ISSN: 2321-7782 (Online)
Volume 2, Issue 2, February 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Paper / Case Study
Available online at: www.ijarcsms.com

Mining Data Streams: A Survey

Twinkle B Ankleshwaria[1]
M.E. Student
Department of Computer Science & Engineering
Modasa, Gujarat – India

J. S. Dhobi[2]
Head of Department
Department of Computer Science & Engineering
Modasa, Gujarat – India

Abstract: ***In recent years, advances in hardware technology have facilitated new ways of collecting data continuously. Tremendous and potentially infinite volumes of data streams are often generated by 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 .Data Stream mining is the process of extracting knowledge structures from such continuous ,rapid data records. Mining data streams raises new problems for the data mining community about how to mine continuous high-speed data items that you can only have one look at. Due to this reason, traditional data mining approach needs to be changed and to discover knowledge or patterns from data streams , it is necessary to develop single-scan, on-line , multilevel , multi-dimensional stream processing and analysis methods. In this paper, we try to discuss the issues of data streams. We present the overview of basic methodologies for stream data processing. We discuss the various data streams mining algorithms. Finally, we conclude with a brief discussion of the big open problems and some promising research directions in the future in this area.***

Keywords: Data streams; Data Streams mining; clustering mining; classification mining; association mining.

I. INTRODUCTION

In the field of information processing Data mining refers to mining knowledge from large amounts of data [1].Similarly, Mining Data Streams is concerned with extracting knowledge from non stopping and continuous stream of information The topic of data streams is very recent one [2]. A data stream is a massive, infinite, temporally ordered, continuous and rapid sequence of data elements [3]. Research on data stream is motivated by emerging applications involving massive data sets such as customer click streams, supermarket, telephone records, stock market, meteorological research, multimedia data, scientific and engineering experiments and sensor data. A new generation of mining algorithms are needed for real –time analysis & query response for these applications since most conventional data mining algorithms can only be applied to static data sets that may be updated periodically in large chunks, but not in continuous streams of data. While data mining has become a fairly well established field now, the data stream problem poses a no. of unique challenges which are not easily solved by traditional data mining methods. Some of issues of data stream [4] like dynamic nature, Infinite size and high speed, unbounded memory requirements, Lack of global view, handling the continuous flow of data impose a great challenge for the researchers dealing with streaming data. Unlike traditional data sets , it is impossible to store an entire data stream or to scan through it multiple times due to its tremendous volume .New concepts may keep on evolving in data streams at different times, to deal with this any data stream processing algorithm have to continuously update their models to adapt to the changes. The main purpose of this paper is to study the various methods and algorithms for mining data streams.

The paper is organized as follows. Section I provides Introduction. Section II introduces the basic Methodologies for Stream Data Processing. Section III describes the various Stream Mining Algorithms. Section IV describes various research

issues that have been raised after the study of Data Stream and finally Section V gives the conclusion of our work which concludes the paper.

II. Basic Methodologies for Stream Data Processing

For effective processing of stream data, new data structures, techniques, and algorithms are needed. Because we do not have an infinite amount of space to store stream data, we often trade off between accuracy and storage. That is, we generally are willing to settle for approximate rather than exact answers. Synopses allow for this by providing summaries of the data, which can return approximate answers. It uses synopsis data structures, which are smaller than the base data sets .From algorithmic point of view; we require algorithms that are efficient in both space and time. Instead of storing all elements using O(N) space synopsis data structures uses much smaller poly logarithmic space of order $O(\log^k N)$, where N is the number of elements in the data stream. In this section, we examine some common synopsis data structures and techniques.

