Evaluation of Faculty Performance Using Improved Apriori and Association Rule Mining

Mr. D. K. Kirange
Dept. of Computer and IT, J T Mahajan College of Engineering,
Faizpur, Tal, Yawal, Dist. Jalgaon, India

Smt Shubhangi D Patil
Department of Information Technology
Government Polytechnic Jalgaon

Mr Kanchan S Bhagat
Dept. of Electronics and Telecommunications,
J T Mahajan College of Engineering,
Faizpur, Tal, Yawal, Dist. Jalgaon, India

***Abstract*: Apriori is the most popular algorithm that is used to extract frequent itemsets from large data sets where these frequent itemsets can be used to generate association rules. Such rules are used as a basis for discovering knowledge such as detecting unknown relationships and producing results which can be used for decision making and prediction.**

When the data size is very large, both memory use and computational cost for Apriori algorithm are very expensive. And in this case the Apriori algorithm performance inefficient. In our research we propose an Adaptive Apriori approach with enhanced speedup and performance. In the proposed algorithm, the intermediate dynamic dataset is created separately using MATLAB by using the database transactions at each level separately. Thus instead of scanning the entire database, we need to scan only the extracted rows and columns at each level.

The candidate itemset generation for Apriori algorithm is improved here. The most of the candidate itemset generation occurs at support less than or equal to 0.5 as the number of frequent transactions is more with less support count. Hence the Adaptive Apriori algorithm performs better for support less than 0.5 only. The proposed Adaptive Apriori algorithm outperforms the basic Apriori because at each level, the transactions with minimum support are eliminated. Hence not considered for higher levels. This helps to reduce the size of the database at each level which saves a lot of time, and a noticeable improvement in the speed by reducing the frequent database scans.

1. Introduction

In Data Mining, for location and fascination of relations in variables in large databases, Association Rule Mining is a standard and well researched technique. Before applying various data mining techniques such as classification, clustering and prediction, for data analysis, association rule mining is used. The association rule mining was first proposed by Agrawal et al. [1]. It is one of the most recommended research area which is applicable in most of the fields like analysis of market trends, forecasting and detection of faults. While analysis of the market trends, association rule mining is used to obtain all association rules like “Items X and Y are bought by the customer at the same time”. Such rules are represented like *X->Y* where *X* and *Y* are sets of items that from a transactional database. The percentages of transactions in the database containing X U Y define the support of association rule *X ->Y*. A database can be analysed by finding interesting relationships and patterns among items in the database by using association rule mining. The process of association rule mining is divided in two steps; first find all frequent items from the dataset and then discovering the relationships among the items in the database. Itemset denotes a set of items. Itemsets with support count more than the minimum support threshold are referred as frequent itemsets. Mostly the performance of the association rule mining is affected by the first step, as next step of association rule mining is simple [2]. Hence mostly association rule mining is mostly called as frequent itemset mining also.

The two most frequently used algorithms of association rule mining are Apriori and FP-Growth [3, 4]. Both of these algorithms are having different approaches for finding frequent itemsets. The Apriori Algorithm generates the frequent itemsets level wise using the apriori property. But the major drawback of the apriori algorithm is that more execution time is needed for generating the candidate itemsets. Also the number of database scans required is more. The number of database scans required for FP growth is less as it creates the tree structure. While performing the association rule mining using Apriori algorithm, the first level candidate itemset are generated and then these are used to generate the second level candidate itemset and so on. The number of database scans as well as time required for frequent itemset mining is more in this case. An adaptive itemset mining is proposed here to overcome this problem. Here the frequent itemset for the last level are generated, and then all transactions contained in the last level are found. These transactions are copied for all lower levels also and new transaction database is created by deleting the already copied transactions for each subsequent level.

