JCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Vol.-7, Issue-5, May 2019 E-ISSN: 2347-2693

Measuring Different Tasks for Unstructured Data and High Speed Data in Data Stream Mining

K. Rajasekhar[1*], P. Venkata Maheswara[2]

[1,2]Dept. of Computer Science, Annamacharya Institute of Technology & Sciences, Tirupati, India

Corresponding Author: kotapati.raja@gmail.com, Tel.: +91-9550469129

DOI: https://doi.org/10.26438/ijcse/v7i5.582589 | Available online at: www.ijcseonline.org

Accepted: 09/May/2019, Published: 31/May/2019

Abstract**—** Data streams are continuous flows of data. Examples of data streams include network traffic, sensor data, call center records and so on. One important problem is mining data streams in extremely large databases (e.g. 100 TB). Satellite and computer network data can easily be of this scale. However, today's data mining technology is still too slow to handle data of this scale. In addition, data mining should be a continuous, online process, rather than an occasional one-shot process. Organizations that can do this will have a decisive advantage over ones that do not.

One particular instance is from high speed network traffic where one hopes to mine information for various purposes, including identifying anomalous events possibly indicating attacks of one kind or another. A technical problem is how to compute models over streaming data, which accommodate changing environments from which the data are drawn. This is the problem of "concept drift" or "environment drift." This problem is particularly hard in the context of large streaming data. How may one compute models that are accurate and useful very efficiently? For example, one cannot presume to have a great deal of computing power and resources to store a lot of data, or to pass over the data multiple times. Hence, incremental mining and effective model updating to maintain accurate modeling of the current stream are both very hard problems.

Keywords**—** Data Stream, Data Stream Mining, Concept Drift/Environment Drift

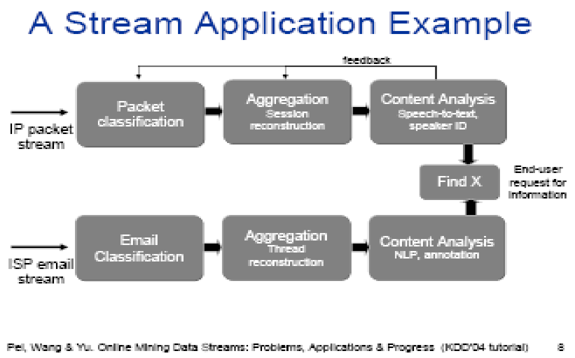
I. INTRODUCTION

Data Stream Mining is the process of extracting knowledge structures from continuous, rapid data records. A data stream is an ordered sequence of instances that in many applications of data stream mining can be read only once or a small number of times using limited computing and storage capabilities. Examples of data streams include computer network traffic, phone conversations, ATM transactions, web searches, and sensor data. Data stream mining can be considered a subfield of data mining, machine learning, and knowledge discovery.

In many data stream mining applications, the goal is to predict the class or value of new instances in the data stream given some knowledge about the class membership or values of previous instances in the data stream. Machine learning techniques can be used to learn this prediction task from labeled examples in an automated fashion. In many applications, the distribution underlying the instances or the rules underlying their labeling may change over time, i.e. the goal of the prediction, the class to be predicted or the target value to be predicted, may change over time. This problem is referred to as concept drift.

The **concept drift** means that the statistical properties of the target variable, which the model is trying to predict, change over time in unforeseen ways. This causes problems because the predictions become less accurate as time passes.

The term *concept* refers to the quantity to be predicted. More generally, it can also refer to other phenomena of interest besides the target concept, such as an input, but, in the context of concept drift, the term commonly refers to the target variable.


A Stream Application Example

Pei, Wang & Yu. Online Mining Data Streams: Problems, Applications & Progress (KDD'04 tutorial) 8

Fig. 1. Example of Stream Application.

II. Related Work

Scaling Data Mining Algorithms:
Most data mining algorithms, assume that the data fits into memory. Although success on large data sets is often claimed, usually this is the result of sampling large data sets until they fit into memory. A fundamental challenge is to scale data mining algorithms as:

1. the number of records or observations increases;
2. the number of attributes per observation increases;
3. the number of predictive models or rule sets used to analyze a collection of observations increases; and,
4. as the demand for interactivity and real-time response increases.

