Models and Issues in Data Stream Mining

Lalit S. Agrawal
Asst. Prof., Department of Computer Application
RCOEM, Nagpur, India
agrawalls@rknec.edu

Dattatraya S. Adane
Professor, Department of Information Technology
RCOEM, Nagpur, India
adaneds@rknec.edu

***Abstract*— With great innovation in technology there is a huge data explosion. Real-time surveillance, internet traffic, sensor data, health monitoring systems, communication networks, online transactions in the financial market and so on contribute as a data sources. Sometime data is huge enough that it cannot be stored in traditional databases. Moreover, this data can be structured, semi-structured or unstructured. In today's scenarios it is always desired to take real time decisions from the data which is coming in with high velocity. Here, suitable solution is Stream processing. Stream processing allows us to analyze and mine data on-the-fly without storing it completely. The start point for the stream processing is the assumption that the potential value of data depends on data freshness. Thus, the stream processing paradigm analyzes data in real time to extract potential value out of it. Data mining techniques for streaming data includes: clustering, classification, frequent pattern mining and outlier detection which can be used to extract important information from streaming data.**

In this paper we will summarize the efforts taken by researchers in the field of data stream mining along with the open research issues. We will also present the comparative study of few algorithms used for data stream processing. Finally we will conclude with the open issues in data stream processing.

Index Terms*— Data Stream, Stream Processing, Data Stream Mining

I. INTRODUCTION

Traditional data mining techniques are suitable for simple and structured data sets like relational databases and data warehouses. But with the advancement in technology many application generates enormous amount of data which cannot be stored and processed for potential value in timely manner. To extract knowledge from such huge and high speed data we need a system to mine data on-the-fly. The overview of data stream mining process is depicted in Fig.1. In recent years different approaches are proposed to overcome the challenges of storing and processing fast and continuous streams. Data stream processing technique is different than traditional data mining approach. Stream data comes with certain characteristics as follows:

- Data arrives continuously from the streams.
- With continuous data it is not possible to store it and processes it efficiently by using multiple passes.
- No assumptions on data ordering in stream can be assumed.
- Data is unbounded in nature.
- Difficult to get global view of data.

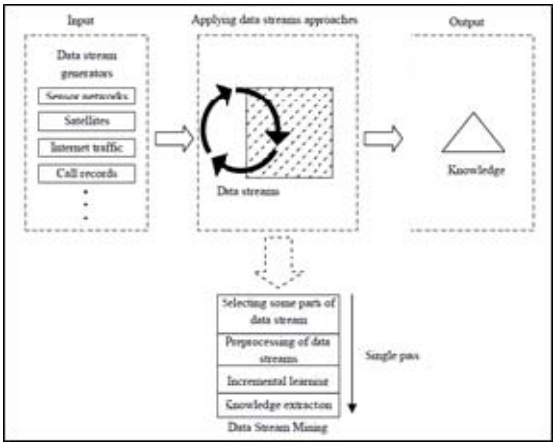

Fig.1. Overview of Data Stream Mining Process

Capturing data from this kind of stream is very important for many real life applications like web stream monitoring, credit card fraud analysis, network traffic monitoring, highway traffic congestion analysis, link statistics in networking and stock market based analysis etc. Systems and models have been proposed to cope up with the challenges of stream processing [1]. There are many differences [2] between stream processing and traditional processing as summarized in Table 1.

TABLE I: COMPARISON BETWEEN STREAM PROCESSING AND BATCH PROCESSING [2]

Parameters	Stream processing	Batch processing
Input	Stream of data	Data chunks
Data size	Infinite	Finite
Storage	Limited Storage	Store entire data
Hardware	Limited memory	Multiple CPUs
Processing	Single pass	Multiple passes
Time	Few milliseconds	Much longer

We will present the important models, comparative study of various algorithms proposed by the researchers so far along with our observation and important research issues in data stream processing.

