Volume 4, Issue 6, June 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Improved Apriori Algorithm Using Fuzzy Logic

Heydar Jafarzadeh*, Mehdi Sadeghzadeh
Department of Computer Engineering,
Science and Research branch, Islamic Azad University,
Ilam, Iran

Abstract—One problem Apriori algorithm and other algorithms in the field association rules mining, this is user must determine the threshold minimum-support. Consider that the user wants to apply Apriori algorithm on a database with millions of transactions, definitely user can not have the necessary knowledge about all the transactions in the database, and therefore would not be able to determine an appropriate threshold. Our goal in this paper improved algorithm Apriori, to achieve it, initially will try to use fuzzy logic to distribute data in different clusters, and then we try to introduce the user the most appropriate threshold automatically. The results show that this approach causes the any rule which can be interesting will not be lost and also any rule that is useless cannot be extracted.

Keywords—Apriori Algorithm, Association Rules, Frequent Patterns, Support, Fuzzy Logic & C-Meance Clustering

I. INTRODUCTION

Knowledge discovery is the nontrivial extraction of implicit, previously unknown, and potentially useful information from data [1]. In order to get this information, we try to find patterns in the given data set. Patterns that are interesting and certain enough according to the user's measures are called knowledge [1]. The output of a program that discovers such useful patterns is called discovered knowledge. Data mining (DM) is a sub process of Knowledge Discovery in Databases in which the different available data sources are analyzed using various data mining algorithms [2]. Data mining is a logical process that is used to search the relevant data from the large amounts of information or data. The goal of this technique is to find patterns that were previously unknown [3]. Association rules mining is one of the most important tasks used in DM, which can be applied in different domains. Association rule discovery has been widely studied throughout the state-of-the-art techniques [4]. Many data mining researchers had improved upon the quality of association rule for business development by incorporating influential factors like utility, number of items sold and for the mining of association data patterns [5]. In this area three algorithms have been considered as the basic algorithms, these algorithms Apriori and FP-Growth and Eclat are called which in this paper we will describe the three algorithms [6]. We believe that the field of mining frequent patterns and association rules mining is still a research area has raised interest among researchers because the researchers are working to provide effective and efficient methods.

1.2 Problem Statement

A problem of classical association rules is that not every kind of data can be used for mining. Rules can only be derived from data containing binary data, where an item either exists in a transaction or it does not exist [7]. These types of calculations are in the crisp sets category, where expression is strictly one item is available in a transaction or not. To overcome this problem, the approach of fuzzy association rules has been developed. It allows the intervals to overlap making the fuzzy set instead of crisp set. Items can then show a partial membership to more than one set, overcoming the above addressed, so-called "sharp boundary problem" [8]. The idea is to cover the weakness crisp sets. There are also other challenges related to the Apriori algorithm and other algorithms in the field of association rules mining is that These algorithms are based on the assumption that users can specify the threshold: minimum-support (minsup). It is impossible that users give a suitable minimum-support for a database to be mined if the users are without knowledge concerning the database. Our aim in this paper is that transactions that are stored in the database as the crisp enter to a fuzzy environment of the so-called fuzzification. The paper is organized as follows: section 2 contains the general model for association rules and an overview of related approaches to the discovery of association rules. We present an efficient fuzzy association rules mining algorithm based on our proposed approach in section 3, then in section 4 we implement a case study based on the proposed approach. Evaluate the effectiveness of the proposed approach experimentally perform in Section 5. Finally, we Conclusions and recommendations in section 6.

II. RESEARCH BACKGROUND

As many of the databases are very large containing a huge number of tuples and attributes, efficient automated tools are necessary for acquiring useful information. Therefore, many data mining tools have been developed that allow a great variety of analysis techniques, mostly derived from classical statistics [9]. Since its introduction in [10], the technique of association rules mining has received great interest by the data mining community and a lot of research has been done resulting in the development of many different algorithms. Association rules are especially useful for conducting a market basket analysis where transaction data can be analyzed. Regularities in data of a supermarket for example can be found in this way. An association rule could be "If a customer buys bread and milk, he will mostly buy butter as well". This information is very useful for business because promotion actions can be designed accordingly [11].

2.1 Theoretical Background

Mining association rules from the database principles that all researchers who are active in this area, according to these principles, try to improve the algorithms. The following will describe the principles.