Not only must distributed, parallel, and out-of-memory versions of current data mining algorithms be developed, but genuinely new algorithms are required. For example, association algorithms today can analyze out-of-memory data with one or two passes, while requiring only some auxiliary data be kept in memory.

Data stream classification:

a) *Single model* incremental classification: Strives to cope with the concept-drift [8, 9, 10]
b) *Ensemble-model* based classification (chunk driven): An ensemble is maintained rather than a single model
i Supervised: The models are trained in supervised fashion
ii.Semi-supervised: The models are trained with semi-supervised technique using both labeled and unlabeled data.
iii. Active learning: Data are chosen selectively for labeling, and those labeled data are used for training [4]

III. Methodology

Data observations with thousands of features or more are now common, such as profiles clustering in recommender systems, personality similarity, genomic data, financial data, web document data and sensor data. However, high-dimensional data poses different challenges for clustering algorithms that require specialized solutions. Recently, some researches have given solutions on high-dimensional problem. In the well-known survey [1] the problem is introduced in a very illustrative way and some approaches are sketched. There is no clear distinction between different sub problems (axis-parallel or arbitrarily oriented) and the corresponding algorithms are discussed without pointing out the underlying differences in the respective problem definitions.

Our main objective is proposing a framework to combine relational definition of clustering space and divide and conquer method to overcome aforementioned difficulties and improving efficiency and accuracy in K-Means algorithm to apply in high dimensional datasets. Despite previous study in dividing whole space into subspaces vertically based on object's features [6] we apply a horizontal method to divide entire space into subspaces based on objects. We conducted some experiments on real world dataset (personality similarity) as an example of special groups of applications that are silent other methods for presenting suitable solution.

A. K-Means Clustering Algorithm

K - Means is regarded as a staple of clustering methods, due to its ease of implementation. It works well for many practical problems, particularly when the resulting clusters are compact and hyper spherical in shape. The Madjid Khalilian, Norwati Mustapha, MD Nasir Suliman, MD Ali Mamat time complexity of *K* - means is$O(N.K.d.T)$ where T is the number of iterations. Since K, d, and T are usually much less than N, the time complexity of K - means is approximately linear. Therefore, K - means is a good selection for clustering large - scale data sets [7].

B. Divide AND Conquer

When the size of a data set is too large to be stored in the main memory, it is possible to divide the data into different subsets that can fit the main memory and to use the selected cluster algorithm separately to these subsets. The final clustering result is obtained by merging the previously formed clusters. This approach is known as divide and conquer [8, 9]. Specifically, given a data set with N points stored in a secondary memory, the divide - and - conquer algorithm first divides the entire data set into r subsets with approximately similar sizes. Each of the subsets is then loaded into the main memory and is divided into a certain number of clusters with a clustering algorithm. Representative points of these clusters, such as the centers of the clusters, are then picked for further clustering. These representatives may be weighted based on some rule, e.g., the centers of the clusters could be weighted by the number of points belonging to them [8]. The algorithm repeatedly clusters the representatives obtained from the clusters in the previous level until the highest level is reached. The data points are then put into corresponding clusters formed at the highest level based on the representatives at different levels.

C. Equivalency AND Similarity

We are going to describe a mathematic base for our method and definition of similarity base on relation definition. In previous studies on divide and conquer[8], dividing data has been done without any prior knowledge about data but in this study a criterion is used to divide data into partitions. This criterion can be selected base on the following discussion.

Theorem.1. Each equivalence relation R can divide S into some partitions (classes) s1,s2,…,sn where

$$a) \bigcup_{i=1}^{n} s_i = S$$

$$b) s_i \neq \varnothing |$$ proof is in context of Discrete

$$c) s_i \cap s_j = \varnothing, i \neq j$$

Mathematics Structure [10, 11].

Theorem.2. Length of vector is an equivalence relation where length is defined by

$$L(V) = \sqrt{\sum_{i=1}^{d} v_i^2} ,$$

, where d and vi are for number of dimensions and value of ith feature respectively.