The remaining parts of this paper are organized as follows: In section 2, we briefly discuss the data stream processing models. Then, section 3 described various

methodologies researchers used so far to solve the problems of stream mining. Section 4 briefly explains about various pre-processing methods. Section 5 and 6 concludes this paper with important research issues and conclusion respectively.

II. DATA STREAMS PROCESSING MODELS

Landmark, Damped and Sliding Windows are the three data stream processing models mentioned in [3]. The Landmark model mines all frequent itemsets over the entire history of stream data from a specific time point called landmark to the present. However, this model is not suitable for applications where people are interested only in the most recent information of the data streams, such as in the stock monitoring systems, where current and real time information and results will be more meaningful to the end users.

The Damped model, also known as Time-Fading model, mines frequent itemsets in stream data in which each transaction has a weight and this weight decreases with age. While calculating the result, older transactions contribute lesser weight than newer transactions. This model considers different weights for new and old transactions. This is suitable for applications in which new data has more effect on mining results and effect of old data get reduced with time.

The Sliding Windows model finds and maintains frequent itemsets in sliding windows. Only part of the data streams within the sliding window are stored and processed at the time when the data flows in. The size of the sliding window may be decided according to applications and system resources. The mining result of the sliding window method totally depends on recently generated transactions in the range of the window. All the transactions in the window need to be maintained in order to remove their effects on the current mining results when they are out of range of the sliding window. The real challenge while using sliding window based model is to avoid load shedding effect. When the data arrival rate in data stream is high than the processing rate few packets or data points are dropped and they will never be processed.

Various methodologies proposed by researchers to solve the problem of mining data stream are: Data stream clustering, classification, frequent pattern mining and outlier detection as depicted in Fig. 2.

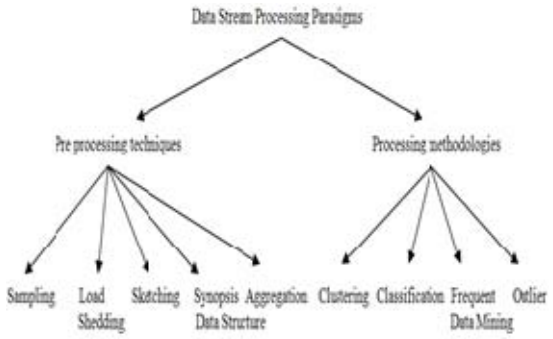

Fig. 2. Data Stream Processing Paradigms

III. METHODOLOGIES AND RELATED WORK

From the statistical and computational approaches the problems of mining data streams can be solved using methodologies mentioned in previous section. Few of the methodologies and related work is presented here:

A. Data stream Clustering

In linear, logistic regression scenarios, we have one variable that we need to compute as a function of several known variables. This type of problem is known as supervised learning problem. But, many a times, it is required to explore the patterns within a given data with no target attribute. Such problems are called unsupervised learning problems. Clustering is popularly known as unsupervised learning problem that aims to find similar group of data.

For a given set of objects if we partition them into one or more groups of similar objects then the smaller groups are known as clusters and this division methodology when applied to data stream is known as data stream clustering. As a data stream mining function, cluster analysis serves as a tool to gain insight into the distribution of data to observe characteristics of each cluster.

B. Data stream classification

In real world, the way business operates changes with time and hence the way in which data stream flows also changes considerably. Typical examples of this are stock market prediction rules and customers' preferences. The underlying data distribution may change as well. The above mentioned scenario leads to a classical problem known as concept drifting. It is defined as with the change in data distribution the model built on old data become inconsistent with the new data and regular updating of the model is necessary. The framework to address the problem of finding patterns over concept drifting streams is presented in [10].

