Penalty Parameter Selection for Hierarchical Data Stream Clustering

Article *in* Procedia Computer Science · December 2016

DOI: 10.1016/j.procs.2016.03.005

Amol Bhagat
Prof Ram Meghe College of Engineering and Management, Badnera, Amravati
49 PUBLICATIONS **135** CITATIONS

Priti Abhay Khodke
Professor Ram Meghe College of Engineering and Management
8 PUBLICATIONS **6** CITATIONS

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 79 (2016) 24 – 31

Procedia Computer Science

7th International Conference on Communication, Computing and Virtualization 2016

Penalty parameter selection for hierarchical data stream clustering

Amol Bhagat[a], Nilesh Kshirsagar[a], Priti Khodke[a], Kiran Dongre[a], Sadique Ali[a]

[a]*Innovation and Entrepreneurship Development Center, Prof Ram Meghe College of Engg and Mgmt, Badnera, Amravati, 444701, India*

Abstract

Extracting useful information from large sets of data is the main task of data mining. Clustering is one of the most commonly used data mining technique. Data streams are sequences of data elements continuously generated at high rate from various sources. Data streams are everywhere and are generated by the applications like cell-phones, cars, security sensors, televisions and so on. Partitioning data streams into sets of meaningful subclasses is required for proper and efficient mining of intended data. Identifying the number of clusters required for the precise clustering of data streams is an open research area. This paper gives the overview of the hierarchical data stream clustering algorithms. It also compares the performance analysis of the different algorithms under hierarchical clustering techniques for data streams. Different data clustering tools are also explained and compared in this paper. It also applies the proper hierarchical clustering algorithm to the standard datasets taken as input and the expected result must be the clustered data which is well versed, properly arranged. This paper addresses the issue of identifying the number of clusters by proposed penalty parameter selection approach. The approaches presented in this paper are helpful for the researchers in the field of data stream clustering and data mining.

Keywords: Clustering algorithms; data mining; data streams clustering; hierarchical clustering; penalty parameter selection

1. Introduction

Data mining or knowledge discovery in databases (KDD) is the process of extracting the valuable information or knowledge from the large datasets or repositories for better decision making. Data mining is the process of analyzing the data from different perspectives and summarizing it into useful information, finding the valuable patterns [1] from large amount of data. It has importance in finding the patterns, forecasting, and discovery of knowledge and used in almost every industry where data is generated. Data Mining represents a process developed to examine large amounts of data routinely collected. The term also refers to a collection of tools used to perform the process. Data mining is used in most areas where data are collected such as marketing, health, communications, etc. [2]. Data mining tools are used for analyzing the data from many dimensions or angles, categorize it and summarize the

doi:10.1016/j.procs.2016.03.005

relationship between them. Industries such as banking, insurance, medicine and retailing commonly uses data mining to reduce cost, enhance research and increase sales.

Data streams have some unique features as: huge or possibly infinite volume, dynamically changing, flowing in and out in a fixed order, allowing only one or a small number of scans and demanding fast response time. Clustering is a process of putting similar data into groups. Clustering can be considered the most important unsupervised learning technique so as every other problem of this kind; it deals with finding a structure in a collection of unlabeled data. Clustering algorithms are used to organize data, categorize data, for data compression and model construction, for detection of outliers etc. Common approach for all clustering techniques is to find clusters centre [4] that will represent each cluster. The data stream is motivated by emerging applications involving massive data sets for example, consumer click streams and telephone records, bulky sets of web pages, multimedia data, and financial transactions and so on. It raises new problems for the data stream community in terms of how to mine continuous arrival of high speed data items [5].

Data stream can be classified into online streams and offline streams. Online Data stream mining used in a number of real world applications, including network traffic monitoring, intrusion detection and credit card fraud detection. Offline data stream mining used in like generating report based on web log streams. Clustering data streams is commonly a difficult task. Data stream clustering is clustering of data that arrives continuously such as telephone records, multimedia data, financial transactions etc. A challenge imposed by continuously arriving data streams is to analyze them and to modify the models that explain them as new data arrives [3]. A data stream clustering algorithm requires much greater functionality in discovering and exploring clusters over different portions of the stream. A data stream clustering algorithm must provide the flexibility to compute clusters over user defined time periods in online time. Data stream clustering techniques are highly helpful to cluster the similar data items in data streams and also to detect the outliers, so they are called cluster based outlier detection [5, 12].