Proof

a)reflexive

$$\therefore \forall V_i \in S : L(V_i) = L(V_i)$$

b)symmetric

$$\therefore \forall V_i, V_j \in S : L(V_i) = L(V_j) \Rightarrow L(V_j) = L(V_{ji})$$

c)transitive

$$\therefore \forall V_i, V_j, V_k \in S : L(V_i) = L(V_j) \wedge L(V_j) = L(V_k) \Rightarrow L(V_i) = L(V_k)$$

Outcome of theorems 1, 2 is the important result that implies length of vector can divide our problem space into subspaces with equivalency property. Although, vectors in one subspace are in the same level but they might be in different directions. The radius of subspace is L (Vi) for all vectors inside of that subspace as Vi belongs to subspace.

Theorem.3. Similarity is a compatible relation where similarity is defined by Minkowski distance [12]:

$$D_n(V_i, V_j) = \left(\sum_{k=1}^{d} \left| V_{ik} - V_{jk} \right|^n \right)^{1/n}$$ or cosine measure [13-15] that is used inner product for similarity among vectors: $\cos(V_i, V_j) = \dfrac{V_i^T \cdot V_j}{|V_i||V_j|}$

Proof

a)reflexive

$$\therefore \forall V_i \in S : D_n(V_i, V_i) = 0, \cos(V_i, V_i) = 1$$

b)symmetric

$$\therefore \forall V_i, V_j \in S : D_n(V_i, V_j) = D_n(V_j, V_i), \cos(V_i, V_j) = \cos(V_j, V_i)$$

Similarity is not an equivalence relation because it doesn't have transitive property. For example consider 3 vectors in space V1, V2, V3. If we define threshold of angle α for being similarity, it is clear that V1 is not similar with V3 base on definition because the angle is 2α more than threshold value.

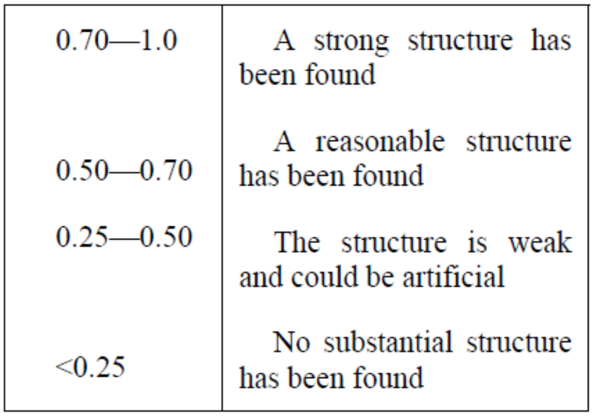
Result of theorems 2, 3 $D\ L\ n \therefore\ \subset$. We are going to device an algorithm in two steps. First it divides entire space into some subspaces based on length of vectors. As it was mentioned, samples in each subspace are equivalent but not necessarily similar. In second phase K-Means algorithm is applied in each subspace. It is suitable to overcome clustering problem for large datasets with high dimensions instead of using clustering on entire space

• **Space Dividing**: Base on length of vector which is equivalency relation, we divide space of problem into some subspaces. This has been developed by K-Means algorithm. Length of vectors are input for K-Means algorithm and output will be some subspaces or partitions which elements inside them are equivalent and ready to clustering. In other word all samples in one partition have almost same size but might be dissimilar.

• **Subspaces Clustering**: After finding subspaces, clustering algorithm is applied on each subspace and outcomes final group. Although samples in different subspaces may be similar base on cosin criterion but they are in different levels [6].

