Global Journal of Computer Science and Technology
Software & Data Engineering
Volume 13 Issue 5 Version 1.0 Year 2013
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Benchmark Algorithms and Models of Frequent Itemset Mining over Data Streams: Contemporary Affirmation of State of Art

By V. Sidda Reddy, Dr. T.V. Rao & Dr. A. Govardhan
K.L. University, India

Abstract - Data mining and knowledge discovery is an active research work and getting popular by the day because it can be applied in different type of data like web click streams, sensor networks, stock exchange data and time-series data and so on. Data streams are not devoid of research problems. This is attributed to non-stop data arrival in numerous, swift, varying with time, erratic and unrestricted data field. It is highly important to find the regular prototype in single pass data stream or minor number of passes when making use of limited space of memory. In this survey the review on the final progress in the study of regular model mining in data streams. Mining algorithms are talked about at length and further research directions have been suggested.

Keywords : *data stream mining; sliding window; training model; linear reparability; data mining, frequent pattern; combinatorial approximation; single-pass algorithms; bit-sequence representation.*

GJCST-C Classification : H.2.8

Strictly as per the compliance and regulations of:

Benchmark Algorithms and Models of Frequent Itemset Mining over Data Streams: Contemporary Affirmation of State of Art

V. Sidda Reddy [α], Dr. T.V. Rao [σ] & Dr. A. Govardhan [ρ]

Abstract - Data mining and knowledge discovery is an active research work and getting popular by the day because it can be applied in different type of data like web click streams, sensor networks, stock exchange data and time-series data and so on. Data streams are not devoid of research problems. This is attributed to non-stop data arrival in numerous, swift, varying with time, erratic and unrestricted data field. It is highly important to find the regular prototype in single pass data stream or minor number of passes when making use of limited space of memory. In this survey the review on the final progress in the study of regular model mining in data streams. Mining algorithms are talked about at length and further research directions have been suggested.

Keywords : data stream mining; sliding window; training model; linear reparability; data mining, frequent pattern; combinatorial approximation; single-pass algorithms; bit-sequence representation.

I. Introduction

Frequent item set mining [48] is popularly known to be very essential in several crucial data mining activities. These activities include clusters [20], classifiers [46], sequences [62], correlations [14] and associations [35]. Several studies that point to mining frequent item sets on statistic databases and several proficient algorithms are suggested.

In recent years, data streams become an active research work in computer applications such as database systems, distributed databases and data mining. Data streams also importance to study online mining of frequent item sets, which is much needed for

These applications include sensor networks, e-business and stock market analysis, trend analysis, fraud detection in telecommunications, network traffic analysis, and web log and click-stream mining. It is ever more tedious to perform data mining and advanced analysis on huge and rapidly arriving data streams to capture remarkable development, model and exceptions. This is recognized to the fast appearance of these new application domains.

Author α : Research scholar, JNTU, Hyderabad, AP, India.
E-mail : siddareddy.v@gmail.com
Author σ : Professor, School of Computing, K.L. University, AP, India.
E-mail : tv_venkat@yahoo.com
Author ρ : Professor and DE, JNTU, Hyderabad, AP, India.
E-mail : govardhan_cse@yahoo.co.in

Data streams are continuous and high speed flow of data items that come in an appropriate order. These are quite different when compared to the data in traditional static databases.

Apart from having data distributions that modify with time, the data streams are unbounded, usually come in high speed and are continuous [23].

Data stream is further divided into offline streams and online streams. Normal bulk arrivals are attributed to offline streams [13]. Making reports on web log streams is regarded as mining offline data streams this is due to the reports that are created on the log data for a specific period of time. Backup devices or queries on updates to warehouse form other examples for offline streams. Queries on these streams can be permitted to treated offline.

Online streams are distinguished by instantaneously restructured data that appear once at a time. Calculating the regularity estimation of the internet packet streams is an application of mining online data streams, since internet packet streams is an instant one at a time packet process. Sensor data, network measurements and stock tickers are the other online data streams. Apart from keeping pace with the high speed of online queries, these must also be developed online. Immediately on their arrival, they need to be processed. It is not possible to process bulk data for mining data streams comes with a new set of questions. Primarily, to keep the whole stream in the main memory or in a secondary storage area it will be impractical. This is because data is streamed non-stop and the quantity of data is limitless. Secondarily it is not feasible to use traditional mining techniques with multiple scans like stored datasets. In data streams, data will be streamed and flow with high speed only once. To keep pace with high data arrival rate the mining streams will need quick, real-time processing. The results will likely to be available in a short time span. Also when the item sets are combine it can intensify mining regular item sets over streams in regards to both processing frequency and memory consumption.

Due to these limitations, research work performed on approximating mining results, with some sensible assurance on the value of approximation.

a) Distinguish between data streams and static data

Static data contains persistent data relations, approach of querying is one-time, data set is passive and randomly accessed, no role of response time on result accuracy, update speed is minimal and responses in passive mode.

