

# Exploring Trends in Real-Time Data Streams through Frequent Itemset Mining in Big Data Analytics

Nalla Chandana

Department of Computer Science Sri Venkateswara University, Tirupati

**Abstract**— In today's digital landscape, the volume of data from social media or business transactions surpasses traditional static databases and continues to expand rapidly. Our study introduces a frequent itemset mining approach using sliding windows to uncover trends in continuous data streams. Leveraging Big Data technologies, specifically the Association rule and apriori algorithm, we distribute computations across multiple clusters to enhance algorithm efficiency. Experimental results demonstrate the effectiveness and scalability of our method in handling vast amounts of streaming data.

## I. INTRODUCTION

In today's data-driven world, extracting insights through techniques like Data Mining is crucial. The choice of technique depends on both the type of data and the desired information. Trend analysis, especially in areas like economic fluctuations and social media trends, has become essential for staying updated. However, the continuous change and vast volume of data pose challenges for traditional extraction methods. Big Data emerges as a solution, enabling distributed computation across clusters. Frameworks like Hadoop and Spark, based on the Map-Reduce paradigm, facilitate large-scale data storage and processing. Despite the growth in libraries and extensions for Data Mining and Machine Learning, adapting algorithms to distributed environments remains complex. Frequent itemset mining, traditionally done using sequential approaches like Apriori, ECLAT, or FP-growth, faces limitations in handling continuous data streams where time sensitivity is crucial. Existing stream mining algorithms struggle with processing data from sources like social media due to memory overflow issues. Distributed algorithms in Big Data frameworks offer a solution by mapping data to different computation clusters and providing redundancy mechanisms to prevent data loss. However, there's a gap in Big Data implementations for frequent itemset stream mining. To address this, our paper introduces a new approach using Spark Streaming, enhancing the FIMOTS algorithm for frequent itemset mining over time-sensitive streams. Unlike a straightforward extension, our proposed algorithm tackles the complexities of distributed processing, overcoming limitations of sequential approaches and scaling well with large data streams.

## II. LITERATURE REVIEW

### 2.1 Efficiently Mining Maximal Frequent Item sets

*Karam Gouda, Mohammed J. Zaki*

We present GenMax, a backtracking search-based algorithm for mining maximal frequent itemsets. GenMax uses a number of optimizations to prune the search space. It uses a novel technique called progressive focusing to perform maximality checking, and diffused propagation to perform fast frequency computation. Systematic experimental comparison with previous work indicates that different methods have varying strengths and weaknesses based on dataset characteristics. We found GenMax to be a highly efficient method to mine the exact set of maximal patterns.

### 2.2 Discovering Frequent Closed Itemset for Association Rules

*Nicolas Pasquier Yves Bastiderafik Taouillotfi Lakkhal*

In this paper, we address the problem of finding frequent itemsets in a database. Using the closed itemset lattice framework, we show that this problem can be reduced to the problem of finding frequent closed itemsets. Based on this statement, we can construct efficient data mining algorithms by limiting the search space to the closed itemset lattice rather than the subset lattice. Moreover, we show that the set of all frequent closed itemsets suffices to determine a reduced set of association rules, thus addressing another important data mining problem: limiting the number of rules produced without information loss. We propose a new algorithm, called A-Close, using a closure mechanism to find frequent closed itemsets. We realized experiments to compare our approach to the commonly used frequent itemset search approach. Those experiments showed that our approach is very valuable for dense and/or correlated data that represent an important part of existing databases.

### 2.3 An Efficient Algorithm for Mining Frequent Closed item sets

*Jian Pei, Jiawei Han, And Runying Mao*

Association mining may often derive an undesirably large set of frequent itemsets and association rules. Recent studies have proposed an interesting alternative: mining frequent closed itemsets and their corresponding rules, which has the same power as association mining but substantially reduces the number of rules to be presented. In this paper, we propose an efficient algorithm, CLOSET, for mining closed itemsets, with the development of three techniques: applying a compressed, frequent pattern tree FP-tree structure for mining closed itemsets without candidate generation, developing a single pre x path compression technique to identify frequent closed itemsets quickly, and exploring a partition-based projection mechanism for scalable mining in large databases. Our performance study shows that CLOSET is efficient and scalable over large databases, and is faster than the previously proposed methods.