A. Random Sampling.

Sampling methods are among the simplest methods for synopsis construction in data streams. Rather than deal with entire data stream, we can think of sampling the stream at periodic intervals. The main advantage of this technique is that it is relatively easy to use this synopsis with a wide variety of application since their representation is not specialized and uses the same multi-dimensional representation as the original data points. To obtain an unbiased sampling of data we need to know the full length of data set in advance, but it is not possible in case of data stream to know the length in advance. For this reason in this case we require to modify our approach. A technique called reservoir sampling can be used to select an unbiased random sample of s elements without replacement [5]. The main idea behind reservoir sampling is relatively simple. We maintain a sample of size at least s, called the "reservoir," from which a random sample of size s can be generated. But, generating such sample from the reservoir can be costly, especially when the reservoir is large. To avoid this, we maintain a set of s candidates in the reservoir, which form a true random sample of the element seen so far in the stream. As the data stream flows, every new element has a certain probability of replacing an old element in the reservoir. For example if we have seen N elements so far in the stream. The probability that a new element replaces an old one, chosen at random, is then s/N. The effectiveness of the reservoir based sampling method can be improved further with the use of concise sampling. More details in this approach can be found in [6].

B. Sliding Windows.

Instead of sampling the data stream randomly, we can use the sliding window model to analyze stream data. The sliding-window model of computation is motivated by the assumption that rather than running computations on all of the data seen so far, or on some sample, we can make decisions based only on recent data [7]. At every time t, a new data element arrives. This element "expires" at time t +w, where w is the window "size" or length. This model is very useful as an advanced technique for producing approximate answers to a data stream query for example stocks or sensor networks, where only recent events may be important uses this model. It also reduces memory requirements because only a small window of data is stored. Many recent on-line analysis software tools like MAIDS use this model for generating summaries of data in stream. We have two types of windows called count- based and time-based which can be used. In former the latest n items are stored where as in latter we can store only those items which have been generated or have arrived in the last T units of time.

C. Histograms.

The histogram is a synopsis data structure that can be used to approximate the frequency distribution of element values in a data stream. In the method of histograms, we essentially divide the data along any attribute into a set of ranges, and maintain the count for each bucket. Thus, the space requirement is defined by the number of buckets in the histogram. A histogram partitions the data into a set of contiguous buckets. Depending on the partitioning rule used, the width (bucket value range) and depth (number of elements per bucket) can vary. It is relatively easy to use the histogram for answering different kinds of

queries such as range queries, since we only need to determine the set of buckets which lie within the user specified ranges. A number of strategies can be devised for improved query resolution from the histogram [8].The equal-width partitioning rule is a simple way to construct histograms, where the range of each bucket is the same. The main drawback of this is that it may not sample the probability distribution function well. A better approach is to use V-Optimal histograms. This method is proposed in [9].Similar to clustering, V-Optimal histograms define bucket sizes that minimize the frequency variance within each bucket, which better captures the distribution of the data. These histograms can then be used to approximate query answers rather than using sampling techniques. The extension of histogram to data stream case is still a challenging task.

D. Multiresolution Methods.

A common way to deal with a large amount of data is through the use of data reduction methods. One of the popular data reduction method is the use of divide and conquer strategies such as multiresolution data structures .The main advantage of such methods is that it not only allows to balance the trade off between accuracy and storage, but also allow to understand the stream data at multiple levels. For example a balanced binary tree, in which each level of tree provides a different resolution. The more far we go from the root of the tree, the more detailed is the level of resolution. Two more sophisticated methods are described as below:

i. Micro-clusters:

A recent micro-clustering method[10] can be used be perform synopsis construction of data streams. For example, we can use a typical hierarchical clustering data structures like Cluster Feature (CF) - tree in BRICH [11] to form a hierarchy of micro-clusters. The advantage of micro-cluster summarization is that it is applicable to the multi-dimensional case, and adjusts well to the evolution of the underlying data stream. With dynamic stream data flowing in and out, summary statistics of data streams can be incrementally updated over time in the hierarchy of microclusters. Information in such microclusters can be aggregated into larger macroclusters depending on the application requirements to derive general data statistics at multiresolution.

ii. Wavelets :

