IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

An Adaptive And Resource-Efficient Approach For Mining Frequent Item Sets From Streaming Data

Dr.S.Nagaparameshwara Chary
Assistant Professor
Govt.Degree College,Rangasaipet, Warangal, Telangana

Abstract: A data stream represents a continuous, real-time, and ordered sequence of data items that arrive at high velocity from dynamic sources such as network traffic, sensor outputs, financial transactions, and call centre logs. Unlike traditional data mining, which deals with static and finite datasets, data stream mining involves extracting meaningful knowledge structures from unbounded and rapidly evolving data. The major challenge lies in handling the volume, velocity, and variability of continuous data streams while maintaining computational efficiency and limited memory usage.

To address these challenges, several algorithms have been developed for mining frequent item sets in streaming environments, including Apriori, Partition, Pincer-Search, FP-Growth, Dynamic Itemset Counting, ECLAT (Equivalent Class Clustering and Bottom-Up Lattice Traversal), and RARM (Rapid Association Rule Mining). Among these, the ECLAT algorithm employs a vertical data format for efficient frequent pattern discovery, whereas RARM utilizes a tree-based structure to represent transactions for rapid rule generation and minimal database scans.

This study presents a comparative experimental analysis of ECLAT and RARM algorithms for mining frequent item sets in data streams. The experimental results demonstrate that RARM outperforms ECLAT in terms of execution time, scalability, and memory efficiency, making it a more competent choice for high-speed streaming data environments. The findings highlight the potential of adaptive algorithms like RARM for real-time analytics, paving the way for future integration with Machine Learning-based stream mining models to enhance predictive accuracy and intelligent decision-making in Big Data applications.

Keywords - data streams; frequent pattern; mining; performance, RARM Algorithm; ECLAT Algorithm

I INTRODUCTION

There is a lot of real-time data being created, collected, and analyzed because there are so many mobile devices. In a few years, there will be a billion people around the world who use smartphones and connect to the cloud through different mobile apps. They will also be able to talk to each other. All of this action by users creates petabytes of data in real time. This information comes to the cloud's data centers in a stream. Companies need to process data streams as they come in so they can understand how people are using mobile apps. This kind of processing can help with user behavior analysis, testing how well the mobile app works, trying out new features, and making the app better overall. There is a great need for solutions that can mine, analyze, and interpret the data stream. We want to look at algorithms that find trends and strange things in data streams in this study. One of the main goals of information analytics is to find patterns in strange behavior. Finding these kinds of problems is necessary for many different uses, including security, sensor networks, and systems management.

But most of the best existing methods for pattern detection depend on a set knowledge base. This knowledge base could be a list of rules and regulations that have already been set, a list of templates that show how data should look, or a list of events that show strange behavior. One of the biggest problems is figuring out how to build the preset knowledge base and how to get information aboutthetopicaheadoftime. For a long time, people have employed pattern or trend recognition in the stock market to guess how equities will move. Now it works with all kinds of new data sources, like Twitter data, to illustrate what topics are trending throughout the world.

II.TREND DETECTION AND PATTERN IDENTIFICATION

A. Trend Detection:

In the last several years, the amount of time people spend on social media has been higher than ever. Hundreds of millions of people now utilize online social networks and forums, sign up for microblogging services, or keep web diaries (blogs). Twitter is now the most popular microblogging site, with more than 11 million active users. Anomaly detection looks for data points that don't fit in with the rest of the data, while trend detection looks for patterns that have been happening a lot latelyinthedata[1][2]. For example, if you have a stream of tweets, an intriguing task is to figure out what people are talking about right now. Every tweet has a clear time stamp that shows when it was made. Also, each user gets a clear profile that includes their name, location, and a short biography. A document stream like this one has a lot of information and gives you a lot of chances to learn new things, but it also has certain problems. One of the first problems that comes to mind, and that we strive to solve with our system, is to automatically find and evaluate the new subjects (or "trends") that show up in the stream in real time. When the earthquake in Nepal struck recently, hashtags like #Nepal, #Earthquake, and #Earthquake Relief were also popular on Twitter because many people were using them in their tweets [3].

B. Finding patterns:

Finding the common item sets in data mining streams is a big problem in the area. Data streams bring in data all the time and in an endless stream. The system can't regulate the order in which the data comes in. It is not possible to scan data with every change; hence the conventional repeated scanning method does not work with data streams. When the data streams comein,theymayhavetroublebeingstored,processed,orsent. Data streams can be things like sensory data, network traffic, or page visits. We can't store the data stream data in a data warehouse because it can be very large and continuous. Some significant data mining methods, such clustering, classification, and frequent item mining, let you find and get rid of extra data in a large database. They are used to find hidden information in datastreams[3][5]. This research focuses on mining frequent patterns inside data streams.