In this research, Adaptive Apriori Algorithm is proposed as a new and efficient way for frequent itemset mining and compared with Apriori algorithm. The proposed Adaptive Apriori algorithm is improved in terms of time required as well as number of database scans. The intermediate dynamic dataset is created separately using MATLAB by using the database transactions at each level separately. Thus instead of scanning the entire database, we need to scan only the extracted rows and columns at each level. The proposed improved Apriori algorithm outperforms the basic Apriori because at each level, the transactions with minimum support are eliminated. Hence not considered for higher levels. This helps to reduce the size of the database at each level which saves a lot of time, and a noticeable improvement in the speed by reducing the frequent database scans.

2. Advances in Association Rule Mining

2.1 Algorithms for Association Rule mining

Many algorithms for generating association rules were presented over time.

Some well-known algorithms are Apriori, Eclat and FP-Growth, but they only do half the job, since they are algorithms for mining frequent itemsets. Another step needs to be done after to generate rules from frequent itemsets found in a database.

- Apriori algorithm

Apriori uses a breadth-first search strategy to count the support of itemsets and uses a candidate generation function which exploits the downward closure property of support.

- Eclat algorithm

Eclat (alt. ECLAT, stands for Equivalence Class Transformation) is a depth-first search algorithm using set intersection. It is a naturally elegant algorithm suitable for both sequential as well as parallel execution with locality-enhancing properties. It was first introduced by Zaki, Parthasarathy, Li and Ogihara in a series of papers written in 1997 [5, 6].

- FP-growth algorithm

FP stands for frequent pattern. [7]

In the first pass, the algorithm counts occurrence of items (attribute-value pairs) in the dataset, and stores them to 'header table'. In the second pass, it builds the FP-tree structure by inserting instances. Items in each instance have to be sorted by descending order of their frequency in the dataset, so that the tree can be processed quickly. Items in each instance that do not meet minimum coverage threshold are discarded. If many instances share most frequent items, FP-tree provides high compression close to tree root.

Recursive processing of this compressed version of main dataset grows large item sets directly, instead of generating candidate items and testing them against the entire database. Growth starts from the bottom of the header table (having longest branches), by finding all instances matching given condition. New tree is created, with counts projected from the original tree corresponding to the set of instances that are conditional on the attribute, with each node getting sum of its children counts. Recursive growth ends when no individual items conditional on the attribute meet minimum support threshold, and processing continues on the remaining header items of the original FP-tree.

Once the recursive process has completed, all large item sets with minimum coverage have been found, and association rule creation begins. [8]

2.2 Summary of the Literature

Mining Association Rules is one of the most important in data mining. Association rules are of interested in database researchers and data mining users. Since 90s, different approaches of data mining have been proposed for discovering useful knowledge from very large semantic datasets. A survey of previous research in the area is provided below:

- Ashraf Sadat Heydari Yazdi,and Mohsen Kahani in their paper titled "A Novel Model for Mining Association Rules from Semantic Web Data" has stated, two general phases in semantic association rule mining system:1)semantic transaction production and 2)running semantic association rule mining algorithm on them. The algorithm is rewritten to deal with semantic transactions and semantic rules, with their predefined format in the ontology will be resulted [9].
- Rakesh Agrawal, Tomasz Imielinski and Arun Swami in their paper titled "Mining Association Rules between sets of Items in Large Database" has stated that if there is a large database of customer transactions, the Memory reclamation algorithm defined in the paper incorporates buffer management, novel estimation and pruning techniques. They also present results of applying this algorithm to sales data customer commercial obtained from a large retailing company, which intriguing shows the effectiveness of the algorithm [10].
- Farah Hanna AL-Zawaidah and Yosef Hasan Jbara in their paper titled "An Improved Algorithm for Mining Association Rules in Large Databases" stated that, mining association rules in large databases is a topic of data mining. The approach proposed in this paper is derived from the conventional Apriori approach with features added to improve data mining performance. The approach to attained desired improvement is to create efficient new algorithm out of the conventional extensive one by adding new features to the Apriori approach. The proposed mining transaction and algorithm can efficiently discover the association rules between the large items in large database. They have performed extensive experiments and compared the performance of their algorithm with existing discovering algorithms found in the liter [11].
- S.C.Punitha, P. Ranjith Jeba Thangaiah and M. Punithavalli in their paper titled "Performance Analysis of Clustering using Partitioning and Hierarchical Clustering Techniques" stated the HAC method which gives various algorithmic aspects, including well-definiteness and computational properties. The basic idea of the algorithm in HAC method is to merge documents based on their similarity into clusters. This method starts with each example in its own cluster and iteratively combines them to form larger and larger clusters. The effectiveness of this technique is improving the search efficiency over sequential scans method [12].

