Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN: 2581 - 611X)
VOLUME:8, ISSUE:2, AUGUST 2025, PAPER NUMBER:38

Improvements on Apriori – Partitioning and Sampling

Shubhangi Kale[1],

[1]Assistant Professor
[1]School of Computer Engineering,
[1]MIT Academy of Engineering, Alandi(D.)
[1]Pune, Maharashtra, India.
[1]spkale@mitaoe.ac.in

Abstract: *Extraction of useful information from huge databases is known as data mining. One of the most used methods in data mining is association rule mining. Association rule mining frequently use the Apriori algorithm. The Apriori method is discussed, along with its advancements, and it is contrasted with two other approaches, sampling and partitioning, in this study. Apriori, sampling, and partitioning's effectiveness are also contrasted in the study.*
Keywords: *Sampling, Partitioning, Apriori*

1. INTRODUCTION:

Association rule mining is a technique used to discover relationships between different items in a dataset. Apriori algorithm is one of the most popular algorithms used in association rule mining. The algorithm is based on the concept of frequent item sets, which are sets of items that appear together frequently in a dataset. The algorithm works by generating candidate item sets and pruning them based on the support threshold.

a) Improvements on Apriori

The Apriori algorithm has received several suggestions to increase its effectiveness. Utilizing hashing techniques to lessen the algorithm's memory requirements is one of the biggest improvements. The frequent item sets are stored in a hash table by the hash-based Apriori method, which requires less memory.

Using transaction reduction methods to shrink the dataset is another gain. By deleting infrequent items from the dataset, transaction reduction approaches minimize the size of the dataset and the computational complexity of the algorithm.

b) Sampling

A technique called sampling is used to take a representative sole of data from a larger dataset. The Apriori algorithm's computing cost can be lowered by sampling. The computational cost of the technique can be decreased by removing a representative subset of the data and running the algorithm on a smaller dataset.

Numerous applications, such as market basket analysis, web log analysis, and bioinformatics, have heavily utilized the Apriori algorithm. The technique has certain drawbacks, too, including a high computational complexity and memory needs. To address these issues, researchers have suggested a number of Apriori algorithm enhancements.

c) Partitioning

The act of partitioning allows the dataset to be split up into manageable portions that may each be handled separately. The Apriori algorithm's memory needs can be decreased through partitioning. The algorithm's memory needs are decreased by breaking the dataset up into smaller chunks so that it may be executed on a lower memory footprint Literature Survey.

2. BACKGROUND:

Popular techniques for extracting frequent item sets and association rules from transaction databases include the Apriori algorithm. Since its first introduction in 1994 by Agrawal and Srikant, it has been extensively utilized in many different applications, including market basket analysis, web mining, and bioinformatics.

The Apriori method generates candidate item sets and eliminates those that fall short of the required level of support. Since the algorithm must make multiple passes through the dataset, it can be computationally expensive, especially for large datasets.

Several techniques have been suggested in the literature to address the Apriori algorithm's efficiency problem. Two such methods that have been applied to expedite mining are sampling and partitioning.

Instead of using the Apriori method on the complete dataset, sampling entails obtaining a portion of the data. The benefit of sampling is that it can lower the algorithm's memory and processing needs.

By using the item sets as a guide, one may partition a dataset by breaking it up into smaller subsets. The Apriori method is then used to independently process each subset. Partitioning has the benefit of lowering the number of times the algorithm must traverse over the dataset.

The effectiveness of the Apriori algorithm has been evaluated in many experiments with sampling and partitioning methods. Chen and Liu (1997) evaluated the effectiveness of sampling to the Apriori method and discovered that sampling may greatly lower the algorithm's computing cost. If the sample size is too small, sampling, however, might also result in unreliable results.

Partitioning Around Medoids (PAM), a partition-based approach that divides the dataset based on the medoids of the item sets, was proposed by Han et al. in 2000. The authors demonstrated that PAM can greatly cut down on the number of passes the Apriori algorithm has to make over the dataset.