Clusters Validity: Quality of clusters should be demonstrated. K-Means uses an iterative algorithm that minimizes the sum of distances from each object to its cluster centroid, over all clusters. This algorithm moves objects between clusters until the sum cannot be decreased further. The result is a set of clusters that are as compact and well-separated as possible. You can control the details of the minimization using several optional input parameters to K-Means, including ones for the initial values of the cluster centroids, and for the maximum number of iterations. To get an idea of how well-separated the resulting clusters are, we can make a silhouette plot using the cluster indices output from K-Means. The silhouette plot displays a measure of how close each point in one cluster is to points in the neighbouring clusters. This measure ranges from +1, indicating points that are very distant from neighbouring clusters, to 0, indicating points that are not distinctly in one cluster or -1, indicating points that are probably assigned to the wrong cluster. A more quantitative way to compare the two solutions is to look at the average silhouette values for the two cases. Average silhouette width values can be interpreted as follows:

Table 1. K-Means values for overall structures

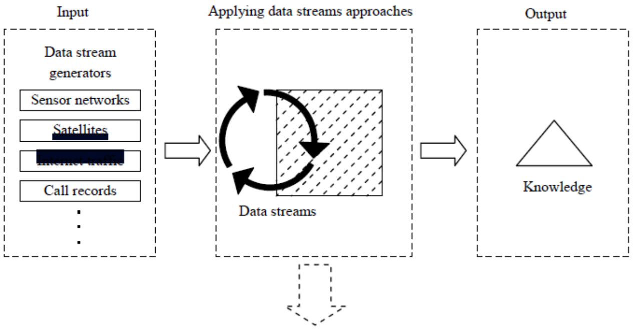


0.70—1.0	A strong structure has been found
0.50—0.70	A reasonable structure has been found
0.25—0.50	The structure is weak and could be artificial
<0.25	No substantial structure has been found

Data stream mining

High volume and potential infinite data streams are generated bye So many resources such as real-time surveillance systems, communication networks, Internet traffic, on-line transactions in the financial market or retail industry, electric power grids, industry production processes, scientific and engineering experiments, remote sensors, and other dynamic environments. In data stream model, data items can be relational tuples like network measurements and call records. In comparison with traditional data sets, data stream flows continuously in systems with varying update rate. Data streams are continuous, temporally ordered, fast changing, massive and potentially infinite. Due to huge amount and high storage cost, it is impossible to store an entire data streams or to scan through it multiple times. So it makes so many challenges in storage, computational and communication capabilities of computational systems. Because of high volume and speed of input data, it is needed to use semi-automatic interactional techniques to extract embedded knowledge from data.

Data stream mining is the extraction of structures of knowledge that are represented in the case of models and patterns of infinite streams of information. The general process of data stream mining is depicted in Figure.2.



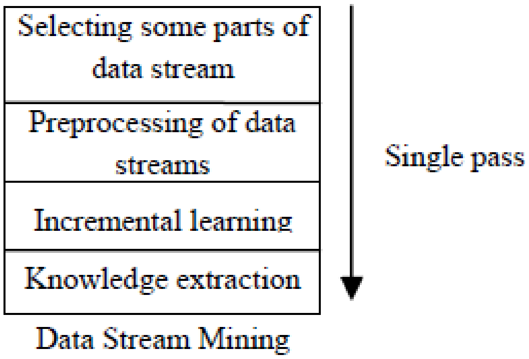


Figure.2. General process of data stream mining

For extracting knowledge or patterns from data streams, it is crucial to develop methods that analyze and process streams of data in multidimensional, multi-level, single pass and online manner. These methods should not be limited to data streams only, because they are also needed when we have large volume of data. Moreover, because of the limitation of data streams, the proposed methods are based on statistic, calculation and complexity theories. For example, by using summarization techniques that are derived from statistic science, we can confront with memory limitation. In addition, some of the techniques in computation theory can be used for implementing time and space efficient algorithms. By using these techniques we can also use common data mining approaches by enforcing some changes in data streams [2].

Some solutions have been proposed based on data stream mining problems and challenges. These solutions can be categorized to data-based and task-based solutions. This classification is depicted in Figure. 3. Data-based techniques refer to summarizing the whole dataset or choosing a subset of the incoming stream to be analyzed. Sampling, load and sketching techniques represent the former one. Synopsis data structures and aggregation represent the later one. Task-based techniques are those methods that modify existing techniques or invent new ones in order to address the computational challenges of data stream processing. Approximation algorithms, sliding window and algorithm output granularity represent this category. with using sampling in the context of data stream analysis is the unknown dataset size. Thus the treatment of data stream should follow a special analysis to find the error bounds. Another problem with sampling is that it would be important to check for anomalies for surveillance analysis as an application in mining data streams. Sampling may not be the right choice for such an application. Sampling also does not address the problem of fluctuating data rates. It would be worth investigating the relationship among the three parameters: data rate, sampling rate and error bounds.

