ISSN No. 0976-5697

Volume 3, No. 2, March-April 2012

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Finding Association Rules Based on Maximal Frequent Itemsets over Data Streams Adaptively

T. Muthamilselvan*
Assistant Professor (Senior)
School of Information Technology & Engineering
VIT University, Vellore, India.
tmuthamilselvan@vit.ac.in

N. Senthil Kumar
Assistant Professor (Junior)
School of Information Technology & Engineering
VIT University, Vellore, India.
senthilkumar.n@vit.ac.in

I. Alagiri
Assistant Professor,
School of Information Technology & Engineering
VIT University, Vellore, India.
ialagiri@vit.ac.in

Abstract: Overflow of data streams are gathered and manipulated in sensor networks, communication networks, Internet traffic, and online transaction in financial market, power grids, and industry production processes, scientific and engineering experiments to yield better analysis. In contrast to conventional data sets, stream data have infiltrated from systems temporally ordered, rapidly fluctuated, massive and potentially infinite. It would be potentially cumbersome and very exponential to store the entire data streams or scan through it multiple times due to its tremendous volume.

This paper proposes the strategies to mine maximal data items and its data itemsets in single scan. Besides it generates association rules based on top maximal itemsets and data itemsets, which contain current and useful information for effective data analysis.

Keywords: Data Streams, Association Rule Mining, Memory Utilization, Frequent Itemsets, Hash Table.

I. INTRODUCTION

Contrary to conventional data sets, stream data are very radical, incremental and changing will fall through randomly at each phase of process. Data streams require abundant memory usage and restricted finitely when new data elements are continuously generated even after the saturated point [1]. Spontaneous stream analysis has been made for every newly generated data elements and produces the up-to-date analysis over the data streams which in turn resources are utilized instantly to prevent any bottleneck. In order to quench the requirements, there may be some sacrifices done to lift the correctness of its analysis by allowing trivial errors which will not bring any big impact of the process of analysis. Hence it has become necessary to develop a single-scan, online, multi-level and multi-dimensional stream processing and analysis methods.

Frequent itemsets are items that occur frequently in a data stream and which in turn also occur in the same transactions. Many algorithms have been proposed to find the frequent itemsets through generating large number of candidate itemsets and multiple pass scan [2]. By these existing algorithms, it has become unrealistic for infinite data stream. The incremental updates of frequent items for stream data makes it difficult because unprecedented fluctuations like frequent and infrequent items occur at random space and it made the whole process critical and not produces any qualitative analysis.

The effective utilization of memory and its maintenance would become a hard task for this process and it makes the real chore to the memory consumption [3]. The transaction which occurs frequently can be kept in main memory so that the data can be accessed at regular interval without any overhead. When the number of frequent itemsets is large, accessing the .information of frequent itemsets in a secondary disk needs more time. Due to this reason, this algorithm is not appropriate for an online data stream. At the same time when the buffer is enlarged, then newly generated transaction can be batch processed together, so that the algorithm would become more efficient and processed with utmost dimensions [4].

An association rule mining is defined as: Let I= {i1, i2, i_n} be a set of n number of possible items called *items*. Let D= {t_1, t2, .t_m} be a set of transactions called the *database*. A unique transaction ID is given to each transaction in D.and each transaction is a subset of the items in I. A *rule* is defined as an implication of the form X Y(s, c) where s is the percentage of records that contain both X and Y in the database, called support of the rule, and c is the percentage of records containing X that also contain Y, called the confidence of the rule. Association rule mining is to find all association rules the support and confidence of which satisfies user-specified minimum support and confidence, respectively.

The behaviors of the data stream are high speed, not predictable, unordered, burst in nature and analyze in main memory. These kinds of issues have to be considered while designing an algorithm for data stream mining [5]. With the limited memory space and effective usages of CPU cycles are to be considered to achieve effective and accuracy results.

II. RELATED WORKS

A. *Frequent item sets Counting:*

Manku and Motwani have proposed and implemented the lossy counting for approximate frequent itemset counts in data streams[6]. The implemented algorithm uses incrementally all the previous historical data to calculate the frequent patterns.

Chi et al developed an algorithm called moment algorithm to update frequent closed itemset incrementally over a sliding window [7]. This algorithm designed to work as in-memory tree based structure. The advantage of this idea is it finds every set of frequent closed itemsets in the current sliding window. The disadvantage is it is not able to handle burst and very high speed streams and the tree can be very huge for a large window.