C. Frequent pattern mining

Frequent pattern mining focuses on discovering frequently occurring patterns from different types of datasets which includes unstructured ones such as transaction and text

datasets, semi-structured ones such as XML datasets, and structured ones such as graph datasets. Many efficient frequent pattern mining algorithms have been developed in past but these algorithms typically require datasets to be stored in persistent storage and involve two or more passes over the dataset. In a streaming environment, a mining algorithm must take only a single pass over the data. Such algorithms can only guarantee an approximate result.

The unique challenges in discovering frequent patterns are: First, frequent pattern mining needs to search the available space with an exponential number of patterns. Second, frequent pattern mining relies on the down-closure property to prune infrequent patterns and generate the frequent ones.

Though the existing one-pass mining algorithms have been shown to be very accurate and faster than traditional multi-pass algorithms, the researches show that they are still computationally expensive meaning that if the data arrives too rapidly, the mining algorithms will not able to handle complete data.

D. Outliers

Outlier can be defined as any observation which deviates too much from the other observations so as to provide suspicions that it was generated by a different mechanism. Outliers may appear in a dataset for numerous reasons like malicious activity, instrumental error, setup error, changes of environment, human error, catastrophe, etc.

Regardless of the reason, outliers may be interesting and/or important to the user because of their diverse nature compared to normal data points. Some people define outliers as problems, some people define them as interesting items, but in any case, they are unavoidable [20]. Finding outliers from the data stream has been the most popular research area among the researchers of data mining community. Outlier can be of type point, contextual or correlated. Various methodologies have been proposed by researchers to detect outliers [20, 24].

IV. PREPROCESSING TECHNIQUES

Few of the characteristics of data stream are transient, uncertain, heterogeneity etc. Data points in the stream are moving continuously and also the sequence of data points can't be assumed. These two properties pose great challenge in processing the data stream in run time. Hence some sort if pre processing is required to facilitate data mining in run time. Various techniques like sampling, load shedding, sketching and aggregation helps in pre-processing of the data stream.

Sampling refers to the process of selecting data item for processing. Sampling methods [18] are among the simplest methods for synopsis construction in data streams. It is also relatively easy to use these synopses with a wide variety of application since their representation is not specialized and uses the same multi-dimensional representation as the original data points. The main problem with sampling in the context of data stream analysis is the unknown size of stream.

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.

Load shedding refers to the process of dropping a sequence of data points from the data streams. This concept of load shedding is useful where data stream is flowing with high velocity and it is very difficult to consider every data point for the analysis purpose. Load shedding has been used successfully in querying data streams. 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 analysis [4].

Sketching is the process of randomly project a subset of the data stream. It is the process of vertically sample the incoming data stream. Sketching has been applied in comparing different data streams and in aggregate queries. The major drawback of sketching is that of accuracy in the context of data stream mining [15].

Creating synopsis of data refers to the process of applying summarization techniques that are capable of summarizing the incoming data 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 characteristic of the dataset, approximate answers are produced when using such data structures [19].

The process in which the input stream is represented in a summarized form is called aggregation. This aggregate data can be used in data stream mining algorithms. The main problem of this method is that highly fluctuating data distributions reduce the method's efficiency [1].

TABLE II: COMPARISONS OF VARIOUS DATA STREAM PROCESSING ALGORITHMS

Algorithm	Mining Approach	Advantages	Disadvantages
Clustering Algorithms			
STREAM [4]	K-Medians	Incremental learning	Low clustering quality in high speed data streams
CluStream [5]	Micro clustering approach	High Accuracy in detecting concept drift	Clustering activity is performed in offline mode

HP Stream [6]	Projection based	High scalability and incremental update	High complexity
D – Stream [7]	Density based clustering	High quality and efficiency	High complexity
E – Stream [8]	Hierarchical approach	High scalability	High fading rate
SPE-Cluster [9]	Partitioning approach	High accuracy	Output is highly dependent on size of sliding window specified
Classification Algorithms			
Ensemble-based Classification [12]	Uses different combination of classifiers	Single pass, considers concept drift	Costly learning
CDM [14]	Decision tress and Bayes network	Dynamic update	High complexity
On-demand Stream Classification [11]	Uses the concept of micro cluster	Dynamic update	High cost and time needed for labelling
VFDT [13]	Decision tress	High speed, less memory	Does not consider concept drift
Frequent Pattern Mining Algorithms			
Lossy counting [15]	False positives based	Single pass	False negatives are not shown
HTM [17]	Sliding window based	Single pass	High complexity
FPDM [16]	False negative based	High-probability to find itemsets which are truly frequent	Does not allow false positive

