ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 3, March 2015

An Efficient Fuzzy Association Rule Mining Technique

Gajendra Singh Rajput[1], Nayna Sharma[2], Shashank Swami[3]

M.Tech scholar, Computer Science & Engineering, VITM, Gwalior

M.Tech scholar, Computer Science & Engineering, VITM, Gwalior

Associate Professor, Computer Science & Engineering, VITM, Gwalior

Abstract-Association rules mining in large databases is a core topic of data mining. Discovering these associations is beneficial to the correct and appropriate decision made by decision makers. Discovering frequent item sets is the key process in association rule mining. One of the challenges in developing association rules mining algorithms is the extremely large number of rules generated which makes the algorithms inefficient and makes it difficult for the end users to comprehend the generated rules. In this paper we proposed efficient fuzzy association rule mining technique to find all co-occurrence relationships among data items. The proposed method which allows considerably reduced the search space with discover the frequent item set and finding fuzzy sets for quantitative attributes in a database and finally employs techniques for mining of Fuzzy Associate Rules Mining (FARM).

Keywords: Data Mining, Association Rule Mining, FP Mining, FARM**.**

INTRODUCTION

Data mining is the key step in the knowledge discovery process. The main tasks of Data mining are generally divided in two categories: Predictive and Descriptive. The objective of the predictive tasks is to predict the value of a particular attribute based on the values of other attributes, while for the descriptive ones, is to extract previously unknown and useful information such as patterns, associations, changes, anomalies and significant structures, from large databases. There are several techniques satisfying these objectives of data mining. The existing mining algorithms have some problems: the existing mining algorithms are mostly designed in forms of several passes so that the whole database needs to be read from disks several times for each user's query under the constraint that the whole database is too large to be stored in memory. This is very inefficient in considering the big overhead of reading the large database even though only partial items are interested

In fact. As a result, they cannot perform efficiently in terms of responding the user's query quickly. Secondly, in many cases, the algorithms generate an extremely large number of association rules, often in thousands or even millions. Further, the association rules are sometimes very large. It is nearly impossible for the end users to comprehend or validate such large number of complex association rules, thereby limiting the usefulness of the data mining results. Thirdly, no guiding information is provided for users to choose suitable settings for the constraints such as support and confidence such that an appropriate number of association rules are discovered. Consequently, the users have to use a try and -error approach to get suitable number of rules. It is very time consuming and inefficient. One of the main challenges in mining association rules is developing fast and the efficient algorithms that can handle large volumes of data set. We proposed in this paper an efficient fuzzy association rule mining technique to allows considerably reduced the search space with discover the frequent item set and finding fuzzy sets for quantitative attributes in a database and finally employs techniques for mining of fuzzy Associate Rules Mining (FARM).

BACKGROUND TECHNIQUES

Association Rule

Association rule mining provides a useful mechanism for discovering correlations among items belonging to customer transactions in a market basket database. Let D be the database of transactions and J = {J1, ..., Jn} be the set of items. A transaction T includes one or more items in J (i.e., $T \subseteq J$). An association rule has the form $X \Rightarrow Y$, where X and Y are non-empty sets of items (i.e. $X \subseteq J$, $Y \subseteq J$) such that $X \cap Y = \emptyset$. A set of items is called an itemset, while X is called the antecedent. The support sprtD(x) of an item (or itemset) x is the percentage of transactions from D in which that item or itemset occurs in the database. In other words, the support sprt () of an association rule $X \Rightarrow Y$ is the percentage of transactions T in a database where $X \cup Y \subseteq T$. The confidence or strength c for an association rule $X \Rightarrow Y$ is the ratio of the number of transactions that contain $X \cup Y$ to the number of transactions that contain X. An itemset $X \subseteq J$ is frequent if at least a fraction sprt() of the transaction in a database contains X. Frequent itemsets are important because they are the building blocks to obtain association rules with a given confidence and support [1-2].