B. Association Rule Mining:

The traditional algorithm for association rule mining is Apriori [8]. It eliminates many candidate sets using prior knowledge of frequent patterns. It requires multiple scans on the given database and consumes high CPU costs. Many algorithms are proposed based on Apriori algorithms to improve its performance but all are not suitable for data stream environment because they need multiple scans

Frequent pattern tree data structure and frequent pattern growth algorithms are used to find frequent patterns [9]. These algorithms give good performance when compared to Apriori algorithm it avoids generation of candidates. It constructs frequent pattern tree in two scans of database. Because of its two scan requirements we can not adapt for data stream mining.

III. COMPREHENSIVE APPROACH

A. Hash Based Technique to find Frequent Item sets:

Due to the high volume of data handled in on- line data streams and limitations in main memory we confined to finding the association rules only to the datasets that have top k frequency. In this methodology, an adaptive technique to find the top k frequent dataitems is used.

The incoming stream is conceptually divided into buckets of width **w.** Let N be the current stream length that is the number of items seen so far. This technique uses a frequency-list data structure for all items with frequency grater than zero. When the bucket boundary is reached (w,2w,3w etc)[10][14]. Frequency count is examined and kept in the memory only the first n highest frequency data itemsets information. Rest of the itemset information is deleted to have the memory space for any new itemsets coming in the next buckets

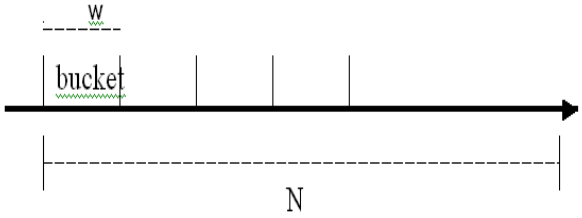

Figure1: Window model with fixed count

Next bucket is scanned and dataitems are retrieved, if retrieved dataitems is already in frequency data structure then its frequency is incremented, otherwise new entry in the data structure is inserted. Whenever there is an entry in the frequency data structure, there is a hash table generated to store the related data items in the same transaction [11][13]. So each top frequent itemsets and its related items are available in the memory.

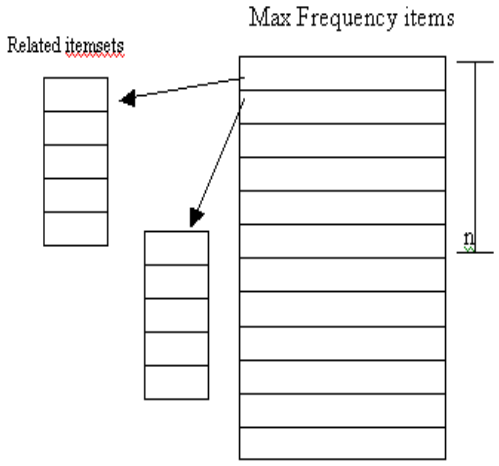

Figure 2. Top n frequent itemsets in hash table

Hence the association rules for maximal frequent itemsets can be generated dynamically.

B. Association Rules Base on Item sets Found:

In generating the association rules, subset rules also be viewed. For example X➔I, Y➔I and Y is subset of X then the rule X➔I is only enough. No need to consider Y➔I.

Since the data streams are dynamic and this system handles current high frequent itemsets, current association rules are generated. On request the current association rules can be generated.

Since data streams are fast changing, snapshots of association rules of some time intervals can also be stored in hard disk. To generate the overall frequent patterns, the newly generated frequent itemsets are added with previous overall frequent patterns counts. In the same way, if an itemsets is frequent in newly generated itemsets but not in previous overall frequent patterns then add the new frequent pattern to the overall frequent patterns.

This approach reduces the data to be analyzed so it is working limited memory, the analysis done on limited dataset so speed of the association rule extraction is increased and reduces CPU resources [12].

