ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2013

An Efficient Algorithm for Mining Frequent Items in Data Streams

Dr. S. Vijayarani[1], Ms. P. Sathya[2]

Assistant Professor, Dept. of CSE, Bharathiar university, Tamil Nadu, India[1]

Research Scholar, Dept. of CSE, Bharathiar University, Tamil Nadu, India[2]

ABSTRACT: Data stream is a continuous, real time, ordered sequence of items. In data stream, data arrives endlessly and the volume of data can be potentially infinite. In recent years advances in hardware and software technologies have resulted in automated storage of data from a variety of process. Data mining techniques are applied in data streams to find out the significant knowledge. The term data mining refers, to find relevant and useful information from large database. Some of the important techniques in data mining are association rule, classification, clustering, frequent episodes, and deviation detection. Frequent pattern mining is used to find important frequent patterns from the large dataset. Click stream analysis, market basket analysis, web link enquiry, genome study, network monitoring and medicine designing are some of the important areas where frequent pattern mining is used. Most commonly used frequent pattern mining algorithms are Apriori, partition algorithm, pincer- search algorithm, fp-growth algorithm, dynamic item set counting algorithm and so on. In this research paper Éclat and Rapid Association Rule mining algorithm are used for finding the frequent item sets in data streams. The experimental results show that the performance of RARM algorithm is better than Éclat.

Keywords: Association rules mining, Data mining, Data streams, Éclat algorithm, frequent pattern mining, RARM algorithm

I. INTRODUCTION

In data streams the data elements arrive incessantly and infinitely [2]. There is no limit on the end points in data stream data. The system has no control over the order in which the data elements arrive. The repeated scanning method is impossible in data streams. Due to the continuous flow of data the user should select the single pass algorithm to find out the frequent item sets [4]. In addition to that the algorithm does not drop any events in that stream data. The data streams unremitting data has lots of problems to store, compute as well as communication capabilities in computing system. Call records, web page visits, sensor reading are some of the examples of tuples in data streams. The hasty growth of uninterrupted data has many challenges to store, computation and communication capabilities in computing system.

In data stream data enters at high speed and continuous way. It is not possible to store them in a data warehouse. To identify a bit that is some portion of information in the database and remove the unnecessary information from the large data base. Important data mining techniques such as clustering, classification, frequent item mining, etc. [1] [12] are used for extracting the hidden knowledge from the data streams. In this paper, we have concentrated on mining frequent items from data streams. In association rule mining, association rules are discovered only when the rules whose support and confidence is greater than or equal to the minimum support and confidence. There are two important steps in association rule mining. The first step is to find the frequent items from the data set. During the second step, the association rules are generated from the frequent items.

This paper will focus on the following sections. In Section 2, we present an overview of the frequent item set mining in data streams and its related works. Section 3 discusses how Éclat algorithm is used for finding the frequent items in data streams. Section 4 gives the rapid association rule mining algorithm (RARM). Section 5 discusses the performance evaluation of the two algorithms. Section 6 discusses the conclusion and future work.

II. FREQUENT ITEM SET MINING IN DATA STEAMS

In a transactional database, the items which occur repeatedly in transactions are called frequent items. Frequent item set mining is used to discover useful patterns in customer's transaction databases. This type of finding item set helps businesses to make important decisions, such as catalogue drawing, cross marketing and consumer shopping performance scrutiny. A customer's transaction database consists of set of transactions, where each transaction is an item set. The frequent item set mining is to find all frequent item set in a given transaction database. There are many kinds of frequent patterns. They are [11], [2], [15]

- Frequent item set
- Subsequent item set

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2013

- Sub structure item set

Frequent item set -> ex: set of items like milk, bread appears together in a transactional data is known as frequent item set. (Items frequently appears together)
Subsequence item set -> ex: PC, digital camera, memory card which means a person first buys a PC, then a digital camera and then a memory card. If these items are appearing together in a shopping database, it is a sequential pattern.
Sub structure items -> ex: graph, tree, and lattice. Different structural forms combined with item sets or subsequence is called as substructure.