• Peter Fule and John F. Roddick in their paper titled "Experiences in Building a Tool for Navigating Association Rule Result Set" stated the model IRSetNav which has capabilities in redundant rule reduction, subjective interestingness evaluation, item and item set pruning, related information searching, text-based item set and rule visualization, hierarchy based searching and tracking changes between data sets using a knowledge base. It also incorporates several techniques that have found to be useful for speeding up the knowledge discovery process. And also reduce iterations in the knowledge discovery process by reducing its iterative nature [13].
• Mohd Helmy Abd, Mohd Norzali Haji Mohd and Mohamad Mohsin in their paper titled "Discovering Web Server Logs Patterns Using Generalized Association Rules Algorithm" focused on the aspect Kalyani A.Kale et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 7 (3), 2016, 1328-1331 www.ijcsit.com 1330 of web usage mining. They stated that as commercial companies as well as academic researchers developed an array of tools that perform several data mining algorithms on log Files coming from web servers in order to identify user behavior on a particular web site. Performing this kind of investigation on the web site can provide information that can be used to better accommodate the user's needs [14].
• Ming-Cheng Tseng ·Wen-Yang Lin and Rong Jeng in their paper titled "Updating generalized association rules with evolving taxonomies" stated the problem of updating the discovered generalized association rules under evolving taxonomies. For this purpose they proposed two algorithms Diff _ET and Diff_ET2 are used for updating generalized frequent item sets. And evaluation showed that both algorithms are effective and have good linear scale-up characteristics [15].
• Zahir Tari and Wensheng Wu in their paper titled "ARM: A Hybrid Association Rule Mining Algorithm" stated that Most of the approaches for association rule mining focus on the performance of the discovery of the frequent item sets. They are based on the algorithms that require the transformation of from one representation to another, and therefore excessively use resources and incur heavy CPU overhead. They Proposes a hybrid algorithm that is resource efficient and provides better performance. In addition, they propose a comparison algorithm (CmpApr) that compares candidate item sets with a transaction, a filtering algorithm (FilterApr) that reduces the number of comparison operations required to find frequent item sets. ARM has better performance and scales linearly [16].

3 Adaptive Apriori Algorithm

The Adaptive Apriori algorithm proposed here is able to overcome the basic Apriori algorithm in terms of number of database scans as well as time required. The size of the database is reduced at each level. This algorithm uses a dynamic technique to reduce the time required for candidate itemset generation. It is claimed that the size of the database is reduced at each level starting from last to first and hence the time required for candidate itemset generation is reduced as compared with basic Apriori algorithm. Here we generate the dynamic intermediate database for each level separately.
For example, when scanning each transaction in the database find all the transactions which contain all items. These transactions are considered for generation for level K itemset. The same transactions are also considered for generating level 1 to level k-1 itemsets. So these are copied for all these levels. Now the database is updated by deleting all these transactions. This updated database is again considered for generating level L k-1 itemsets. Hence the size of the database is reduced at level of candidate itemset generation as well as the time required is also minimized.

Algorithm:
Input D, a database of transactions
Min_sup, the minimum threshold support
Output Lk Maximal frequent itemsets in D
Ck Set of Candidate k-itemsets.
Method:
1. Generate the Intermediate Database
a. Find all transactions to be considered for level K containing all itemsets.
b. Copy these transactions in the database to be considered for level 1 to K
c. Delete these transactions from the database D and update the database.
d. Now consider this updated (reduced in size) database for finding all transactions for level 1 to K-1
e. Repeat the steps subsequently and update the database.
2. Consider this updated database for candidate itemset generation at each step.
3. L1 =Frequent items of length 1.
4. For (k=1; Lk! =ɸ; k++) do.
a. Consider D as intermediate updated database for level k
b. Ck+1=candidates generated from Lk.
c. For each transaction t in database D do.
d. Increment the count of all candidates in Ck+1 that are contained in t.
e. Lk+1 =candidates in Ck+1 with minimum support
f. end do
• Return the set Lk as the set of all possible frequent itemsets

In this algorithm, the intermediate database is generated to reduce the time required for candidate itemset generation.