Data streams contains transient data relations, the querying process is continuous, data set with continuous updates and sequentially accessed, response time influences the performance and results accuracy, update speed is maximal since response type is active

II. Taxonomy of Issues in Mining Data Streams

Regular item set issue is a daunting problem. For instance, there can be a massive number of regular item sets because of combinational explosion. The test here lies in how effectively can we detail, find and stock up these regular item sets. Due to their exclusive features, these data streams have added many more new challenges in regular item set mining.

Lately, in a couple of data sources, the data generation has become rapid than before. This quick production of non-stop data streams has questioned storage, communication and computation capacity in the computing systems. It is quite demanding to store, mine and query these data sets. Pulling out knowledge structures present in models and patterns in the uncontrollable streams of information is what mining data streams are all about.

The demands and the research problems faced in the data stream mining is motivational. Several demands concerned with this arena are explained here.

a) Handling the continuous flow of data

This problem is concerning the data management. These high data rates cannot be handled by the traditional database management systems. As a result, it is unreasonable to keep all the information in relentless media. In addition, it is very high-priced to aimlessly check the data several times. The demanding task here is to access the data item once and get frequent item sets. To deal with these fluctuations it is important to see to new indexing, storage and querying techniques.

b) Data reduction and synopsis construction

To comply with the earlier mentioned problems the data stream analysis, classification, querying and clustering applications will need some type of specification techniques. These techniques will be used to give out fairly accurate answers from huge data sets generally by means of synopsis construction and data reduction. This can be done by choosing a subset of incoming data or by making use of sketching, aggregation techniques, load shedding.

c) Minimizing energy consumption

In the resource-constrained environment a massive quantity of data streams are produced. For example: Sensor network. Devices here have short battery life. Energy efficient designs are very important because sending every generated stream to a central site is energy inefficient in addition to its lack of scalability problem.

d) Unbounded memory requirements

Data streams have unbound data but the storage that can be used to find or preserve frequent item sets is inadequate. Machine learning techniques show the core basis of data mining algorithms. When the analysis algorithm is implemented, it is important that the machine learning methods have the information in the memory. Because of this large number of produced streams, it is extremely essential to design space effective techniques that can have one look or less over the incoming stream. The frequency of an item set depends on time; this is another result of unbounded data. It is very demanding to use inadequate storage and find dynamic frequent item sets from unbounded data.

e) Transferring data mining results

Representation of knowledge is a new important research essential. The structures must be transferred to the user after getting it from models and patterns from data stream generators. Kargupta et al [6] have dealt with this issue. He has used Fourier transformations to capably transfer the mining results in a limited bandwidth links.

f) Modeling changes of result over time

In a couple of cases, the user is disinterested in the mining data stream results. But how do these results change over time. For instance, clusters formed by changes in data stream, may symbolize the changes in the dynamics of the coming stream. This change would help many temporal-based analysis applications like disaster recovery, emergency and video-based surveillance.

g) Real-time response

There is a demand on response time as the data stream applications are very time specific. Algorithms that come slower than the data arriving rate in constrained situations are of no use.

h) Visualization of data mining results

Research is still on in revelation of traditional data mining results on a desktop. It is a real challenge to see visualization in the small screens of the PDA. If a businessman is seeing the results of data being streamed and analyzed on his PDA, the results should be so effective that it should facilitate him to take quick decisions.

III. Taxonomy of Data Preprocessing in Mining Data Streams

a) Sampling

Data size can be lessened by using the random sampling technique. This can be done while capturing its important attributes. By using this form of summarization in a data stream and other summation can be built by using this example itself [2]. The data size is mandatory to get unbiased sample of data. The sampling approach is the simplest approach that has periodic time intervals which gives nice way to dwindle the data stream. This approach can have huge information loss in the stream as the data rates change.

b) Sketching

Sketching is a process of constructing a statistical synopsis of a data stream that makes use a little amount of memory and has frequency moments.

c) Histograms

Histograms are synopsis structures that can total the distribution value of the dataset. Tasks such as data mining, approximate query answering and query size estimation uses histograms.

d) Wavelets

These are methods for estimating data with a known likelihood. Wavelets co-efficient are predictions of a known signal (set of data values) into an orthogonal set of source vectors.

e) Concept Drifts

This is a crucial field in data mining. We need to study the judgment of swift and precise concept drift, the successful use of concept drift acquisition, how to save and serious use of concept and inclination of concept drift.

f) Sliding Window Models

The recent most data streams are examined in the sliding window models. The recent data items and summarized versions are examined in detail. Many techniques have accepted by many techniques. A preset newly generated data items, intended for data mining are taken up and knowledge discovery is performed on this. To make this synopsis structure several other synopsis structures are conversed to the entire data stream. It is assumed that the sliding window model is inspired to use the most recent data in the data stream [9]. Hence only a set history of the data stream is taken into consideration for the analysis and processing.

g) Landmark-window models

Regular sets are worked out from adjacent transactions in a stream which is known between a particular transaction in the past called a landmark and the existing transaction. While one steadily increases with the arrival of new transactions, the other endpoint (landmark) remains fixed. Transactions generally come in batches for this model that is generally appropriate for data streams.

h) Damped window model

Compared to the preceding windows the most new one have crucial importance in the damped windows. Older transactions do not add much towards the item set frequencies. For instance, the sliding window model will compute the average. The importance of data go slow exponentially into the past in the damped window model.