### 2.4 Hierarchical Document Clustering Using Frequent item sets

*Benjamin C.M. Fung, Ke Wang and Martin Ester*

A major challenge in document clustering is the extremely high dimensionality. For example, the vocabulary for a document set can easily be thousands of words. On the other hand, each document often contains a small fraction of words in the vocabulary. These features require special handlings. Another requirement is hierarchical clustering where clustered documents can be browsed according to the increasing specificity of topics. In this paper, we propose to use the notion of frequent itemsets, which comes from association rule mining, for document clustering. The intuition of our clustering criterion is that each cluster is identified by some common words, called frequent itemsets, for the documents in the cluster. Frequent itemsets are also used to produce a hierarchical topic tree for clusters. By focusing on frequent items, the dimensionality of the document set is drastically reduced. We show that this method outperforms best existing methods in terms of both clustering accuracy and scalability.

### 2.5 Finding Recent Frequent Itemsets Adaptively Over Online Data Streams

*Joong Hyuk Chang, Won Suk Lee*

A data stream is a continuous, high-speed flow of data elements, where knowledge changes rapidly over time. Identifying recent changes in such streams is crucial for analysis. However, many mining algorithms overlook the distinction between recent and obsolete data, affecting the accuracy of results. This paper presents a data mining method that adaptively identifies recent frequent item sets in online data streams. By decaying old occurrences over time and employing optimization techniques, processing time and memory usage are minimized. Experimental analysis validates the effectiveness of the proposed method.

## III. PROPOSED WORK

1. The System implementing a distributed version of the FIMoTS algorithm is made, where this approach outperforms the rest of approaches using sliding windows.
2. The original FIMoTS algorithm is mainly iterative and performs recurrent updates over the tree structure, but in distributive platforms, it is advisable to keep the communication among clusters as low as possible.
3. However, the maintenance of the tree structure needs direct communication with all the clusters. making it as efficient as possible.

### Advantages

1. High performance and accuracy.
2. Data flow is passed well.
3. Time Consuming.

### 3.1 Dataset Collection

A dataset, typically presented in tabular format, comprises columns representing variables and rows corresponding to specific data points. Each value within the dataset, known as a datum, provides information on variables like height and weight. We've opted for a publicly available frequent itemset dataset with a manageable number of inputs and cases. The dataset's organization

facilitates medical professionals in relating traditional statistical methods to novel machine learning techniques. Moreover, its compact size ensures quick computational processing on most modern computers.

### 3.2 Pre-Processing

The sklearn.preprocessing package provides several common utility functions and transformer classes to change raw feature vectors into a representation that is more suitable for the downstream estimators.

### 3.3 Implement the Model

The implementation model represents how a system (application, service, interface, etc.) works. It is often described with system diagrams and pseudocode to be later translated into real code. It is shaped by technical, organizations, and business constraints. Here we use Association rule and Apriori algorithm .

Association rule mining is a procedure which is meant to find frequent patterns, correlations, associations, or causal structures from data sets found in various kinds of databases such as relational databases, transactional databases, and other forms of data repositories.

When we go grocery shopping, we often have a standard list of things to buy. Each shopper has a distinctive list, depending on one's needs and preferences. A housewife might buy healthy ingredients for a family dinner, while a bachelor might buy beer and chips. Understanding these buying patterns can help to increase sales in several ways. If there is a pair of items, X and Y, that are frequently bought together:

- Both X and Y can be placed on the same shelf, so that buyers of one item would be prompted to buy the other.
- Promotional discounts could be applied to just one out of the two items.
- Advertisements on X could be targeted at buyers who purchase Y.
- X and Y could be combined into a new product, such as having Y in flavours of X.

#### 3.3.1 Support:

This says how popular an itemset is, as measured by the proportion of transactions in which an itemset appears. In Table 1 below, the support of {apple} is 4 out of 8, or 50%. Itemsets can also contain multiple items. For instance, the support of {apple, beer, rice} is 2 out of 8, or 25%.

If you discover that sales of items beyond a certain proportion tend to have a significant impact on your profits, you might consider using that proportion as your *support threshold*. You may then identify itemsets with support values above this threshold as significant itemsets.

#### 3.3.2 Confidence:

This says how likely item Y is purchased when item X is purchased, expressed as  $\{X \rightarrow Y\}$ . This is measured by the proportion of transactions with item X, in which item Y also appears. In Table 1, the confidence of {apple  $\rightarrow$  beer} is 3 out of 4, or 75%.

One drawback of the confidence measure is that it might misrepresent the importance of an association. This is because it only accounts for how popular apples are, but not beers. If beers are also very popular in general, there will be a higher chance that a transaction containing apples will also contain beers, thus inflating the confidence measure. To account for the base popularity of both constituent items, we use a third measure called lift.

#### 3.3.3 Lift:

This says how likely item Y is purchased when item X is purchased, while controlling for how popular item Y is. In Table 1, the lift of {apple  $\rightarrow$  beer} is 1, which implies no association between items. A lift value greater than 1 means that item Y is *likely* to be bought if item X is bought, while a value less than 1 means that item Y is *unlikely* to be bought if item X is bought.