A. Related Works

Syed Khairuzzaman Tanbeer et.al [14] proposed a prefix-tree structure called CPS-tree. The CPS tree uses a new technique called as dynamic tree restructuring technique to handle the stream data. The main disadvantage of this algorithm is every time a new item arrives, it reconstructs the tree. So it causes more memory space as well as time.

Pauray S.M. Tsai [13] proposed a new technique called the weighted sliding window (WSW) algorithm. This algorithm calculates the weight of each transaction in each window. The candidate item set generation may take more time and memory. For candidate generation, an apriori algorithm is used.

Hue-Fu Li et.al [7] proposed an efficient bit-sequence based algorithm called MFI-Trans SW (Mining Frequent Item sets with in a Transaction Sensitive Sliding window). MFI algorithm worked on three phases. If the window size is increased, the memory usage of MFI-TransSW is also increased. If the window size increases, the processing time of phase 1 and phase 2 of MFI-TimeSW is also increased.

Yo Unghee Kim et.al [17] proposed an efficient algorithm WSFI mine (Weighted Support Frequent Item sets mining) with normalized weight over data stream. This WSFI-mine algorithm can mine all frequent item sets in one scan from the database.

Chowdhury Farhan Ahmed et. al [6] they recommended a novel algorithm for sliding window based high utility pattern mining over data stream called as HUPMS (High Utility Pattern Mining in Stream data). This algorithm is only suitable for interactive mining

Jing Guo,Peng Zhang et. al [9] discussed how to mine frequent patterns across multiple data streams. In this paper they selected real time news paper data for analysis. In multiple streams it is important to discover collaborative frequent patterns and comparative frequent patterns.

Anushree Gowtham Ringe et. al [3], This work was mainly focused on, how to prevent the misuse of sensitive data, in a stream. In this paper they proposed a novel technique for preserving the privacy of data stream.

III. ÉCLAT ALGORITHM IN DATA STREAMS

Equivalence Class Clustering and bottom up Lattice Traversal is an acronym for ECLAT algorithm [2] [5]. The name implies, that the algorithm uses bottom up searching method to find out the frequent item set. This algorithm is also used to perform item set mining. It uses tid set intersection to compute the support of a candidate item set. Compared with other algorithms like Apriori, FP- growth, partition algorithm, the ECLAT algorithm does not required candidate generation phase and pruning phase.

Algorithm 1: Éclat algorithm

Procedure: intersect TID sets (T, σ)
Input: T is available as vertical data base
Output: list of frequent item sets
1. Initialize: P= {<$1_{j,}$ t (1_j)> for all 1itemset in T
2. For each item set X with < X, t(X)> in P do
   $P_X \leftarrow \theta$
3. For each item set < Y, t(Y)> in P such that Y is lexicographically > X do
4. $N_{xy} = X \cup Y$
5. T ($N_{xy)}$ = t(X) $\cup$ t(Y)
6. If support of Nxy >= σ
7. Then $P_X \leftarrow P_X \leftarrow$ {<$N_{xy,}$ t ($N_{xy)}$>}
8. End
9. End
10. Intersect TID sets ($P_{X,}$ σ)
End

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2013

Éclat algorithm first put some id value to all the item set in a database. It attempts to improve the support counting step by indexing the data base. This algorithm does not create candidate generation process directly. The support of the candidate item set can be computed to intersect the tid list of suitably chosen subsets. Éclat counts the supports of all item sets much more efficiently using the intersection of tid lists. Compared with other algorithms, the memory usage is much lower as at any depth.

IV. RARM ALGORITHM IN DATA STREAMS

Rapid Association Rule Mining algorithm is an abbreviation for RARM [2]. This algorithm uses tree structure to represent the set of transactions. This algorithm avoids candidate generation process. It uses a data structure called Support Ordered Trie Item set (SOTrieIT) to generate large 1 item sets and large 2 item sets. This algorithm scans the data base only once. This algorithm first scans the database and forms some tree structure.

