Association Rule Mining on Spambase Dataset using Tanagra

Pushpa Devi, Vikrant Singh Bhardwaj, K.L. Bansal

Abstract*: There is huge amount of data being generated every minute on internet. This data is of no use until we cannot extract useful information from it. Data mining is the process of extracting useful information or knowledge from this huge amount of data that can be further used for various purposes. Discovering Association rules is one of the most important tasks among all other data mining tasks. Association rules contain the rules in the form of IF then THAN form. The leftmost part of the rule i.e. IF is called as the Antecedent which defines the condition and the rightmost part i.e. ELSE is called as the Consequent which defines the result. In this paper, we present the overview and comparison of Apriori, Apriori PT and Frequent Itemsets algorithm of association component in Tanagra Tool. We analyzed the performance based on the execution time and memory used for different number of instances, support and Rule Length in Spambase Dataset. The results show that when we increase the support value the Apriori PT takes the less execution time and Apriori takes less memory space. When numbers of instances are reduced Frequent Itemsets outperforms well both in case of memory and execution time. When rule length is increased the Apriori algorithm performs better than Apriori PT and Frequent Itemsets.

Keywords: Apriori, Apriori PT, Frequent Itemsets, support, Confidence, Tanagra.

I. INTRODUCTION

Data mining is the process of extracting knowledge, patterns and trends from the large set of databases by posing various queries [1]. The major goals of data mining include **predicting the future based on past experiences**, **improving capabilities of marketing, detecting the abnormal patterns** and **mining hidden knowledge from the data.** Among the various techniques used for data mining Association Rule mining is one of the most widely used technique for finding the frequent patterns. In a Transactional database **frequent patterns** are those Itemsets which are frequently visited at least for a user defined number of times which is known as support threshold [7]. **Support** and **Confidence** are the two strategically measures of the association rule mining which controls the process of association rule mining.

Revised Manuscript Received on April 30, 2020.
***** Correspondence Author
Pushpa Devi*, Department of Computer Science, Himachal Pradesh University, Summerhill, Shimla -5, Himachal Pradesh, India. Email: pushpasuryavanshi0@gmail.com
Vikrant Singh Bhardwaj, Department of Computer Science, Himachal Pradesh University, Summerhill, Shimla -5, Himachal Pradesh, India. Email: vikrantsingh5508@gmail.com
Dr. Kishori Lal Bansal, Professor Department of Computer Science, Himachal Pradesh University, Summerhill, Shimla -5, Himachal Pradesh, India. Email: kishorilalbansal@yahoo.co.in

Support gives the indication how frequently the items appear in a dataset. Confidence indicates the number of times the if-then statements are found true [14]. There are various algorithms available for association rule mining for example Apriori, A- Priori Prefix Tree and Frequent Pattern Growth etc.

The main aim of this paper is to evaluate the performance of **Apriori**, **Apriori PT** and **frequent Itemsets** algorithms using the Tanagra 1.4.50 Tool. Although a brief introduction of these algorithms or sub-components of association component in **Tanagra** Tool is available within the tool itself but Tanagra tutorial available on official website gives more details about the listed algorithms for this study.

A. Association Rule Mining Algorithms:

- **Apriori**

Apriori algorithm was proposed by R AGRAWAL AND R SRIKANT in 1994. This algorithm uses prior information of frequent item set properties hence the Apriori name is given to it [11]. The algorithm is highly efficient in generating frequent item sets.

Apriori algorithm uses a “bottom-up” approach to generate frequent item sets, where frequent subsets are extended Level-by-Level one item at a time and this step is known as **Candidate Generation** [2]. The algorithm will terminate when no further successful extensions are found.

- **Apriori PT**

The implementation of **Apriori** algorithm is fast (in Tanagra) but needs a large memory space. The need of a lot of memory limits **Apriori** algorithm’s performances when we take large dataset or generates numerous rules. For this reason, a new approach is added to TANAGRA: the launching and the control of an external program [15].

