IOSR Journal of Computer Science (IOSR-JCE)
e-ISSN: 2278-0661, p-ISSN: 2278-8727
PP 28-33
www.iosrjournals.org

A Data Mining Approach For Handling Evolving Data Streams

Ms. Tasneem Hasan
Computer Science & Engineering Department, ITM College of Engineering Nagpur, India

ABSTRACT: *Efficient analysis of data streams is becoming a key area of data mining research, as the number of applications demanding such processing increases. With recent advancement in technology, need for analysis of such unbounded streams is increasing day by day. Data mining process helps to excavate useful knowledge from rapidly generated raw data streams. In context with the continuously generated data, mining data streams is emerging challenging task in which several issues like limited space, limited time, accuracy, handling evolving data need to be considered. In this paper the main method of research is clustering which is focused to handle evolving data streams. Most of the previously proposed methods inherit the drawbacks of k means method and fail to handle the issues. A hybrid data mining approach encompassing windowing, grid and density clustering and divide and merge method is proposed. A dynamic data stream clustering algorithm (DDS) is used in which a dynamic density threshold is designed to accommodate the changing density of grids with time in data stream. At last divide and merge approach is used to handle varying data points and further refine the quality of result obtained. Experimental results on large data set demonstrate the utility of this approach.*

Keywords -*Concept drift, Clustering, Data mining, Data streams, Threshold****.***

I. INTRODUCTION

With the emerging applications involving massive data e.g. customer click, retail chain transaction, stock trading, emails etc leads to the generation of continuous huge bulk of information streams. These data streams contains continuous fast and varying data points on which various data mining method like clustering ,classification etc could be applied. Huge size of continuously flowing data has put forward a number of challenges in data steam analysis. The application of various data mining methods on these uncertain data streams helps to excavate hidden information which can be used in processing of data streams.

A data stream mining is an approach to extract meaningful information (knowledge) from raw data streams and find changes and evolution of stream overtime. A general form of data mining technique involves clustering which is important and difficult research topic in data mining, artificial intelligence, machine learning involving wide range of application such as website analysis, stock market analysis, astronomers, news organization etc. Various clustering algorithms are available for processing data streams but majority of them work on static data streams. Many issues can be envisioned with handling of data streams which are summarized as follows: [1]

- Large space: Data streams have enormous volumes of continuously incoming data.
- Dynamic data: Data streams are fast, changing, uncertain and require fast response to incorporate changes in data and reflect it in output.
- Noise: Any approach applied to data streams should be able to deal with noise and outliers.
- Single scan: Since data streams have infinite volume of information which is fast and changing, hence stream data should be read only once.
- Light weight: Techniques applied to vast data streams should process stream in less time and memory.

Data streams have an important characteristic that it is unpredictable, dynamic and uncertain in nature. A difficult problem with learning in many real-world scenarios is that the concepts are often not stable but change with time. Concept of interest may depend on some hidden context, not given explicitly. A typical example is weather prediction rules that may

vary radically with the season. Often the cause of changes in continuously flowing data streams is hidden, not known a priori, making the learning task more complicated. Knowledge discovery and data mining from time-changing data streams and efficient concept drift handling on data streams have become important topics in the machine learning community and have gained increased focus recently [2].

II. RELATED WORK

Data mining is a process of efficiently extracting meaningful information from underlying raw data. These are combination of techniques that make use of computer programs to automatically excavate model that represents useful patterns [2]. A major challenge imposed by continuously arriving data stream is to analyze them and respond to newly arrived data by improvising models to adapt to the changing data and reflect these changes to produce correct results and improve overall prediction accuracy [3]. Due to recent technological advances and availability of streaming data in diverse range of fields increases importance of streaming data analysis [4]. Various data mining techniques that could be applied on data streams can be divided into two major categories, supervised approach and unsupervised approach[2].

Supervised approach includes classification which is constrained to work only on labelled data set. Unsupervised approach encompasses clustering in which data set is unlabelled. As mentioned earlier data streams are uncertain so we cannot predict whether incoming data points would be labeled or unlabelled. Clustering approach is assumed to be more suitable for data streams because since it works on unlabelled data sets, it would also produce exceptionally good results if applied to labeled datasets.

