International Journal of Research in Science and Technology http://www.ijrst.com

(IJRST) 2019, Vol. No. 9, Issue No. III, Jul-Sep **e-ISSN: 2249-0604, p-ISSN: 2454-180X**

A REVIEW ON STATIC AND STREAMED DATA[1]

Atul Antil

Undergraduate student, University of California, Sandiego, US

Received: 25 April 2019; Accepted: 20 June 2019; Published: 05 July 2019

ABSTRACT

This paper focuses on different continuous example mining strategies, their difficulties associated with static just as stream information condition. Data, the most valuable resource is the majority of the occasions covered up inside stacks of crude information. It is required to be cleaned to recover data, changed over into a structure which can be broken down for deciding. Many research papers were perused for the comprehension of different systems that mine incessant thing sets either in static or stream information conditions. This paper condenses all the well-known procedures accessible with us for continuous thing set mining and gives a few recommendations for upgrading them. The issues for static mining simply consider the existence complexities yet stream information mining is substantially more unpredictable and testing when contrasted with static. The issues like idea floating, nature of information, its handling model, insufficiency, size, its maintenance in distribution centres and so on are additionally required to be considered while working with constant information. The calculations for stream information mining ought to be gradual and asset versatile in nature with the goal that they can deal with the change and modify their preparing parameters as per the accessibility of assets separately. The accessible assets which will be useful in shared conditions (where assets are shared by various procedures).

1. INTRODUCTION

Data mining is one of the important phases of KDD process (a procedure for acquiring knowledge out of data). It is the process where we analyse huge data from store house to extract useful information with the help of intelligent techniques that aid in making decisions [1]. The paper will examine two procedures of information mining: visit examples and affiliation governs under static just as stream information condition.

2. DATA MINING TECHNIQUES: FREQUENT PATTERNS, ASSOCIATION RULES

Before explaining frequent patterns and association rules, one should have idea about two terminologies: Support and Confidence.

Support is the count of the number of transactions in which item appears in a dataset. Minimum Support is a value specified for finding frequent patterns. On the off chance that itemsets are happening together in a database more noteworthy than or equivalent to the predetermined help esteem, at that point, those itemsets are called visit itemsets [1].

Support (M=>N) = P (M»N) (2.1)

A confidence defines if a person purchases any item M, how many times he purchases item N with it1.

Minimum confidence value is set to validate associations.

Confidence(M=>N) =Support (M»N)/Support (M) (2.2)

Frequent patterns, as the term describes, are those patterns that occur together time and again in dataset. The patterns can be subsets of item sets, sequences, graphs, structures. When transactional database is discussed, frequent item sets are discovered depending upon the value of support. On the off chance that the check of event of a thing set is more noteworthy than or equivalent to the estimation of

[1] *How to cite the article:* Antil A., A Review on Static and Streamed Data; *International Journal of Research in Science and Technology,* Jul-Sep 2019, Vol 9, Issue 3, 1-6

least limit (support), that thing set will be considered as successive itemsets [1].

Association rule mining is done to discover hidden relationships among data items which can help in making decisions and predictions [2]. These are of the form A=>B(S,C), here A,B represents frequent item sets where A∩B=Ø and S and C represent minimum support and confidence values respectively. Associations are mined based upon confidence value. First, produce all non-empty subsets of a frequent item set. Then, for each subset, if The most famous application of association rule mining is Market Basket Analysis, a technique of analysing customer's behaviour by discovering relations among various items they place in their shopping baskets.

3. STATIC AND STREAM DATA MINING

Data streams are of two types: In Online data streams, we deal with real time data that needs to be updated regularly. This data comes one by one continuously in a sequence. For example: sensor data, stock tickers. In Offline data streams, data is collected at regular intervals in large amounts and then processed afterwards. For example, in web logs, data at certain intervals is logged and then processed offline. All the data mining techniques (data clustering, data classification, association rules and frequent patterns) are applicable to stream data but the process of extracting meaningful information from stream data is very complex. This paper reviews various approaches to find frequent patterns both in static as well as stream data environment [3].