IV. Mining Frequent Item Set Over Data Streams: A Contemporary Affirmation of the State of Art

A regular item set mining algorithm over the data stream has been developed by Giannella et al [56]. Tilted windows have been used to estimate the regular patterns for the newest transactions built on the fact that the interest of the users lie on the newest transactions. Frequent item sets are symbolized by a tree data structure called as FP-stream, an increasing algorithm is used to maintain this. The earlier historical data is made use of to estimate the regular patterns increment by the implemented algorithm. A statistical technique was proposed by Laur et al [61] that is an unfair estimation of the support of sequential patterns or recall as selected by the user and restricts the degradation of the different principle. Using statistical support the conventional minimal support condition for checking regular sequential patterns was replaced. The theoretical foundations of the data stream analysis were checked by Gaber M M [49]. He also checked the mining data stream systems and techniques. The issues in the streaming was outlined and contemplated upon.

Observation: Algorithms [56] [13] [6] [49] that deal with mining frequent or sequential patterns on the data streams spotlight on how to technically deal with large historical data. There is no assurance that the algorithms can be run in limited memory capacity, when the data becomes huge and does not follow the running time.

MOMENT, an algorithm was suggested by Chi et al [63]. For continuous streaming data this showed sliding window to mine closed regular itemset. CET an internal data structure is used to check and store the closed item sets and boundary nodes. Teng [4] suggested a FTP-DS model that has sliding windows to ease the heavy information volume brought in by the continuous data streams. Data streams from the item sets give in temporal patterns to the FTP-DS mines. The sliding window data model is used by the FTP-DS an approximate mining algorithm. Li et al [11] created a DSM-FI technique to mine frequent item sets over the whole historical stream information. A distinctive top-down frequent item set discovery scheme and compact

pattern representation is accepted by the DSM-FI. Another algorithm on lossy counting was shown my Manku [57] and Motwani [13]. For the very first time mining, frequent item sets over the whole streaming data was used.

Observation: Historical Meta patterns got through data scanning are stored in internal data structures. This is a familiar feature that is explained in [4][11][63][13]. Additional procedures are created to re-mine the internal data structures when the user instigates a request for mining results.

Using a data structure called FP-Stream to keep historical knowledge such as item set frequencies Giannella et al [53] proved an approximation algorithm in random time intervals for mining frequent item sets. Linear Potential Function (LPF) algorithm to estimate propagation in Bayesian networks was shown by Zhang and Zhang [32]. An algorithm for keeping up frequent item sets in stream data in the supposition that every transaction holds a weight that is related to its age, permitting transactions on various ages to be dealt with different approaches was developed by Chang and Lee [38]. Stream an online algorithm was given by Asai et al [65] targeted mining patterns from semi-structured stream data like the XML data.

Observation: [53] [32] [38] [65] are the methodologies that permit handling of huge or infinite data streams in an estimated manner.

Cheung et al [42] [2] showed algorithms FUP and FUP2 for augmenting the frequent item sets. Similar algorithm was proposed by Thomas et al [12]. The mentioned models benefit from the tie up between the original database (DB) and incrementally changed transactions (db) and they also assume batch updates. Also known as Apriori algorithm FUP is a multiple-step algorithm. ZIGZAG was an algorithm proposed by Veloso et al [52] for mining regular item sets in developing databases. ZIGZAG was later extended by Otey et al [67] as parallel and distributed algorithms. [47] [2] and [12] have a striking similarity with Zigzag when both accelerate by using DB and db.

Observation: The main thing noted in the FUP is that some regular item sets will remain regular and some earlier irregular item sets will become regular when db is added to DB. Such item sets are called as winners. In addition some earlier regular item sets will become irregular. Such item sets are called as losers. The main method of FUP is to use the data in db to separate some winners and losers and thereby decrease the size of the candidate in Apriori algorithm. The working of the Apriori algorithm heavily relies on the size of the candidate set. Apriori's working is improved *to a large extent* by the FUP. Previous transactions from the database can be removed by extending FUP to FUP2. Hence FUP2 is used in the sliding-window data model and FUP will deal only with accumulative data model. FUP2 and the algorithm told in [12] are alike except that supplementing the regular item sets a negative border is retained. The regular item sets of db are mined first in the algorithm. Also the regular item sets in DB are updated in parallel. The regular item sets in the restructured database are calculated with a possible scan of the restructured database which is based on the change of the regular item sets in DB, the negative border in DB and the regular item sets in db. The algorithm [12] works well as the changed database is scanned at most once. Both accumulative and sliding window can make use of algorithm [12]. The [47], [2] and [12] come under the exact mining algorithm. ZIGZAG comes with many dissimilar features. Zigzag increases the speed of the support counting of common item sets in the reconstructed database and it does not find the regular item set in db itself. Hence with the lowest support ZIGZAG can take care of the batch update with random block size. It also takes in the techniques given by the GENMAX algorithm [31] and in every update it sustains maximum regular item sets. Sometimes this is not enough, at that point of time a second step is used in ZIGZAG. Here the restructured database is checked to find all regular item set and their supports.