A temporary file is created at the time of the execution. This file is then transmitted to the Christian BORGELT’s APRIORI.EXE executable file. Then the rules will be automatically downloaded and displayed in TANAGRA tool.

- **Frequent Itemsets**

Frequent Itemsets mining is one of the main objectives of the data mining. A frequent item set basically refers to the set of items that frequently appear together in a transactional data set. For example, 90% of the customers who buy milk and bread will purchase butter too. This type of rules can be used for the promotion of the products. Frequent Itemsets provides the valuable information which can be used to better understand the relationship among various Itemsets. Frequent Itemsets in TANAGRA is based on the Borgelt's “Apriori.exe” program [16].

II. LITERATURE REVIEW

K. Vanitha [2] analyzed performance of Apriori and FP-growth algorithms. The parameter used for comparison was execution time for different number of instances and confidence. The comparison is done on Super market data set. Author's performance study shows that FP-Growth algorithm is more efficient and scalable than Apriori. **M. Dhanda, S. Guglani, and G. Gupta** [3] stated that in case of large dataset Apriori algorithm is not an efficient algorithm as it consumes more time. Authors have illustrated the disadvantages of Apriori algorithm and how to improve its efficiency by utilizing various attributes. Authors concluded that in a business environment the efficiency of Association rule mining can be improved by using attributes like profit, quantity. **M. Tiwari, M.B. Jha, and O. Yadav** [4] compared the performance of four classification algorithms, and four clustering algorithms on three different datasets named as zoo, Labor and Supermarket dataset. The performance of algorithms is evaluated on three test modes: **hold out method**, **percentage split**, and **full training.** Author's work is focused to allow the retailer to increase customer understanding and make knowledge- driven decisions in order to provide an efficient customer service. **V. Jain, G.S. Narula, and M. Singh** [5] developed new methods to discover knowledge from online store database. They used association rule mining and clustering algorithms to classify the customers based on their reviews about the product. **T.A. Kumbhare and S.V. Chobe** [6] presented an overview of **AIS, SETM, Apriori, Apriori tid, Apriori hybrid, FP-Growth** association rule mining algorithms. Also, compared these algorithms based on some performance criteria like accuracy, data support and execution speed. This paper concluded that FP-growth algorithm performs better than other algorithms. **J. Ashri** [7] focused on the performance of the multidimensional database. Author has used the Apriori algorithm as the base algorithm and optimized the results using empirical algorithmic approach. WEKA, MATLAB, MySQL Database Engine and MySQL J Connector for JDBC tools are used for implementation. **A.K. Mishra, S.K. Pani, and B.K. Ratha** [8] compared Apriori and FP-growth algorithms using weka tool. Authors have analyzed the frequent itemsets generation and number of cycles performed over the Apriori algorithm and Filter Associator in the context of association analysis. **S.O. Fageeri, R. Ahmad, and H. Alhussian** [9] analyzed the performance of Binary-Based algorithm against ELACT algorithm. The parameters used are execution time and memory usage using the CPU profiler of Java VisualVM. The Results show that Binary-Based algorithm performs better than ELACT algorithms. **M. Sinthuja, N. Puviarasan, and P. Aruna** [10] compared the performance of Apriori, FP-Growth and ELACT algorithms based on two parameters runtime and usage of memory on spect heart data, primary tumor and mushroom dataset. **M. Shridhar and M. Parmar** [11] studied couple of good improved methodologies used for Apriori calculation. Authors also conducted a theoretical comparison of Apriori, Improved Apriori and FP-Growth algorithms. It is observed that I/O cost and size of Candidate Itemsets is reduced in Improved Apriori and FP-growth is faster than Apriori in case of large datasets. **D. Ai, H. Pan, X. Li, Y. Gao, and D. He** [12] presented a general survey of various association rule mining algorithms that are applicable to high-dimensional datasets. According to the previous studies the merits and characteristics of the algorithms are explained as well as areas for improvement and optimization strategies that might be better adapted to high-dimensional datasets are pointed out. Author concluded that many methodologies are not able to manage the issue of high dimensionality. Many of the methods with their merits and faults have been proposed but new algorithms that can better balance scalability and interpretability are still not found. **M.A. ALBAHAR and K. G. MOHAMED** implemented a Distributed Denial of Service (DDoS) attack detection system using Association rule learning in Simple Payment Verification (SPV). KDD dataset is used to analyze the detecting accuracy and processing time. To evaluate the effectiveness of the methods Authors used NSL-KDD and GureKDD datasets. The results of this paper show that Association rule learning algorithm is capable of detecting the DDoS attack in SPV.