Association Rules: We state the problem of mining association rules as follows: $I = \{i_1, i_2, \ldots, i_m\}$ is a set of items, $T = \{t_1, t_2, \ldots, t_n\}$ is a set of transactions, each of which contains items of the itemset I. Thus, each transaction t_i is a set of items such that $t_i \subseteq I$. An association rule is an implication of the form: $X \rightarrow Y$, where $X \subset I, Y \subset I$ and $X \cap Y = \phi$. X and Y is a set of items, called itemset [2]. An example for a simple association rule would be $Bread \rightarrow Cheese$. This rule says that if bread was in a transaction, butter was in most cases in that transaction too. In other words, people who buy bread often buy butter as well. Such a rule is based on observations of the customer behavior and is a result from the data stored in transaction databases [10]. Looking at an association rule of the form $X \rightarrow Y$, X would be called the antecedent, Y the consequent. It is obvious that the value of the antecedent implies the value of the consequent. The antecedent, also called the “left hand side” of a rule, can consist either of a single item or of a whole set of items. This applies for the consequent, also called the “right hand side”, as well.

Frequent Itemset: An itemset whose support is greater than or equal to a minimum-support threshold is known as frequent itemsets. In many situations, we only care about association rules involving sets of items that appear frequently in baskets. For example, we cannot run a good marketing strategy involving items that no one can buys, thus data mining starts with the assumption that we only care about sets of items with high support; i.e., they appear together in many baskets. Then find the association rules only involving a high-support set of items i.e., $\{X_1, X_2, \ldots, X_n, Y\}$ must appear in at least a certain percent of the baskets, called the support threshold [12]. Support and confidence are the two most important quality measures for evaluating the interestingness of a rule.

Support: The support of the rule $X \rightarrow Y$ is the percentage of transactions in T that contain $X \cap Y$. It determines how frequent the rule is applicable to the transaction set T. The support of a rule is represented by the formula (1).

$$\sup(X \rightarrow Y) = \frac{X \cup Y}{N} \tag{1}$$

Where $X \cup Y$ is the number of transactions that contain all the items of the rule and N is the total number of transactions. The support is a useful measure to determine whether a set of items occurs frequently in a database or not. Rules covering only a few transactions might not be valuable to the business. The above presented formula computes the relative support value, but there also exists an absolute support. It works similarly but simply counts the number of transactions where the tested itemset occurs without dividing it through the number of tuples [6].

Confidence: The confidence of a rule describes the percentage of transactions containing X which also contain Y. formula (2)

$$conf(X \Rightarrow Y) = P(Y|X) = \frac{\sup(X \rightarrow Y)}{\sup(X)} \tag{2}$$

This is a very important measure to determine whether a rule is interesting or not. It looks at all transactions which contain a certain item or itemset defined by the antecedent of the rule. Then, it computes the percentage of the transactions also including all the items contained in the consequent [6].

2.1.1 The Process

The process of mining association rules consists of two main parts. First, we have to identify all the itemsets contained in the data that are adequate for mining association rules. These combinations have to show at least a certain frequency to be worth mining and are thus called frequent itemsets. The second step will generate rules out of the discovered frequent itemsets.

2.1.1.1 Mining Frequent Patterns

Mining frequent patterns from a given dataset is not a trivial task. All sets of items that occur at least as frequently as a user-specified minimum-support have to be identified at this step. An important issue is the computation time because when it comes to large databases there might be a lot of possible itemsets all of which need to be evaluated. Different algorithms attempt to allow efficient discovery of frequent patterns [6]. Some of those will be presented in this paper.

2.1.1.2 Discovering Association Rules

After having generated all patterns that meet the minimum-support requirements, rules can be generated out of them. For doing so, a minimum-confidence has to be defined. The task is to generate all possible rules in the frequent itemsets and then compare their confidence value with the minimum-confidence (which is again defined by the user). All rules that meet this requirement are regarded as interesting.

2.2 Experimental Background

Several algorithms have been developed since the introduction of the idea of mining association rules. Those algorithms are attempts to improve the efficiency of frequent pattern and association rule discovery. In this area three algorithms have been considered as the basic algorithm, these algorithms Apriori and FP-Growth and Eclat are called which in this paper we will describe the three algorithms.