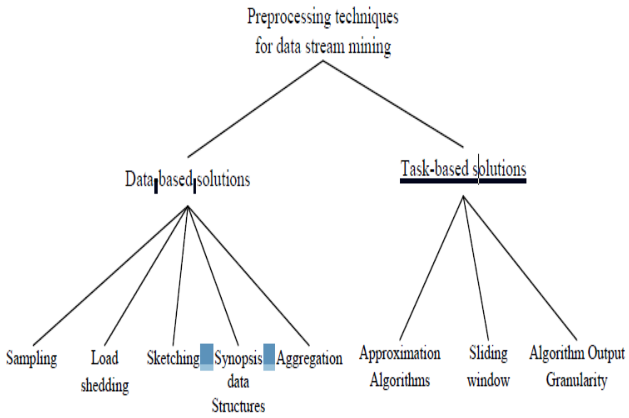


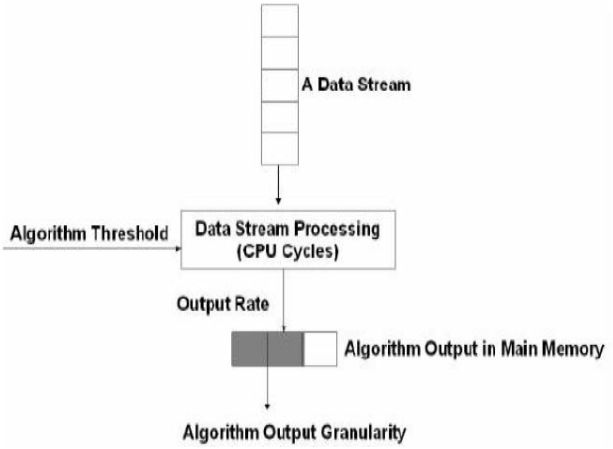
Figure.3. Classification of data stream pre-processing methods

Load shedding refers to the process of dropping a sequence of data streams. Load shedding has been used successfully in querying data streams. It has the same problems of sampling. Load shedding is difficult to be used with mining algorithms because it drops chunks of data streams that could be used in the structuring of the generated models or it might represent a pattern of interest in time series analysis [2]. Sketching is the process of randomly project a subset of the features. It is the process of vertically sample the incoming stream. Sketching has been applied in comparing different data streams and in aggregate queries. The major drawback of sketching is that of accuracy. It is hard to use it in the context of data stream mining.

Creating synopsis of data refers to the process of applying summarization techniques that are capable of summarizing the incoming stream for further analysis. Wavelet analysis, histograms, quantiles and frequency moments have been proposed as synopsis data structures. Since synopsis of data does not represent all the characteristics of the dataset, approximate answers are produced when using such data structures. The process in which the input stream is represented in a summarized form is called aggregation. This aggregate data can be used in data mining algorithms. The main problem of this method is that highly fluctuating data distributions reduce the method’s efficiency.

Approximation algorithms have their roots in algorithm design. It is concerned with design algorithms for computationally hard problems. These algorithms can result in an approximate solution with error bounds. The idea is that mining algorithms are considered hard computational problems given its features of continuality and speed and the generating environment that is featured by being resource constrained. Approximation algorithms have attracted researchers as a direct solution to data stream mining problems. However, the problem of data rates with regard with the available resources could not be solved using approximation algorithms. Other tools should be used along with these algorithms in order to adapt to the available resources. Approximation algorithms have been used in [1]. The inspiration behind sliding window is that the user is more concerned with the analysis of most recent data streams. Thus the detailed analysis is done over the most recent data items and summarized versions of the old ones.