This paper proposes the hierarchical clustering of a data stream subject to the condition that intra-cluster similarity is maximized and the inter-cluster similarity is minimized. Data stream clustering techniques are highly helpful to cluster the similar data items in data streams and also to detect the outliers [5]. Outlier detection aims to detect the objects which are having different behavior than normal objects. For this the performance parameters used in different hierarchical clustering techniques are analyzed and some of the parameters are selected as penalty parameters. The proper penalty parameter based clustering mechanism is utilized for clustering the data streams. The main focus of this paper is on penalty parameters selection for hierarchical data stream clustering by analyzing existing hierarchical clustering algorithms, by identifying the performance parameters used by existing clustering algorithms, by extracting the valuable patterns from the massive amount of data and datasets, by finding out the appropriate ways of representing and dealing with the evolution of clusters in a data stream, and by determining best penalty parameters for getting the required number of clusters.

2. Related Work

Clustering problem is addressed in many contexts by many researchers in many disciplines and it became a broad and useful in exploratory data analysis. Clustering is also useful in exploratory pattern analysis, grouping, decision making, and machine learning situations such as data mining, document retrieval, image segmentation and pattern classification and also for outlier detection. There are various clustering algorithms available for clustering all are having their pros and cons. This motivated to analyse the penalty parameters such that the clustering quality is maintained without sacrificing. Tian Zhang et al. proposed an agglomerative hierarchical clustering method named BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies), and verified that it was especially suitable for large databases. BIRCH incrementally and dynamically clusters incoming multi-dimensional metric data points [7] so that best quality clusters can be produced with available resources. BIRCH can typically produce a good cluster with a single scan of the data, and improve the quality further with a few additional scans of the data. BIRCH was also the first clustering algorithm proposed in the database area that can handle noise effectively. This method has been designed so as to minimize the number of I/O operations. BIRCH process begins by partitioning objects hierarchically using tree structure and then applies other clustering algorithms to refine the clusters. It incrementally and dynamically clusters incoming data points and try to produce the best quality clustering with the available resources like available memory and time constraints. BIRCH is one of the best-known large-data clustering algorithms, and is generally viewed as a benchmark to compare against other clustering algorithms. However, BIRCH does not provide formal guarantees on the quality of the clustering thus produced.

Sudipto Guha et al. proposed a new hierarchical clustering algorithm called CURE [8] that is stronger to outliers, and identifies clusters having non-spherical shapes and wide variances in size. This is achieved in CURE process by representing each cluster by a certain fixed number of points that are generated by selecting well scattered points from the cluster and then shrinking them toward the center of the cluster by a specified fraction. To handle large databases, CURE employs a combination of random sampling and partitioning. Along with the description of CURE algorithm, the author also described, type of features it uses, and why it uses different techniques. CURE is an agglomerative hierarchical clustering algorithm that creates a balance between centroid and all point approaches. It is confirmed by the experiments that the quality of clusters produced by CURE is much better than those found by other existing algorithms. Furthermore, it is demonstrated that random sampling and partitioning enable CURE to not only outperform other existing algorithms but also to scale well for large databases without sacrificing clustering quality. CURE is more robust to outliers, and identifies clusters having non-spherical shapes and wide variances in size. CURE achieves this by representing each cluster by a certain fixed number of points that are generated by selecting well scattered points from the cluster and then shrinking them toward the centre of the cluster by a specified fraction.

ROCK is a robust agglomerative hierarchical clustering algorithm based on the notion of links. It is also appropriate for handling large data sets. For merging data points, ROCK employs links between data points not the distance between them. ROCK algorithm is most suitable for clustering data that have Boolean and categorical attributes. ROCK not only generate better quality cluster than traditional algorithms but also exhibit good scalability property. CHEMELEON is an agglomerative hierarchical clustering algorithm that uses dynamic modelling [13]. It is a hierarchical algorithm that measures the similarity of two cluster based on dynamic model. The merging process using the dynamic model facilitates discovery of natural and homogeneous clusters. The algorithm is proven to find clusters of diverse shapes, densities, and sizes in two-dimensional space. CHEMELEON is an efficient algorithm that uses a dynamic model to obtain clusters of arbitrary shapes and arbitrary densities.