2.2.1Aoriori Algorithm

Agrawal et al, 1994. Proposed the well-known algorithm, Apriori [13], to mine large itemsets to find out the association rules among items.

2.2.1.1 Discovering Frequent Itemsets using Apriori Algorithm

Generation of frequent itemsets makes use of the fact that any subset of a frequent itemset must as well be frequent. The number of items contained in an itemset is called its size; an itemset of size k is called a k -itemset. Within the itemset, the items are kept in lexicographic order. Each itemset has a count field associated with it, storing the support value. Firstly, the database is passed over in order to count the occurrences of single elements. If a single element has a support value that is below the defined minimum-support, It does not have to be considered anymore because it hence can never be part of a frequent itemset. A subsequent pass k consists of two phases:

1. The discovered large itemsets of pass $k-1$, i.e. the sets $Lk-1$, are used to generate the candidate itemsets, Ck for the current pass.
2. The database is scanned once more in order to determine the support for the candidate itemsets Ck. If the support is above the minimum-support, the candidates will be added to the frequent itemsets. Discovering the right candidates is crucial in order to prevent a long counting duration [13].

2.2.1.2 Discovering Association Rules using Apriori Algorithm

As stated before, association rules are allowed to have multiple elements in the antecedent as well as in the consequent. Only frequent itemsets are used to generate the association rules. The procedure starts with finding all possible subsets of the frequent itemset L. For each of those subsets, a rule is setup in the form $A \rightarrow (L-A)$. If the confidence of the rule is as least as big as the user-defined minimum-confidence, the rule is considered to be interesting. All subsets of L are explored in order not to miss any possible dependencies. But, if a subset A of L does not generate an interesting rule, the subsets of A do not have to be explored [13].

2.2.2 FP-Growth Algorithm

Han and Pei [14] proposed a novel frequent pattern tree (FP-Tree) structure, which contains all the compact information for mining frequent itemsets, and then proposed the FP-Growth, which adopts a pattern segment growth approach to prevent generating a large number of candidate itemsets. Their mining method only scans the whole database twice and does not need to generate candidate itemsets, and so is very efficient. For a faster execution, the data should be preprocessed before applying the algorithm.

2.2.2.1 Preprocessing the Data

The FP-Growth algorithm needs the following preprocessing in order to be efficient: An initial scan over the dataset computes the support of the single items. As items that have themselves a support value below the minimum-support can never be part of a frequent itemset, they can be discarded from the transactions [15]. The remaining items are recombined so that they appear in a decreasing order with respect to their support. Each node of the FP-Tree consists of three fields [16]:

Item-Name: In this field, the name of the item that the node represents is stored.
Count: The field count represents the accumulated support of the node within the current path.
Node-Link: In order to build the structure of the tree, links have to be built between the nodes.
The field Node-Link stores the ancestor of the current node, and null if there is none. Having that done, mining the database is not necessary anymore, now the FP-Tree is used for mining. The support of an itemset can easily be determined by following the path and using the minimum value of count from the nodes.

2.2.2.2 Mining the FP-Tree using FP-Growth

The FP-Tree provides an efficient structure for mining, although the combinatorial problem of mining frequent patterns still has to be solved. For discovering all frequent itemsets, the FP-Growth algorithm takes a look at each level of depth of the tree starting from the bottom and generating all possible itemsets that include nodes in that specific level. After having mined the frequent patterns for every level, they are stored in the complete set of frequent patterns.

2.2.3 Eclat Algorithm

Both the Apriori and FP-Growth methods mine frequent patterns from a set of transactions in horizontal data format (i.e., {TID: itemset}), where TID is a transaction-id and itemset is the set of items bought in transaction TID. Alternatively, mining can also be performed with data presented in vertical data format (i.e., {item: TID_set}). Zaki, 2000, proposed Equivalence CLASS Transformation (Eclat) algorithm by exploring the vertical data format [17]. The first scan of the database builds the TID_set of each single item. Starting with a single item ($k = 1$), the frequent $(k+1)$-

itemsets grown from a previous k-itemset can be generated according to the Apriori property, with a depth-first computation order similar to FP-Growth [14]. The computation is done by intersection of the TID_sets of the frequent k-itemsets to compute the TID_sets of the corresponding (k+1)-itemsets. This process repeats, until no frequent itemsets or no candidate itemsets can be found [6].