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 3, March 2015

Fuzzy Logic

Fuzzy logic is a form of many-valued logic which deals with reasoning that is approximate rather than fixed and exact. Compared to traditional binary sets (where variables may take on true or false values), fuzzy logic variables may have a truth value that ranges in degree between 0 and 1. Fuzzy logic has been extended to handle the concept of partial truth, where the truth value may range between completely true and completely false. Furthermore, when linguistic variables are used, these degrees may be managed by specific functions. Any axiomatizable fuzzy theory is recursively enumerable. In particular, the fuzzy set of logically true formulas is recursively enumerable in spite of the fact that the crisp set of valid formulas is not recursively enumerable, in general. Moreover, any axiomatizable and complete theory is decidable.

It is an open question to give supports for a *Church thesis* for fuzzy mathematics the proposed notion of recursive enumerability for fuzzy subsets is the adequate one. To this aim, an extension of the notions of fuzzy grammar and fuzzy Turing machine should be necessary (see for example Wiedermann's paper). Another open question is to start from this notion to find an extension theorems to fuzzy logic [4-7].

Frequent Pattern growth algorithm

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 [8-10].

Related Works on Frequent Itemset Mining

The approach proposed by Chui et. al computes the expected support of itemsets by summing all itemset probabilities in their U-Apriori algorithm. Later, they additionally proposed a probabilistic filter in order to prune candidates early.
The UF-growth algorithm is proposed. Like U-Apriori, UF-growth computes frequent itemsets by means of the expected support, but it uses the FP-tree approach in order to avoid expensive candidate generation. In contrast to our probabilistic approach, itemsets are considered frequent if the expected support exceeds minSup. The main drawback of this estimator is that information about the uncertainty of the expected support is lost; ignore the number of possible worlds in which an itemset is frequent. Proposes exact and sampling-based algorithms to find likely frequent items in streaming probabilistic data. However, they do not consider itemsets with more than one item. The current state-of the art (and only) approach for probabilistic frequent itemset mining (PFIM) in uncertain databases was proposed. Their approach uses an Apriori-like algorithm to mine all probabilistic frequent itemsets and the poisson binomial recurrence to compute the support probability distribution function (SPDF).
A probabilistic database denotes a database composed of relations with uncertain tuples, where each tuple is associated with a probability denoting the likelihood that it exists in the relation. This model, called "tuple uncertainty", adopts the possible worlds semantics. A probabilistic database represents a set of possible "certain" database instances (worlds), where a database instance corresponds to a subset of uncertain tuples. Each instance (world) is associated with the probability that the world is "true". The probabilities reflect the probability distribution of all possible database instances.The approach proposed was the first approach able to solve probabilistic queries efficiently under tuple independency by means of dynamic programming techniques [1-2] and [4].

PROPOSED FUZZY ASSOCIATION RULE MINING TECHNIQUES

In this paper we proposed efficient fuzzy association rule mining technique to find all co-occurrence relationships among data items. The proposed method which allows considerably reduced the search space with discover the frequent item set and finding fuzzy sets for quantitative attributes in a database and finally employs techniques for mining of Fuzzy Associate Rules Mining (FARM).

Definitions:

- ***Support***

The rule $X \Rightarrow Y$ holds with support s if s% of transactions in D contains $X \cup Y$. Rules that have a s greater than a user-specified support is said to have minimum support.

- ***Confidence***

The rule $X \Rightarrow Y$ holds with confidence c if c% of the transactions in D that contain X also contain Y. Rules that have a c greater than a user-specified confidence is said to have minimum confidence.

- ***Itemset:*** An itemset is a set of items. A k-itemset is an itemset that contains k number of items.
- ***Frequent itemset:*** This is an itemset that has minimum support.
- ***Candidate set:*** This is the name given to a set of itemsets that require testing to see if they fit a certain requirement [1] and [5].