5. Performance Analysis

5.1 Task 1: Analyzing the Effect of Size of the Dataset on Execution Time

5.1.1 Dataset

We have used the Turkiye Student Evaluation Data Set This data set contains a total 5820 evaluation scores provided by students from Gazi University in Ankara (Turkey). There is a total of 28 course specific questions and additional 5 attributes. [17]

5.1.2 Evaluations and Results

In this work we have considered various number of database instances from the Turkiye Student Evaluation Data Set. The performance of the algorithm in terms of time required for various support thresholds is computed. The proposed adaptive Apriori algorithm performs better than basic Apriori algorithm. The results for different number of transactions are depicted in Table 1. As depicted in table 1, the execution time required for adaptive apriori is less as compared to basic apriori algorithm for support values 0.2 and 0.4 where maximum rules are generated. For the support range of 0.6 to 1, there are no transactions in the having such a greater support. Hence the time required is comparatively less for both the algorithms.

5.2 Task 2: Analyzing the effect of Dimensionality of the Dataset

It is well known that the size of the database for defining candidates has great effect on running time and memory need. The usefulness of the adaptive Apriori algorithm in terms of dimensionality of the dataset is demonstrated. We presented experimental results, showing that the proposed algorithm always outperform Apriori.

5.2.1 Dataset

To evaluate the performance of the proposed algorithm, we have tested it on Turkey student's database of faculty evaluations. Different 10 standard datasets from UCI machine repository of Data are used [18].

	Execution Time				
Support	**0.2**	**0.4**	**0.6**	**0.8**	**1**
	Number of Transactions = 1000				
Apriori	60.00146	18.55668	0.433772	0.420394	0.424153
Adaptive Apriori	**54.00471**	**11.84865**	**0.923352**	**0.855684**	**0.898021**
	Number of Transactions = 2000				
Apriori	180.218668	**58.058553**	**1.28324388**	**1.281934849**	**1.275869409**
Adaptive Apriori	**99.19852**	95.06378	2.84107	2.534055	2.512093
	Number of Transactions = 3000				
Apriori	344.0570988	108.51069	2.48507891	2.47334696	2.480932981
Adaptive Apriori	**186.895**	**66.87206**	5.387782	5.641123	5.73613
	Number of Transactions = 4000				
Apriori	571.786027	181.51928	**4.1153185**	**4.10401992**	**4.11470736**
Adaptive Apriori	**308.1967**	**109.1488**	10.08284	10.3227	10.34951
	Number of Transactions = 5000				
Apriori	863.084	177.7463	**6.215065**	**6.24219**	**6.198376**
Adaptive Apriori	**449.9942**	**163.5153**	17.24945	18.03369	16.48372

Table 1: Analysis of the effect of Size of the Dataset for Turkiye Student Evaluation Data Set

5.2.2 Evaluations and Results

The Turkey student's database of faculty evaluations is used for analyzing the effect of database dimensionality on the performance of the adaptive Apriori algorithm. The performance of the algorithm is evaluated for 5 to 25 dimensionality of the data. Table 2 shows the time required for the association rule analysis for different dimensions. Various support thresholds are considered for evaluating the performance.