III. METHODOLOGY

The Methodology followed for this study basically includes two steps:

1. Theoretical Study
2. Implementation

A. Theoretical Study:

The theoretical study includes: study of Data Mining concepts and techniques, association rule mining algorithms and overview of Tanagra Tool. Research papers and Internet are the major contributors of this study.

B. Implementation:

Tanagra 1.4.50 is the open source tool used for the implementation purpose and the dataset used is spambase freely available on https://www.openml.org/d/44 (downloaded on 10:33 AM 10/14/2019). Dataset consists of 4601 instances and 58 attributes.

IV. ANALYSIS OF RESULTS:

The aim of this paper is to compare the performance of the Apriori, Apriori PT and Frequent Itemsets algorithms that are present in the TANAGRA 1.4.50 tool.

The results are analyzed based on two parameters execution time and memory used. Execution time of the algorithms is provided by the tool itself and we use the operating system information for measuring the memory usage [17]. The following tables present the test results of the Apriori, Apriori PT and Frequent Itemsets.

1. For varying support

Table 1.1 Runtime in ms

Support	**Confidence**	**Runtime in ms**		
		Apriori	Apriori PT	Frequent Itemsets
0.33	0.75	**31ms**	46ms	47ms
0.35	0.75	**15ms**	47ms	47ms
0.45	0.75	**15ms**	47ms	47ms

The test results of Table 1.1 show that when we increase the support value from 0.33 to 0.35 runtime for Apriori algorithm has decreased from 31ms to 15ms and remain constant i.e. 15ms when support value increased to 0.45. Increase of 1ms is noticed with increase in support value from 0.33 t0 0.35 and remain constant when support value is increased from 0.35 to 0.45. In case of frequent Itemsets the execution time remains same for varying support value. As a result of this table Apriori takes less time than other two algorithms for varying support value.

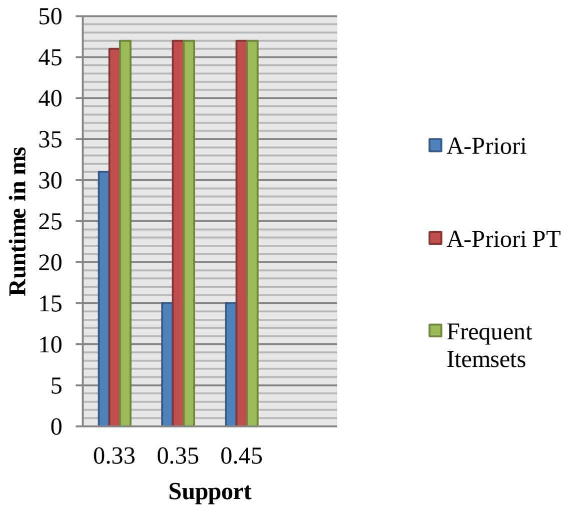


Fig 1.1 Support vs. Runtime

Figure 1.1 is the graphical representation of the Table 1.1 which shows the test results of Apriori, Apriori PT and Frequent Itemsets algorithms when compared based on varying support value against runtime.

Table 1.2 Memory usage in kb

Support	Confidence	Memory usage in kb		
		Apriori	Apriori PT	Frequent Itemsets
0.33	0.75	12,620k	11,484k	**11,452k**
0.35	0.75	11,996k	**11,544k**	11,552k
0.45	0.75	**11,872k**	11,956k	11,964k

The test result of Table 1.2 shows that there is not a much difference in the size of memory Apriori, Apriori PT and Frequent Itemsets occupy.