V. OBSERVATIONS AND OPEN RESEARCH ISSUES

Data stream mining has been the active research area for a decade. Many approaches have been proposed to overcome the challenges posed by data streams. But, heterogeneous schema, dynamic relationships among the data points, asynchronous data points and cross correlation among the streams adds more complexity to stream processing.

Data points in stream are important for certain period of time. If they are not processed within given time frame than particular data point may be of no use to the analyst. Classification algorithms mentioned in Table 2 are highly dependent on trained datasets for decision making. Many a time trained datasets or labelled datasets are not available. In this scenario, classification based techniques are more likely to fail. Data points in stream are highly volatile in nature and hence in the absence of labelled dataset it does not support concept drift.

Methods based on clustering algorithms don't depend on trained datasets hence they are popularly termed as unsupervised learning methods. Clustering algorithms are briefly mentioned in Table 2. To perform clustering activity in run time without the availability of trained datasets makes it more complex and the resultant time complexity of such method is very high.

Frequent pattern mining highly suffers from false positives and false negatives. Data points are highly transient in nature and may change over a period of time hence the underlying pattern mining criteria has to evolve from time to time basis to overcome these problems.

There is a minute gap between normal data point and outlier. Many times, the outlier behaves like a normal data point. Clustering, classification and frequent pattern mining can be treated as methods to find the outliers from the data streams.

Many techniques have been proposed in the past to overcome the challenges imposed by data streams but still there is a lot of scope for improvement. Potential stream mining issues are summarized as below:

Incremental in Nature - Data streams are infinite. It is not possible to wait for the results until the end of data processing. So, the technique should be such that it can build and update the model with new incoming data without a need to build the model from scratch every time. For this summaries or sketches of streaming data can be used.

Single Scan of Data – Due to tremendous volume of data stream it is not possible to store streaming data as oppose to traditional data sets that can be stored in memory [1]. This leads to the one time access of streaming data points. So, processing technique should be such that it can update the processing model or can store the summaries of data for further analysis in single scan of streaming data.

Low Time and Space Complexity – Data streams flow at very high speed so to match this speed of data and to have good results, processing time per data point must be small [1, 22].

Handling Concept Evolution - Concept evolution occurs due to the change in the characteristics of data. So a processing model must be adaptive to concept evolution to reflect the real nature of streaming data in results. It could be done by providing more importance to new data and less importance to old data [1, 22, 23].

Robustness to Outliers – A data stream processing model must be able to identify the outlying data points because outliers can distort the complete structure of data [21].

VI. CONCLUSION

With the advancement in technology, there is a tremendous increase in the user's volume resulting into huge data generation in the form of data streams. Data streams have received a great deal of attention in the last decade and several attempts have been made to solve the problems associated with data streams. Researchers have taken effort to solve the problem of data stream mining however to the

best of our knowledge most of the solutions proposed were catering to specific application domain. There exists no work that presents a comprehensive framework to address all the issues related to data stream mining.

Research in data streams is still in its early stage. In this paper, we have tried to identify and consolidate various data stream mining terminologies, methods and algorithms. We have also mentioned various research issues in data stream mining to provide the young buddy researchers with a picture of the contemporary state of data stream mining research.

REFERENCES

[1] Babcock, Brian, et al. "Models and issues in data stream systems."Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. ACM, 2002.

[2] Hu, Han, et al. "Toward scalable systems for big data analytics: A technology tutorial." Access, IEEE 2 (2014): 652-687.