Cluster analysis or clustering is a primary data mining tool which helps user understand natural grouping of attributes. By partitioning data streams into homogenous clusters in which objects have high intra class similarity and low inter class similarity, data miners can interpret about data properties which can then be utilized to develop efficient classification systems for new data or predictive model for hidden unforeseen events [5]. Due to versatility of clustering present issue intends to provide broad overview on various commonly used clustering approaches.

Guha, Mishra, Motwani, and L. O'Callaghan [6] adapted k means approach for clustering. K means have various advantages like it is simple, results is easy to interpret, but it is sensitive to noise and requires more space. Here data is divided into chunks, each of which is then separately clustered in weighted centers. This approach was again improved to produce STREAM [7] which uses moving window and produces centers at each particular stage. This approach does not need to specify in advance the number of clusters expected at the output and instead evaluates the performance by using a combination of sum of squared distances and the number of centers used.

CLUSTREAM [8] has two phases online and offline. Online phase generates micro clusters as output. Micro clusters absorb coming data points and cluster them based on time window. In offline phase we get macro clusters which are generated based on cluster number, duration, etc using k means approach.

Edwin Lughofer [9] used divide and conquer technique to devise dynamic split and merge method which would be able to handle dynamic nature of data. Merging is performed using weighted averaging of cluster centers based on homogeneity condition between two clusters. Splitting is done when clusters are shrinking inside one large cluster to form separate clusters. It copes up with dynamic data to provide high purity and cleaner cluster.

Density based approach for clustering was utilized in [10] to present DENSTREAM which is extension of DBSCAN and similar to CLUSTREAM. It defines structure of potential micro cluster and outlier micro cluster. When new data points arrives it checks if it can be fused with potential micro cluster, if not then try fusing it with outlier micro cluster else create new outlier micro cluster. When request for clustering occurs it uses DBSCAN to generate final output clusters.

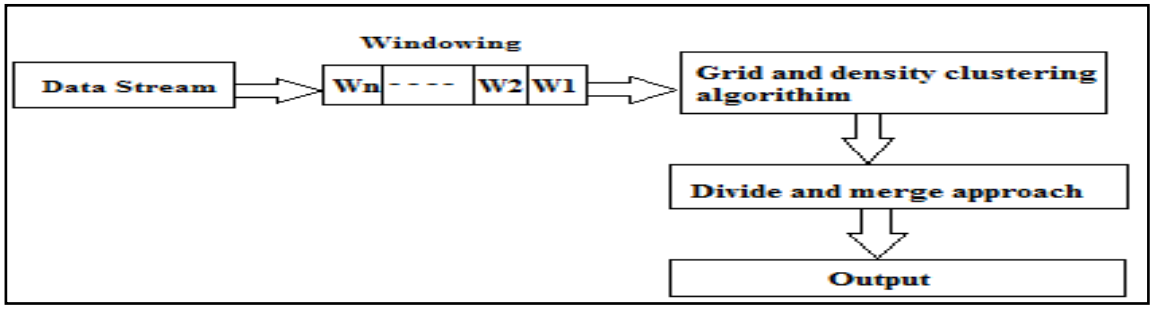
To overcome the drawback of DENSTREAM, rDENSTREAM algorithm was proposed [11]. It adds retrospect phase in above two steps on DENSTREAM. Here result of macro clustering of potential micro cluster is regarded as classifier, outlier micro cluster are not discarded but stored in buffer which is later used to relearn the classifier to provide improved clustering accuracy and limit the loss of knowledge point

III. PROPOSED WORK

3.1 System Architecture

The grid-based clustering method uses a multi resolution grid data structure. It forms the grid data structure by
dividing the data space into a number of cells and perform the clustering on the grids. Clustering depends on the number of grid cells and independent of the number of data objects .Grid-based method could be natural choice for data stream in which the infinite data streams map to finite grid cells. The synopsis information for data streams is contained in the grid cells. Grid can be defined as division of data space into finite number of cells that form a grid structure.