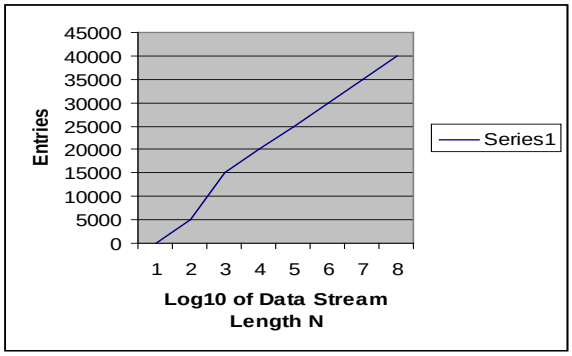

Figure 3: Memory Consumption of Data Stream

Figure 3 shows the considerable amount of space required for the given steams and evaluate the basis of every possible stream to manipulate for the further process of frequent itemset generation. The cumulative estimation of the streams is gradually determined by the overflowing of available data and yields the utmost estimated value to the process calculation. This process of estimation would brings

the core value to the result and improve the sophisticated ways of analysis.

IV. CONCLUSIONS

Our proposed strategies have enhanced the capabilities of consuming minimal CPU cycles and utilization of limited memory usage because we have not considering items of all frequency, instead we concentrated more on maximal frequency. Appropriate mechanism has been devised to give the suitable consequence for the batch-processing transactions and supply the moderate outflow for processing. The adaptive nature of maximal frequent itemsets entry is to mine the stream data and yield the total reduction of qualitative analysis for further investigation.

V. REFERENCES

[1]. Mohamed Medhat Gaber, Arkady Zaslavsky and Shonali Krishnaswamy, "Mining Data Streams: A Review", In proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, Vol. 34, No. 2, pp.18-26, June 2005.

[2]. Ho Jin Woo and Won Suk Lee, "estMax: Tracing Maximal Frequent Item Sets Instantly over Online Transactional Data Streams", TKDE, Vol. 21, No. 10, pp.1418-1431, October 2009.

[3]. James Cheng, Yiping Ke, Wilfred Ng: "A survey on algorithms for mining frequent itemsets over data streams", Knowledge Information System, Vol.16, No.1, pp.1-27, 2008.

[4]. Wang Jiinlong, Xu Conglfu, Cben Weidong, Pan Yunhe, "Survey of the Study on Frequent Pattern Mining in Data Streams", Vol. 6, pp. 5917-5922, 2004.

[5]. Nan Jiang and Le Gruenwald The University of Oklahoma, School of Computer Science, Norman, USA, "Research Issues in Data Stream Association Rule Mining", Vol. 35, No. 1, pp.14-19,2006.

[6]. G. S. Manku and R. Motwani, "Approximate frequency counts over data streams", In Proceedings of the 28th International Conference on Very Large Data Bases, Hong Kong, China,Vol.3,pp.346- 357,August, 2002.

[7]. Y. Chi, H. Wang, P. S. Yu and R. R. Muntz. Moment, "Maintaining Closed Frequent Itemsets over a Stream Sliding Window", In Proc. of ICDM, Vol. 3, pp. 1647 – 1651, 2004.

[8]. Rakesh Agrawal and Ramakrishnan Srikant, "Fast Algorithms for Mining Association Rules", Proceedings of the 20th VLDB Conference, pp.487-499, 1994.

[9]. J.Han, J.Pei, and Y.Yin, "Mining frequent patterns without candidate generation", In SIGMOD, Vol. 29, No.2 , pp.1-12, 2000.

[10]. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A Framework for Projected Clustering of High Dimensional Data Streams", Proc. 2004 Int. Conf. on Very Large Data Bases, Toronto, Canada, pp.81-92, 2004.

[11]. B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, "Models and issues in data stream systems", In Proceedings of PODS, pp. 1-16, 2002.

[12]. Y. Chi, H. Wang, P. S. Yu and R. R. Muntz. Catch the Moment, "Maintaining Closed Frequent Itemsets over a Data Stream Sliding Window", In KAIS, Vol.10, No. 3, pp. 265-294, 2006.

[13]. A. C. Gilbert, Y. Kotidis, S. Muthukrishnan, M. Strauss, "One-Pass Wavelet Decompositions of Data Streams", TKDE ,Vol. 15, No.3, pp.541-554, 2003.

[14]. B. Babcock, M. Datar, R. Motwani, L. O'Callaghan, "Maintaining Variance and k-Medians over Data Stream Windows", Proceedings of the 22nd Symposium on Principles of Database Systems, pp.234-243, 2003.