III. RESEARCH METHODOLOGY

Always provide proper recommendations to users is dependent on the extraction of relevant and interesting rules from databases, on the other hand mining interesting rules is dependent to determining the appropriate threshold that is defined by the user. A great many algorithms and techniques have been designed for discovering association rules from data, such as [6]. These algorithms are Apriori-like, where the Apriori algorithm is based on the assumption that users can specify the threshold: minimum-support. However, it is almost impossible that users give a suitable minimum-support for a database to be mined if the users have not any knowledge concerning about the support of items in the database [18]. Because if the user introduce a small number for the threshold leading to the production of large volumes of rules is useless, on the other hand if the user introduce a large number for the threshold, It is possible to lose a lot of interesting rules. There is a big gap between the research conducted is that none of them would have been given a user may not be able to determine the appropriate threshold. Therefore it is essential that a proper threshold automatically computes the items in the database and presented to the user. This paper presents a fuzzy approach for the identification of association rules that are related to minimum-support and are in the database proposed. Also, using membership functions that are available in Fuzzy Logic (FL) [19], attempt to assign membership degrees to different items in the database, expected to achieve acceptable results than Apriori algorithm classical.

3.1 Fuzzification

Due to the problems that the article was in relation to crisp sets, to overcome these problems, crisp data should be defined as a fuzzy, so as not to lose any data. Hence, we use the method of fuzzy clustering, to be able to enter crisp data into fuzzy sets. Among the algorithms that have been there for fuzzy clustering, we will use the popular fuzzy clustering algorithm C – Means [20, 21]. The first algorithm, the user receives as input the crisp data, next the user wants to determine the number of clusters, and then distributes all the data into clusters with different membership degrees.

3.2 Extracting Fuzzy Association Rules

After the data were transferred to the fuzzy environment, begins for mining fuzzy association rules. The work done by the Apriori approach. As was previously discussed in mining association rules. Used two criteria to measure the called support and confidence. The following criteria are used in fuzzy environment.

3.2.1 Fuzzy Support

The first step is to determine the number of occurrences of each item in the fuzzy database. Total membership degree for each data item will calculate, and we introduce it as the rate of occurrence of each item. See formula (3):

$$fuzzysum = \sum_{i=1}^{n} \mu(x) \tag{3}$$

Then, to produce 2-itemset,we should do compare all the membership degree of every tow field, among them select the minimum, finally, we introduce a total minimum two fields as desired repetition rate. Then, to produce the 3-itemset will compare membership degree of every three field and among them select the minimum; finally, we introduce a total minimum three fields as desired repetition rate. It will continue as long as that not produced any new frequent datasets. Formula (4) it formally shows:

$$fuzzy\,\sup(A \to B) = \sum_{i=1}^{n} \min\left(f_A(x), f_B(x)\right) \tag{4}$$

As previously noted, only those items that are defined as frequent as they are greater than or equal minimum-support.

3.2.2 Fuzzy Confidence

After extracting fuzzy frequent patterns, we will attempt to generate fuzzy association rules; this requires the use of a fuzzy confidence measure. Rules that are able to pass through the filter will be introduced as an interesting association rules. Which calculate the fuzzy confidence, used the formula (5).

$$fuzzyconf\,(A \Rightarrow B) = \frac{\sum\left[fuzzy\,\sup(A \to B)\right]}{\sum(\min(A))} \tag{5}$$

At this stage first we calculate the amount of repetitions our view items (A, B), and we divide it on total minimum repetition rate right on our rule (A). This formula works exactly the same formula (2), The difference here is applied to fuzzy environment. That they can satisfy those rules as an interesting association rules are introduced.

3.3 calculating the fuzzy minimum-support using proposed approach

We try algorithm automatically minimum-support for fuzzy support introduction to user. When experts want a set of data with an index to compare. And there is no such index. Use of statistical techniques and trying to use a formula such as Average Middle, Variance, Standard Deviation, etc. To define an index this can compare the data with the index, and extracting the desired information.