We use a dataset on grocery transactions from the *arules* R library. It contains actual transactions at a grocery outlet over 30 days. The network graph below shows associations between selected items. Larger circles imply higher support, while red circles imply higher lift: Associations between selected items. Visualized using the *arulesViz* R library.

Several purchase patterns can be observed. For example:

1. The most popular transaction was of pip and tropical fruits
2. Another popular transaction was of onions and other vegetables
3. If someone buys meat spreads, he is likely to have bought yogurt as well
4. Relatively many people buy sausage along with sliced cheese
5. If someone buys tea, he is likely to have bought fruit as well, possibly inspiring the production of fruit-flavored tea

Recall that one drawback of the confidence measure is that it tends to misrepresent the importance of an association. To demonstrate this, we go back to the main dataset to pick 3 association rules containing beer:

The {beer -> soda} rule has the highest confidence at 20%. However, both beer and soda appear frequently across all transactions (see Table 3), so their association could simply be a fluke. This is confirmed by the lift value of {beer -> soda}, which is 1, implying no association between beer and soda.

On the other hand, the {beer -> male cosmetics} rule has a low confidence, due to few purchases of male cosmetics in general. However, whenever someone does buy male cosmetics, he is very likely to buy beer as well, as inferred from a high lift value of 2.6. The converse is true for {beer -> berries}. With a lift value below 1, we may conclude that if someone buys berries, he would likely be averse to beer.

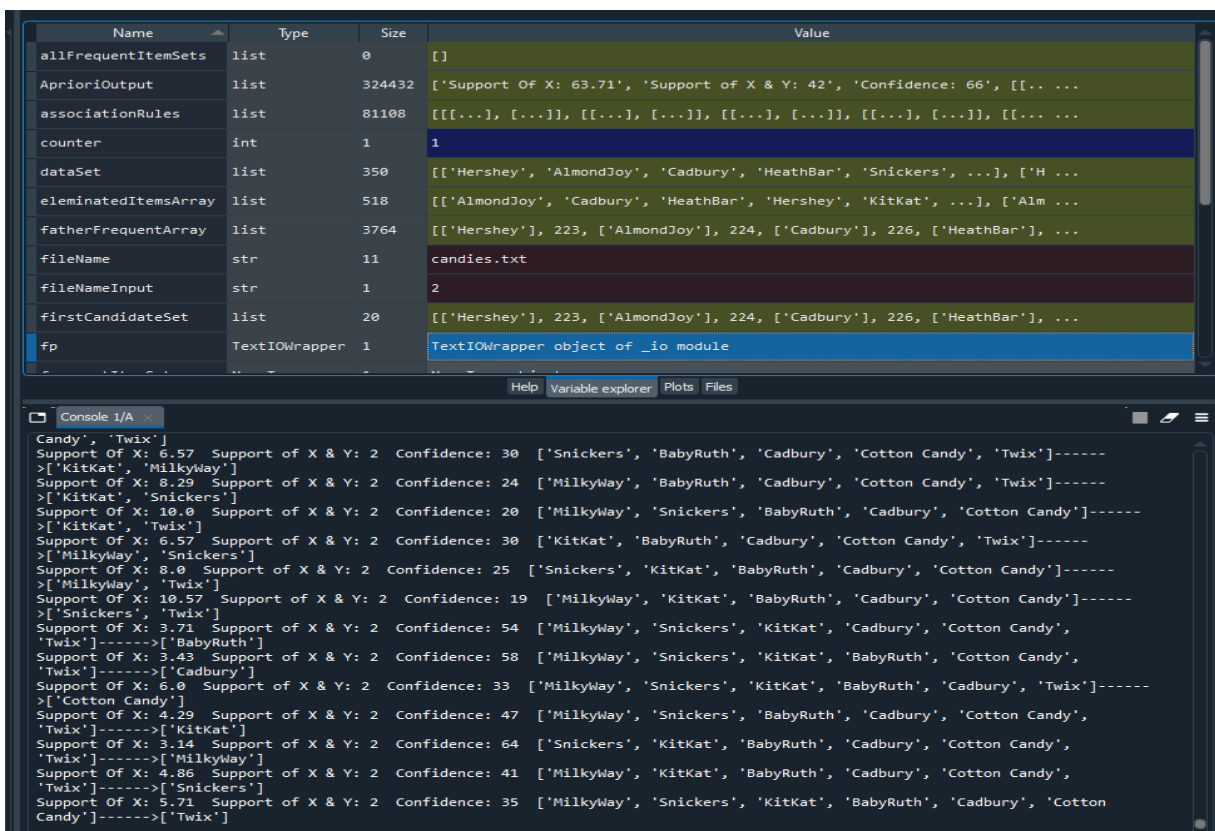
### 3.3.4 Evaluation

Model evaluation aims to estimate the generalization accuracy of a model on future (unseen/out-of-sample) data. In this model precision, support, accuracy and confusion matrices are used to evaluate the fake news