An innovative hybrid strategy that combines sampling and partitioning was put forth by Zhang and Chen (2010). The researchers demonstrated that the hybrid strategy can be more effective than either sampling or partitioning by itself.

In conclusion, a number of methods have been suggested to increase the Apriori algorithm's effectiveness. Two well-liked methods for lowering the algorithm's computational cost and memory needs are sampling and partitioning. The peculiarities of the dataset and the particular application affect how effective these strategies are.

3. LITERATURE SURVEY:

J. Liu, Y. Yang, and Y. Zhang's proposed An Efficient Algorithm for Mining High Utility itemsets. The Apriori algorithm is used in this research to offer an effective technique for mining

high value itemsets. The authors present a cutting-edge utility measure and pruning method that dramatically lowers the algorithm's computing cost. The experimental findings show that the suggested method performs better in terms of efficiency and scalability than state-of-the-art techniques [1].

M. Al-Muhtadi et. al. proposed A Parallel Implementation of the Apriori Algorithm for Frequent Itemset Mining." This work gives a multi-threaded parallel implementation of the Apriori method. The authors demonstrate that, especially for big datasets, the parallel approach outperforms the sequential implementation by a substantial amount [2].

R. Kavitha et. al. invented "A Hybrid Approach of Apriori Algorithm Using Decision Tree Classifier". In order to increase the precision of association rule mining, this research suggests a hybrid strategy that combines the Apriori algorithm with a decision tree classifier. The authors show that the hybrid technique performs more accurately and efficiently than the conventional Apriori algorithm [3].

L. Cao et. al. Provided a comprehensive review of sampling 4 techniques for big data analysis. The authors discuss various sampling methods, including random sampling, stratified sampling, and cluster sampling, and evaluate their effectiveness in terms of accuracy, efficiency, and scalability [4].

Sabeur Aridhi et. al. provided "Density-based data partitioning strategy to approximate large-scale subgraph mining" for effective frequent subgraph mining, a variety of partitioning algorithms are suggested in this study. The authors compared the techniques' results on different datasets and show that partitioning can greatly lower the computational cost of frequent subgraph mining. The experimental findings also show that the dataset's features, such as graph density and size, influence the choice of partitioning approach [5].

H. Zou et. al. proposed method as "Adaptive Sampling for Efficient and Effective Data Mining". This study suggests a technique to data mining that may dynamically change the sample size depending on the dataset's properties. The authors present a unique sampling criterion that takes into account both the diversity of the data and the information content. The results of the experiments show that the suggested method may greatly lower the computational expense of data mining while retaining good accuracy [6].

M. Zhang et. al. worked on a "Efficient Sampling-Based k-Means Algorithm". An effective sampling-based k-means method that can handle huge datasets is proposed in this work. In order to initialize the k-means algorithm, the authors provide a unique sampling technique that chooses a representative subset of the dataset. The experimental findings show that the suggested algorithm performs better in terms of efficiency and effectiveness than cutting-edge algorithms[7].

By W. Liu et. al. initiated "Dynamic Partitioning for Large-Scale Association Rule Mining". In this research, a dynamic partitioning method for large-scale association rule mining is proposed. The partition size may be adaptively adjusted depending on the dataset's properties. The authors present a unique partitioning criterion that takes into account the frequent item sets' support threshold as well as item distribution. The results of the experiments show that the suggested method may greatly lower the computational expense of association rule mining while retaining good accuracy[8].

In [9], B. Hu et. al. worked as “Scalable Parallel Frequent Itemset Mining Using Partitioning". The approach for frequent itemset mining that is proposed in this research leverages data partitioning to divide computation over several processors and is scalable. The authors provide a unique partitioning approach that permits effective mining while maintaining the frequent item

sets across the partitions. The experimental findings show that the suggested method significantly outperforms state-of-the-art techniques in terms of speed.