When the support value is 0.33 Frequent Itemsets took the less memory space i.e. 11,452k than other two algorithms. When we increased the support value from 0.33 to 0.35 Apriori PT took the less memory space than Apriori and Frequent Itemsets algorithms.

At last when support is increased from 0.35 to 0.45 Apriori algorithms took the less memory than other two. As a result of this table we can conclude that for varying support value different algorithms take less memory than others.

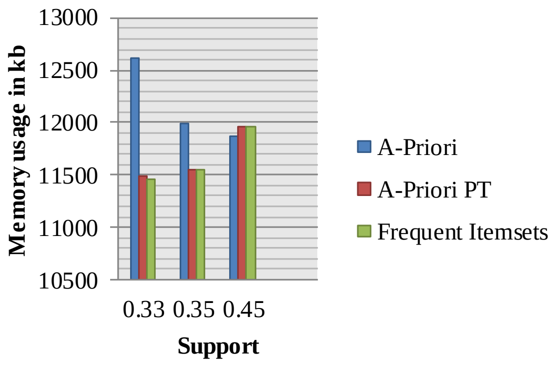


Fig 1.2 Support vs. Memory usage

Figure 1.2 is the graphical representation of the Table 1.2 which shows the test results of Apriori, Apriori PT and Frequent Itemsets algorithms when compared based on varying support value against memory usage.

2. Number of instances

Table 2.1 Runtime in ms

No. Of Instances	Support	Runtime in ms		
		Apriori	Apriori PT	Frequent Itemsets
4601	0.33	**31ms**	46ms	47ms
3499	0.33	**31ms**	**31ms**	46ms
1500	0.33	63ms	31ms	**15ms**

Table 2.1 shows that for 4601 instances Apriori took less execution time i.e. 31ms than Apriori PT and Frequent Itemsets. For 3499 instances Apriori and A- priori PT requires 31ms while Frequent Itemsets require 46ms. For 1500 instances the execution time for Apriori increased to 63ms, Apriori PT execution time remain constant and execution time of Frequent Itemsets reduced with the reduction of number of instances.

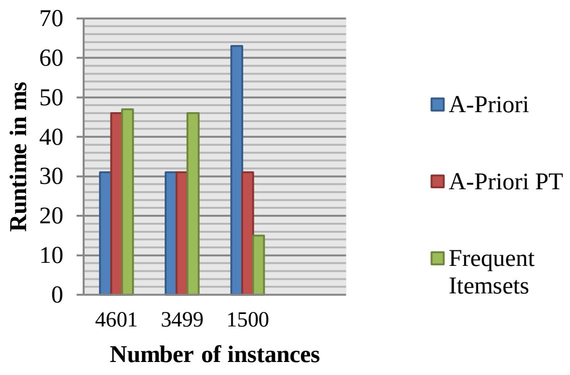


Fig 2.1 Number of instances vs. Runtime

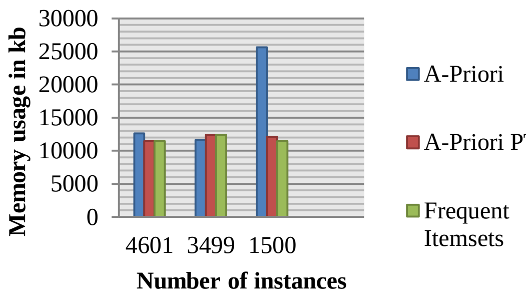
Figure 2.1 is the graphical representation of the Table 2.1 which shows the test results of Apriori,

Apriori PT and Frequent Itemsets algorithms when compared based on different number of instances against runtime.