	Execution Time

Support	0.2	0.4	0.6	0.8	1
	Dimensionality =5				
Apriori	0.0878574	0.004569	**0.001478**	**0.001736**	0.001633
Adaptive Apriori	**0.025232**	0.003867	0.001929	0.003161	**0.002217**
FP Growth	0.1232197	0.00362	0.003217	0.002417	0.002544
	Dimensionality =10				
Apriori	0.3510666	0.304392	0.29935	**0.005706**	**0.005723**
Adaptive Apriori	0.6639832	**0.225795**	**0.221598**	0.010449	0.010321
FP Growth	**0.3199109**	0.311084	0.3085	0.269089	0.266512
	Dimensionality =15				
Apriori	145.96017	148.9162	145.7916	0.015913	0.009564
Adaptive Apriori	**125.81109**	**129.9267**	**109.4859**	**0.01579**	**0.016269**
FP Growth	639.97902	636.913	635.8278	429.2786	423.8414

Table 2: Analyzing the Effect of dimensionality of the data for Turkiye Dataset.

Table 3 shows the time required for the different standard datasets from UCI machine repository data.

Iris Dataset	**Dimensionality = 4**			**Number of Transactions = 150**		
Support	**Number of Database Scans**			**Execution Time**		
	Apriori	**Adaptive Apriori**	**FP Growth**	**Apriori**	**Adaptive Apriori**	**FP Growth**
0.2	3600	**1991**	450	0.312203739	**0.195854**	0.047071
0.3	3600	**1991**	450	0.267747833	**0.192933**	0.049225
0.4	2100	**1991**	450	**0.147016486**	0.158127	0.045218
0.5	2100	**1991**	450	**0.137998971**	0.157415	0.044302
0.6	2100	**1991**	450	**0.133466377**	0.159509	0.044051
0.7	600	1393	450	**0.030306572**	0.058252	0.044425
0.8	600	1393	450	**0.030456875**	0.058248	0.045289
0.9	600	1393	450	**0.030666438**	0.052029	0.044083
1.0	600	1393	450	**0.030549572**	0.060731	0.043545
Lenses Dataset	**Dimensionality = 5**			**Number of Transactions = 24**		
Support	**Number of Database Scans**			**Execution Time**		
	Apriori	**Adaptive Apriori**	**FP Growth**	**Apriori**	**Adaptive Apriori**	**FP Growth**
0.2	2280	**880**	72	0.199759913	**0.185549**	0.057842
0.3	1272	**783**	72	0.094016486	**0.091628**	0.049595
0.4	552	599	72	**0.035011319**	0.042783	0.046774

0.5	456	576	72	**0.027471341**	0.036979	0.046023
0.6	168	360	72	**0.008935743**	0.013931	0.045832
0.7	144	336	72	**0.007338691**	0.011637	0.045422
0.8	144	336	72	**0.007225798**	0.017166	0.04833
0.9	144	336	72	**0.007893223**	0.0128	0.049934
1.0	144	336	72	**0.007563814**	0.011935	0.045771

Letter Recognition Dataset	**Dimensionality = 16**			**Number of Transactions = 500**		
Support	**Number of Database Scans**			**Execution Time**		
	Apriori	**Adaptive Apriori**	**FP Growth**	**Apriori**	**Adaptive Apriori**	**FP Growth**
0.2	1099500	**299320**	1500	89.13398964	**86.82006**	18763.49776
0.3	420500	**123034**	1500	33.2626408	**34.18503**	18806.16984
0.4	188500	**65908**	1500	**14.44004619**	15.27756	18846.4248
0.5	94000	**53535**	1500	**6.456744988**	6.739529	18789.65476
0.6	40000	**39599**	1500	**2.322393815**	2.662026	18846.39542
0.7	13000	23117	1500	**0.650219151**	1.027188	18872.8769
0.8	10000	21617	1500	**0.457103205**	0.775256	18764.42162
0.9	8500	20117	1500	**0.377820751**	0.672706	18788.46186
1.0	8500	20117	1500	**0.377295675**	0.683889	18806.28676

Solar Flare 1 Dataset	**Dimensionality =10**			**Number of Transactions = 323**		
Support	**Number of Database Scans**			**Execution Time**		
	Apriori	**Adaptive Apriori**	**FP Growth**	**Apriori**	**Adaptive Apriori**	**FP Growth**
0.2	26486	**10665**	969	**3.125596501**	3.234232	4.179150629
0.3	26486	**10665**	969	3.052655066	**2.548542**	4.276742374
0.4	9690	**8500**	969	**0.946360482**	1.120249	4.889476951
0.5	9690	**8500**	969	1.067578674	**1.045011**	4.081552725