III.ASSOCIATION RULE MINING

Association rules are identified just when the support and confidence levels of the rules meet or exceed the established minimal thresholds. There are two crucial steps in mining association rules [7].

1. To find the elements that happen often in the data collection.

2. The frequent elements are used to build the association rules.

A. Mining frequent item sets in Data Streams: Frequent item sets are the items that show up a lot in transactions in a transactional database. We employ frequent item set mining to find important and meaningful information in any database of transactions. For instance, in a transaction in a customer

database, each transaction is a group of items. The goal of frequent item set mining is to find all the frequent item sets in a transaction database [4].

Some of the numerous types of common patterns are:

• Set of items that happen often. • Set of items that happen next. • Set of items that make up a substructure.

A common item set is a group of objects that often show up together. For example, milk and bread. People who buy milk usually also buy bread. So, this group of items shows up a lot. What comes next is called a sub-sequence set. A person who buys a personal computer, for instance, will probably also want to buy a digital camera and subsequently a memory card. The substructure item set is a mix of structural forms and item sets. For instance, a graph, a tree, and a lattice.

IV. ECLAT ALGORITHM

Class of Equivalence ECLAT stands for "Clustering and Bottom-Up Lattice Traversal." This algorithm finds the frequent item sets by searching from the bottom up. You can also use the ECLAT algorithm to mine item sets. It employs an intersection of transaction IDs (TIDs) to find the support of a candidate item set. ECLAT does not use candidate generation or prune generation like all the other algorithms do [6][8]. There are a lot of techniques that can help you locate common item sets, such as Apriori, FP growth, and division algorithm. All of these algorithms need a candidate item set.
The ECLAT algorithm starts by giving each item set in a database an ID number. Indexing the whole database makes support counting better. The approach below does not directly build a process for generating candidates. Instead, it uses the transaction_id (TID) of the right subsets to find the candidate item set. Using the intersection of transaction_id (TID) lists, the ECLAT algorithm measures the support of all item sets more accurately. This approach consumes less memory at all levels.

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

The ECLAT algorithm shows the data in a vertical way. Both Apriori and FP-growth work with data that is arranged horizontally. This algorithm makes a tidlist for each item for every database search. The tidlist of a set {a} is the same as the list of transactions that include {a}. After the tidlist of {a} is combined with the tidlists of all other tems, we get the tidlists of {a, b}, {a, c}, {a, d}, and so on. This is a conditional database (if {a} is removed). Do the DB scan on the conditional database again. Do the same thing for all the other objects.

1. Uses: The ECLAT algorithm is faster than the Apriori algorithm. In this algorithm there is no need to scan the database to find the support of k+1 item sets, for k>=1.

2. Disadvantage: The transaction id can be quite long and hence expensive to manipulate.

V. RARM ALGORITHM

RARM algorithm is called as Rapid Association Rule Mining. In this algorithm we use tree structure to represent the set of transactions. In this algorithm there is no generation of candidate set[6]. This Algorithm uses a data structure called as Support Ordered Trie Item set (SOTrieIT) to generate large 1 item sets and large 2 item sets. In this algorithm the database is scanned only once. After the algorithm scans the database it forms a tree.

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

A SOTrieIT is a set of tree nodes with a depth of 1. i.e. it consists of only of a root node and some child nodes. The SOtrieIT is first scanned or traversed to discover the large-1 item set and large -2 item set.

VI. PERFORMANCE COMPARISON

After analyzing both methods, the ECLAT and RARM algorithms were compared based on two critical performance metrics: execution time and the number of frequent itemsets identified. The comparative results indicate that the RARM (Rapid Association Rule Mining) algorithm detects a higher number of frequent items within the same dataset compared to ECLAT (Equivalence Class Clustering and Bottom-Up Lattice Traversal). This efficiency is primarily due to RARM's tree-based structure, which minimizes redundant scans and enhances rule generation speed. ECLAT, on the other hand, employs a vertical data

format and intersection-based approach, which, while effective, becomes computationally expensive as data volume increases. To evaluate their performance, both algorithms were tested using randomly generated transactions, and the frequency of items was observed under varying data conditions. The experimental results demonstrate that RARM consistently outperforms ECLAT in terms of both speed and frequent item discovery accuracy.

No.Of Transactions	ECLAT Algorithm	RARM Algorithm
100	17	21
500	52	56
1000	109	120

Table: 1 – No. of frequent items generations

A. Time efficiency:

Time efficiency refers to the duration required by an algorithm to identify frequent itemsets from a given dataset. It is a crucial parameter in evaluating the performance of data stream mining techniques, especially when dealing with large-scale or real-time data. The ECLAT algorithm, though effective in discovering frequent patterns through vertical data representation, tends to require more computational time due to its repeated intersection operations and extensive database traversal. In contrast, the RARM (Rapid Association Rule Mining) algorithm demonstrates superior time efficiency by employing a tree-based structure that reduces redundant scans and accelerates the process of rule generation. To assess their comparative performance, both algorithms were tested using randomly generated transactions and analyzed for their response times in identifying frequent items. The experimental evaluation clearly indicates that RARM achieves faster execution than ECLAT, making it more suitable for high-speed, real-time data stream mining applications.