1. Discovering Frequent Itemsets using Apriori Algorithm

The proposed of our method is the Apriori algorithm. Our contributions are in providing novel scalable approaches for each building block. We start by counting the support of every item in the dataset and sort them in decreasing order of their frequencies. Next, we sort each transaction with respect to the frequency order of their items. We call this a horizontal sort. We also keep the generated candidate itemsets in horizontal sort.

Efficiently generating candidates

Let us consider generating candidates of an arbitrarily chosen size, k + 1. We will assume that the frequent k-itemsets are sorted both horizontally and down word. The (k − 1) × (k − 1) technique generates candidate (k+1) itemsets by taking the union of frequent k-itemsets. If the first k−1 elements are identical for two distinct frequent k-itemsets, fi and fj , we call them near-equal and denote their near-equality by fi = fj . Then, classically, every frequent itemset fi is compared to every fj and the candidate fi ∪ fj is generated whenever fi = fj. However, our method needs only ever compare one frequent itemset, fi, to the one immediately following it, f_{i+1}.

A crucial observation is that near-equality is transitive because the equality of individual items is transitive. So, if $f_i = f_{i+1}, \dots, f_{i+m-2} = f_{i+m-1}$ then we know that (∀ j, k) < m, $f_{i+j} = f_{i+k}$.

Recall also that the frequent k-itemsets are fully sorted (that is, both horizontally and down word), so all those that are near-equal appear contiguously. This sorting taken together with the transitivity of near-equality is what our method exploits.

In this way, we successfully generate all the candidates with a single pass over the list of frequent k-itemsets as opposed to the classical nested-loop approach. Strictly speaking, it might seem that our processing of $\binom{m}{2}$ candidates effectively causes extra passes, but it can be shown using the A Priori Principle that m is typically much less than the number of frequent itemsets. First, it remains to be shown that our one pass does not miss any potential candidates. Consider some candidate c = $\{i_a, \dots, i_k\}$. If it is a valid candidate, then by the A Priori Principle, fi = $\{i_1, \dots, i_{k-2}, i_{k-1}\}$ and $f_j = \{i_1, \dots, i_{k-2}, i_k\}$ are frequent. Then, because of the sort order that is required as a precondition, the only frequent itemsets that would appear between fi and fj are those that share the same (k − 2)-prefix as they do. The method described above merges together all pairs of frequent itemsets that appear contiguously with the same (k − 2)-prefix. Since this includes both f_i and f_j , c = $f_i \cup f_j$ must have been discovered.

2. Discovering Fuzzy Sets using Clustering

The traditional way to discover the fuzzy sets needed for a certain data set is to consult a domain expert who will define the sets and their membership functions. This requires access to domain knowledge which can be difficult or expensive to acquire. In order to make an automatic discovery of fuzzy sets possible, an approach has been developed which generates fuzzy sets automatically by clustering. This method can be used to divide quantitative attributes into fuzzy sets, which deals with the problem that it is not always easy do define the sets a priori.

The proposed method uses a known clustering algorithm to find the medoids of *k* clusters. The whole process of automatically discovering fuzzy sets can be subdivided into four steps:

1. Transform the database to make clustering possible (the value of all the attributes has to be positive integer).
2. Find the *k* medoids of the transformed database using a clustering method.
3. For each quantitative attribute, fuzzy sets are constructed using the medoids.
4. Generate the associated membership functions.

After discovering *k* medoids, we can compute *k* fuzzy sets out of them. We define {*m1* ,*m2* , ... ,*mk* } as the *k* medoids from a database. The *i* -th medoid can be defined as *mi*={*ai1* ,*ai2* , ... ,*ai n*} . If we want to discover the fuzzy sets for the *j* -th attribute, ranging from *min j* to *max j* , our mid-points will be {*ai1* , *ai2* ,... , *ai n*} . The fuzzy sets will then show the following ranges: $\{min_j - a2_j\}$, $\{a_{1j} – a_{3j}\}$, $\{a_{(i-1)j} - a_{(i+1)j}\}$, ... ,$\{ a_{(k-1)j} - max_j \}$. Finally, the membership functions for the fuzzy sets have to be computed.