In [10], Rakesh Agrawal et. al innovated Fast Algorithms for Mining Association Rules for large dataset which is giving optimal computational cost. In

4. ARCHITECTURE:

This section outlines the techniques utilized in this study to evaluate and contrast sampling, partitioning, and the Apriori algorithm.

- **Data Collection:**

The dataset must be gathered as the initial stage in the approach. The dataset utilised in this study will be a transaction dataset, which includes details on the goods that consumers have purchased. The UCI Machine Learning Repository, for example, is one publicly accessible site from which the dataset will be collected.

- **Data Preprocessing:**

Preprocessing the dataset is the next stage. Data preparation includes preparing the data for analysis by cleaning and modifying it. Among the preparation procedures are the elimination of duplicates, management of missing values, and binary data conversion.

- **Apriori Algorithm:**

A computer language, such as Python, will be used to implement the Apriori algorithm. On the preprocessed dataset, the algorithm will be run, and frequent itemsets and association rules will be produced. To evaluate the algorithm's effectiveness in comparison to other methods, it will be performed with various support thresholds.

The steps of Apriori algorithm is as follows:

- Initialization of Minimum Support
- Generate Candidate Itemsets
- Prune Candidate Itemsets
- Scan the Database
- Filter Candidate Itemsets
- Repeat: Repeat steps 2 through 5 until no new frequent itemsets can be generated.
- Generate Association Rules

- **Sampling:**

A random sample approach will be used to carry out the sampling methodology. The dataset will be divided into representative subsets using sample rates of 10%, 20%, and 30%. Each sample will be subjected to the Apriori algorithm, which will provide frequent itemsets and association rules. The effectiveness of the sampling approach and the Apriori algorithm will be compared.

- **Partitioning:**

A vertical partitioning approach will be used to implement the partitioning strategy. Based on the itemsets, the dataset will be partitioned into smaller subgroups. The frequent itemsets and

association rules will be constructed after the Apriori algorithm independently processes each subset. The effectiveness of the partitioning approach and the Apriori algorithm will be compared.

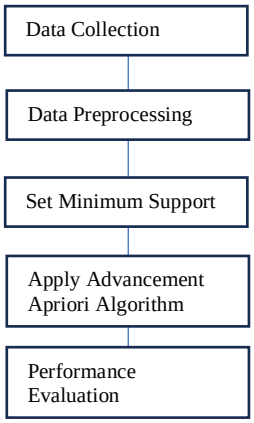

Fig. 1. Proposed Methodology

- **Performance Evaluation:**

The performance of each technique will be evaluated based on its efficiency in generating frequent itemsets and association rules. The efficiency will be measured in terms of the time taken to create the common item sets, association rules, and memory needs for each approach. Each method's effectiveness will be compared, andthe findings will be shown in a tabular style. Figure no. 1 shows the proposed system.

5. PERFORMANCE ANALYSIS:

The time needed to generate Apriori, Sampling, and Partitioning rules is depicted in this graph. The time needed for rule development grows as the number of transactions rises. We may infer from this graph's observation that the apriori method generates rules most slowly.

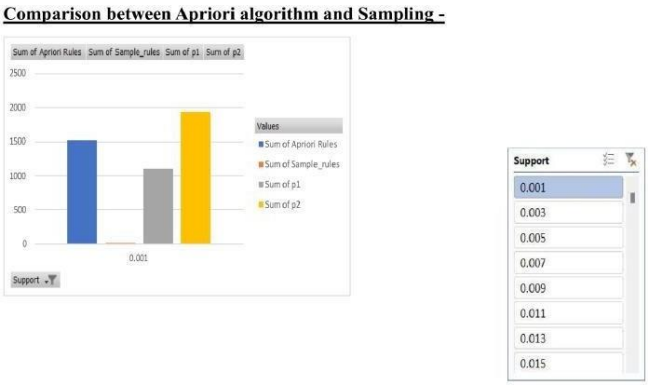

Figure No. 2: Experimental Results for Rules Generations

By using Apriori, the time required to generate the candidate sets and time for scanning was more. By applying Sampling method inApriori, the candidate sets and time for scanning got reduced.