An algorithm was created by Chi et al [16] where closed regular item sets are mined over data stream sliding windows. A condensed data structure called a closed enumeration tree (CET) was brought in to keep a vigorous selected set of item sets over the sliding window. There is a boundary in the closet regular item sets and the other item sets. The boundary movements in the CET will show the concept drifts in data streams. Sliding – window data model are the exact mining algorithms used by both the ZIGZAG and Moment.

1-pass algorithm, Count Sketch, was shown by Charikar et al [37] that resend the most regular items whose frequencies to convince an entry with high probabilities. A randomized algorithm was developed Manku et al [13] for upholding regular items over a data stream for a time t, the regular items defined over the whole data stream up to t.

Observation: No false negativity is promised by these algorithms and restriction the error of the calculated frequency. The Lossy Counting Algorithm is further improved to handle regular item sets. A process called as trie takes care of all regular item sets, this trie is restructured by batches of communications in the data stream. The algorithm Manku et al is attempt for a tunable negotiation between error bounds and memory usage. Accumulative data model is used by the Lossy Counting, Sticky Sampling and Count Sketch.

An algorithm presented by Chang et al [38] estDec, here frequency is defined by the age of a function. During a random time intervals, data streams mined by regular item sets, Giannella et al [56] proposed this algorithm. FP stream is helpful in storing

and updating historic data for the regular item sets and their occurrence with time and a period function is made use of to update the entries so that the newest entries are prejudiced more.

Observation: estDec and Giannella's algorithms are very prejudiced and are basically approximate mining algorithms. An error level in Giannella's algorithm is wee bit different as it gives error levels for information at different levels.

The numbers of patterns available have been vast. Showing the pattern in a condensed form is being worked upon. For example, Pei et al [19] and Zaki et al [18] show well-organized Closet and Charm. An item set is known as closed if all its supersets do not get the support that it gets [30]. Fp-tree was another efficient information structure that was coined by Han et al [57] for efficiently saving transactions for a specified low support threshold. A perfect algorithm known as FP-growth for mining Fp-tree was suggested.

Observation: Mining data streams does not get much support from the Fp-growth algorithm. In this case it goes through each window and is prohibitively expensive for big windows.

Windows methodology has gained a considerable amount of interest from regular item sets in data streams [43] [15] [7-9] [17] [59]. Therefore false negative based approach for mining of regular item sets over data streams has been proposed by Yu et al [1]. Lee at al [73] suggested the generation of k candidate sets from (k-1) candidate sets without checking their frequency. Jiang et al [17] recommended an algorithm for increasing holding the closed regular item sets over data streams. Extra passes over the data can be avoided this way and also this will not conclude in too many added candidate sets. Chi et al [63] advised the Moment algorithm for having a closed regular item sets over the sliding windows. When it comes to bigger slide sizes, Moment does not really help. Increasing mining on regular item sets is supported by Cats Tree [43] and Can Tree [15]. A good amount of work has been done by counting the candidate item sets more capably.

Observation: Park et al [8] suggested the hash-based counting method which was made use by several before mentioned regular item sets algorithms [7-2, 18, 8]. On the contrary Brin et al [27] suggested an active algorithm known as DIC for competently counting item sets frequencies. Fp-tree data structures are made use of by the fast verifiers in this model. The thought of putting conditions is to get must faster delta maintenance and counting patterns.

An association rule generation and summarization is another important area of work. Won et al [66] suggested a system for a hierarchical rule generation driven by ontology. In addition, Kumar et al [29] also suggested a single pass algorithm centered on the hierarchy-aware counting and transaction pruning for mining association rules, after the structure is given. Liu et al [64] recommended a methodology to systematize and sum up the found association rules. This methodology simplifies the rules and keeps stays on exceptions to the generalization.

Observation: [66] focuses on holding the level of items and the categorization of rules making use of a hierarchical association rule collecting the groups the created rules from the item space to the hierarchical space. To find comprehensive association, simplified methods [29] can be used over the history of organization. Based on appealing measures [21] [71] many research projects in this arena have looked upon the rule ordering. For example, Li et al [21] suggested rule position depending on assurance, sustenance and the number of items on the left hand side to trim the found rules.