We can get our membership function looking at the definition of the sets above. For the fuzzy set with mid-point *akj* , the membership function looks as follows: If $x \le a_{(k-1)j}$, the membership of *x* is 0. Also for $x \ge a_{(k-1)j}$, μ_x=0 because in both cases, the value lies outside the range of the fuzzy set. If *x* takes exactly the value of the mid-point a_{kj} , the membership is 1. For all other cases, we have to use a formula in order to compute the specific membership.

Generate membership functions (triangular function):

$$f_{ij}(x : \min{}_j, a_{\frac{k}{2}j}, \max{}_j) = \begin{cases} 1, & if\ a_{\frac{k}{2}j} = x \\ \dfrac{x - \min{}_j}{a_{\frac{k}{2}j} - \min{}_j}, & if\ \min{}_j \le x \le a_{\frac{k}{2}j} \\ \dfrac{\max{}_j - x}{\max{}_j - a_{(\frac{k}{2}+1)j}}, & if\ a_{(\frac{k}{2})j} \le x \le \max{}_j \\ 0, & ortherwise \end{cases}$$

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 3, March 2015

A distinction between two types of fuzzy sets has been introduced. These two types are called equal space fuzzy sets and equal data points fuzzy sets. Equal space fuzzy sets are symmetrical and all occupy the same range in the universal set. In contrary, equal data points fuzzy sets cover a certain number of instances and thus are not symmetrical.

3. Algorithm for Fuzzy Association Rule Mining

The algorithm first searches the database and returns the complete set containing all attributes of the database. In a second step, a transformed fuzzy database is created from the original one. The user has to define the sets to which the items in the original database will be mapped. After generating the candidate itemsets, the transformed database is scanned in order to evaluate the support and after comparing the support to the predefined minimum support, the items with a too low support are deleted. The frequent itemsets F_k will be created from the candidate itemesets C_k. New candidates are being generated from the old ones in a subsequent step. *Ck* is generated from C_{k-1} as described for the Apriori algorithm in step 1. The following pruning step deletes all itemsets of *Ck* if any of its subsets does not appear in C_{k-1}.

Candidate Pruning

The candidate generation so effective is its aggressive candidate pruning. We believe that this can be omitted entirely while still producing nearly the same set of candidates. Stated alternatively, after our particular method of candidate generation, there is little value in running a candidate pruning step.

In recent, the probability that a candidate is generated is shown to be largely dependent on its best testset that is, the least frequent of its subsets. Classical A Priori has a very effective candidate generation technique because if any itemset $c \setminus \{c_i\}$ for $0 \leq i \leq k$ is infrequent the candidate $c = \{c_0, \ldots, c_k\}$ is pruned from the search space. By the A Priori Principle, the best testset is guaranteed to be included among these. However, if one routinely picks the best testset when first generating the candidate, then the pruning phase is redundant.

In our method, on the other hand, we generate a candidate from two particular subsets, $fk = c \setminus \{c_k\}$ and $f_{k-1} = c \setminus \{c_{k-1}\}$. If either of these happens to be the best testset, then there is little added value in a candidate pruning phase that checks the other k−2 size k subsets of c. Because of our least-frequent-first sort order, f_0 and f_1 correspond exactly to the subsets missing the most frequent items of all those in c. We observed that usually either f_0 or f_1 is the best testset.

We are also not especially concerned about generating a few extra candidates, because they will be indexed and compressed and counted simultaneously with others, so if we do not retain a considerable number of prunable candidates by not pruning, then we do not do especially much extra work in counting them, anyway. Finally, the association rules are generated from the discovered frequent itemsets.

The Fuzzy mining Associate rules are composed of two steps:

1. Find all itemsets that have fuzzy support (FS<X,A>) above the user specified minimum support. These itemsets are called frequent itemsets.

2. Use the frequent itemsets to generate the desired rules. Let X and Y be frequent itemsets. We can determine if the rule X => Y holds by computing the fuzzy confidence FC<<X,A>,<Y,B>> and this value is larger than the user specified minimum confidence value.