0.6	9690	**8500**	969	**1.014192453**	1.121368	4.218683084
0.7	7429	**8500**	969	**0.677490853**	0.885675	4.119156918
0.8	4522	7534	969	**0.357810673**	0.59566	4.78865433
0.9	3553	7534	969	**0.323234554**	0.607484	4.960056604
1.0	3230	7211	969	**0.252225897**	0.513883	4.851191688

Solar Flare 2 Dataset	**Dimensionality =10**			**Number of Transactions = 1066**		
Support	**Number of Database Scans**			**Execution Time**		
	Apriori	**Adaptive Apriori**	**FP Growth**	**Apriori**	**Adaptive Apriori**	**FP Growth**
0.2	31890	**26988**	3189	3.146150947	**2.951359**	4.412993786
0.3	31890	**26988**	3189	3.071047247	**2.965903**	4.337874486
0.4	17008	26046	3189	**1.32052341**	1.867663	4.332998204
0.5	14882	23920	3189	**1.157835469**	1.688708	4.33721969
0.6	14882	23920	3189	**1.151850754**	1.749445	4.46004593
0.7	14882	23920	3189	**1.173066155**	1.708596	4.334316417
0.8	14882	23920	3189	**1.153577446**	1.688501	4.411691583
0.9	10630	22857	3189	**0.91047477**	1.623334	4.516422866
1.0	10630	22857	3189	**0.725768021**	1.330733	4.351089877

Table 3: Analyzing the Effect of dimensionality of the data for UCI Dataset

6 Conclusion and Future scope

Association rule mining plays the major role in the field of data mining. The association rule mining is divided in two steps. Firstly it finds all frequent itemsets and then it generated the association rules. Apriori algorithm is one of the most important algorithms proposed for frequent itemset mining. But the Apriori algorithm required more time for generation of frequent itemsets as well as the number of database scans is more.

In this research the improved Apriori algorithm is proposed which is more efficient in terms of time required as well as number of database scans. In this algorithm, the intermediate database is created at each level. Hence scanning the entire database at each subsequent level is avoided. Which reduces the time required for candidate itemset generation as well as the number of database scans. The performance of the proposed algorithm is evaluated using the Turkiye standard student faculty evaluation dataset as well as the real time dataset. Different 10 standard datasets from UCI machine repository of data are used to demonstrate the efficiency of the algorithm for different dimensionality of the data.

Though this research work has been successful in addressing the problem of less execution time and lower number of database scans with the help of proposed modules, this research work has not taken the following aspects into account:

1. Setting up dynamic minimum support values according to the data
2. Generation of only user interested association rules.
3. Parallelizing the data among different processor.

In future, further research work can be preceded in the aforementioned aspects.