Wavelets [12] are a well known technique which is often used in databases for hierarchical data decomposition and summarization. It is a technique from signal processing that can be used to build a multiresolution hierarchy structure over an input signal, in this case, the stream data. The basic idea in the wavelet technique is to create a decomposition of the data characteristics into a set of wavelet functions and basic functions. The simplest basis is the Haar wavelet. Using this basis corresponds to recursively performing averaging and differencing at multiple levels of resolution. This technique is particularly simple to implement, and is widely used in the literature for hierarchical decomposition and summarization. They are especially good at dealing with spatial and multimedia data. Wavelets have been used as approximations to histograms for query optimization. The property of the wavelet method is that the higher order coefficients of the decomposition illustrate the broad trends in the data, where as the more localized trends are captured by the lower order coefficients. Also, wavelet-based histograms can be dynamically maintained over time. Though, wavelets are a popular multiresolution method for data stream compression but, the one-pass requirement of data streams makes the problem of wavelet decomposition somewhat more challenging. In [13] a method is proposed which provides a fast algorithm whose space requirement is linear in the size of the synopsis and logarithmic in the size of the data stream.

E. Sketches

The idea of sketches is essentially an extension of the random projection technique to the time series domain. It can operate in a single pass. The estimation of the frequency moments can be done by synopses that are known as sketches. These build a small space summary for a distribution vector (e.g., histogram) using randomized linear projections of the underlying data vectors. Sketches provide probabilistic guarantees on the quality of the approximate answer. From a database perspective, sketch partitioning [14] was developed to improve the performance of sketching on data stream query optimization in this

technique; we intelligently partition the join attribute domain-space and use it in order to compute separate sketches of each partition. The resulting join estimate is computed as the sum over all partitions. This method has also been discussed in more detail for the problem of multi-query processing [15]. One of the key advantages of sketch based methods is that they require space which is sub linear in the data size being considered. Another advantage of sketch based methods that it is possible to maintain sketches in the presence of deletions. This is often not possible with many synopsis methods such as random samples. One more interesting trick for improving join size estimation is that of sketch skimming, it is described in [16].

F. Randomized Algorithms.

Randomized algorithms, in the form of random sampling and sketching, are often used to deal with massive, high-dimensional data streams. The use of randomization often leads to simpler and more efficient algorithms in comparison to known deterministic algorithms. If a randomized algorithm always returns the right answer but the running times vary, it is known as a Las Vegas algorithm. In contrast, a Monte Carlo algorithm has bounds on the running time but may not return the correct result. We mainly consider Monte Carlo algorithms. One way to think of a randomized algorithm is simply as a probability distribution over a set of deterministic algorithms.

III. DATA STREAM MINING ALGORITHMS

The data stream paradigm has recently emerged in response to the continuous data problem in data mining. Due to the continuous, unbounded, and high speed characteristics of dynamic data, there is a huge amount of data in both offline and online data streams. New algorithms have to be invented to address the computational challenges of data streams. A number of algorithms for extraction of knowledge from data streams were proposed in the various domains of data mining .In this section, we present an overview of different stream mining algorithms in the domains of Clustering, Classification and Association.

A. Clustering.

Imagine a huge amount of dynamic stream data. Many applications require the automated clustering of such data into groups based on their similarities. Although there are many powerful clustering algorithms for static data sets, clustering data streams places additional constraints on such algorithms, as any data stream model requires algorithm to make a single pass over the data, with bounded memory and limited processing time. Several algorithms have been developed for clustering data streams described as below:

STREAM –A k-median based Stream Clustering Algorithm is presented by Guha, Mishra, Motwani and O'Callaghan [17]. It consists of two phases and follows divide and conquer approach. In first phase, it divides the data streams in buckets and then finds k clusters in each bucket by applying k-median clustering. It stores cluster centers only and cluster centers are weighted based on the number of data points belongs to corresponding cluster and then discard the data points. In second phase weighted cluster centers are clustered in small number of clusters. Though its space and time complexity is low but it cannot adapt to concept evolution in data.