RESULTS ANALYSIS

An illustrative of the proposed methodology with an example is given to understand well the concept of the proposed algorithm and how the process of the generating fuzzy association rule mining is performed step by step. The process is started from a given transactional database as shown :

Step-1:

Suppose that δ arbitrarily equals to 3; that means qualified transaction is regarded as a transaction with no more than 3 items purchased in the transaction. Result of this step is a set of qualified transaction as, where **M**={T1,T2,T3, T4,T5,T6 ,T7 ,T9}.

A Qualified Data Transaction (M)

Trans_ID	***List of Items***
T1	*I1, i2 , i5*
T2	*I2, i4*
T3	*I2, i3*
T4	*I1, i2, i4*
T5	*I1, i3*
T6	*I2, i3*
T7	*I1, i3*
T9	*I1, i2, i3*

Step-2:

The process is started by looking for support of 1-itemsets for which *k* is set equal to 1.

Step-3:

Since δ=3, then $k \in \{1,2,3\}$. It is arbitrarily given β1= β2 =0.5, β3=0.2. That means the system just considers support of *k*-itemsets that is greater than 0.5, for k=1,2, and greater than 0.2, for k=3.

Step-4:

Every *k*-itemset is represented as a fuzzy set on set of qualified transactions as given by the following results:

1-itemsets:

{*i*1}={0.31/T1, 0.31/T4, 0.5/T5, 0.5/T7, 0.31/T9},
{*i*2}={0.31/T1, 0.5/T2, 0.5/T3, 0.31/T4, 0.5/T6, 0.31/T9},
{*i*3}={0.5/T3, 0.5/T5, 0.5/T6, 0.5/T7, 0.31/T9},
{*i*4}={0.5/T2, 0.31/T4},
{*i*5}={0.31/T1}.

From Step-5 and Step-6, **{*i*5}** cannot be considered for further process because *support*(**{*i*5}**)< β1.

2-itemsets:

{*i*1, *i*2}={0.31/T1, 0.31/T4, 0.31/T9},
{*i*2, *i*4}={0.5/T2, 0.31/T4},

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 3, March 2015

{*i*2, *i*3}={0.5/T3, 0.5/T6, 0.31/T9},
{*i*1, *i*4}={0.31/T4},
{*i*1, *i*3}={0.5/T5, 0.5/T7,0.31/T9}.
From Step-5 and Step-6, **{*i*1, *i*4}** cannot be considered for further process because *support*(**{*i*1, *i*4}**)< β2.
3-itemsets:
{ *i*1,*i*2,*i*3}={0.31/T9},
Step-5:
Support of each k-itemset is calculate as given in the following results:
1-itemsets: 2-itemsets
support(**{*i*1}**) = 1.99, *support*(**{*i*1, *i*2}**)=0.99
support(**{*i*2}**) = 2.49, *support*(**{*i*2, *i*4}**)=0.83
support(**{*i*3}**) = 2.33, *support*(**{*i*2, *i*3}**)=1.33
support(**{*i*4}**) = 0.83, *support*(**{*i*1, *i*4}**)=0.31
support(**{*i*5}**) = 0.31, *support*(**{*i*1, *i*3}**)=1.33
3-itemsets:
support(**{ *i*1,*i*2,*i*3}**)=0.31
***L1*(β1=0.5)**
L1

1-itemsets	**Support**
{ *i1*}	1.99
{ *i2*}	2.49
{ *i3*}	2.33
{ *i4*}	0.83

Table 9. *L2* (β2=0.5)
L2

2-itemsets	**Support**
{ *i1* ,*i2*}	0.99
{ *i2* ,*i4*}	0.83
{ *i2* ,*i3*}	1.33
{ *i1* ,*i3*}	1.33