Since there is in fact no appropriate index given that a minimum-minsup for fuzzy support defined, we also use statistical techniques and the formula contained in these domains, we use the formula average, to be able to define an appropriate threshold for mining frequent patterns, so the minimum threshold is not too low, leading to the production of large volumes of patterns to be useless. And not too high, causing the loss patterns may be useful. After the use of fuzzy c-means clustering approach, crisp data were transferred into a fuzzy environment, we use the formula (6) to get minimum-support threshold and then to introduce users.

$$fuzzy\min\sup(A,B)=\frac{\sum\left[sum\left(f_x(A),f_x(B)\right)\right]}{|T|} \tag{6}$$

IV. A CASE STUDY USING THE PROPOSED APPROACH

In this section we offer all phases of mining fuzzy association rules using the proposed approach for automatically determining the threshold and also with the use of fuzzy c-means clustering algorithm. For this case study, we use the database of a company distributing health products. Table 1 shows part do of the crisp data set the company's sales of some items.

TABLE 1
Part of the company distributing health products data set

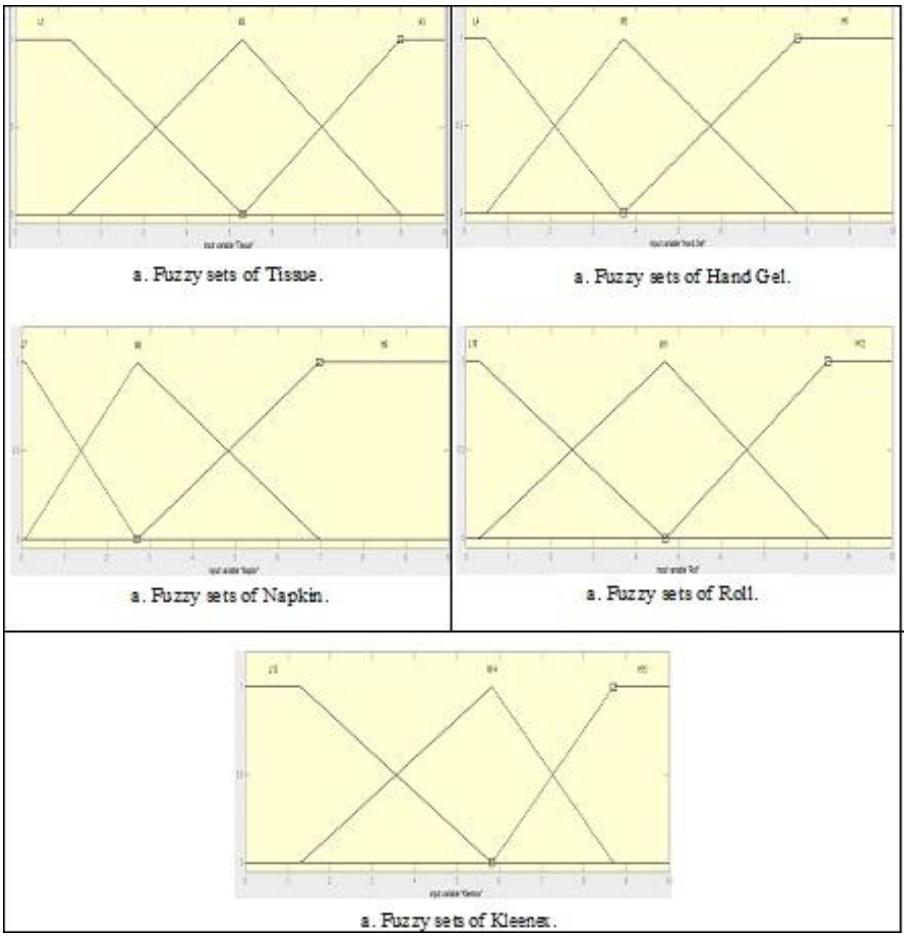
TID	Tissue	Hand Gel	Napkin	Roll	Kleenex	...
1	2	3	2	5	6	...
2	5	4	7	1	9	...
3	6	0	7	8	1	...
4	3	9	0	4	2	...
5	9	8	2	0	3	...
6	9	7	7	5	0	...
7	1	6	3	9	9	...
8	0	1	4	0	8	...
...	...	...	...	...	...	...

We need to transfer this data using fuzzy c-means clustering method inside fuzzy environment; we do this by using MATLAB software. We define for each item three clusters, namely: “Low {L}”, “Medium {M}” and “High {H}”. Table 2 shows the minimum, maximum and centre values of the fuzzy data.

