCAN cannot cluster data sets well with large differences in densities.

**3.3. Hierarchical Methods**

These methods construct the clusters by recursively partitioning the instances in either a top-down or bottom-up fashion. These methods can be subdivided as following: **Agglomerative**: Start with the points as individual clusters and, at each step, merge the closest pair of clusters. This requires defining a notion of cluster proximity. **Divisive**: Start with one, all-inclusive cluster and, at each step, split a cluster until only singleton clusters of individual points remain. In this case, we need to decide which cluster to split at each step and how to do the splitting. The result of the hierarchical methods is a dendrogram, representing the nested grouping of objects and similarity levels at which groupings change. A clustering of the data objects is obtained by cutting the dendrogram at the desired similarity level. The merging or division of clusters is performed according to some similarity measure, chosen so as to optimize some criterion.

**Single-link clustering (min**)- methods that consider the distance between two clusters to be equal to the shortest distance from any member of one cluster to any member of the other.

**Complete-link clustering (max)** - methods that consider the distance between two clusters to be equal to the longest distance from any member of one cluster to any member of the other cluster.

**Average-link clustering**- methods that consider the distance between two clusters to be equal to the average distance from any member of one cluster to any member of the other cluster.

**Disadvantages** of the single-link clustering and the average-link clustering can be summarized as follows: Single-link clustering has a drawback known as the “chaining effect“: A few points that form a bridge between two clusters cause the single-link clustering to unify these two clusters into one.

Average-link clustering may cause elongated clusters to split and for portions of neighbouring elongated clusters to merge.

The complete-link clustering methods usually produce more compact clusters and more useful hierarchies than the single-link clustering methods, yet the single-link methods are more versatile.

**Advantages**

1. Versatility: The single-link methods, for example, maintain good performance on data sets containing non-isotropic clusters, including well separated, chain-like and concentric clusters.
2. Multiple partitions hierarchical methods produce not one partition, but multiple nested partitions, which allow different users to choose different partitions, according to the desired similarity level. The hierarchical partition is presented using the dendrogram.

**Disadvantages**

Inability to scale well — The time complexity of hierarchical algorithms is at least O(m2) (where m is the total number of instances), which is non-linear with the number of objects. Clustering a large number of objects using a hierarchical algorithm is also characterized by huge I/O costs.

Hierarchical methods can never undo what was done previously. Namely there is no back-tracking capability.

Algorithm

1. Compute the proximity matrix, if necessary.
2. Repeat Merge the closest two clusters.
3. Update the proximity matrix to reflect the proximity between the new cluster and the original clusters.
4. Until Only one cluster remains

**4. Data Streaming Algorithms**

Since stream data naturally imposes a one-pass constraint on the design of the algorithms, it becomes more difﬁcult to provide such a ﬂexibility in computing clusters over different kinds of time horizons using conventional algorithms.

Therefore, a natural design to stream clustering would be separate out the process into an online micro-clustering component and an ofﬂine macro-clustering component. The online microclustering component requires a very efﬁcient process for storage of appropriate summary statistics in a fast data stream. The ofﬂine component uses these summary statistics in conjunction with other user input in order to provide the user with a quick understanding of the clusters whenever required

* 1. **Microclustering**

The micro-clustering framework is designed to capture summary information about the data stream, in order to facilitate clustering and analysis over different time horizons. This summary information is deﬁned by the following structures:

**• Micro-clusters**: We maintain statistical information about the data locality in terms of microclusters. These micro-clusters are deﬁned as a temporal extension of the cluster feature vector . The additivity property of the micro-clusters makes them a natural choice for the data stream problem.

• **Pyramidal Time Frame**: The micro-clusters are stored at snapshots in time which follow a pyramidal pattern. This pattern provides an effective trade-off between the storage requirements and the ability to recall summary statistics from different time horizons.

The summary information in the micro-clusters is used by an ofﬂine component which is dependent upon a wide variety of user inputs such as the time horizon or the granularity of clustering

* 1. **Clustream**

The CluStream method is a method of clustering data streams, based on the concept of microclusters. Microclusters are data structures which summarize a set of instances from the stream, and is composed of a set of statistics which are easily updated and allow fast analysis. CluStream has two phases. In the online phase, a set of microclusters are kept in main memory; each instance coming from the input stream can then be either appended to an existing microcluster or created as a new microcluster. Space for the new microcluster is created either by deleting a microcluster (by analyzing its expiration timestamp) or by merging the two closest microclusters. The offline phase will apply a weighted k-means algorithm on the microclusters, to obtain the final clusters from the stream. The advantage is that the amount of information archived is controlled by a user specified maximum number of micro-clusters with the algorithm attempting to capture as much detail as memory constraints allow. Its biggest disadvantage is that the radius of clustering continuously increases with the inflowing of data, and as it doesn't eliminate “old data” online, more and more data will increase the cost of process.