4. CHALLENGES IN STREAM AND STATIC DATA MINING

Data Treatment Model: Data stream arises in limitless and uninterrupted manner and that too in large volumes. The issue is to draw out exchanges from an enormous information stream that would help in affiliation standard mining. Three systems were presented for information treatment. In Landmark model, a point known as a milestone is chosen. Every one of the exchanges starting there to the current is dug for discovering incessant examples. In the Damped model, every exchange is allowed some worth and this worth decreases with their timestamp. Ongoing exchange is having more an incentive when contrasted with more established. In Sliding window model, a sliding window is kept up in which a bit of stream is stacked in and handled.

Memory Management: Sufficient space for accommodating itemsets and their frequencies when a large volume of data arrives at once is the biggest issue. Moreover, with the arrival of fresh stream, the frequencies of itemsets vary most of the times. So, it is essential to gather up least amount of information. But this information should be sufficient to yield association rules.

Choice of Algorithm: The algorithm should be chosen according to the requirement of results. A few calculations give precise outcomes and some give surmised results with false positives or false negatives.

Concept Drift Problem: The itemsets which is frequent can become infrequent with the coming transactions and vice versa. Due to this varying nature of data, predictions of association rules can become inaccurate. This problem is known as Concept Drift and to handle it, incremental algorithms are required.

Resource aware algorithms: This algorithm required which can adjust their getting ready rate according to the openness of advantages. This thought will uncommonly be obliging in the earth where resources are shared by different techniques.

Each application has its own needs and issues. Clients ought to have the option to change the mining parameters as per their prerequisites notwithstanding when the calculation is running. Mining multidimensional data stream is another issue which increases the complexity. The applications need to generate responses in accordance with user's queries. If data is arriving from more than one source, it leads to the increased communication cost. Integrating the frequency counts is also an issue [4,5].

Authors discussed issues at all phases of KDD and in business environment. Data pre-processing is the most time consuming task and much complex in stream data as compared to static data because it is continuous, arrives every second in large volumes. In requirement gathering, most of the times clients are not able to formulate their business problems and business questions. They are unable to specify what they want from developers. The pre-processing challenges include noisy, varying data and outliers. In real world scenario, data is mostly incomplete, biased and even sometimes is not readily available. Selecting important information from large set of data is also a challenge. Data transformation is the most time consuming phase. Problem of concept drift in data is very difficult to handle. It is very hard to summate the information every second. Performing various operations like aggregation, association is not an easy task. Unlike static data mining, there are very few tools available which can assess the quality of stream mining algorithms. The retention period of data in data warehouse and making updates without compromising its availability is very big challenge. The system should be able to handle missing and noisy data. Data should be mined at different levels. The system should be scalable and support data migration, should be capable of handling multiple data types. Complexity of algorithms is another important issue [6–8].

5. STATIC DATA MINING TECHNIQUES FOR FREQUENT PATTERNS

5.1 Apriori Algorithm

It is a fundamental system for extricating regular examples by creating applicants. As the name infers, it requires the earlier information of incessant itemsets properties. It is a steady approach where incessant k-thing set is utilized to produce visit (k+1)- itemsets. At first, the database is examined for discovering tally of each of the 1-itemsets. Then dependent on the edge esteem; visit 1-itemsets are extricated. A cross join on the resultant is associated with getting all possible 2-itemsets mixes. Again database is checked for the counts of those itemsets and the methodology goes over until there are no new visit itemsets. To reduce the number of contenders, count uses Apriori property, moreover called plummeting end property, says, "If an itemsets isn't visited, its supersets will never visit". Hence, the estimation works in two phases: joining (cross join is performed on k-itemsets to make k+1 itemsets) and pruning (throwing out uncommon itemsets reliant on Apriori property). The shortcoming of using this count is that the database is required to be analysed different time which fabricates the execution time. The or age of the colossal number of up-and-comers manufactures the space complexity [1].

5.2 FP Growth

It is a methodology that focuses visit itemsets subject to segment and vanquish strategy. FP-Growth works in two phases: Creating and Mining FP tree. While making the tree, the database things are sifted and sorted out as a piece of tree in the dropping solicitation of their methods each trade. Things are separate close by these checks. The root is continually NULL. If some game plan of trade is as of now existing, by then the remainder of the things are joined underneath it and the count of subset things is extended by one. A tree is mined by building up its unforeseen model which joins the approaches to land at the centre point through the root. A subtree is assembled and models are made by connecting the thing with its way. Search space is diminished as a result of the time of prohibitive models. It gives extraordinary results for even long structures. Since there is no need for candidate age, space flightiness is decreased [1].