Algorithm 2: RARM algorithm

1. Let Y be a set of SOTrieIT
2. For (k=1; k<=2; k++) do begin
3. Obtain all k-item sets of the transaction and store them in C_k
4. For each item set $X \in C_k$ do begin
5. Traverse Y to locate nodes along the path that represents X
6. If such a set of nodes exists in Y then
7. Increment the support count of the leaf node
8. Sort updated node among siblings according to its new support count in descending order
9. Else
10. Create a new set of nodes with support
Counts of 1 that represent a path to X
11. Insert them into Y according to their support
Counts in descending order from the left
End if
End for
End

V. PERFORMANCE EVALUATION

Experimental results of éclat algorithm and RARM algorithm are discussed in this section. It is implemented in Microsoft visual studio 2008 with SQL server 2000. For testing frequent pattern mining over transactional data, synthetic data streams data sets [10] [18] are used. The synthetic data used in this paper is Kosakshi from IBM data generator. This data set contains 88054 transactions and 46 attributes.

A. Number of Frequent Items:

There are five windows W1, W2, W3, W4 and W5 are used in this work. The performance of the Éclat and RARM algorithms are compared by the two factors namely execution time and the number of frequent items discovered in each window. Window sliding concept is used in this work. After finding the frequent items in W1 the next window W2 automatically slides. Different sizes of transactions 100,500 and 1000 are tested and their results are obtained.

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2013

TABLE I
NUMBER OF FREQUENT ITEMS

Windows	Number Of Transaction	Éclat Algorithm	RARM Algorithm
W1	100	3.94	2.12
	500	7.32	5.55
	1000	14.34	12.19
W2	100	3.25	1.98
	500	6.99	5.09
	1000	13.98	11.97
W3	100	3.34	1.24
	500	7.55	5.10
	1000	14.29	12.00
W4	100	3.12	2.00
	500	7.29	5.12
	1000	13.74	11.30
W5	100	3.24	1.34
	500	7.56	5.36
	1000	14.93	11.87

Table 1 illustrates the number frequent items identified by Éclat and RARM algorithms. From the results, it can be shown the more number of frequent items are identified by RARM algorithm compared to Éclat algorithm.

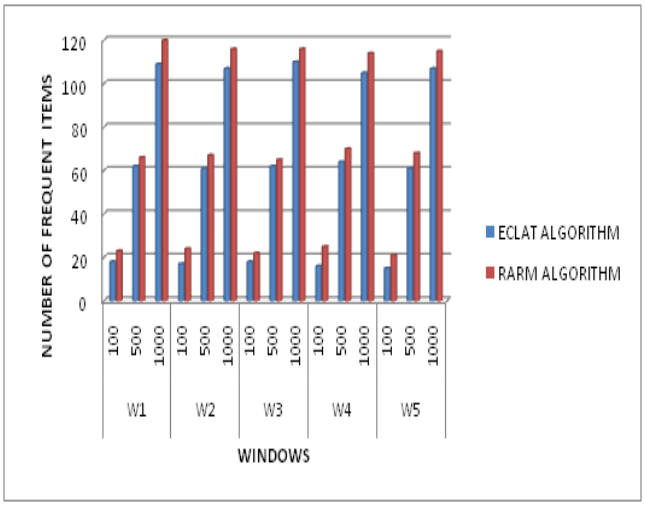

Fig 1- Number of frequent items

B. Time Efficiency:

Another performance factor used for measuring the efficiency of Éclat and RARM is execution time. Execution time is nothing but how much time required for identifying the frequent items in each window.

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2013

TABLE III
EXECUTION TIME

windows	Number Of Transaction	Éclat Algorithm	RARM Algorithm
W1	100	18	23
	500	62	66
	1000	109	120
W2	100	17	24
	500	61	67
	1000	107	116
W3	100	18	22
	500	62	65
	1000	110	116
W4	100	16	25
	500	64	70
	1000	105	114
W5	100	15	21
	500	61	68
	1000	107	115

Table II shows the execution time required for Éclat and RARM algorithms. From this we come to know that Éclat algorithm needs more time for finding the frequent items compared to RARM algorithm.