Linkage algorithms [10] are agglomerative hierarchical methods that consider merging of clusters based on distance between clusters. In single-linkage shortest distance between two subsets is utilized. It is sensitive to the presence of outliers. It displays total insensibility to shape and size of clusters. Single-link is sensitive to the presence of outliers and the difficulty in dealing with severe differences in the density of clusters. On the other hand, displays total insensibility to shape and size of clusters. In Average-linkage average distance between two subsets is used. It is sensitive to the shape and size of clusters. It easily fails when clusters have complicated forms departing from the hyper spherical shape. Average-linkage is sensitive to the shape and size of clusters. Thus, it can easily fail when clusters have complicated forms departing from the hyper spherical shape. Complete-linkage uses longest distance between two subsets. It is not strongly affected by the outliers, but can break large clusters and has trouble with convex shapes. Complete-linkage is not strongly affected by outliers, but can break large clusters, and has trouble with convex shapes.

Bisecting k-Means (BKMS) [12] is a divisive hierarchical clustering algorithm. It was proposed by Steinbach et al. in the context of document clustering. Bisecting k-means always finds the partition with the highest overall similarity, which is calculated based on the pair wise similarity of all points in a cluster. As reported, the bisecting k-means frequently outperforms the standard k-means and agglomerative clustering approaches. In addition, the bisecting k-means time complexity is O(nk) where n is the number of items and k is the number of clusters. Advantage of BKMS is low computational cost. BKMS is identified to have better performance than k-means (KMS) agglomerative hierarchical algorithms for clustering large datasets. The comparison of all these algorithms is shown in table 1.

3. Clustering Methodologies

The clustering problem is defined as “for a given set of data points, partition them into one or more groups of similar objects. The similarity of the objects with one another is typically defined with the use of some distance measure or objective function [6]”. Some of the clustering methodologies [12] are discussed in this section. A **partition clustering method** constructs k partitions of the data, where each partition represents a cluster and k<=n.

Most of the partitioning methods are based on the distance between objects. It classifies the data into k-groups which satisfy the requirements: each group must contain at least one object and each object must belong to exactly one group. A partitioning method creates an initial partitioning. It then uses an iterative relocation technique that attempts to improve the partitioning by moving objects from one group to another. The general criterion of a good partitioning clustering is that the objects in the same cluster are 'close' or related to each other whereas objects of different clusters are 'far apart'.

Table 1. Comparison of hierarchical clustering algorithms.

Algorithm	Sensitivity to outliers	Model Type	Time Complexity	Space Complexity	Features	Limitations
BIRCH [7]	Handles noise effectively	Dynamic	O (n)	---	Scales linearly: finds a good clustering with a single scan and improves the quality with a few additional scans.	Handles only numeric data, and sensitive to the order of the data record. Favors only clusters with spherical shape and similar sizes
CURE [8]	Less sensitive to outliers	Static	O (n^2logn)	O (n)	Recognizes arbitrarily shaped clusters. Robust to the presence of outliers. Handles large data sets.	Information of total interconnectivity of objects in two clusters is ignored.
ROCK [9]	---	Static	O ($n^2 + m_m m_a + n^2$ logn)	O (min{n^2, $n m_m m_a$})	Most suitable for clustering data that have Boolean and categorical attributes.	Static modeling of the clusters to be merged.
CHAMELEON [10]	-------------	Dynamic	O (*n*(log2 n + *m*))	-------------	Obtain clusters of arbitrary shapes and arbitrary densities.	---
Single-Linkage [10]	Sensitive to outliers	---	O (n^2 log n)	O (n^2)	It displays total insensibility to shape and size of clusters.	---
Average-linkage [10]	---	---	---	---		Fails when clusters have complicated forms with hyper spherical shape.
Complete-linkage [10]	Not strongly affected by outliers	---	O (n^3)	---	Not strongly affected by the outliers.	It has trouble with convex shapes.
Leaders-sub-leaders [11]	---	---	O (ndh) h=2	O ((L-SL)d)	Computationally less expensive	----
Bisecting K-means [12]	---	---	O (nk)	---	Finds the partition with the highest overall similarity	----

A **hierarchical clustering method** creates a hierarchical decomposition of the given set of data objects. It is further classified as agglomerative hierarchical clustering and divisive hierarchical clustering. An agglomerative approach is also known as bottom-up approach which considers each object as a separate cluster initially and then successively merges the objects that are closer to one another until all objects are merged into a single cluster. An agglomerative hierarchical clustering consists of following steps.