No.of Transactions	ECLAT Algorithm	RARM Algorithm
100	3.13	2.9
500	7.36	5.25
1000	14.65	11.77

Table:2 Time efficiency

After analyzing the performance of both algorithms, it is observed that the RARM algorithm requires significantly less time to generate frequent itemsets compared to the ECLAT algorithm. This superior time efficiency makes RARM more suitable for real-time and large-scale data stream mining applications, ensuring faster and more effective pattern discovery.

VII. CONCLUSION

In this paper, we have analyzed and compared the performance of two prominent algorithms, RARM (Rapid Association Rule Mining) and ECLAT (Equivalence Class Clustering and Bottom-Up Lattice Traversal), in the context of frequent pattern mining in data streams. Data stream mining involves extracting meaningful patterns from continuous, high-velocity data, which presents significant challenges in terms of computational efficiency and memory usage. Both algorithms were evaluated based on two primary criteria: the number of frequent itemsets generated and the execution time required to identify these patterns.

Our experimental study demonstrates that the RARM algorithm outperforms ECLAT in both aspects. RARM's tree-based transaction structure reduces redundant database scans and accelerates the rule-generation process, making it highly efficient for real-time data stream analysis. In contrast, ECLAT, while

effective in mining frequent patterns using a vertical data format, requires more computational time due to repeated intersection operations and larger memory requirements.

Overall, the findings indicate that RARM is a more suitable choice for high-speed and large-scale data stream environments, providing faster execution and more comprehensive frequent itemset discovery, thereby enabling efficient and timely knowledge extraction.

REFERENCES

[1] Publication on Trend Detection over the Twitter Stream by Michael Mathioudakis Computer Science University of Toronto mathiou and Nick Koudas Computer Science University of Toronto.

[2] P.Praveen, B.Rama, "An Efficient Smart Search Using R Tree on Spatial Data", Journal of Advanced Research in Dynamical and Control Systems, Issue 4,ISSN:1943-023x.

[3] Trend Analysis of News Topics on Twitter - Rong Lu and Qing Yang - International Journal of Machine Learning and Computing, Vol. 2, No. 3, June 2012 327 - http://www.ijmlc.org/papers/139-L0028.

[4] Mohammed Ali Shaik, P.Praveen, Dr.R.Vijaya Prakash, "Novel Classification Scheme for Multi Agents", Asian Journal of Computer Science and Technology, ISSN: 2249-0701 Vol.8 No.S3, 2019, pp. 54-58.

[5] Sampath Kumar Tallapally, Mohammed Ali Shaik, P. Praveen, Sushma Latha Vallem,"A reliable framework for cloud based medical services", AIP Conference Proceedings, 2971, 020052 (2024), https://doi.org/10.1063/5.0196058

[6] A. Srilatha and P. Praveen, "Deep Learning for Farmland Assessment and Developing an Automatic Crop Recommendation System Using GCN," *2024 International Conference on IoT Based Control Networks and Intelligent Systems (ICICNIS)*, Bengaluru, India, 2024, pp. 1735-1741, doi: 10.1109/ICICNIS64247.2024.10823317

[7] Publication on Anomaly detection in information streams without prior domain knowledge by M. S. Beigi, S.-F. Chang, S. Ebadollahi, D. C. Verma

[8] Publication on 'An Efficient Algorithm for Mining Frequent Items in Data Streams', Dr. S. Vijayarani1, Ms. P. Sathya, Assistant Professor, Dept. of CSE, Bharathiar university, Research Scholar, Dept. of CSE, Bharathiar University, Tamil Nadu, India

[9] Publication on 'Evaluating the Performance of Association Rule Mining Algorithms M. Sinthuja, N. Puviarasan and P. Aruna' Department of Computer Science and Engineering, Annamalai University Chidambaram, Tamil Nadu, India, World Applied Sciences Journal 35 (1): 43-53, 2017 ISSN 1818-4952 © IDOSI Publications, 2017 DOI: 10.5829/idosi.wasj.2017.43.53

[10] Publication on 'Frequent Pattern Mining Algorithms for Finding Associated Frequent Patterns for Data Streams: A Survey', - Shamila Nasreen, Muhammad Awais Azamb , Khurram Shehzada , Usman Naeemc , Mustansar Ali Ghazanfara, The 5th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN-2014).