TABLE 2
Centers of the data set with minimum and maximum values for each field

	Tissue	Hand Gel	Napkin	Roll	Kleenex
Minimum	0	0	0	0	0
Maximum	9	9	7	9	9
Center1	1.2357	0.5138	0.0805	0.3193	1.3106
Center2	5.3046	3.7102	2.7039	4.6852	5.8287
Center3	8.9862	7.7719	6.9779	8.5037	8.6995

Figure 1 represents each field and its membership functions (Note: all fields have three fuzzy classes including: Low (L), Medium (M) and High (H). For abbreviation, each fuzzy class (fuzzy set) is mapped into numbers, for example, Tissue.L→1, Tissue.M→2...).



a. Fuzzy sets of Tissue.

a. Fuzzy sets of Hand Gel.

a. Fuzzy sets of Napkin.

a. Fuzzy sets of Roll.

a. Fuzzy sets of Kleenex.

Figure 1 the membership functions for each field used in this case study

We use fuzzy clustering; data were distributed among the three clusters. Table 3 shows the distribution of some of the data within clusters.

TABLE3

Part of fuzzy data set for the company distributing health products

Tissue			Hand Gel			Napkin		
L {1}	M {2}	H {3}	L {4}	M {5}	H {6}	L {7}	M {8}	H {9}
0.939	0.05	0.011	0.074	0.906	0.02	0.117	0.866	0.017
0.007	0.989	0.004	0.007	0.987	0.006	0	0	1
0.02	0.93	0.05	0.977	0.019	0.004	0	0	1
0.598	0.35	0.052	0.02	0.05	0.93	0.999	0.001	0
0	0	1	0.001	0.003	0.996	0.117	0.866	0.017
0	0	1	0.013	0.052	0.935	0	0	1
0.996	0.003	0.001	0.061	0.352	0.587	0.01	0.984	0.006
0.932	0.051	0.017	0.964	0.031	0.005	0.084	0.77	0.146

From this point onwards that we used from Apriori algorithm for mining frequent patterns and generating association rules. According to the formula (6) fuzzy support threshold obtained for this case study has equal to 0.3333 (Fuzzy minimum-support=0.3333). How to calculate the fuzzy confidence are presented in Table 4.

TABLE 4
Part of fuzzy confidence calculating.

Frequent Patterns	Form (x→y)	Support (x,y)	Support (x)	Conf= Support (x,y)/ Support (x)
{1,2}	(1→2)	0.4804	3.4906	0.1376
{4,1}	(4→1)	1.1127	2.1165	0.5258
{4,2}	(4→2)	1.0599	2.1165	0.5008
{5,6}	(5→6)	0.4912	2.399	0.2048
{1,2,6}	(1,2→6)	0.3885	0.4804	0.8088
{1,6,2}	(1,6→2)	0.3885	1.22	0.3184
{1,6,5}	(1,6→5)	0.4369	1.22	0.3581
{5,6,1}	(5,6→1)	0.4369	0.4912	0.8895

Table 5 shows some association rules produced using the proposed approach where the measurement criteria have been considered as: Fuzzy minimum-support=0.3333 and Fuzzy minimum-confidence=0.5.

TABLE 5
Part of association rules produced using the proposed approach

Association Rules	Support (x,y)	Support (x)	Confidence (x→y)
(4→1)	1.1127	2.1165	0.5258
(5→1)	1.3639	2.399	0.5685
(4→2)	1.0599	2.1165	0.5008
(3→4)	1.1262	2.1378	0.5268
(4→3)	1.1262	2.1165	0.5321
(3→6)	2.0107	2.1378	0.9405
(6→3)	2.0107	3.4843	0.5771
(1,2→6)	0.3885	0.4804	0.8088
(2,6→1)	0.3885	0.3885	1

Then, this information can be expressed in a form that is understandable to users. For example:
If Hand Gel = Low Then Tissue = High, Confidence=0.5321.
If Tissue = Medium and Napkin=High Then Hand Gel = Low, Confidence= 0.8421.
If Napkin = Medium and Roll = High and Hand Gel = Low Then Kleenex = High, Confidence= 0.6565.