Table 2.2 Memory usage in kb

N0. Of Instances	Support	Memory usage in kb		
		Apriori	Apriori PT	Frequent Itemsets
4601	0.33	12,620k	11,484k	**11,452k**
3499	0.33	**11,680k**	12,404k	12,380k
1500	0.33	25,640k	12,076	**11,460k**

The test result of Table 2.1 shows that for 4601 instances Frequent Itemsets took the less memory space i.e. 11,452k than other two algorithms. When we decreased the number of instances to 3499 Apriori took less memory space than Apriori PT and Frequent Itemsets algorithms. For 1500 instances Apriori algorithms took less memory than other two. As a result of this table we can conclude that when we decrease number of instances different algorithms take less memory than others.



Fi g 2.2 Number of instances vs. Memory usage

Figure 2.2 is the graphical representation of the Table 2.2 which shows the test results of Apriori, Apriori PT and Frequent Itemsets algorithms when compared based on different number of instances against memory usage.

3. Rule Length

Table 3.1 Runtime in ms

Rule Length	Support	Runtime in ms		
		Apriori	Apriori PT	Frequent Itemsets
4	0.33	**31ms**	46ms	47ms
7	0.33	**47ms**	**47ms**	**47ms**
10	0.33	**31ms**	46ms	47ms

Table 3.1 shows that when rule length is 4 Apriori took less execution time i.e. 31ms than Apriori PT and Frequent Itemsets. When rule length is 7 Apriori all three algorithms took same amount of time 47ms.When rule length is 10, Apriori took least execution time.

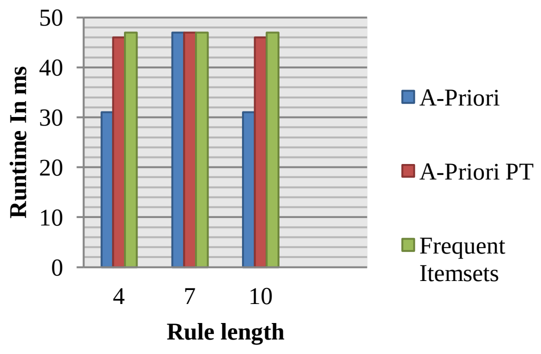


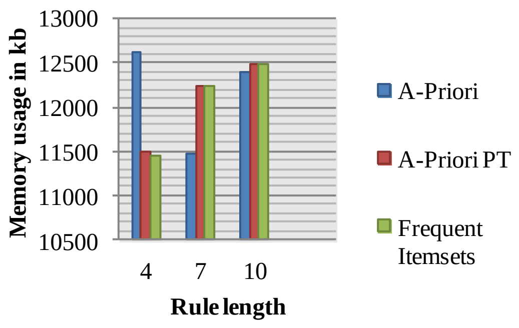
Fig 3.1 Rule length vs. Runtime

Figure 3.1 is the graphical representation of the Table 3.1 which shows the test results of Apriori, Apriori PT and Frequent Itemsets algorithms when compared based on different Rule Length against runtime.

Table 3.2 Memory usage in kb

N0. Of Instances	Support	Memory usage in kb		
		Apriori	Apriori PT	Frequent Itemsets
4	0.33	12,620k	11,484k	**11,452k**
7	0.33	**11,480k**	12,244k	12,236k
10	0.33	**12,400k**	12,476k	12,484k

The test result of Table 3.2 shows that when rule length is 4 Frequent Itemsets took the less memory space i.e. 11,452k than other two algorithms. When we increased the rule length to 7 Apriori took less memory space than Apriori PT and Frequent Itemsets algorithms. For rule length ten Apriori algorithm took less memory than other two algorithms.



3.2 Rule length vs. Memory usage

Figure 3.2 is the graphical representation of the Table 3.2 which shows the test results of Apriori, Apriori PT and Frequent Itemsets algorithms when compared based on different Rule Length against Memory Usage.