CluStream [18] clustering evolving data streams is proposed by Aggarwal et al. It divides the clustering process in following two online component and offline components. Online component stores the summary of data in the form of micro-clusters. Micro-cluster is the temporal extension of clustering feature of BIRCH [11]. Summary statistics of data are stored in snapshots form which provides the user flexibility to specify the time interval for clustering of micro-clusters. Offline component apply the k- means clustering to cluster micro-clusters into bigger clusters.

ClusTree [19] Any time Stream Clustering is proposed by Kranen et a..It divides the clustering process in following two online and offline components. Online component is used to learn micro cluster. The micro clusters are arranged in hierarchical tree structure. Any variety of offline components can be utilized. It is a self adaptive algorithm and delivers a model at any time.

HPStream [20] is proposed by Aggarwal et al. for clustering of high dimensional data streams. It uses a Fading Cluster Structure (FCS) to stores the summary of streaming data and it gives more importance to recent data by fading the old data with time. For handling high dimensional data it selects the subset of dimensions by projecting on original high dimensional data stream. Number of dimensions and dimensions are not same for each cluster. This is due to the fact that relevance of each dimension in each cluster may differ. It is incrementally updatable and highly scalable on number of dimensions. But it cannot discover the cluster of arbitrary shapes and require domain knowledge for specifying the number of clusters and average number of projected dimensions parameters

E-Stream [21] is a data stream clustering technique which supports following five type of evolution in streaming data: Appearance of new cluster, Disappearance of an old cluster, Split of a large cluster, merging of two similar clusters and change in the behavior of cluster itself. It uses a fading cluster structure with histogram to approximate the streaming data. Though its performance is better than HPStream algorithm but it requires many parameters to be specified by user.

HUE-Stream [22] extends E-Stream which is described earlier, in order to support uncertainty in heterogeneous data. A distance function with probability distribution of two objects is introduced to support uncertainty in categorical attributes. To detect change in clustering structure, the proposed distance function is used to merge clusters and find the closest cluster of a given incoming data and proposed histogram management to split cluster in categorical data.

POD-Clus (Probability and Distribution-based Clustering) [23] is a model based clustering technique for streaming data. It is applicable to both clustering by example and clustering by variable scenarios. For summarizing the cluster information and update it incrementally, it uses a cluster synopsis which comprises the mean, standard deviation, and number of points for each cluster. It supports concept evolution by allowing new cluster appearance, splitting of a cluster, merging of two clusters and disappearance of a cluster.

B. Classification

There are several methods for the classification of static data. Classification is a two step process consisting of model construction from training data and classification where the model is used to predict the class labels of tuples from new data sets in this case stream data. In a traditional setting, the training data reside in a relatively static database so it can be scanned multiple times, but in stream data, the data flow is so quickly that storage and multiple scans are infeasible. Another distinguishing characteristic of data streams is that they are time-varying, as opposed to traditional database systems, where only the current state is stored. This change in the nature of the data takes the form of changes in the target classification model over time and is referred to as concept drift. Concept drift is an important consideration when dealing with stream data. Several methods have been proposed for classification of stream data as shown below.

Hoeffding Tree Algorithm [24] proposed by from Domingos and Hulten's gives the streaming decision tree induction which is called Hoeffding Tree. The name is derived from the Hoeffding bound that is used in the tree induction. The main idea is, Hoeffding bound gives certain level of confidence on the best attribute to split the tree, hence we can build the model based on certain number of instances that we have seen. The main advantage of this algorithm is high accuracy with a small sample, it never performs multiple scans of the same data, and it is incremental. Apart from the advantages the main drawback of the algorithm is it cannot handle concept drift, because once a node is created, it can never change.