In the past 10 years, a good number of algorithms have been suggested on the mining regular patterns in data streams. These can be divided into three special categories, landmark based [4, 13, 1, 9] damped or time decay based [39, 33] and a sliding window based [26-34] algorithms. DSM-FI [11] is an innovative algorithm which converts all operations into minor transactions and is put in the synopsis data structure known as the item-suffix regular item set forest that is based on the prefix-tree. In order to get fairly accurate results of regular patterns over the landmark window the authors made use of Chernoff Bound [1]. Lattice structure was made use by Zhi-Jun et al this is also known as the regular enumerate tree that is separated into many equal classes of the accumulated patterns with the similar transaction ids in a single class. estDec was recommended by Change and Lee reproduced the time decay model where every transaction has an influence that lessens with age [39]. Algorithm [33] which is alike estDec, it is said that mining increasing number of regular item set instantaneously over data stream based on a damped model.

Observation: The data that enters from a particular time called landmark till the existing time is taken into consideration in the landmark model. This time can be the starting or the restarting time. Earlier and existing transactions of the input stream are considered similar. Regular models are further separated into equivalent classes and only these regular patterns symbolize the two borders of every class are preserved, other regular models are trimmed.

Significance is given to data elements based on their arrival order in the time decal model. This is done in order to highlight the recently arrived data. A decay rate is shown to lessen the effect of the previous transaction in the set of regular pattern.

Numerous sliding window based algorithms recommended for regular item set mining over data streams. Raw transactions of sliding windows can be stored using DS Tree [26] and CPS-Tree [74]. Fixed tree

structure is used by DS Tree in canonical order of the branches and CPS-Tree is recreated to keep a check on the memory usage. Both these use the FP-Growth [418] for mining as suggested for static databases. MFI-Trans based on priori algorithm was recommended in [54]. Every regular item sets is mined over the newest window of transactions. Bit string is made use of for all items to keep its happening data in the window. A sliding based algorithm has been recommended in [28]. This has a window content that is kept vigorously using a set of easy lists.

Lin et al [51] suggested a fresh technique for mining regular patterns on time receptive sliding window. Here window is divided into batches and mining for this item set is done discretely. A limit in frequent closed item set and other item set is kept; this helps the Moment algorithm to locate the closed regular item sets.

The regular item set in one pane of the window are taken into consideration for further check to look out for regular item sets in the entire window in SWIM [77] a pane based algorithm. The union of these regular patterns of all panes is maintained. It also increasing updates support and trims irregular ones. Transactions are stored as a prefix tree for each plane. The authors devised algorithm for mining [4-10] constantly to maintain the non-derivable regular item sets of the sliding windows. Closed regular item sets and non-derivable can be viewed as a synopsis of all regular item sets.

estWin algorithm was recommended by Chang and Lee it locates the newest regular patterns adaptively on transactional data streams making use of the sliding window model. Monitoring lattice is made use of as a prefix tree to check the set of regular item sets over a data stream. Every node of the lattice will symbolize an item set that can be created by using items kept in the nodes in the way from the root to the node. All the data is kept in the final node of the item set. A Less amount and small support called as minimum important is used by the algorithm to know the new regular item set to approximate their support. The existing set of regular item set is updates going to the related item set in the monitoring lattice, when a novel transaction come from the input data stream. With this noteworthy item sets are known by second traversal of the related ways of the monitoring lattice making use of a transaction. In order to remove their effect when the algorithm expires keeps the set of transaction of the window in the memory. Connected ways of the monitoring lattice must be gone through to delete the previous transaction from the window. The algorithm trims irrelevant item sets from the tree after reaching a stable number of transactions making use of a method called as force pruning. When all the subsets become irrelevant and are kept in the tree an item set is put to the prefix tree. The support of the novel important item set in the old transaction of the window is estimated by the algorithm. The item set of length k (k-item set) calculated support is alike to the lowest support in all its subsets with length k-1. Information like possible count 9pcnt), error (err), actual count (acnt) and first transaction (mid) are available in each node of the prefix tree. Pcnt is known as the calculated frequency of the item set in the older transaction of the window before the item set was put to the prefix tree. Acnt is the checked frequency of the item set post its insertion in the tree. Err is termed as the error of the calculated support of the item set estimated using the subsets. Id of transaction that results the item set to be put in the tree is known as Mtid. The total of Acnt and Pcnt gives support of an item set in the prefix tree. To know more about data computing the probable count and the error in support of important item sets and other features of the estWin algorithm we can refer to [34]. On putting the item set, fresh transactions come in, previous transactions are removed and the error is minimized. Due to the removal of previous transactions the error will lessen.