## IV. IMPLEMENTATION

Select the dataset:

1. Auto Mobile
2. Candies
3. Computer Accessories
4. Food
5. Mobile Accessories



## V. CONCLUSION AND FUTURE WORK

Our proposal aims to analyse trends in large data streams, particularly beneficial for dynamic environments like social media and business analysis. We've focused on frequent itemset mining to uncover significant trends, considering both massive data volumes and continuous streams. Our algorithm, leveraging MapReduce and the Association rule and Apriori algorithm, outperforms non-distributed versions and handles variable window lengths and changing support thresholds seamlessly. Future plans include real-time social media analysis and extending the algorithm to association rule mining for studying frequent item co-occurrences in data streams.

## REFERENCES

- [1] Gouda, Karam, and Mohammed Javeed Zaki. "Efficiently mining maximal frequent itemsets." In Proceedings 2001 IEEE International Conference on Data Mining, pp. 163-170. IEEE, 2001.
- [2] Pasquier, Nicolas, Yves Bastide, Rafik Taouil, and Lotfi Lakhal. "Discovering frequent closed itemsets for association rules." In International Conference on Database Theory, pp. 398-416. Springer, Berlin, Heidelberg, 1999.
- [3] Pei, Jian, Jiawei Han, and Runying Mao. "Closet: An efficient algorithm for mining frequent closed itemsets." In ACM SIGMOD workshop on research issues in data mining and knowledge discovery, vol. 4, no. 2, pp. 21-30. 2000.
- [4] Fung, Benjamin CM, Ke Wang, and Martin Ester. "Hierarchical document clustering using frequent itemsets." In Proceedings of the 2003 SIAM international conference on data mining, pp. 59-70. Society for Industrial and Applied Mathematics, 2003.
- [5] Chui, Chun-Kit, Ben Kao, and Edward Hung. "Mining frequent itemsets from uncertain data." In Pacific-Asia Conference on knowledge discovery and data mining, pp. 47-58. Springer, Berlin, Heidelberg, 2007.
- [6] Chang, Joong Hyuk, and Won Suk Lee. "Finding recent frequent itemsets adaptively over online data streams." In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 487-492. 2003.
- [7] Gouda, Karam, and Mohammed J. Zaki. "Genmax: An efficient algorithm for mining maximal frequent itemsets." *Data Mining and Knowledge Discovery* 11, no. 3 (2005): 223-242.
- [8] Grahne, Gösta, and Jianfei Zhu. "Efficiently using prefix-trees in mining frequent itemsets." In FIMI, vol. 90, p. 65. 2003.
- [9] Calders, Toon, and Bart Goethals. "Mining all non-derivable frequent itemsets." In European Conference on Principles of Data Mining and Knowledge Discovery, pp. 74-86. Springer, Berlin, Heidelberg, 2002.
- [10] Chi, Yun, Haixun Wang, Philip S. Yu, and Richard R. Muntz. "Moment: Maintaining closed frequent itemsets over a stream sliding window." In Fourth IEEE International Conference on Data Mining (ICDM'04), pp. 59-66. IEEE, 2004.
- [11] Agarwal, Ramesh C., Charu C. Aggarwal, and V. V. V. Prasad. "A tree projection algorithm for generation of frequent item sets." *Journal of parallel and Distributed Computing* 61, no. 3 (2001): 350-371.
- [12] Pei, Jian, Jiawei Han, and Laks VS Lakshmanan. "Mining frequent itemsets with convertible constraints." In Proceedings 17th International Conference on Data Engineering, pp. 433-442. IEEE, 2001.
- [13] Deng, Zhi-Hong, and Sheng-Long Lv. "Fast mining frequent itemsets using Nodsets." *Expert Systems with Applications* 41, no. 10 (2014): 4505-4512.
- [14] Moustakides, George V., and Vassilios S. Verykios. "A maxmin approach for hiding frequent itemsets." *Data & Knowledge Engineering* 65, no. 1 (2008): 75-89.
- [15] Ye, Yanbin, and Chia-Chu Chiang. "A parallel apriori algorithm for frequent itemsets mining." In Fourth International Conference on Software Engineering Research, Management and Applications (SERA'06), pp. 87-94. IEEE, 2006.