1. Start with one point (singleton).
2. Recursively add two or more suitable clusters.
3. The process stops when k number of clusters is achieved.

The divisive approach also known as top-down approach starts with all objects into a same cluster. It then splits up into smaller clusters at each successive iteration until the each object is in one cluster or termination condition meets. The divisive hierarchical clustering consists of following steps.

1. Start with a big cluster.
2. Recursively divides into smaller clusters.
3. The process stops when k number of clusters is achieved.

The generalized steps in any hierarchical clustering algorithm are listed below.

1. Start by assigning each item to a cluster, so that if u have N items, you now have N clusters, each containing just one item. Let the distances between the clusters the same as the distances between the items they contain.
2. Find the closest pair of clusters and merge them into a single Cluster, so that now you have one cluster less.

3. Compute distances between the new cluster and each of the old clusters.
4. Repeat 2 and 3 until all items are clustered into a single cluster of size N.

One of the main reasons hierarchical clustering is used by many researchers and practitioners is that there is no need to specify the desired number of clusters, and also it can be easily illustrated with dendrogram (graphical representation). Hierarchical clusters when merged or split can never be undone. This rigidity is useful that it leads to smaller computational costs by not worrying about a combinatorial number of different choices.

Table 2. Advantages and disadvantages of clustering methodologies.

Algorithm	Advantages	Disadvantages
Partitioning Clustering Algorithm	1. Relatively scalable and simple. 2.Suitable for datasets with compact spherical clusters that are well-separated	1. Severe effectiveness degradation in high dimensional spaces 2. Poor cluster descriptors 3.Reliance on the user to specify the number of clusters in advance 4. High sensitivity to initialization phase, noise and outliers 5. Inability to deal with non-convex clusters of varying size and density.
Hierarchical Clustering Algorithm	1. No need to define number of clusters in advance. 2. Calculates a whole hierarchy of clusters. 3. Good result visualizations Joint into the methods. 4. Uses dendrogram for graphical representation. 5. Embedded flexibility regarding the level of granularity. 6. Point linkages, e.g. taxonomy trees can be solved.	1. Inability to make corrections once the splitting/merging decision is made. 2. Lack of interpretability regarding the cluster descriptors. 3. Vagueness of termination criterion. 4. Prohibitively expensive for high dimensional and massive datasets. 5. Severe effectiveness degradation in high dimensional spaces
Density Based Clustering	1. It discovers clusters of arbitrary shapes. 2.It handles noise and outliers efficiently.	1.High sensitivity to the setting of input parameters 2. Poor cluster descriptors 3. Unsuitable for high-dimensional datasets
Grid-based Clustering	1. It is fast as there is no distance computation. 2. Clustering is performed on summaries 3. Easy to determine which clusters are neighbouring. 4. Shapes are limited to union of grid cells. 5. Query independent, parallelizable, incremental update.	1. All the clusters boundaries are either horizontal or vertical and no diagonal boundary is detected.
Model-based Clustering	1. Clusters can be characterized by a small number of parameters. 2. Result satisfies the statistical assumptions of the generative models.	1. Computationally expensive if the number of distributions is large or the data set contains very few observed data points. 2. It needs large data sets. 3. It is hard to estimate the number of clusters.
Constraint-based Clustering	1. Effectively improved the baseline MMC (maximum margin clustering) 2. It takes very less computational time. 3. Increasing number of constraints reduced the computational time.	1. It is not effective when the data is not uniformly distributed in the output domain. 2. Clustering speed is slow. 3. High dimensional data clustering is difficult.