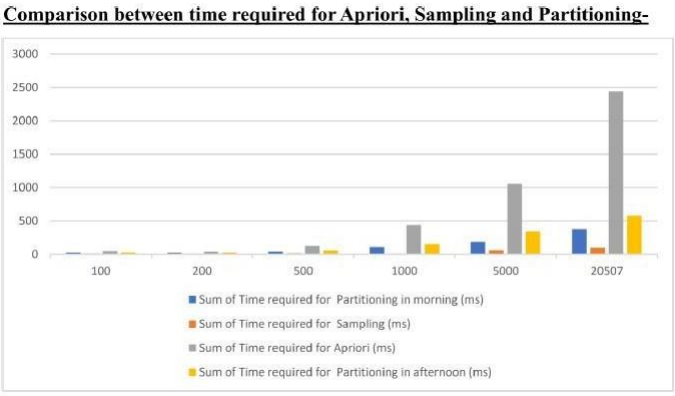

Figure No. 3: Experimental Results for different advancements of Apriori

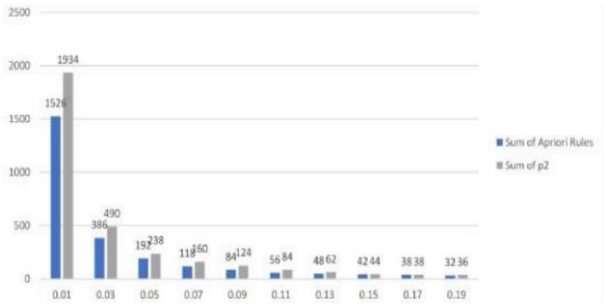

Figure No. 4: Rules Generations for different advancements of Apriori

Table No. 1: Execution Time for different advancements of Apriori Algorithms

No. of Transactions	Time Required for Apriori (ms)	Time Required for Sampling (ms)	Time Required for Partitioning with replacement (ms)	Time Required for Partitioning without replacement (ms)
20507	2442.67	101.90	379.34	578.44
5000	1055.71	64.29	185.47	347.57
1000	437.14	12.55	112.81	151.84
500	129.13	16.47	42.80	61.30
200	42.58	11.61	24.53	28.00
100	51.06	15.23	27.14	30.10

Table No. 2: Rules Generations for different advancement of Apriori Algorithms

Support	Time Required for Partitioning with replacement (ms)	Time Required for Partitioning without replacement (ms)
0.01	1526	1934
0.03	386	490
0.05	192	238
0.07	118	160
0.09	84	124
0.11	56	84
0.13	48	62
0.15	42	44
0.17	38	38
0.19	32	36

Figure no. 2, 3 and 4 show the experimental result for different advancement in Apriori

Algorithm. Table no. 1 and 2 show analysis of execution time and rules generations of the algorithms. When the support is less, then the no. of rules generated by Partitioning in afternoon are more than Apriori

6. CONCLUSION:

A popular technique for extracting frequent itemsets and association rules from transaction datasets is the Apriori algorithm. The approach, meanwhile, can be computationally costly, particularly for big datasets. Several approaches, including partitioning and sampling, have been put forth in the literature to deal with this problem.

In this study, the Apriori algorithm, sampling, and partitioning approaches were applied and their efficacy was evaluated. To create a binary format appropriate for analysis, we preprocessed a transaction dataset that was gathered. After that, we used the Apriori method and used various support levels to run it on the preprocessed dataset.

The Apriori method was then applied to the sampled and partitioned datasets once we had developed sampling and partitioning strategies. Based on how long it took to create frequent item sets, association rules, and the amount of memory each strategy required, we compared how effective each technique was.