5.3 ECLAT

it is an improvement of Apriori figuring. It uses vertical data position (item: transaction id set). It resembles Apriori, basically the table is convoluted. The thing sets having check not actually least help farthest point will be discarded. 2-itemsets will be delivered by the intermingling of trade id sets of 1-itemsets. Cross join is performed to make three thing sets. The 2-itemset subsets of 3-itemset are evaluated from past table. From the dropping end property, 2-itemsets which are not visit, their 3-itemset will in like manner be uncommon. along these lines, those 3-itemsets are casted out. Count repeats till no new visit itemsets is created. As a result of this way of thinking, different yields of database are not required since trade id set contains all the required information for checking reinforces. Nevertheless, length of TID-set requires immense memory space. Estimation time is moreover affected during intersection point process1.

Numerous upgrades were made on these fundamental calculations to mine continuous examples efficiently. Another paper introduced a half and half Apriori calculation for regular itemsets mining utilizing both level (unique Apriori) and vertical Apriori(éclat). Both the methodologies were utilized all the while utilizing the idea of multithreading. The calculation utilized string based performing multiple tasks. As strings are light weighted, they require less space and CPU. Setting exchanging and interchanges are more affordable. Along these lines, multithreading approach is energized for upgraded use of CPU by diminishing its free time [9]. Half and half of Apriori and FP-Growth was introduced [10].More enhancements in Apriori were presented in [11,12].

6. STREAM DATA MINING TECHNIQUES

6.1 RAQ-FIG

RAQ-FIG (Resource Adaptive Quality Assuring Frequent Item Generation) by [13] is an improvement of MFI_TRANSW developed [13]. Here, data items in a particular transaction are presented in a bit sequence. If an item is in i-th transaction, i-th bit in the sequence is fixed to 1 and the rest to 0. After it, frequent ‘k’ itemsets are mined by executing logical AND operation on items. The RAQ-FIG algorithm includes the concept of resource adaptation and enhances the quality of results through its capability to adjust its processing parameters according to the availability of resources. The algorithm uses sliding window model to mine the most recent data. The model works by putting transactions into batches. A sliding window is composed of a number of basic windows. The main idea behind basic window is that rather than discarding single old transaction, whole batch can be discarded during window sliding. Figure 1 shows the concept of sliding window composed of various basic windows [13].

$$\text{Size of sliding window} = x * \text{number of basic windows allowed in a sliding window}$$

Initially, all the items of a transaction are presented in a bit sequence. The number of bits in a sequence depends upon the number of transactions. The concept will be clear with example. Say the number of transactions be four (t1, t2, t3, t4), then the number of bits will be four. w, x, y and z be the items present in various transactions in Table 1.