References

1. Agrawal, R., T. Imielinski, and A. Swami, “Mining association rules between sets of items in large databases.” In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, SIGMOD ’93, New York, NY, USA, 1993, pp. 207–216.
2. Han, J. and M. Kamber, “Data *Mining. Concepts and Techniques”* (2nd ed. ed.). Morgan Kaufmann.,2006
3. Agrawal, R. and R. Srikant, “Fast algorithms for mining association rules in large databases.” In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB ’94, San Francisco, CA, USA, pp. 487–499. Morgan Kaufmann Publishers Inc., 1994
4. Han, J., J. Pei, and Y. Yin., “Mining frequent patterns without candidate generation.” In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD ’00, New York, NY, USA, 2000, pp. 1–12.
5. Mohammed Javeed Zaki, Srinivasan Parthasarathy, M. Ogihara, Wei Li: New Algorithms for Fast Discovery of Association Rules. KDD 1997.
6. Mohammed Javeed Zaki, Srinivasan Parthasarathy, Mitsunori Ogihara, Wei Li: Parallel Algorithms for Discovery of Association Rules. Data Min. Knowl. Discov. 1(4): 343-373 (1997)
7. Han (2000). "Mining Frequent Patterns without Candidate Generation". Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. SIGMOD '00: 1–12. doi:10.1145/342009.335372.
8. Witten, Frank, Hall: Data mining practical machine learning tools and techniques, 3rd edition.
9. Ashraf Sadat Heydari Yazdi, Mohsen Kahani, “A Novel Model for Mining Association Rules from Semantic Web Data” in Engineering Faculty Ferdowsi University of Mashhad, 978-1-4799-3351- 8/14/$31.00 ©2014 IEEE
10. Rakesh Agrawal, Tomasz Imielinski and Arun Swami, “Mining Association Rules between Sets of Items in Large Databases” in IBM Almaden Research Center650 Harry Road, San Jose, CA 95120 2012
11. Farah Hanna AL-Zawaidah , Yosef Hasan Jbara,” An Improved Algorithm for Mining Association Rules in Large Databases” in World of Computer Science and Information Technology Journal (WCSIT) ISSN: 2221-0741 Vol. 1, No. 7, 311-316, 2011.
12. S.C. Punitha, P. Ranjith Jeba Thangaiah and M. Punithavalli, “Performance Analysis of Clustering using Partitioning and Hierarchical Clustering Techniques” in International Journal of Database Theory and Application Vol.7, No.6 (2014), pp.233-240
13. Peter Fule and John F. Roddick, “Experiences in Building a Tool for Navigating Association Rule Result Set” Copyright c 2004, Australian Computer Society, Australasian Workshop on Data Mining andWeb Intelligence (DMWI04), Dunedin, New Zealand.Conferences in Research and Practice in Information Technology, Vol. 32.
14. Mohd Helmy Abd Wahab, Mohd Norzali Haji Mohd, Mohamad Farhan Mohamad Mohsin, “Discovering Web Server Logs Patterns Using Generalized Association Rules Algorithm” in Intech ISBN: 978-953-307-067-4, 2010
15. Ming-Cheng Tseng ·Wen-Yang Lin · Rong Jeng, “Updating generalized association rules with evolving taxonomies” in Appl Intell (2008) 29: 306–320 DOI 10.1007/s10489-007-0096-5
16. Zahir Tari and Wensheng Wu, “ARM: A Hybrid Association Rule Mining Algorithm” in Springer journal, 2006
17. Website:
https://archive.ics.uci.edu/ml/datasets/Turkiye+Student+Evaluation
18. Gunduz, G. & Fokoue, E., “UCI Machine Learning Repository” Irvine, CA: University of California, School of Information and Computer Science, 2012.

Publications

1. Ms. Shubhangi Patil., Dr. Ratnadeep R Deshmukh., “Assigning Subjects to teachers using Apriori Algorithm”, International Journal of Electronics Communication and Computer Technology (IJECCT) Volume 5 Issue 2 (March 2015), ISSN:2249-7838
2. Ms. Shubhangi Patil., Dr. Ratnadeep R Deshmukh, “Review and Analysis of Apriori algorithm for Association Rule Mining”, International Journal of Latest Trends in Engineering and Technology (IJLTET) Volume 6 Issue 4 March 2016
3. Ms. Shubhangi Patil., Dr. Ratnadeep R Deshmukh., D. K. Kirange, “Adaptive Apriori Algorithm for Frequent Itemset Mining”, 2016 IEEE International Conference System Modeling & Advancement in Research Trends (SMART) IEEE Explore DOI: 10.1109/SYSMART.2016.7894480.
4. Ms. Shubhangi Patil., Dr. Ratnadeep R Deshmukh., D. K. Kirange, “Adaptive Apriori Algorithm for Frequent Itemset Mining”, International Conference on Knowledge Engineering, Dr. B. A. M. U. Aurangabad. ICKE 2016.

Research Guide

Dr. R. R. Deshmukh.
Head & Professor,
Department of Computer Science & IT,
Dr. Babasaheb Ambedkar Marathwada University,
Aurangabad

Research Student

Smt. Shubhangi D. Patil.