Very Fast Decision Tree [25] proposed by Domingos et al. makes several modifications to the Hoeffding tree algorithm to improve both speed and accuracy. It splits the tree using the current best attribute. Such a technique has the property that its output is (asymptotically) nearly identical to that of a conventional learner. Although VFDT algorithm works wee with data streams, it still cannot handle concept drift in data streams. To adapt concept drift in data streams, VFDT algorithm was further developed into the Concept-adapting Very Fast Decision Tree algorithm (CVFDT) it runs VFDT over fixed sliding windows in order to have the most updated classifier.

On Demand Classification [26] proposed by Aggarwal et al. have adopted the concept of micro clusters introduced in Clustream [18]. The classification process is divided into two components, first which performs summarization of data in micro clusters and second performs classification on it.

ANNCAD Algorithm [27] proposed by Law et al. is an incremental classification algorithm termed as Adaptive Nearest Neighbour Classification for Data Streams. The algorithm uses Haar Wavelets Transformation for multi-resolution data representation. A grid-based representation at each level is used. To address the concept drift problem of data streams, an exponential fade factor is used to decrease the weight of old data in the classification process. This algorithm have achieved accuracy over VFDT and CVFDT but the drawback of this algorithm is it cannot handle the sudden change in concept drift.

Ensemble based Classification [28] proposed by Wang et al. is a generic framework for mining concept drifting data stream. It uses weighted ensemble classifiers to handle concept drifting. The idea is to train an ensemble or group of classifiers from sequential chunks of the data stream. Each classifier is weighted and only the top K-classifiers are kept. The final output is based on the decision made by the weighted votes of the classifiers.

C. Association

Algorithms for association rule mining usually consist of two steps. The first step is to discover frequent itemsets. In this step, all frequent itemsets that meet the support threshold are discovered and the second step is to derive association rules. In this step, based on the frequent itemsets discovered in the first step, the association rules that meet the confidence criterion are derived. However, traditional association rule mining algorithms are developed to work on static data and, thus, cannot be applied directly to mine association rules in stream data. Recently many researches are conducted on how to get frequent items, patterns and association rules in a data stream environment. Some of the algorithms are described below.

In [29] Chang has proposed a method for finding recent Frequent itemsets adaptively over online data streams. It uses damped model which is also called as Time –Fading model in their algorithm, which mines frequent itemsets in stream data. This model considers different weights for new and old transactions. This is suitable for applications in which old data has an effect on the mining results, but the effect decreases as time goes on.

In [30] Lin has proposed a method for mining frequent Itemsets from data streams. For this it uses Sliding Window model in their algorithm. 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 vary according to applications and system resources.

In [31] Yang has proposed an algorithm for mining short association rules in one data base. In this the authors uses the exact algorithms to generate the result frequent itemsets. In exact algorithms, the result sets consist of all of the itemsets the support values of which are greater than or equal to the threshold support. The main drawback of this algorithm is it can only mine short itemsets, which cannot be applied to large itemsets.

In [32] Manku has proposed an algorithm lossy counting for calculating approximate frequency counts in data streams. It uses a lattice data structures to store itemsets .It splits an input stream of elements into fixed-size windows and processes each window sequentially. For each element in a window, it inserts an entry into a table, which monitors the number of occurrences of the elements, or, if the element is already in the table, it updates its frequency. At the end of each window, the algorithm removes elements of small frequency from the table. The main drawback of this algorithm is space bound, multiple scans and final output is based on previous data.

In [33] Cormode has proposed an algorithm for counting frequent items. The algorithm maintains a small space data structures that monitors the transactions on the relation, and whenever required, quickly outputs all hot items from the item set without rescanning the relation in the data base.

IV. Research Issues

The study of Mining Data Streams has raised many challenges and research issues as described below:

- Unbounded memory requirements.
- Handling the continuous flow of data.
- Transferring data mining results over a wireless network with a limited bandwidth.
- Integrating expert knowledge into data stream models.
- Scalability of data stream mining systems.
- Moving from data stream algorithms towards data stream tools.
- Modeling changes of results over time.
- Visualization of data mining results.
- Privacy preserving in data stream mining.