V. COMPARE THE PROPOSED APPROACH WITH BASIC ALGORITHMS

In this section we will do some comparisons between the proposed approach and the basic algorithms mentioned in this paper. Since we used fuzzy logic techniques and fuzzy clustering algorithm, we claim that all rules have been analyzed with more sensitive. Figure 2 shows the number of association rules generated by all four methods (With input and Minimum-support and Minimum-confidence identical).

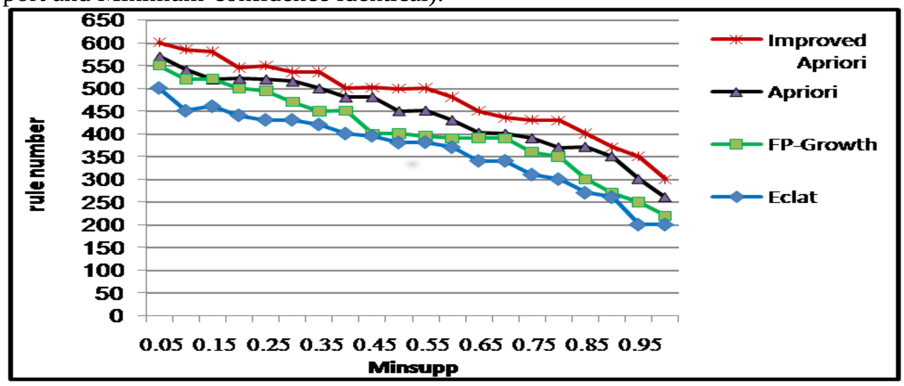


Figure 2 the number of association rules generated by all four methods

Also, since we minimum-support threshold determine automatically, we claim that will extract all appropriate frequent patterns. Figure 3 also illustrates a comparison between the number of frequent items generated using the proposed approach and the various thresholds.

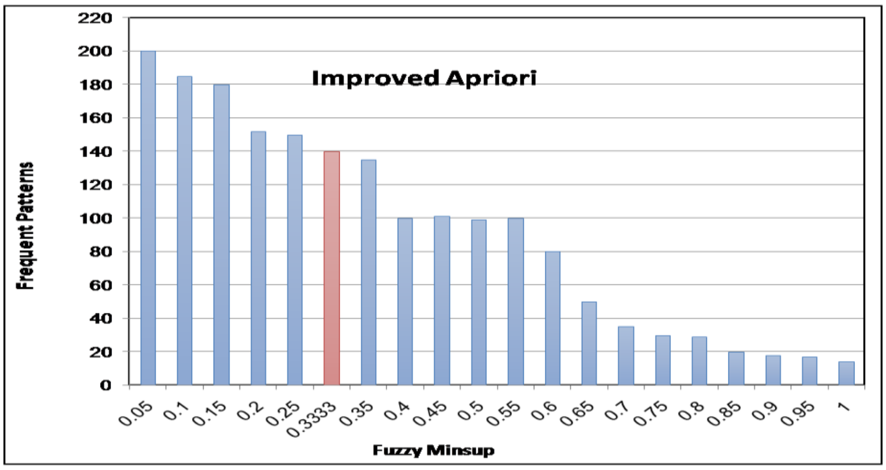


Figure 3 comparison between the numbers of frequent items generated using the proposed approach and the various threshold

VI. Conclusions and Recommendations

In this paper, we did a short overview on basic methods and algorithms in the fields of mining frequent patterns and generating association rules. Because according to the description given in this paper, a user cannot determine an appropriate threshold for mining association rules from a large database. We tried to improve Apriori algorithm by offering the proper threshold automatically based on existing items in the database and also utilizes the fuzzy logic techniques. We tried to through fuzzy logic techniques define the dependencies between items as membership functions, to be able to derive all interesting association rules that are hidden in the database. The results show that the Apriori algorithm is promising improved and all interesting association rules possible be extracted. If we can determine the number of clusters and interval clusters to be fully systematic and precise, sure, the distribution of items in the cluster can be obtained more accurately which leads to the production of the more interesting association rules. However, it is recommended that the use of other statistical techniques to determine the threshold of minimum-support, and the results achieved to compare with the approach proposed in this paper. As a final offer, it seems that determining minimum-confidence threshold automatically, also be useful in improving mining association rules.