* 1. **Denstream**

Density-based methods construct a density proﬁle of the data for clustering purposes. The data is separated out into density-connected regions. These densities connected regions may be of different shapes and sizes. One of the advantages of density-based algorithms is that an implicit shape is not assumed for the clusters. In density-based clustering, connected regions of high density may often have arbitrary shapes. Another aspect of density based clustering is that it does not pre-decide the number of clusters. Rather, a threshold on the density is used in order to determine the connected regions. The main challenge in the stream scenario is to construct density-based algorithms which can be efﬁciently executed in a single pass of the data, since the process of density estimation may be computationally intensive. One of the methods extends the micro-clustering technique to this case, by relaxing the constraint on the number of micro-clusters, and imposing a constraint on the radius and “weight” of each micro-cluster. DenStream is a density-based stream clustering algorithm that extends the DBSCAN algorithm. Similar to CluStream, DenStream uses micro-clusters to capture synopsis information of data streams; its online component continually updates the microclusters collection. Each micro-cluster has a center and a radius that are derived from its clustering feature vector.Given threshold values for the weight and radius, there are three types of micro-clusters: a core micro-cluster, a potential core micro-cluster, and an outlier micro-cluster. For the ofﬂine components, it applies DBSCAN on these kinds of microclusters; a cluster is created as a group of micro-clusters that are dense and close to another

* 1. **HP-stream**

The micro-clustering method can also be extended to the case of high dimensional projected stream clustering. The algorithms is referred to as HPSTREAM. The basic idea is to use an (incremental) algorithm in which we associate a set of dimensions with each cluster. HPStream maintains micro-clusters to capture the summary information about the data stream. Furthermore, each micro-cluster consists of a set of relevant attributes, which can be considered its subspace. When a new data instance arrives, the average distance between the new instance and each cluster is computed. Only relevant attributes of the clusters are utilized in the distance computation. Then, the new instance is assigned to the closest cluster if their distance does not exceed a limiting range, a multiple of the cluster’s radius. Moreover, the statistical properties of the closest cluster are also updated. HPStream only maintains a ﬁx number of micro-clusters. When the number of clusters reaches a maximum value, it removes the oldest cluster to give space for a new one.HPStream improves CluStream to work for high-dimensional data streams

**References**

[1] Fahad, A.; Alshatri, N.; Tari, Z.; Alamri, A.; Khalil, I.; Zomaya, A.Y.; Foufou, S.; Bouras, A.:”A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis”, IEEE Transactions on Emerging Topics in Computing,Year: 2014, Volume: 2, Issue: 3 Pages: 267 – 279.

[2] Kehe Wu; Wenjing Zeng; Tingting Wu; Yanwen An “Research and improve on K-means algorithm based on hadoop” ,IEEE International Conference on Software Engineering and Service Science (ICSESS), Year: 2015 ,Pages: 334 – 337.

[3] Weihua Hu; Mingzhong Cheng; Guoping Wu; Liang Wu," Research on Parallel Data Stream Clustering Algorithm Based on Grid and Density”, Computer Science and Mechanical Automation (CSMA), 2015 , Pages: 70 – 75.

[4] Yogita, Y. ,“Clustering techniques for streaming data-a survey”, IEEE 3rd International Advance Computing Conference (IACC), Year:2013 Page(s):951 – 956.

[5] Amini, A.; Teh Ying Wah; Saybani, M.R.; Yazdi, S.R.A.S..”A study of density-grid based clustering algorithms on data streams”, Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD),Year: 2011, Volume: 3 Pages: 1652 – 1656.

[6] Kumar, M.; Sharma, A.;” Mining of data stream using DenStream clustering algorithm”, IEEE International Conference in MOOC Innovation and Technology in Education (MITE) ,Year: 2013 Pages: 315 – 320.

[7] Rui Xu;Donald Wunsch II ;“Survey of Clustering Algorithms “,IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 16, NO. 3, MAY 2005,Pages:645-675.

[8] Amini A, Wah TY, Saboohi H. On density-based data streams clustering algorithms: A survey. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 29(1): 116–141 Jan. 2014.

[9] Hai-LongNguyen , Yew-KwongWoon , Wee-KeongNg;” A survey on data stream clustering and classiﬁcation”,Year:2014.

[10] Sanjay Chakraborty, Prof. N.K.Nagwani , Lopamudra Dey;” Performance Comparison of Incremental K-means and Incremental DBSCAN Algorithms”, Volume 27– No.11, August 2011,Pages:14-18.

[11] Yogita, Durga toshniwal, “Clustering Techniques for Streaming Data–A Survey”,Year:2012,Pages:951-956.

[12] Woong-Kee Loh, Young-Ho Park,” A Survey on Density-Based Clustering Algorithms”,Year:2014,Pages”775-780.

[13] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu. A framework for clustering evolving data streams. In Proc. VLDB, pages 81–92, 2003.

[14] Prajesh P Anchalia,” Improved MapReduce k-Means Clustering Algorithm with Combiner”,Year:2014,Pages:385-390.