V. Conclusion

In the present era, continuous generation of data streams has lead to recent trend in the field of data mining termed as Data Stream Mining. So in this paper, we discussed various issues raised by data streams and presented an overview of various methodology used for generating synopsis data structures from continuous data streams. We also reviewed some algorithms developed in Clustering, Classification and Association domain of data mining, for dealing with the data stream. From our study we can conclude that data stream evolve massive volumes of temporally changing data So, traditional techniques of data mining cannot be applied straightforwardly. Research in data stream mining is in early stage. If the problems posed by data streams are solved and if more effective and interactive mining techniques which are user friendly are developed, it is likely that in the near future data stream mining will play an important role in business world, as it deals with many applications which involves mining from continuous data streams.

References

1. J. Han and M. Kamber, Data Mining: Concepts and Techniques,J. Kacprzyk and L. C. Jain, Eds. Morgan Kaufmann, 2006, vol. 54, no. Second Edition.
2. CharuC Agrawal, Data Streams: Models and Algorithms,. Springer (Science +Business Media, LLC, 233 Spring Street, New York, NY 10013,USA), 2007.
3. A.BIFET, G. HOLMES,R.KRIKBY AND B.PFAHRINGER,DATA STREAM MINING-A PRACTICAL APPROACH,2011.
4. Madjid Khalilian , Norwati Mustapha, " Data Stream Clustering: Challenges and Issues", Proceedings of the International MultiConference of Engineers and Computer Scientists,2010Vol1,IMECS 2010,March 17-19,2010,HongKong.
5. Vitter J. S. (1985) Random Sampling with a Reservoir. ACMTransactions on Mathematical Software, Vol. 11(1), pp 37–57.
6. Gibbons P., Mattias Y. (1998) New Sampling-Based Summary Statistics for Improving Approximate Query Answers. ACM SIGMOD Conference Proceedings.
7. M.Datar,A.Gionis, P. Indyk, andR.Motwani.Maintaining stream statistics over sliding windows. SIAM Journal on Computing, 31(6):1794–1813,2002.
8. Poosala V., Ganti V., Ioannidis Y. (1999) Approximate Query Answering using Histograms. IEEE Data Eng. Bull.
9. JagadishH.,KoudasN.,Muthukrishnan S., PoosalaV., SevcikK., and Suel T. (1998)OptimalHistogramswithQualityGuarantees. VLDB Conference.
10. Aggarwal C., Han J., Wang J., Yu P. (2003) A Framework for Clustering Evolving Data Streams. VLDB Conference.
11. Zhang, T.,Ramakrishnan ,R, Livny , M.Brich: An efficient data clustering method for very large data bases, In SIGMOD, Monteral , Canada , ACM(1996).
12. Keim D. A., Heczko M. (2001) Wavelets and their Applications in Databases. ICDE Conference.
13. Guha S., Kim C., Shim K. (2004) XWAVE: Approximate Extended Wavelets for Streaming Data. VLDB Conference, 2004.
14. Dobra A., Garofalakis M., Gehrke J., Rastogi R. (2002) Processing complex aggregate queries over data streams. SIGMOD Conference, 2002.
15. Dobra A., Garofalakis M. N., Gehrke J., Rastogi R. (2004) Sketch-Based Multi-query Processing over Data Streams. EDBT Conference.
16. Ganguly S., Garofalakis M., Rastogi R. (2004) Processing Data Stream Join Aggregates using Skimmed Sketches. EDBT Conference.