Table 10: *L2* (β3=0.2)
L2

2-itemsets	**Support**
{ *i1* ,*i2*,*i3*}	0.31

Step-6:
From the results as performed by Step-4 and 5, the sets of frequent 1-itemsets, 2-itemsets and 3-itemsets are given in Table 8, 9 and 10, respectively.
Step-7:
This step is just for increment the value of k in which if *k* > δ, then the process is going to Step-9.
Step-8:
This step is looking for possible/candidate *k*-itemsets from *Lk-1*. If there is no any more candidate k-itemset then go to Step-9. Otherwise, the process is going to Step-3.
Step-9:
The step is to calculate every confidence of each possible association rules as follows:

$Cf(i_1 => i_2) = [Support(\{i_1, i_2\})]/[Support(\{i_1\})] = 0.98/1.98 = 0.49$,
$Cf(i_2 => i_4) = [Support(\{i_2, i_4\})]/[Support(\{i_2\})] = 0.84/2.5 = 0.35$
$Cf(i_1 \wedge i_2 => i_3) = [Support(\{i_1, i_2, i_3\})]/[Support(\{i_1, i_2\})] = 0.35/0.98 = 0.35$
$Cf(i_1 \wedge i_2 => i_3) = [Support(\{i_1, i_2, i_3\})]/[Support(\{i_1\})] = 035/1.98 = 0.17$

Let a fuzzy association rule represents association between two *fuzzy* itemsets, *A* and *B*, where *A* and *B* are two fuzzy sets on set of items as given by μA={0.5/*i1*, 1/*i2*} and μB={1/*i2*, 0.5/*i3*}, respectively. Confidence of the fuzzy association rule is calculated by (10) as follows. First, from *A* and *B*, Φ*A* and Φ*B* can be determined by Φ*A* ={*i1*,*i2*} and Φ*B* ={*i2*,*i3*}, respectively. Implementation of the proposed algorithm had been partially experimented by developing a software where tested transactional database.

Performance Analysis
Quality Measures
This experiment shows how the new FARM approach gives more interesting rules than the previous one using ARM algorithm. The figure 1 shows the difference between the number of large item sets generated from the previous method and the new FARM approach using different fuzzy support values. Number of large item sets increases as the minimum support decreases. The figure shows the graph against fuzzy support and Frequent item sets. From the results, it is clear that the approach with normalization produces less frequent item sets (or even rules) than the converse. And this is because during the normalization process, the fuzzy degree of fuzzy sets is averaged thus making the data more dissimilar

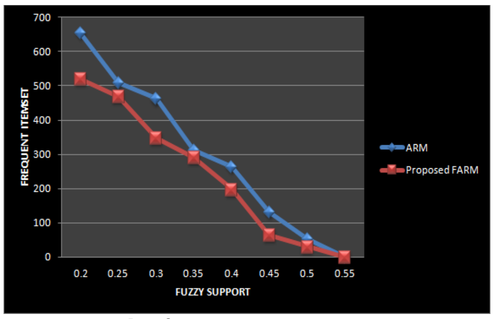

Figure 1: The frequent Item sets comparison

Figure2 shows the execution time of our proposed algorithm with different number of attributes. The graph is plotted against Number of Attributes and Execution time represented in seconds. Execution time increases as the number of attributes are increased simultaneously. The FARM algorithms have lesser timings while comparing with the ARM algorithm execution time, when the number of attributes is increased then the number of rules also increases with more attributes but fixed transactions.