Some clustering methods are developed on the basis of ‘density’ are called as **density-based clustering**. The cluster continue to grow as long as the density (number of objects or data points) in the neighborhood exceeds some threshold. Such a method can be used to filter out outliers and discover clusters of arbitrary shape. DBSCAN, OPTICS, and DENCLUE are some of the examples of such algorithms. **Grid-based clustering** methods quantize the object space into a finite number of cells that form a grid-structure. All of the clustering operations are performed on a grid structure. The main advantage of this approach is its fast processing time which is independent of the number of objects and dependent on the number of cells in each dimension in quantized space. STING, and Wave Cluster are examples of such algorithms. A **Model-based clustering** method hypothesizes a model for each of the clusters and find the best fit of the data to the given model. This algorithm locates clusters by constructing a density function that reflects the spatial distribution of the data points. It automatically determines the number of clusters based on a standard statistics. It is also a robust clustering method as they take outliers into account. EM, COBWEB, and SOM are the methods based on model-based clustering.

The **constraint-based clustering** approach performs clustering by incorporation of user-specified or application-oriented constraints. A constraint expresses a user’s expectation or describes properties of the desired clustering results and provides an effective way for communicating with the clustering process. The choice of clustering algorithm depends both on the type of data available and on the particular purpose of the application. Some clustering algorithms integrate the ideas of several clustering methods so that it is sometimes difficult to classify a given algorithm as uniquely belonging to one clustering method category. Table 2 compares the advantages and disadvantages of the above discussed methodologies. Due to certain advantages of hierarchical clustering, hierarchical clustering is used in this paper.

4. Proposed Penalty Parameter-based Hierarchical Data Stream Clustering

The data stream as is a continuously arriving data streams and has huge data, the challenge is to analyse them and to modify the models as the new data arrives. The data streams require online mining, in which the mining of the data is done in a continuous fashion. The system need to have the capability to perform an offline analysis as well based on the user interests. The main problem in data stream mining means evolving data is more difficult to detect in this techniques therefore unsupervised methods should be applied. However, clustering techniques can lead us to discover hidden information [14]. Data streams can be solved using the methodologies of data stream clustering, data stream classification, frequent pattern mining, sliding window, association technique and so on. Fig. 1 shows the proposed hierarchical clustering architecture. The data recorded initially is a raw data taken from standard datasets. The raw data need to be pre-processed before applying the clustering algorithm. Pre-processing includes: data cleaning which fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies; data integration of multiple data bases, data cubes, or files; data reduction using normalization and aggregation; data transformation which obtains reduced representation in volume but produces the same or similar

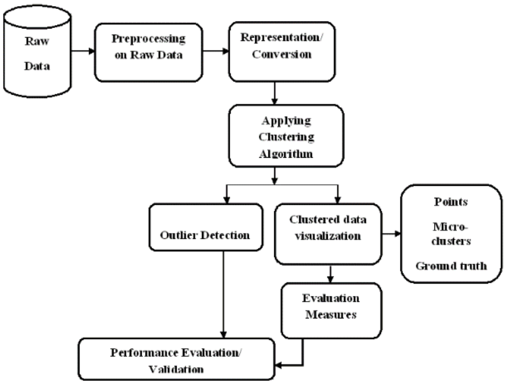
Raw Data
Preprocessing on Raw Data
Representation/ Conversion
Applying Clustering Algorithm
Outlier Detection
Clustered data visualization
Points
Micro-clusters
Ground truth
Evaluation Measures
Performance Evaluation/ Validation

Fig. 1. Proposed hierarchical data stream clustering architecture.

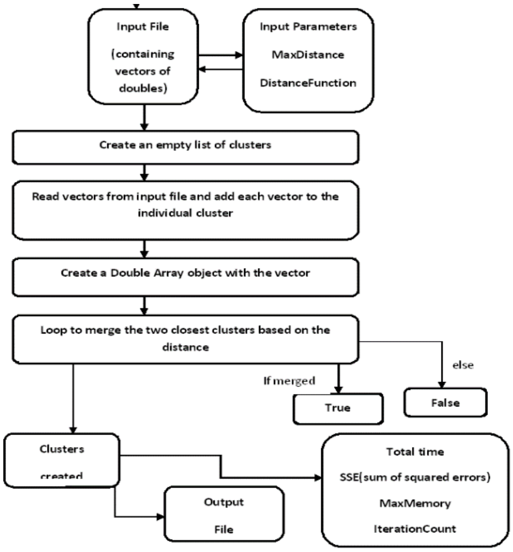
Input File
(containing vectors of doubles)
Input Parameters
MaxDistance
DistanceFunction
Create an empty list of clusters
Read vectors from input file and add each vector to the individual cluster
Create a Double Array object with the vector
Loop to merge the two closest clusters based on the distance
else
If merged
True
False
Clusters created
Output File
Total time
SSE(sum of squared errors)
MaxMemory
IterationCount