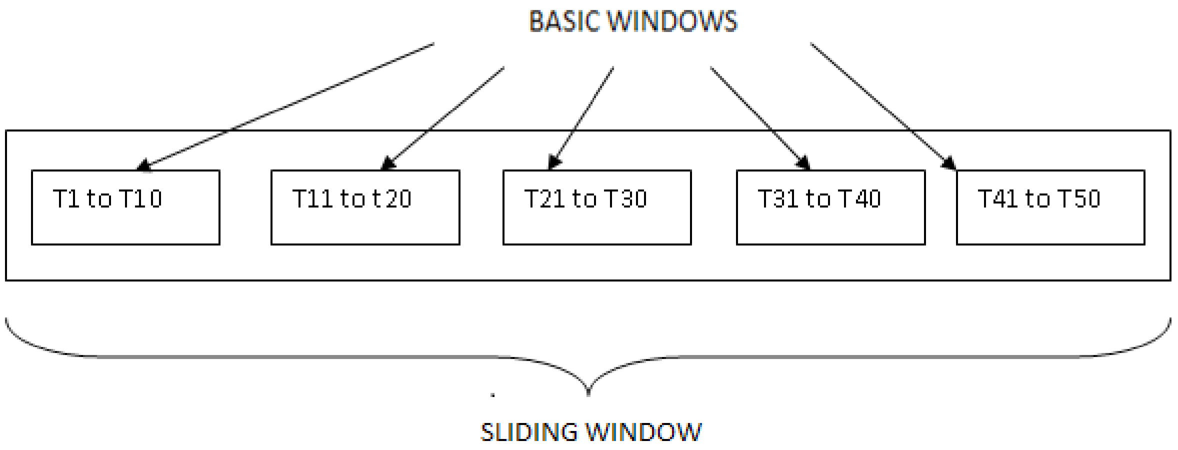
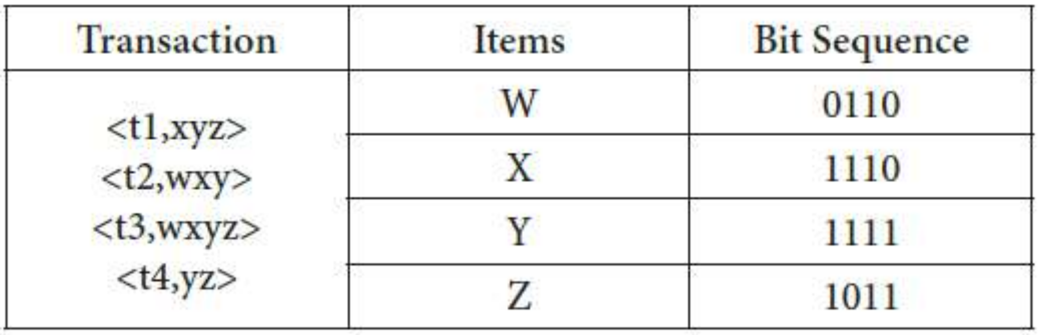


Figure 1. Chandrika: Sliding window composed of basic windows [13].

Table 1. Bit representation of items [13]



Transaction	Items	Bit Sequence
<t1,xyz> <t2,wxy> <t3,wxyz> <t4,yz>	W	0110
	X	1110
	Y	1111
	Z	1011

Since ‘w’, is present only in t2 and t3, corresponding bits are set to 1 and rest to 0. Representation of xyz= 1110 & 1111 & 1011 = 1010, hence x, y and z are present in t1 and t3. In this way, other item sets can be generated. 3-itemsets can be used to generate 4-itemsets. Using bit representation; the computation of frequent item sets becomes easier. One needs to just calculate the number of 1’s after AND operation and then compare it with support value. Current resource usage is traced with reference to space used, processing speed and input frequency. The quality of results is analysed with the ratio of error value to the minimal support value. Initially, the error is set by user and later on, the value varies depending upon resource availability. In this way, by controlling error, the quality of results is improved. All these measurements are made repeatedly. If it is felt that resources are not sufficient, adaptation parameter gets adjusted accordingly, for example, decreasing or increasing sliding windows size. Current resource usage is traced with reference to space used, processing speed and input frequency. The parameter which measures the current resource consumption is called as ‘Adaptation Factor’ (AF)

$$AF = (\text{processing time in seconds} * \text{speed of stream}) + \frac{\pi}{180} * \text{memory allocated}$$

The desired value is near 1. Modification in the size of sliding window depends upon this factor [13].

6.2 Extracting frequent itemsets at timespan

Another algorithm improved the temporal accuracy of subsisting algorithm (Mining frequent item sets over arbitrary time intervals by1ignoring outdated data and introducing a concept of ‘Shaking Point’. The base algorithm, FP streaming uses tilted time window treatment model and stocks the data in FP-tree which is updated with the arrival of new transaction. FP streaming tree consists of three components: A Transaction tree, which stocks the dealings for the current window; a Pattern tree, which stocks all the frequent patterns of previous windows and a tilted time window. Various improvements are made to FP-streaming algorithm. The concept of tail pruning is casted out for maintaining veracity of results. The input is read and filled in batch within a fixed time point after which algorithm discards the input and processes the batch. Initially, when an item arrives, its time stamp is set to current time. Each item is treated as node of a tree. With the arrival of each new transaction, if there is any new item, it is added to the tree. Frequencies of nodes are updated. The frequencies of items in the current item set increases and the remaining nodes in the tree are updated with 0 frequencies and this frequency is added to tilted time window. During processing, after some fixed number of batches, some nodes having continuously 0 frequencies will be encountered, these nodes are called obsolete items. A variable named as fading factor is assigned to each node. A user defined parameter called ‘fading support’ is passed. The timestamp of the node whose count is 0 will never be updated. After ‘x’ batches, Shaking Point will be performed where fading factor of every node is computed. It is equal to the difference between current timestamp and time-stamp of node. If this factor touches or is larger than the fading support, then that node and its full super tree will be wiped out. This helps to avoid useless procedures and saves time [2]