[3] Jea, Kuen-Fang, and Chao-Wei Li. "A sliding-window based adaptive approximating method to discover recent frequent itemsets from data streams."Proceedings of the International MultiConference of Engineers and Computer Scientists. Vol. 1. 2010.

[4] O'callaghan, Liadan, et al. "Streaming-data algorithms for high-quality clustering." icde. IEEE, 2002.

[5] Aggarwal, Charu C., et al. "A framework for clustering evolving data streams."Proceedings of the 29th international conference on Very large data bases-Volume 29. VLDB Endowment, 2003.

[6] Chai, Y., H. Wang, and P. Yu Loadstar. "Load shedding in data stream mining."Proc. Int. Conf. on Very Large Data Bases (VLDB). 2005.

[7] Chen, Yixin, and Li Tu. "Density-based clustering for real-time stream data."Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2007.

[8] Udommanetanakit, Komkrit, Thanawin Rakthanmanon, and Kitsana Waiyamai. "E-stream: Evolution-based technique for stream clustering." Advanced Data Mining and Applications. Springer Berlin Heidelberg, 2007. 605-615.

[9] Chen, Ling, Ling-Jun Zou, and Li Tu. "A clustering algorithm for multiple data streams based on spectral component similarity." Information Sciences 183.1 (2012): 35-47.

[10] Wang, Haixun, et al. "Mining concept-drifting data streams using ensemble classifiers." Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2003.

[11] Aggarwal, Charu C., et al. "On demand classification of data streams."Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2004.

[12] Wang, Haixun, et al. "Mining concept-drifting data streams using ensemble classifiers." Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2003.

[13] Chi, Yun, Haixun Wang, and Philip S. Yu. "Loadstar: load shedding in data stream mining." Proceedings of the 31st international conference on Very large data bases. VLDB Endowment, 2005.

[14] Kwon, YongChul, et al. "Clustering events on streams using complex context information." Data Mining Workshops, 2008. ICDMW'08. IEEE International Conference on. IEEE, 2008.

[15] Manku, Gurmeet Singh, and Rajeev Motwani. "Approximate frequency counts over data streams." Proceedings of the 28th international conference on Very Large Data Bases. VLDB Endowment, 2002.

[16] Yu, Jeffery Xu, et al. "False positive or false negative: mining frequent itemsets from high speed transactional data streams." Proceedings of the Thirtieth international conference on Very large data bases-Volume 30. VLDB Endowment, 2004.

[17] Chi, Yun, Yirong Yang, and Richard R. Muntz. "HybridTreeMiner: An efficient algorithm for mining frequent rooted trees and free trees using canonical forms." Scientific and Statistical Database Management, 2004. Proceedings. 16th International Conference on. IEEE, 2004.

[18] Vitter, Jeffrey S. "Random sampling with a reservoir." ACM Transactions on Mathematical Software (TOMS) 11.1 (1985): 37-57.

[19] Chakrabarti, Kaushik, et al. "Approximate query processing using wavelets."The VLDB Journal—The International Journal on Very Large Data Bases 10.2-3 (2001): 199-223.

[20] Chandola, Varun, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey." ACM computing surveys (CSUR) 41.3 (2009): 15.

[21] Sadik, Shiblee, and Le Gruenwald. "Online outlier detection for data streams."Proceedings of the 15th Symposium on International Database Engineering & Applications. ACM, 2011.

[22] Tatbul, Nesime. "Streaming data integration: Challenges and opportunities."Data Engineering Workshops (ICDEW), 2010 IEEE 26th International Conference on. IEEE, 2010.

[23] Jiang, Nan, and Le Gruenwald. "Research issues in data stream association rule mining." ACM Sigmod Record 35.1 (2006): 14-19.

[24] Abe, Naoki, Bianca Zadrozny, and John Langford. "Outlier detection by active learning." Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2006.