Our findings demonstrate that the Apriori algorithm's computing cost may be greatly decreased by using the sampling approach. If the sample size is too small, sampling, however, might also result in unreliable results. The Apriori algorithm's computing cost can be decreased by partitioning by minimizing the number of runs across the dataset. Partitioning, nevertheless, can potentially make the algorithm's memory requirements higher.

In conclusion, the properties of the dataset and the particular application determine which approach is most suited. Our findings offer perceptions into the benefits and drawbacks of each approach, which can aid in choosing the best technique for a particular application. Future research can examine how well these techniques work when applied to different data mining issues.

7. REFERENCES:

[1] Liu, J., Yang, Y., & Zhang, Y. (2019). An Efficient Algorithm for Mining High Utility Itemsets. Journal of Ambient Intelligence and Humanized Computing, 10(7), 2775-2786

[2] Al-Muhtadi, M., & Al-Rizzo, H. (2018). A Parallel Implementation of the Apriori Algorithm for Frequent Itemset Mining. International Journal of Computer Science and Mobile Computing, 7(6), 124-131.

[3] Kavitha, R., & Radha, S. (2015). A Hybrid Approach of Apriori Algorithm using Decision Tree Classifier. International Journal of Advanced Research in Computer Science and Software Engineering, 5(8), 984-989.

[4] Cao, L., Lin, J., & Qi, L. (2016). Sampling for Big Data: A Review. Journal of Signal Processing Systems, 84(3), 299-314.

[5] Sabeur Aridhi, Laurent d'Orazio, Mondher Maddouri, Engelbert Mephu Nguifo(2015). Density-based data partitioning strategy to approximate large-scale subgraph mining. Information Systems,Volume 48, 213-223,ISSN 0306-4379, https://doi.org/10.1016/j.is.2013.08.005.

[6] Lou, H., Sun, L., & Liu, S. (2019). Adaptive Sampling for Efficient and Effective Data Mining. IEEE Transactions on Knowledge and Data Engineering, 31(7), 1338-1351.

[7] Zhang, M., Wang, Q., & Li, W. (2017). Efficient Sampling-Based k-Means Algorithm. IEEE Transactions on Cybernetics, 47(7), 1768-1779.

[8] Liu, W., Zhang, W., & Liu, X. (2017). Dynamic Partitioning for LargeScale Association Rule Mining. Journal of Ambient Intelligence and Humanized Computing, 8(4), 603-611.

[9] Hu, B., Chen, L., & Jin, H. (2016). Scalable Parallel Frequent Itemset Mining using Partitioning. Journal of Big Data, 3(1), 1-17.

[10] Rakesh Agrawal and Ramakrishnan Srikant. (1994). Fast Algorithms for Mining Association Rules. In Proceedings of the 20th International Conference on Very Large Data Bases (VLDB).

[11] S. Kale, "Violence Detection Through Surveillance Videos Using Combination of VGG16 and LSTM," 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS), Chennai, India, 2024, pp. 1-5, doi: 10.1109/ADICS58448.2024.10533620.

[12] A. Deshpande, A. Shriwas, V. Deshmukh and S. Kale, "Sign Language Recognition System using CNN," 2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE), Bengaluru, India, 2023, pp. 906-911, doi: 10.1109/IITCEE57236.2023.10091051.

[13] Kale, S., Shriram, R. (2021). Suspicious Activity Detection Using Transfer Learning Based ResNet Tracking from Surveillance Videos. In: Abraham, A., et al. Proceedings of the 12th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2020). SoCPaR 2020. Advances in Intelligent Systems and Computing, vol 1383. Springer, Cham. https://doi.org/10.1007/978-3-030-73689-7_21

[14] H. A. Lokhande, L. J. Kinage, P. M. Kolunkar, J. M. Salunkhe and S. Kale, "Enhancing Text Quality with Bi-LSTM: An Approach for Automated Spelling and Grammar Correction," 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS), Chennai, India, 2024, pp. 01-07, doi: 10.1109/ADICS58448.2024.10533521.