Grid cell can be dense, sparse or empty. If a grid cell does not contain even a single data point then such grids are called empty grids and are discarded. Threshold specifies the



minimum number of points to classify whether the grid is dense or sparse. If a grid cell contains density value greater than or equal to the threshold then that cell

Figure 1. Proposed System Architecture

is considered as dense grid. If the density value of grid cell is less than threshold then that grid would be considered as sparse grid.

The density of grid is calculated using its dynamic weight. It is suitable for updating the grid cell feature with negligible time and storage overhead. The figure below depicts the proposed system architecture.

As shown in fig. 1 there are following three modules. The first module is windowing which takes raw data streams as input and applies sliding window technique to form windowed stream. Huge bulk of information cannot be given directly to clustering algorithm as it would lead to space overhead and consume more time. Windowing simply partitions the huge data stream into blocks or windows. This windowed stream would be provided input to next module. The next module is grid and density based clustering where the windows obtained from previous stage are given one by one to the clustering algorithm.

A dynamic data stream algorithm (DDS) which aim to cluster uncertain data points is used here. Here first grid mapping is done where infinite data stream is mapped onto a finite grid cells. For every data point that has been located on grid cells, calculate its dynamic weight which depends on rate of processing, arrival time of data point and existence probability of data point. Using dynamic weight, dynamic density is calculated. For every

non empty grid cell dynamic density can be defined as summation of dynamic weight of all data points that belong to that grid cell.

Using the dynamic density and threshold classification of grid cells into dense and sparse grid is done. By comparing the value of dynamic density with threshold each cell can be stamped as either dense or sparse. If the value of dynamic density of a cell in grid is greater than or equal to threshold then classify that cell as dense If the value of dynamic density of a cell in grid is less than threshold then it is considered as sparse.

Finally clustering is done based on density of grid cells. Every empty grid cells are discarded. For every dense grid if its neighbor is also dense then include them in one cluster and finds its neighbor. If neighbor is sparse then consider it boundary point for cluster. For every sparse grid if its neighbor is also sparse then consider it as noise point.

To handle the dynamic and unpredictable nature of data streams re clustering is done on the output obtained from grid and density clustering method. The intermediate clusters obtained from previous step would be provided input to divide and merge technique to produce final output. A specific problem in evolving cluster models arise whenever two or more local data clouds (each one modeled by a cluster) are moving together or are delaminating inside one cluster. In such cases, where concept drift occurs clusters should be dynamically merged or split in order to account for correct cluster partitions.

3.2 Clustering Algorithm

This section explains the steps of grid and density based dynamic data stream (DDS) clustering algorithm used. Steps of our DDS algorithm are as follows:

Step1: Locate current data points on grid.
Step2: Calculate uncertain weight [UW(X,t)].
Step 3: Discard empty grids cells.
Step4: For every non empty grid cell
1) Compute Uncertain density [Density (G,t)].
2) If Density greater than or equal to Threshold Stamp grid cell as dense grid Else, Stamp cell as sparse grid.
Step 5: For every dense grid cell, find its neighbor grid NEIGH.
Initialize cluster ID=0,
Case I: NEIGH is a dense.
a) Assign all the dense NEIGH and core grid with a new cluster ID.
b) Cluster Id++.
Case II: NEIGH is a sparse, Compute boundary point process.
Step 6: For every sparse grid. If its neighbor is also sparse then, noise point process.

IV. Results and Discussion

4.1 Evaluation Measurement

A hybrid approach was used which includes DDS algorithm and split and merge to handle evolving data streams. Initially clusters were formed using DDS algorithm. This output obtained was again re clustered using split and merge to handle dynamic nature of data streams.

We are using Cover type dataset.7 clusters of cover type were obtained. The following are the classes obtained in cover type which are numbered accordingly: Spruce/Fir tree, Lodgepole Pine tree, Ponderosa Pine tree, Cottonwood/Willow tree, Aspen tree, Douglas-fir tree, Krummholz tree.The accuracy can be calculated using following formula

Accuracy = (Classified Instances / Total number of instances) *100% (1)

4.2 Performance Evaluation and comparison

The experimental result shows that the proposed approach clearly outperforms other existing methods.Table1 gives the class wise values of accuracy obtained by using equation above equation.