Fig. 2. Proposed Hierarchical Data Stream Clustering Algorithm.

analytical results; representation the data is put up into a suitable form for further processing; clustering is applied to the data considering the matching algorithm, presentation of result and the choice of parameters; and validation of the performance and comparison with the other clustering algorithms. The steps in the proposed hierarchical clustering algorithm are shown in Fig 2. The proposed adaptive hierarchical clustering process with penalty parameter selection is shown in Fig 3.

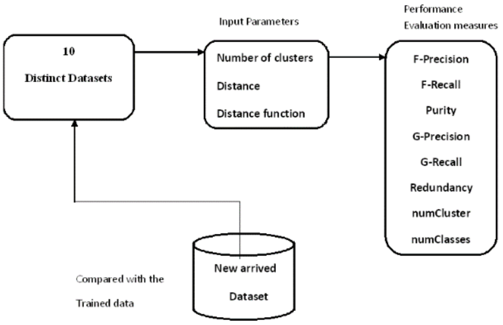

Fig. 3. Proposed Adaptive Hierarchical Data Stream Clustering using Penalty Parameters.

The 10 distinct datasets are considered in order to have a trained data. The datasets considered must be distinct in order of their attributes and instances. So, every time when the new dataset is taken as an input file, it is compared with the trained data with the number of attributes, number of instances and type of datasets. The input parameters such as number of clusters, distance and distance functions are provided as an input to the clustering algorithms. Based on the input parameters the output generated as a clustered data and is evaluated on the basis of performance evaluation measures such as F-measures, G-measures, purity, redundancy, numCluster and numClasses.

5. Performance Evaluation and Results

Various performance measures are utilized for the evaluation of the proposed clustering methodology for some standard data streams. The results of evaluation are presented in this section. In the field of information retrieval, precision is the fraction of retrieved documents that are relevant to the query:

$$\text{Precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

Table 3. Evaluation results of the proposed adaptive hierarchical clustering algorithm.

Data Stream	Precision	Recall	Purity	G-Pre	G-Recall	Redundancy
RandomRBF Generator Events –n	1.0	0.83	0.99	1	0.91	0
Diabetes.arff	0.99	0.83	0.99	1	0.91	0
Autos.arff	0.99	0.83	0.99	1	0.92	0
Mushroom.arff	0.99	0.83	0.99	1	0.91	0
Splice.arff	0.95	0.79	0.95	1	0.91	0.05
Vowel.arff	0.81	0.68	0.81	1	0.88	0.24

Precision takes all retrieved documents into account, but it can also be evaluated at a given cut-off rank, considering only the topmost results returned by the system. Recall in information retrieval is the fraction of the documents that are relevant to the query that are successfully retrieved.

$$\text{Recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

The F-score is often used in the field of information retrieval for measuring search, document classification, and query classification performance. The F-score is also used in machine learning. The F-score has been widely used in the natural language processing literature, such as the evaluation of named entity recognition and word

$G=\sqrt{\text{precision . recall}}$

segmentation. While the F-measure is the harmonic mean of Recall and Precision, the G-measure is the geometric mean. Purity is a simple and transparent evaluation measure. Each cluster is assigned to the class which is most frequent in the cluster, and then the accuracy of this assignment is measured by counting the number of correctly assigned documents and dividing by N Formally:

$$\text{Purity}(\Omega, €) = \frac{1}{N}\sum_{i=1}^{n} \max | w_k \cap c_j |$$

where $\Omega = \{w_1, w_2, \ldots\ldots, w_k\}$ is the set of clusters and $€ = \{c_1, c_2, \ldots..c_j\}$ is the set classes. W_k is interpreted as the set of documents in W_k and C_j as the set of documents in C_j in above Equation. One way to measure redundancy between two variables is in terms of their correlation coefficient; however, this captures only linear dependencies among random variables. Table 3 shows various results obtained as per these measures for different data streams using the proposed hierarchical clustering algorithm. It can be observed from the table the proposed approach has more than 95% accuracy with on average less than 5% accuracy.