New-Moment was used by H F Li [45] to preserve a set of regular closed item sets in data streams with transaction sensitive sliding window. Chi recommended MOMENT [63] which was a usual algorithm that can reduce the size of the data structure. A new way was suggested by N Jiang [70] for mining regular closed item sets over data streams. Compact data structure was brought in by Y Chi [16], this is a closed enumeration tree that keep a set of item sets that are dynamically selected in a sliding window. There is a border between the item sets and regularly closed item sets. FPCFI-DS was an algorithm suggested by F J Ao [10] for mining closed regular item sets in data streams. This makes use of single-pass lexicon graphical order FP-Tree-based algorithm that is merged with the ordering policy to mine the closed regular item sets in the primary window and the tree is updated for every sliding window.

Another mining task for recommended by J Y wang [T8-9] for mining top-k regular closed item sets whose length was greater than min_l.

Observation: compared to the sliding window based regular item set the data streams can be categorized as two groups. [26] [74] [54] and [28] is the first group. The window is kept above the newest transactions and frequent patterns in the current window are extracted by the mining process. This happens when the user submits a request. Hence the intention here is to keep the transactions making use of the low memory and do the mining process competently. [51] [9-8] [77] [41] and [34] is the second group where the mining results are constantly updated by making use of the fresh transaction to the window and previous transactions are removed from the window. Hence quick updation of the mining result and less memory usage gives the desired result. As you can see the mining

result with immediate effect the algorithms of the second group are proven to be more useful. To give out quickly the estimated algorithm that can find set of regular item sets in the sliding window with elevated quality of result is adequate. The first group [26-6] fails to adaptively maintain and update the mining result. The result becomes invalid when fresh transactions arrive from the stream. This results in the re-execution of the mining task. Conversely, relating a mining algorithm to the entire window will require significant processing and memory requirements specifically in big window size in respect to algorithms of second group [28] [51] [9-8] [77] and [410] where mining results are updated adaptively. Compared to the high arrival rated of the input streams the functioning of these algorithms are poor.

In [69], [75] and [44] sequential pattern-mining algorithms in data streams are presented. The issues relating to the mining distributed data stream was dealt in many areas of data mining such as mining regular item sets [55], [5], clustering [24],[3] and association rules [7], [50]. In contrast, sequential patterns of mining in several data streams are dealt in [72] and [42]. Collective Data Mining (CDM) has been suggested in [22].

Observation: The methods [69], [75] and [44] cannot deal with the distributed streams, as accumulating data streams in one location before giving them out come with a high-priced cost and can invade owner's privacy. Old mining results will not be preserved in MILE algorithm, which happens to be one-time fashioned algorithm. Therefore, this might take excessive amount of time in re-mining. In IAs pam algorithm this issue has been done away with. It was shown in [42] that increasingly mine across-streams sequential type to hold the fresh mining results. Nevertheless centralized setting strategy is made use here and it depends on a sliding window at the same working node to read the sample data streams. CDM has a purpose to guarantee that incomplete models made from local data at various sites are right. A global model can be formed from these models.

V. Conclusion

The deliberations above we observe that the present mining strategies have an increasing and one pass mining algorithm that are appropriate to mine data streams. Very few of them tackle the concept drifting problem. Fairly accurate results are obtained from these algorithms as a result of large data streams and less memory. Unlike traditional databases, here we cannot keep the frequency counts of all item sets in the entire data streams. Precise mining results are obtained by some suggested algorithms by keeping up to a minute subset of regular item sets from data streams, retaining their precise frequency counts. Sliding window data processing model helps in preserves only some section of the regular item sets in the sliding windows. Different other ways to maintain is maximal frequent item sets, closed frequent item sets and short frequent item sets.

The existing stream data mining techniques need users to describe one or more parameters before its implementation. But many of these streams do not tell how these parameters can be adjusted while they run. Waiting for the mining algorithm to cease to reset the parameters is not a good idea as it will take a long time for the execution as the data is massive.

Some recommended techniques permit the user to regulate some parameters online. There is no assurance that these parameters are of key importance and they might not be helpful in the mining scenario. Auto adjustment of mining algorithms and users adjusting online was also considered. The research in this field is in gestation stage. To tackle all those discussed issues in this paper would propel the process of developing association rule mining applications in data stream systems. Previous researches focused on creating competent mining techniques. A new aspect of data mining streams is necessary to find scalable regular item set mining models with the following factors.

1. Utility Mining
2. Weighted Supports
3. Multiple Supports
4. Transitional States
5. Temporal Validity

If many of these issues are yet too solved and a perfect and efficient system has to be developed, it is certain that in coming future data stream item set will be very important in the business world. Focusing on this we guesstimate more research in this direction.

References Références Referencias

1. Information Sciences, 176(14), 2006, pp. 1986–2015.
2. A General Incremental unquie for Maintaining Discovered Association Rules. Proceedings of the Fifth International Conference on Database Systems for Advanced Ap- plications (DASFAA), Melbourne, Australia, April 1, 4, 1997.
3. A Generic Local Algorithm for Mining Data Streams in Large Distributed Systems
4. A regression-based temporal pattern mining scheme for data streams. In: Proceedings of the 29th VLDB Conference (Berlin, Germany, 2003) 93–104.
5. A Sketch-Based Architecture for Mining Frequent Items and Item sets from Distributed Data Streams. In: Proc. of the 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (2011), pp. 245-253.