Table 1. Comparison In Terms of Accuracy

Class	DDS	Hybrid(DDS + Split and Merge)
1	90.99%	95.98%
2	90.99%	95.99%
3	90.97%	95.97%
4	90.97%	95.97%
5	90.98%	94.99%
6	90.97%	95.87%
7	90.97%	95.97%

As we can see the accuracy of DDS algorithm is less. This may happen due to dynamic nature of data streams, sum points might have been misclassified or some points might have been not considered while clustering.

After applying split and merge the points get accurately placed in the actual class in which they belong. This leads to increase in accuracy. In this way the proposed approach tackles the problem of evolving data streams to provide more accurate clusters. The table 1 clearly depicts the increase in accuracy after applying split and merge. The overall accuracy of DDS algorithm is 90.98 % and after applying split and merge the overall accuracy is 95.98 %.

V. CONCLUSION

Data mining, also called Knowledge-Discovery is one of the hot topics in the field of knowledge extraction. With the growing use of technology efficient handling of data streams is needed. On the basis of the previous clustering method it can be concluded that none of the clustering methods accurately handle varying data streams. Previous clustering method fails to handle all the issues of data streams simultaneously. Hence a hybrid data mining approach has been proposed that combines the advantages of individual methods to form a more robust system.

The main research methodology used here for handling evolving data streams is clustering because clustering can handle both labeled as well as unlabelled data streams efficiently. Windowing helps in conserving memory, DDS clustering algorithm helps in fast processing of data streams and divide and merge further increases the accuracy by handling varying data points by re clustering them. The proposed method achieves the accuracy 90.98% and 0.86 error rate for DDS algorithm and accuracy of 95.98% and 0.36 error rate after applying split and merge.

REFERENCES

[1] Yi-Hong Lu, Yan Huang, "Mining data streams using clustering" ,Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, IEEE, Guangzhou, August 2005,pp 2079-2083.
[2] Kapil Wankhade, Tasneem Hasan, Ravindra Thool, " A Survey: Approaches for Handling Evolving Data streams", In proceedings of International Conference on Communication Systems and Network Technologies,IEEE,INDIA,2013,pp 621-625.
[3] K.Prasanna Lakshmi, Dr.C.R.K.Reddy, "A Survey on different trends in Data Streams",Proceeding of International Conference on Networking and Information Technology, IEEE, 2010, pp 451-455.
[4] Amr Magdy, Noha A. Yousri, and Nagwa M. El-Makky, "Discovering Clusters with Arbitrary Shapes and Densities in Data Streams ", In proceedings of 10th International Conference on Machine Learning and Applications, IEEE, 2011, pp 279-282.
[5] S. Guha, N. Mishra, R. Motwani, and L. O'Callaghan, "Clustering data streams," Foundations of Computer Science, Annual IEEE Symposium, 2000, pp 359-366.
[6] L. O'callaghan, N. Mishra, A. Meyerson, S. Guha, and R. Motwani, " Streaming-data algorithms for high-quality clustering", Proceedings of the 18th International Conference on Data Engineering (ICDE.02), IEEE, 2002. pp 685–694.
[7] C. Aggarwal, J. Han, J. Wang, and P. Yu, "A framework for clustering evolving data streams", In Proceedings of the 29th international conference on Very large data bases, 2003, Volume 29, pp 81–92.
[8] Edwin Lughofer, "Dynamic Evolving Cluster Models using On-line Split-and-Merge Operations", In proceedings of 10th International Conference on Machine Learning and Applications, IEEE, 2011, pp 20-26.
[9] Cao F, Ester M, Qian W and Zhou A, "Density-Based Clustering over an Evolving Data Stream with Noise", Proceedings of the SIAM Conference on Data Ming, 2006, pp 326-337.

[10] Liu Li-xiong, Huang Hai, Guo Yun-fei, Chen Fu-cai, "rDenStream,A Clustering Algorithm overan Evolving Data Stream", IEEE, 2009.
[11] Chen Jia, Cheng Yu Tan, Ai Yong, "A Grid and Density-based Clustering Algorithm for Processing Data Stream", In proceedings of Second International Conference on Genetic and Evolutionary Computing, IEEE, 2008, pp 517-521.