V. CONCLUSION

In this paper three association rule mining algorithms namely, **Apriori**, **Apriori PT** and **Frequent Itemsets** are compared. Parameters used are **execution time** and **memory used** for different number of instances, support and Rule Length. The results show that when we increase the support value the Apriori algorithm takes less execution time than Apriori PT and Frequent Itemsets and different algorithms for different support values take less memory space than others. When numbers of instances are reduced to 1500 that is the one third of the total instances Frequent Itemsets outperforms well both in case of memory and execution time. When rule length is increased the Apriori algorithm performs better than Apriori PT and Frequent Itemsets both in terms of execution time and memory usage.

REFERENCES

1. B. Thuraisingham, Data mining: technologies, techniques, tools, and trends. CRC press, 2014.
2. K. Vanitha, "EVALUATING THE PERFORMANCE OF ASSOCIATION RULE MINING ALGORITHMS", *Journal of Global Research in Computer Science*, vol. 2, pp.101-103, June 2011.
3. M. Dhanda, S. Guglani, and G. Gupta, "Mining Efficient Association Rules Through Apriori Algorithm Using Attributes", *IJCST*, Vol. 2, Issue 3, September 2011.
4. M. Tiwari, M.B. Jha, and O. Yadav, "Performance analysis of Data Mining algorithms in Weka", *IOSR journal of computer engineering*, Vol. 6, Issue 3, pp.32-41, Sep-Oct. 2012.
5. V. Jain, G.S. Narula, and M. Singh, "Implementation of data mining in online shopping system using tanagra tool", *International Journal of Computer Science and Engineering*, Vol. 2, Issue 1,pp. 47-58, Feb 2013.
6. T.A. Kumbhare, and S.V. Chobe,"An overview of association rule mining algorithms", *International Journal of Computer Science and Information Technologies*", Vol. 5, Issue 1, pp.927-930, 2014.
7. J. Ashri, " PERFORMANCE EVALUATION OF ASSOCIATION RULE MINING ON PARTITIONED DATA*", International Journal of Enterprise Computing and Business Systems*", Vol. 4, Issue 2, 2014.
8. A.K. Mishra, S.K. Pani, and B.K. Ratha, "ASSOCIATION RULE MINING WITH APRIORI AND FPGROWTH USING WEKA", *Internal Journal of Advanced Technology in Engineering and Science*, Vol. 3,Issue 1, September 2015.
9. S.O. Fageeri, R. Ahmad, and H. Alhussian, "A performance analysis of association rule mining algorithms",2016 *3rd International Conference on Computer and Information Sciences (ICCOINS)*,2016 ,pp. 328-333.
10. D. Ai, H. Pan, X. Li, Y. Gao, and D. He," Association rule mining algorithms on high-dimensional datasets", *Artificial Life and Robotics*, Vol *23*, Issue 3, pp.420-427, 30 May 2018
11. M. Sinthuja, N. Puviarasan, and P. Aruna, "Evaluating the performance of association rule mining algorithms", *World Applied Sciences Journal*, Vol. *35*, Issue 1, pp. 43-53, 2017
12. M. Shridhar and M. Parmar, "Survey on Association Rule Mining and Its Approaches", *World Applied Sciences Journal*, Vol. *5*, Issue 3, Mar. 2017
13. M.A. ALBAHAR and K. G. MOHAMED,"USING ASSOCIATION RULE LEARNING TO DETECT DDOS ATTACK IN SPV", *Journal of Theoretical and Applied Information Technology,* Vol. 97, Issue 5, 15th March 2019
14. https://searchbusinessanalytics.techtarget.com/definition/association-rules-in-data-minin [23/10/2019 5:20 PM]
15. http://eric.univlyon2.fr/~ricco/tanagra/fichiers/en_Tanagra_A_Priori_Prefix_Tree.pdf [23/10/2019 5:50 PM]
16. http://eric.univlyon2.fr/~ricco/tanagra/fichiers/en_Tanagra_Itemset_Mining.pdf [23/10/2019 8:13 PM]
17. http://eric.univlyon2.fr/~ricco/tanagra/fichiers/en_Tanagra_Perfs_Comp_SVM.pdf [23/10/2019 10:16 PM]