6. Alternative Interest Measures for Mining Associations. In IEEE TKDE, 15:57-69, 2003.
7. An Approach for Distributed Streams Mining Using Combination of Naïve Bayes and Decision Trees
8. An effective hash-based algorithm for mining association rules. In SIGMOD, pages 175-186, 1995.
9. An efficient algorithm for frequent itemset mining on data streams, Proc. ICDM, 2006, pp. 474–491.
10. itemsets in data Streams. Proceedings of the IEEE 8th International Conference on Computer and Information Technology, pp.37-42, 2008.
11. An efficient algorithm for mining frequent itemsets over the entire history of data streams. In: 1st International Workshop on Knowledge Discovery in Data Streams (Pisa, Italy, 2004) 20–24.
12. An Efficient Algorithm for the Incremental Updation of Association Rules in Large Databases. Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97), Newport Beach, California, USA, August 14, 17, 1997.
13. Approximate frequency counts over data streams. In: Proceedings of the 28th VLDB Conference (Hong Kong, China, 2002).
14. Beyond Market Basket: Generalizing Association Rules to Correlations. In Proc. of SIGMOD, 1997.
15. Cantree: A tree structure for efficient incremental mining of frequent patterns. In ICDM, pages 274-281, 2005.
16. Catch the Moment: Maintaining Closed Frequent Itemsets in a Data Stream Sliding Win- dow. Knowledge and Information Systems, 10(3): 265-294.
17. Cfi-stream: mining closed frequent itemsets in data streams. In SIGKDD, pages 592-597, 2006.
18. CHARM: An efficient algorithm for closed itemset mining. In SIAM, 2002.
19. CLOSET: An efficient algorithm for mining frequent closed itemsets. In SIGMOD, pages 21-30, 2000.
20. Clustering by Pattern Similarity in Large Datasets. In Proc. of SIGMOD, 2002.
21. CMAR: Accurate and efficient classification based on multiple class-association rules. In ICDM, pages 369-376, 2001.
22. Collective Data Mining: A New Perspective towards Distributed Data Mining. In: Advances in Distributed and Parallel Knowledge Discovery, editied by H. Kargupta and P. Chan, chapter, 5, AAAI Press / The MIT Press (2000).
23. Data Streams and Histograms; ACM Symposium on Theory of Computing; 2001.
24. Discovering Trend-Based Clusters in Spatially Distributed Data Streams. International Workshop of Mining Ubiquitous and Social Environments (2010), pp 107-122.
25. Discovery of Frequent Episodes in Event Sequences. In DMKD, 1:259-289, 1997.
26. DSTree: a tree structure for the mining of frequent sets from data streams", in Proceedings of IEEE International Conference on Data Mining, 2006, pp. 928–932.
27. Dynamic itemset counting and implication rules for market basket data. In SIGMOD, pages 255-264, 1997.
28. EclatDS: An Efficient Sliding Window Based Frequent Pattern Mining Method for Data Streams", Intelligent Data Analysis, Vol. 15(4), 2011, pp. 571-587.
29. Efficient algorithm for hierarchical online mining of association rules
30. Efficient mining of association rules using closed itemset lattices. Information Systems, pages 25-46, 1999.
31. Efficiently Mining Maximal Frequent Itemsets. Proceedings of the 2001 IEEE International Conference on Data Mining, San Jose, California, USA, November 29 , December 2, 2001.
32. Encoding probability propagation in belief networks IEEE Transactions on Systems, Man and Cybernetics (Part A) 32(4) (2002) 526–31.
33. est Max: Tracing Maximal Frequent Itemsets Instantly over Online Transactional Data Streams", IEEE Transactions on Knowledge and Data Engineering, 21 (10), 2009, pp. 1418-1431.
34. estWin: Online data stream mining of recent frequent itemsets by sliding window method", Journal of Information Science, Vol. 31, 2005, pp. 76–90.
35. Fast Algorithms for Mining Association Rules.
36. Fast algorithms for mining association rules" in Proceedings of International Conference on Very Large Databases, 1994, pp. 487–499.
37. Finding Frequent Items in Data Streams. Proceedings of the 2002 International Collo- quium on Automata, Languages and Programming, Malaga, Spain, July 8, 13, 2002.
38. Finding recent frequent itemsets adaptively over online data streams. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD2003) (Washington, DC, 2003) 226–35.
39. Finding recently frequent itemsets adaptively over online transactional data streams Frequent Itemset Mining Implementations. In Proc. of the ICDM 2003 Workshop on Frequent Itemset Mining Implementations, 2003.
40. Frequent pattern mining: current status and future directions", Data Mining and Knowledge Discovery, Vol. 15, 2007, pp. 55–86.
41. Incremental Mining of Across-streams Sequential Patterns in Multiple Data Streams. Journal of Computers Vol. 6 (2011), pp.449-457.