6.3 WMS (Weighted Minimum Support)

In WMS, algorithm, itemsets are assigned weights depending upon the time of their occurrence in the transactions. Database is divided into zones according to different time spans. Weights are assigned according to formula.

Db(t1,t2) is the number of transactions between the time t1 and t2. DB is the total number of transactions in DB and minimum support is the common support initially assigned for all transactions [15]

6.4 SWFP (Sliding Window Frequent Patterns)

SWFP used sliding window treatment model to find in-frequent patterns along with frequent item sets. To decrease the effect of old items, decomposition technique was used. The patterns which are not useful for final result are casted out which cuts down time and space complexity. The experimental results prove the efficiency and extensibility of SWFP [16].

6.5 More Algorithms

One more algorithm focused on the uncertain nature of stream data to improve the veracity of results. The items which are not frequent right now may become frequent in the coming transactions. Two tree based algorithms were used, first was capable of giving approximate results and the second one gave exact results. The concept of pre-minimum-support was introduced, where frequent patterns were found with support little less than the exact support threshold value. In this way, early casting out of items was avoided. Frequent item sets are stored in a tree structure called UF-stream. Then, the second algorithm helps in outputting exact results without any requirement of pre-minimum-support [17].

A new algorithm ‘output granularity’ was introduced. Stream data mining suffers from a problem that the rate of input stream is much greater than the processing and mining process of algorithms. Due to this, one is always left with estimated results. To solve this problem, author followed resource adaptation approach to maintain accuracy in results by reducing the rate of stream along with sampling and aggregation depending upon available resources. The main focus of algorithm was on the number of transactions that can reside in the primary memory before cumulative additions. The author explained the algorithm for clustering. The main component of algorithm was Resource Adaptation (RA) which tries to compute the minimal input rate and match the processing rate of algorithm with it. The algorithm can be applied for classification, clustering and frequent patterns [18]. Another algorithm was introduced for incremental mining in distributed environment. Here support is calculated at local servers and then sent to the main (global) servers or to their above hierarchy

7. DISCUSSION

Various algorithms in both static and stream data environments are studied. All are having some advantages as well as shortcomings. Multiple scans of database are required to find the support counts of item sets. Éclat emerged as an improvement of Apriori which uses vertical data format. FP-Growth uses a different approach of divide and rule. A tree is created and frequent item sets are mined. Due to limited number of database scans and zero candidates, it is efficient as compared to Apriori. Many improvements were made to these algorithms. The algorithm in time sensitive environment discussed in previous section helps in reducing the size of tree through the execution of shaking point but the problem is that data within a certain time span was taken for processing and rest was discarded. There is rise in execution time when shaking point was executed after some fixed number of batches. RAQ-FIG came out with a powerful approach of adjusting its processing parameters according to the availability of resources to save the system from being halted by getting stressed due to excess usage of resources. Many new algorithms emerged which introduced the use of sliding windows, weighted transactions, finding the frequent patterns at various granularities.

8. CONCLUSION

Mining frequent patterns in stream data environment is complex than that of static data environment due to their unbounded, continuous and varying nature. The data is generated every second in large amounts which makes its processing difficult. The algorithm should include the concept of resource adaptation. It should output the results or vary the processing speed according to the available resources. There is lots of challenges in mine data streams. Concept drift is the biggest problem. Various algorithms for mining stream data like FP-streaming, RAQ-FIG, MFI-TRANSW, WMS are available but there is no single algorithm which deals with all the challenges of stream data environment. Most of the algorithms give approximate results and very few generate exact results. The algorithms for data stream mining should be incremental in nature and address the problem of concept drift. Their incremental nature will make them adaptable to the changes in the distribution of input. There are many more challenges which are required to be overcome to mine frequent patterns efficiently

Financial Support and Sponsorship: Nil

Conflict of interest: None