The algorithm output granularity (AOG) introduces the first resource-aware data analysis approach that can cope with fluctuating very high data rates according to the available memory and the processing speed represented in time constraints. The AOG performs the local data analysis on a resource constrained device that generates or receive streams of information. AOG has three main stages. Mining followed by adaptation to resources and data stream rates represent the first two stages. Merging the generated knowledge structures when running out of memory represents the last stage. AOG has been used in clustering, classification and frequency counting [1]. The function of the AOG algorithm is depicted in figure.4.


A Data Stream
Algorithm Threshold
Data Stream Processing (CPU Cycles)
Output Rate
Algorithm Output in Main Memory
Algorithm Output Granularity

.

Figure.4. The AOG algorithm [15]

3. Classification of data stream challenges

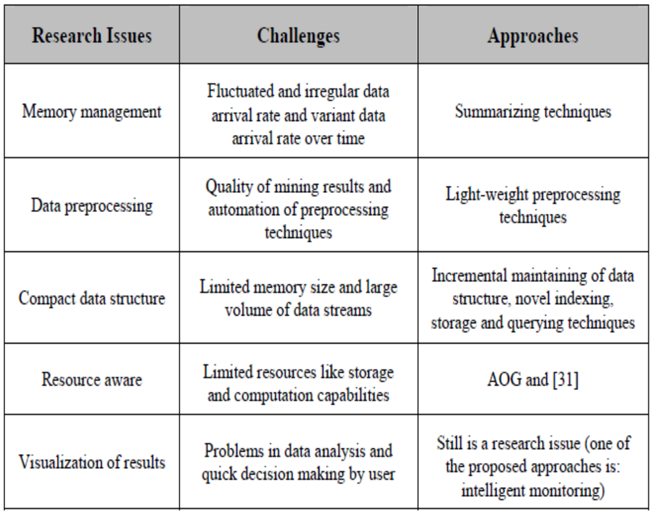
There are different challenges in data stream mining that cause many research issues in this field. Regarding to data stream requirements, developing stream mining algorithms is needed more studying than traditional mining methods. We can classify stream mining challenges in 5 categories; Irregular rate of arrival and variant data arrival rate over time, Quality of mining results, Bounded memory size and huge amount of data streams, Limited resources, e.g. ,memory space and computation power and To facilitate data analysis and take a quick decision for users. In the following each of them will be described.

One of the most important issues in data stream mining is optimization of memory space consumed by the mining algorithm. Memory management is a main challenge in stream processing because many real data have irregular arrival rate and variation of data arrival rate over time. In many applications like sensor networks, stream mining algorithms with high memory cost is not applicable. Therefore, it is necessary to develop summarizing techniques for collecting valuable information from data streams [5]. Data pre-processing is an important and time consuming phase in the knowledge discovery process and must be taken into consideration when mining data streams. Designing a light-weight pre-processing techniques that can guarantee quality of the mining results is crucial. The challenge here is to automate such a process and integrate it with the mining techniques.

By considering the size of memory and the huge amount of data stream that continuously arrive to the system, it is needed to have a compact data structure to store, update and retrieve the collected information. Without such a data structure, the efficiency of mining algorithm will largely decrease. Even if we store the information in disks, the additional I/O operations will increase the processing time.
While it is impossible to rescan the entire input data, incremental maintaining of data structure is indispensable. Furthermore, novel indexing, storage and querying techniques are required to manage continuous and changing flow of data streams.

It is crucial to consider the limited resources such as memory space and computation power for reaching accurate estimates in data streams mining. If stream data mining algorithms consume the available resources without any consideration, the accuracy of their results would decrease dramatically. In several papers this issue is discussed and their solutions for resource-aware mining are proposed. One of the proposed solutions is AOG which use a control parameter to control its output rate according to memory, time constraints and data stream rate [1]. Also in another algorithm is proposed that not only reduces the memory required for data storage but also retains good approximation given limited resources like memory space and computation power.

Visualization is a powerful way to facilitate data analysis. Absence of suitable tools for visualization of mining result makes many problems in data analysis and quick decision making by user. This challenge still is a research issue that one of the proposed approaches is intelligent monitoring [2]. We summarized these challenges and related research issues inTable1.