17. L. callaghan, N. Mishra, A. Meyerson, S. Guha, and R. Motwani, "Streaming-Data Algorithms for High-Quality Clustering," in Proceedings of IEEE International Conference on Data Engineering,
18. C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A framework for clustering evolving data streams," in Proceedings of the 29th international conference on Very large data bases - Volume 29, ser. VLDB '03. VLDB Endowment, 2003, pp. 81–92.
19. Kranen,Assent,Baldauf,Seidl, "Self Adaptive any time clustering",ICMD, 2009
20. 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, ser. VLDB. VLDB Endowment, 2004, pp. 852–863.
21. K. Udommanetanakit, T. Rakthanmanon, and K. Waiyamai, "E-stream: Evolution-based technique for stream clustering," in Proceedings of the 3rd international conference on Advanced Data Mining and Applications, ser. ADMA '07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 605–615.
22. W. Meesuksabai, T. Kangkachit, and K. Waiyamai, "Hue-stream: Evolution-based clustering technique for heterogeneous data streams with uncertainty." in ADMA (2), ser. Lecture Notes in Computer Science, vol. 7121. Springer, 2011, pp. 27–40.
23. P. P. Rodrigues, J. a. Gama, and J. Pedroso, "Hierarchical clustering of time-series data streams," IEEE Trans. on Knowl. and Data Eng., vol. 20, no. 5, pp. 615–627, May 2008.
24. Domingos P. and Hulten G. (2000) Mining High-Speed Data Streams. In Proceedings of the Association for Computing Machinery Sixth International Conference on Knowledge Discovery and Data Mining.
25. Hulten G., Spencer L., and Domingos P. (2001) Mining Time-Changing Data Streams. ACM SIGKDD Conference.
26. Aggarwal C., Han J., Wang J., Yu P. S., (2004) On Demand Classification of Data Streams, Proc. 2004 Int. Conf. on Knowledge Discovery and Data Mining (KDD'04), Seattle, WA.
27. Law Y., Zaniolo C. (2005) An Adaptive Nearest Neighbor Classification Algorithm for Data Streams, Proceedings of the 9th European Conference on the Principals and Practice of Knowledge Discovery in Databases, Springer Verlag, Porto, Portugal.
28. Wang H., FanW., Yu P. and Han J. (2003) Mining Concept-Drifting Data Streams using Ensemble Classifiers, in the 9th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), Washington DC, USA.
29. Chang, 2003] Joong Hyuk Chang, Won Suk Lee, Aoying Zhou; Finding Recent Frequent Itemsets Adaptively over Online Data Streams; ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining; August 2003.
30. [Lin, 2005] Chih-Hsiang Lin, Ding-Ying Chiu, Yi-Hung Wu, Arbee L. P. Chen; Mining Frequent Itemsets from Data Streams with a Time-Sensitive Sliding Window; SIAM Int'l Conf. on Data Mining; April 2005.
31. [Yang, 2004] Li Yang, Mustafa Sanver; Mining Short Association Rules with One Database Scan; Int'l Conf. on Information and Knowledge Engineering; June 2004.
32. G. S. Manku and R. Motwani. Approximate frequency counts over data streams. In In Proceedings of the 28th International Conference on Very Large Data Bases (VLDB), 2002.
33. Cormode, 2003] Graham Cormode, S.Muthukrishnan; What's Hot and What's Not: Tracking Most Frequent Items Dynamically; ACM Transactions on Database Systems; March 2005.

AUTHOR(S) PROFILE

Twinkle B Ankleshwaria, has received her B.E. degree in Computer Engineering from Mumbai University, India in 2007 and now she is pursuing M.Tech in CSE branch from Government Engineering College, Modasa-India. She has four years of teaching experience. Her area of research includes data mining privacy preserving

Prof. J. S. Dhobi, has completed his BE and ME in Computer Engineering. He has published many papers at national and international level. He has more than 15 years experience of teaching at undergraduate and graduate level.His area of research includes Computer Network, Data Mining, Web Mining and Security. Currently he is working as head of Computer Science and Engineering department at Govt,College of Engineering, Modasa-India.