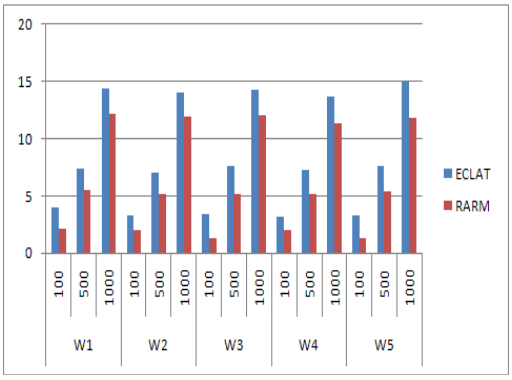

Fig 2- Execution Time

VI. CONCLUSION AND FUTURE WORK

In this research work we have analyzed the performance of two algorithms namely Éclat and Rapid Association Rule Mining for mining frequent items in data streams. Execution time and number of frequent items are the main performance factors of this work. From the experimental results, we observed that, the RARM algorithm required minimum execution time and it also identified more number of frequent items compared to Éclat algorithm. Mining techniques will then be very significant in order to conduct advanced analysis, such as determining trends and finding interesting patterns, on streaming data. Data streams data are from various sources, and it has much confidential information also, so we can protect these confidential data by applying a privacy technique in future [8]. It is often a challenge to perform privacy for continuously arriving data.

REFERENCES

[1]. Arun k Pujari "Data mining techniques second edition" ISBN 978 81 7371 672 0 universities press 2010
[2]. Aggarwal, C. In C. Aggarwal (Ed.), "Data streams: Models and algorithms". Springer, 2007.
[3]. Anushree Gowtham Ringe, Deeksha Sood and Turga Toshniwa"Compression and privacy preservation of data streams using moments",Information journal of machine learning and computing. 2011
[4]. Albert Bifet, Geoff Holmes, Richard Kirkby and Bernhard Pfahringer" Data Stream Mining A Practical Approach" May 2011

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2013

[5]. Borgelt C "efficient implementations of apriori and éclat". In: 2nd workshop of frequent item set mining implementations. 2004.
[6]. Chowdary Farha ahmed, Byeong-Soo Jeong, "Efficient mining of high utility patterns over data streams with a sliding window model". Springerlink.com, 2011
[7]. Hue-Fu Li, Suh-Li "Mining frequent item sets over data streams using efficient window sliding technique", Elsevier publication. 2009.
[8]. Han J, Cheng H, Xin D, Yan X "Frequent pattern mining: current status and future directions". 2007
[9]. Jing Guo, Peng Zhang, Jianlong Tan and li Guo "Mining frequent patterns across multiple data streams", 2011.
[10]. J. Han and M. kamber, "Data Mining: Concepts and techniques," Series Editor Morgan Kaufmann Publishers, ISDN 1-55860-489-8. 2000
[11]. Koh, J.-L., and Shieh, S.-F, "An efficient approach for maintaining association rules based on adjusting FP-tree structures". In Lee Y-J, Li J, Whang K-Y, Lee D (eds) Proc. of DASFAA 2004. Springer-Verlag, Berlin Heidelberg New York, 417–424
[12]. Margaret H. Dunham"Data Mining: Introductory and Advanced Topics".
[13]. Pauray S.M.Tsai, "Mining frequent item sets in data streams using the weighted sliding window model", Elsevier publication 2009.
[14]. Syed Khairuzzaman Tabeer, Chowdary Farha ahmed, Byeong-Soo Jeong, Young Koo Lee"Efficient frequent pattern mining over data streams" 2008.
[15]. Tanbeer, S. K., Ahmed, C. F., Jeong, B.-S., and Lee, Y.-K. 2008. "CP-tree: a tree structure for single-pass frequent pattern mining"S. In Proc. of PAKDD, Lect Notes Artif Int, 1022-1027.
[16]. Yo unghee Kim, Won Young Kim and Ungmo Kim "Mining frequent item sets with normalized weight in continuous data streams". Journal of information processing systems. 2010.
[17]. www.borgelt.net/slides/fpm.pdf

BIOGRAPHY

Dr. S.Vijayarani has completed MCA, M.Phil and PhD in Computer Science. She is working as Assistant Professor in the School of Computer Science and Engineering, Bharathiar University, Coimbatore. Her fields of research interest are data mining, privacy and security issues. She has published papers in the international journals and presented research papers in international and national conferences.

Ms. P.Sathya has completed M.Sc in Software Systems. She is currently pursuing her M.Phil in Computer Science in the School of Computer Science and Engineering, Bharathiar University, Coimbatore. Her fields of interest are Data Streams in data mining and privacy preserving in Data mining.