6. Conclusion and Future Scope

In this paper detailed analysis of hierarchical clustering algorithms is provided. A novel scheme for data stream clustering using hierarchical clustering by adaptable penalty parameter selection is proposed. The main reason the hierarchical clustering is selected is because there is no need to specify the desired number of clusters and also its graphical representation named dendrogram is easy to illustrate. The presented work evaluates the performance of the hierarchical data stream clustering. The performance parameters used are precision, recall, purity, G-precision, G-Recall, and. From experimental results it is observed that the hierarchical clustering algorithm outperforms the other clustering algorithms in case of accuracy and purity. The proposed adaptive hierarchical clustering algorithm can be applied to the high dimensional datasets. Finding the appropriate ways of representing and dealing with the evolution of clusters in a data stream is still not addressed.

References

1. Bhagat A. P., Harle B. R. Materialized view management in peer to peer environment. *Proceedings of the International Conference & Workshop on Emerging Trends in Technology* 2011. p. 480-484.
2. Yoon Shinsook, Ryu Chang-Keun. Clustering for Context Inference in the Data Stream Mining. *International Journal of Software Engineering and Its Applications* 2015. Vol. 9, No. 1, p. 105-112.
3. Vijayarani S., Jothi P. An Efficient Clustering Algorithm for Outlier Detection in Data Streams. *International Journal Of Advanced Research in Computer and Communication Engineering* 2013. Vol. 2, Issue 9.
4. Vijaya P.A., Murty M. Narasimha, Subramanian D.K. Leaders–Sub-leaders: An efficient hierarchical clustering algorithm for large data sets. *Pattern Recognition Letters*, 2004. p. 505-513.
5. Almeida J.A.S., Barbosa L.M.S., Pais A.A.C.C. and Formosinho S.J. Improving Hierarchical Cluster Analysis: A new method with outlier detection and automatic clustering. *Chemo metrics and Intelligent Laboratory Systems*, 2007. p. 208-217.
6. Zhang Xiangliang, Furtlehner C., Germain-Renaud C, Sebag M. Data Stream Clustering With Affinity Propagation. *IEEE Transactions on Knowledge and Data Engineering*, 2014. vol. 26, no. 7, p.1644,1656.
7. Zhang T., Ramakrishnan R., Livny M. BIRCH: An efficient data clustering method for very large databases. *in proc. SIGMOD, Montreal, QC, Canada*, 1996. p. 103-114.
8. Guha S., Rastogi R., Shim K. CURE: An Efficient Clustering Algorithm for Large Databases. *Proc. 1998 ACM Special Interest Group on Management of Data*, 1998. p. 73-84.
9. Guha, S., Rastogi, R., Kyuseok Shim. ROCK: A robust clustering algorithm for categorical attributes. *Proceedings 15th International Conference on Data Engineering, 1999.*, p.512-521.
10. Karypis G., Han E-H, Kumar V. CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling. *Computer*, 1999. vol. 32, no. 8, p. 68-75.
11. Rodrigues P.P., Gama J., Pedroso J. P. Hierarchical Clustering of Time-Series Data Streams. *IEEE Transactions on Knowledge and Data Engineering*, 2008. vol. 20, no. 5, p. 615-627.
12. Badase, P.S., Deshbhratar, G.P., Bhagat, A.P. Classification and analysis of clustering algorithms for large datasets. *Proceedings of 2015 IEEE International Conference on Innovations in Information, Embedded and Communication Systems*, 2015. p. 1-5.
13. Demsar J., Curk T., Erjavec A. Orange: Data Mining Toolbox in Python. *Journal of Machine Learning Research*, 2013, vol. 14, p. 2349–2353.
14. Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 2011. vol. 12, p. 2825–2830.