Research Issues	Challenges	Approaches
Memory management	Fluctuated and irregular data arrival rate and variant data arrival rate over time	Summarizing techniques
Data preprocessing	Quality of mining results and automation of preprocessing techniques	Light-weight preprocessing techniques
Compact data structure	Limited memory size and large volume of data streams	Incremental maintaining of data structure, novel indexing, storage and querying techniques
Resource aware	Limited resources like storage and computation capabilities	AOG and [31]
Visualization of results	Problems in data analysis and quick decision making by user	Still is a research issue (one of the proposed approaches is: intelligent monitoring)

TABLE 2. CLASSIFICATION OF DATA STREAM MINING CHALLENGES

IV. RESULTS AND DISCUSSION

We compare our method according to its quality and speed up using a specific data set. In this section data set is described and proposed algorithm is compared with K-Means algorithm and Hierarchical Clustering considering quality of clustering and speed up.

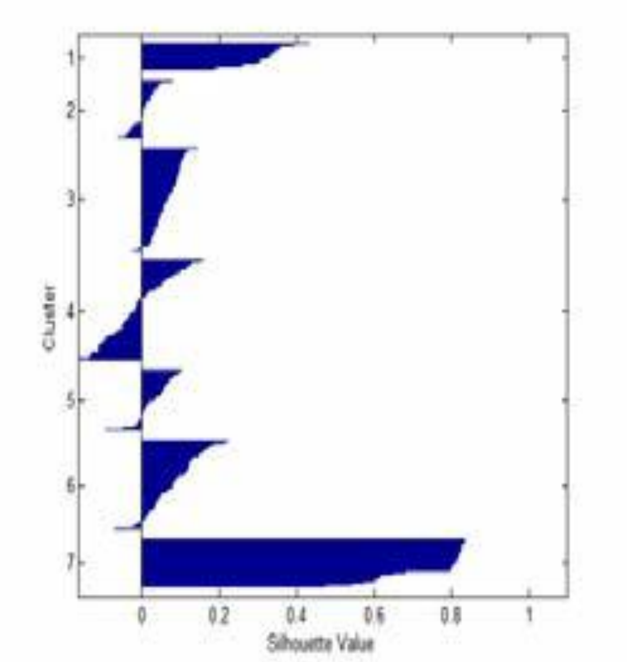


Figure.5. K-Means silhouette value with mean=0.1242

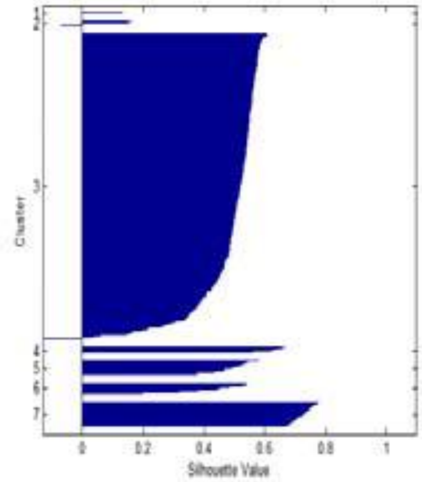


Figure.6. Hierarchical Clustering silhouette value with mean=0.502

The proposed analytical framework

This research ends in an analytical framework which is shown in Table 2. This framework tries to show the efficiency of data mining applications in developing the novel data stream mining algorithms. These algorithms are classified base on the data mining tasks. We described the details of these algorithms based on pre-processing steps and the following steps. In addition, this framework can direct future works in this field. Some of the most important results that have been reached during this research are:

(1) Mining data streams has raised a number of research challenges for the data mining community. Due to the resource and time constraints many summarization and approximation techniques have been adopted from the fields of statistics and computational theory.

(2) There are many open issues that need to be addressed. The development of systems that will fully address these issues is crucial for accelerating the science discovery in the fields of physics and astronomy, as well as in business and financial applications.