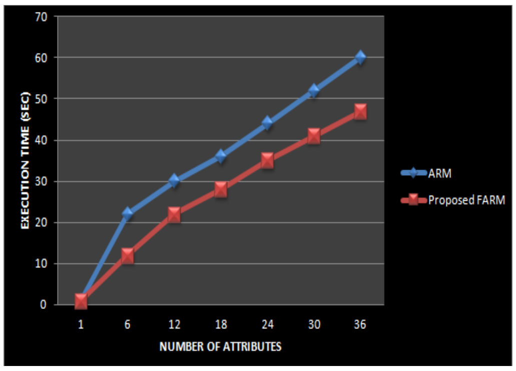

Figure 2: Performance Measures: Number of Attributes

CONCLUSION

This paper introduced an algorithm for generating fuzzy association rules mining. The algorithm is based on the concept that the larger number of items purchased in a transaction means the lower degree of association among the items in the transaction. Based on the concept, two new formulas of calculating degree of support and confidence were proposed utilizing the fuzzy set theory. The generalized formulas were also proposed in the relation to the fuzzy association rules. Finally, an illustrated example was given to clearly demonstrate and understand steps of the algorithm. The proposed approach is capable of making web recommendation more accurately and effectively against the conventional method. Combining the similarity between rules and active user and confidence of the weighted rules and the recommendation engine has selected only the most relevant pages. Therefore, it increases the efficiency of the recommendation engine. The simulation results show that the performance of proposed FARM outperforms the existing collaborative recommendation algorithm by means of time consumption, performance and quality.

REFERENCES

[1] Manish Saggar, Ashish Kumar Agarwal and Abhimunya Lad, "Optimization of Association Rule Mining using Improved Genetic Algorithms"IEEE 2004.
[2] Anandhavalli M, Suraj Kumar Sudhanshu, Ayush Kumar and Ghose M.K., "Optimized association rule mining using genetic algorithm", Advances in Information Mining, ISSN: 0975–3265, Volume 1, Issue 2, 2009, pp-01-04.
[3] Peter P. Wakabi-Waiswa and Dr. Venansius Baryamureeba, "Extraction of Interesting Association Rules Using Genetic Algorithms", Advances in Systems Modelling and ICT Applications, pp. 101-110.
[4] Farah Hanna AL-Zawaidah, Yosef Hasan Jbara and Marwan AL-Abed Abu-Zanona, "An Improved Algorithm for Mining Association Rules in Large Databases", World of Computer Science and Information Technology Journal (WCSIT) ISSN: 2221-0741 Vol. 1, No. 7, 2011, pp. 311-316.
[5] Rupali Haldulakar and Prof. Jitendra Agrawal, "Optimization of Association Rule Mining through Genetic Algorithm", International Journal on Computer Science and Engineering (IJCSE), Vol. 3 No. 3 Mar 2011, pp. 1252-1259.
[6] M. Ramesh Kumar and Dr. K. Iyakutti, "Genetic algorithms for the prioritization of Association Rules", IJCA Special Issue on "Artificial Intelligence Techniques - Novel Approaches & Practical Applications" AIT, 2011, pp. 35-38.
[7] C. C. Aggarwal, Y. Li, J. Wang, and J. Wang. Frequent pattern mining with uncertain data. In Proc. of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, 2009.
[8] P. Agrawal, O. Benjelloun, A. Das Sarma, C. Hayworth, S. Nabar, T. Sugihara, and J. Widom. "Trio: A system for data, uncertainty, and lineage". In Proc. Int. Conf. on Very Large Databases, 2006.
[9] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In Proc. ACM SIGMOD Int. Conf. on Management of Data, Minneapolis, MN, 1994.
[10] 1S. Dehuri,A. K. Jagadev, A. Ghosh and R. Mall, "Multi-objective Genetic Algorithm for Association Rule Mining Using a Homogeneous Dedicated Cluster of Workstations", American Journal of Applied Sciences 3 (11), 2006, pp. 2086-2095.

BIOGRAPHIES

Gajendra Singh Rajput[1] is the M.Tech Scholar in Department of Computer Science & Engineering from Vikrant institute of Technology & Management, Gwalior.
He has research interest's area in Data Mining and Network Security.

Nayna Sharma[2] is the M.Tech Scholar in Department of Computer Science & Engineering from Vikrant institute of Technology & Management, Gwalior.
She has research interest's area in Data Mining and Network Security.

Shashank Swami[3] is the Associate Professor in Department of Computer Science & Engineering from Vikrant institute of Technology & Management, Gwalior.
He has research interest's area in Data Mining and Network Security. He is the Guide of M. Tech Scholar.