References

[1] Frawley, W. J. & Piatetsky-Shapiro, G. & Matheus, Ch.J. (1992). Knowledge Discovery in Databases: an Overview. AAAI/MIT Press, 1992.

[2] Liu, B. (2007). Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data. Springer, 2007.

[3] Aggarwal, Sh. & Rani, B. (2013). Optimization of Association Rule Mining Process using Apriori and Ant Colony Optimization Algorithm. International Journal of Current Engineering and Technology 620 – 623.

[4] Kiran, R. & Reddy, P. (2010). Mining Rare Association Rules in the Datasets with Widely Varying Items Frequencies. In: Proceedings of 15th International Conference on Database Systems for Advanced Applications (DASFAA 2010) Tsukuba, Japan: Springer, pp. 49-62.

[5] Jnanamurthy, H.K. & Vishesh, H.V. & Vishruth, J. & Preetham, K. & Radhika, M.P. (2013). Discovery of Maximal Frequent Item Sets using Subset Creation, International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.3, No.1. 27-38.

[6] Han, J. & Cheng, H. & Xin, D. & Yan, X. (2007). Frequent pattern mining: current status and future Directions. Data Mining Knowledge Discovery. 15: 55-88.

[7] Zhang, M. & He, C. (2010). Survey on Association Rules Mining Algorithms. Advancing Computing, Communication, Control and Management, 111-118.

[8] Hong, T.P. & Chen, C.H. & Wu, Y.L. & Lee, Y.C. (2006). A GA-based Fuzzy Mining Approach to Achieve a Trade-off between Number of Rules and Suitability of Membership Functions. Soft Computing-A Fusion of Foundations, Methodologies and Applications, 10 (11), 1091-1101.

[9] Cakir, O. & Aras, M.E. (2012). A recommendation engine by using association rules. Procedia – social and Behavioral Sciences 62: 452 – 456.

[10] Agrawal, R. & Imielinski, T. & Swami, A. (1993). Mining Association Rules between Sets of Items in Large Databases. Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data.

[11] Tseng, V.Sh. & Wang, M-H. & Hwung, J. (2010). A New Method for Image Classification By Using Multi Level Association Rules. Institute of Computer Science and Information Engineering National Cheng Kung University 1 – 7.

[12] Miss, N.D. & Sheetal, D. (2013). Scalable and Efficient Improved Apriori Algorithm, International Journal of Innovative Research in Computer and Communication Engineering Vol. 1, Issue 2, 289-292.

[13] Agrawal, R. & Srikant, R. (1994). Fast algorithms for mining association rules. In: Proceedings of the 1994 international conference on very large data bases (VLDB'94), Santiago, Chile, pp 487–499

[14] Han, J. & Pei, J. & Yin, Y. (2000). Mining frequent patterns without candidate generation. In: Proceeding of the 2000 ACM-SIGMOD international conference on management of data (SIGMOD'00), Dallas, TX, pp 1–12.

[15] Borgelt, C. (2005). An Implementation of the FP-growth Algorithm. ACM Press, New York, NY, USA, 2005.

[16] Hipp, J. & Guentzer, U. & Nakhaeizadeh, Gh. (2000). Algorithms for Association Rule Mining - A General Survey and Comparison. ACM SIGKDD Explorations Newsletter, Volume 2, Issue 1, 2000.

[17] Zaki, M.J. (2000). Scalable algorithms for association mining. IEEE Trans Know l Data Eng 12: 372–390

[18] Shichao, Z. & Jingli, Lu. Chengqi, Z. (2003). A fuzzy logic based method to acquire user threshold of minimum-support for mining association rules, Information Sciences, 1-15, doi:10.1016/j.ins.2003.09.017

[19] Zadeh, L.A. (1965). Fuzzy sets Information and Control, (8) 3: 338-353.

[20] Bezdek, J.C. (1981). Pattern Recognition with Fuzzy Objective Function Algorithms. Kluwer Academic Publishers Norwell, MA, USA.

[21] Shihab, A.I. & Burger, P. (1998). The Analysis of Cardiac Velocity MR Images Using Fuzzy Clustering. In: Proceeding of SPIE Medical Imaging Physiology and Function from Multidimensional Images. Vol. 3337: pp. 176–183.