V. CONCLUSION AND FUTURE SCOPE

In this paper we reviewed and analyzed data mining applications for solving data stream mining challenges. At first we presented a comprehensive classification for data stream mining algorithms based on data mining applications. In this classification, we separate algorithms with pre-processing from those without pre-processing. In addition, we classify pre-processing techniques in a distinct classification. In the following, the layered architecture of the classification represents almost all of the challenges that are mentioned in various researches. Combining advantages of K-Means and divide and conquer strategy can help us in both efficiency and quality. Besides simulating of HC is possible with recursive intrinsic divide and conquer method and creating nested clusters. HC algorithm can construct and can be used for big datasets that yields low quality. In this paper we present a method to use both advantages of HC and K-Means by introducing equivalency and compatible relation concepts. By these two concepts we defined similarity and our space and could divide our space by a specific criterion. Many directions exist to improve and extend the proposed method. Different applications can be used and examined the framework. Text mining is an interesting arena. Based on this method data stream processing can be improved. Data type is another direction to examine this method.

In this study K-Means has been used for second phase whereas we can use other clustering algorithms e.g. genetic algorithm, HC algorithm, Ant clustering[1], Self Organizing Maps, etc. Determining number of sub spaces can be studied as important direction for the proposed method. Application in data stream mining so far, there are still wide areas for further researches.

REFERENCES

[1] Latifur Khan[1],Wei Fan, Data Stream Mining and Its Applications, June, 2012.

[2] Chen, S., Wang, H., Zhou, S., Yu, P (2008). Stop chasing trends: Discovering high order models in evolving data, In: Proc. ICDE, pp. 923–932 (2008).

[3] Gaber MM, Zaslavsky A, Krishnaswamy S. Mining data streams: a review. *ACM SIGMOD Rec* 2005,

[4] Mohamed Medhat Gaber, Advances in data stream mining, Volume 2, Januar y / Februar y 2012.

[5] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A framework for projected clustering of high dimensional data streams," in Proceedings of the Thirtieth international conference on Very large data bases-Volume 30, 2004, p.863.

[6] Gaber MM, Zaslavsky A, Krishnaswamy S. Mining data streams: a review. *ACM SIGMOD Rec* 2005,

[7] Nicolás García-Pedrajas · Aida de Haro-GarcíaScaling up data mining algorithms: review and taxonomy,Received: 4 June 2011 / Accepted: 26 September 2011 / Published online: 13 January 2012

[8] Madjid Khalilian, Norwati Mustapha, MD Nasir Suliman, MD Ali Mamat," A Novel K-Means Based Clustering Algorithm for High Dimensional Data Sets",Vol.1, IMECS 2010,march 2010.

[9] R.S. Walse , G.D. Kurundkar , P. U. Bhalchandra,"A Review: Design and Development of Novel Techniques for Clustering and Classification of Data" in IJSRCSE, Vol.06 , Special Issue.01 , pp.19-22, Jan-2018.

[10] A.Jenita Jebamalar "Efficiency of Data Mining Algorithms Used In Agnostic Data Analytics Insight Tools" Int. J. Sc. Res. in Network Security and Communication ,Volume-6, Issue-6, December 2018.

[11] Himanshi , Komal Kumar Bhatia, "Prediction Model for Under-Graduating Student's Salary Using Data Mining Techniques" Int. J. Sc. Res. in Network Security and Communication , Volume-6, Issue-2, April 2018.

Authors Profile

Mr. K. Rajasekhar, received B.Tech (CSE) from JNTUH, Hyderabad, India in 2007 and M.Tech (CSE) from Andhra Univesity College of Engnineering(A), Andhra University, Visakhapatnam in the year 2010. He has 8 years of teaching experience. Currently he is working as Assistant Professor, Department of Computer Science and Engineering, Annamacharya Institute of Technology and Sciences, Tirupati, India. His areas of interests are Data Mining, Artificial Intelligence, Advanced Computer Networks, Cloud Computing and Wireless Ad-Hoc Networks. He is a member of IAENG.

P. Venkata Maheswara, received B.Tech degree in Computer Science and Engineering from JNTU-Anantapur in 2011 and M.Tech degree in Computer Networks and Information Security from JNTU-Anantapur in 2013. He is also a member of IAENG. He has 5 years of teaching experience, present working as a Assistant Professor of Computer Science and Engineering Department in Annamacharya Institute of Technology & Sciences, Tirupati. His research includes Computer Networks, Data mining and Cloud computing.