42. Incremental mining of frequent patterns without candidate generation or support. In 7th Database Engineering and Applications Symposium International Proceedings, pages 111-116, 2003.
43. Incremental Mining of Sequential Patterns over a Stream Sliding Window. In: Proc. of the 6th IEEE International Conference on Data Mining (2006), pp.677-681.
44. Incremental updates of closed frequent itemsets over continuous data streams. Expert Systems with Applications, Vol.36, pp.2451-2458, 2009.
45. Integrating Classification and Association Rule Mining. In Proc. of KDD, 1998.
46. Maintenance of Discovered Association Rules in Large Databases: An Incremental Up - dating Technique. Proceedings of the Twelfth International Conference on Data Engineering, New Orleans, Louisiana, USA, February 26 , March 1, 1996.
47. Mining Association Rules between Sets of Items in Large Databases. In Proc. of SIGMOD, 1993.
48. Mining Data Streams: A Review, Mining Distributed Evolving Data Streams using Fractal GP Ensembles. In: Proc. of the 10th European Conference on Genetic Programming (2007), pp. 160-169.
49. Mining frequent itemsets from data streams with a time-sensitive sliding window", in Proceedings of SDM International Conference on Data Mining, 2005.
50. Mining Frequent Itemsets in Evolving Databases. Proceedings of the Second SIAM International Conference on Data Mining, Arlington, VA, USA, April 11, 13, 2002.
51. Mining frequent itemsets over arbitrary time intervals in data streams. In: Technical Report TR587 (Indiana University, IN 2003).
52. Mining frequent itemsets over data streams using efficient window sliding techniques", Expert Systems with Applications, Vol. 36, 2009, pp. 1466–1477.
53. Mining Frequent Itemsets over Distributed Data Streams by Continuously Maintaining a Global Synopsis. Data Mining and Knowledge Discovery Vol. 23 (2011), pp. 252-299.
54. Mining Frequent Patterns in Data Streams at Multiple Time Granularities,
55. Mining frequent patterns without candidate generation. In: SIGMOD' 00 (ACM Press, Dallas, TX, 2000) 1–12.
56. Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach", Data Mining and Knowledge Discovery, Vol. 8, 2004, pp. 53-87.
57. Mining maximal frequent itemsets from data streams. Information Science, pages 251-262, 2007
58. Mining non-derivable frequent itemsets over data stream", Data & Knowledge Engineering, Vol. 68, 2009, pp. 481-498.
59. Mining Sequential Patterns on Data Streams: a Near-Optimal Statistical Approach.
60. Mining Sequential Patterns. In Proc. of IDCE, 1995.
61. Moment: maintaining closed frequent itemsets over a stream sliding window. In: Rajeev Rastogi et al. (eds) Proceedings of 4th IEEE International Conference on Data Mining (Brighton, UK, 2004) 59–66.
62. Multi-level organization and summarization of the discovered rules. In Knowledge Discovery and Data Mining, pages 208-217, 2000.
63. Online algorithms for mining semi structured data streams.
64. Ontology-driven rule generalization and categorization for market data. Data Engineering Workshop, 2007 IEEE 23rd International Conference on, pages 917-923, 2007.
65. Parallel and Distributed Methods for Incremental Frequent Itemset Mining. IEEE Transactions on Systems, Man, and Cybernetics, Part B, 34(6): 2439, 2450.
66. Querying and Mining Data Streams: You Only Get One Look. In Tutorial of SIGMOD, 2002.
67. Random Sampling Over Data Streams for Sequential Pattern Mining. In: Proc. of the 1st European Workshop on Data Streams (2007), pp. 61-66.
68. Research issues in data stream association rule mining. SIGMOD Record 35 (1), pp.14-19, 2006.
69. Selecting the right interestingness measure for association patterns. In Knowledge Discovery and Data Mining, pages 32-41, 2002.
70. Sequential Pattern Mining in Multiple Streams. In: Proc. of the 5th IEEE International Conference on Data Mining (2005).
71. Sliding window filtering: an efficient method for incremental mining on a time-variant database. Information Systems, pages 227-244, 2005.
72. Sliding window-based frequent pattern mining over data streams", Information Sciences, Vol. 179, 2009, pp. 3843-3865.
73. Stream Sequential Pattern Mining with Precise Error Bounds. In: Proc. of the IEEE International Conference on Data Mining (2008), pp. 941-946.
74. The space complexity of approximating the frequency moments. Compute. Syst. Sci., 58(1):137–147, 1999.
75. Verifying and mining frequent patterns from large windows over data streams", in Proceedings of International Conference on Data Engineering, 2008, pp. 179–188.

This page is intentionally left blank