IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 15, Issue 6 (Nov. - Dec. 2013), PP 01-08
www.iosrjournals.org

A Survey on Fuzzy Association Rule Mining Methodologies

Aritra Roy[1], Rajdeep Chatterjee[2]
[1]*(School of Computer Engineering, KIIT University, India)*
[2]*(School of Computer Engineering, KIIT University, India)*

Abstract : *Fuzzy association rule mining (Fuzzy ARM) uses fuzzy logic to generate interesting association rules. These association relationships can help in decision making for the solution of a given problem. Fuzzy ARM is a variant of classical association rule mining. Classical association rule mining uses the concept of crisp sets. Because of this reason classical association rule mining has several drawbacks. To overcome those drawbacks the concept of fuzzy association rule mining came. Today there is a huge number of different types of fuzzy association rule mining algorithms are present in research works and day by day these algorithms are getting better. But as the problem domain is also becoming more complex in nature, continuous research work is still going on. In this paper, we have studied several well-known methodologies and algorithms for fuzzy association rule mining. Four important methodologies are briefly discussed in this paper which will show the recent trends and future scope of research in the field of fuzzy association rule mining.*

Keywords: *Knowledge discovery in databases, Data mining, Fuzzy association rule mining, Classical association rule mining, Very large datasets, Minimum support, Cardinality, Certainty factor, Redundant rule, Equivalence , Equivalent rules*

I. INTRODUCTION

Rule mining is an aspect of data mining and also a process of Knowledge Discovery in Databases (KDD) in which the different available data sources are analyzed [1], [37].An expert system merges knowledge, facts and reasoning techniques in producing a decision. Expert system [2] has a knowledge base as a key component. Knowledge base stores domain specific knowledge in the form of rules or heuristic rules. Rules or heuristic rules explain procedures of reasoning used to solve a certain problem. From this aspect the concept of “Association Rule Mining” came. Association rule mining finds interesting association or correlation relationship among a large data set of items [3].

II. CLASSIFICATION OF ASSOCIATION RULE MINING ALGORITHMS

Association rule mining algorithms can be divided in two basic classes; these are BFS like algorithms and DFS like algorithms [4]. In case of BFS, at first the minimum support is determined for all itemsets in a specific level of depth, but in case of DFS, it descends the structure recursively through several depth levels. Both of these can be divided further in two sub classes; these are counting and intersecting. Apriori algorithm [5] falls under the counting subclass of BFS class algorithms. Apriori algorithm was the first attempt to mine association rules from a large dataset [6]. The algorithm can be used for both, finding frequent patterns and also deriving association rules from them. FP-Growth algorithm [7] falls under the counting subclass of DFS class algorithms. These two algorithms are the popular example of the classical association rule mining. Association rule mining can be classified as in the following figure,

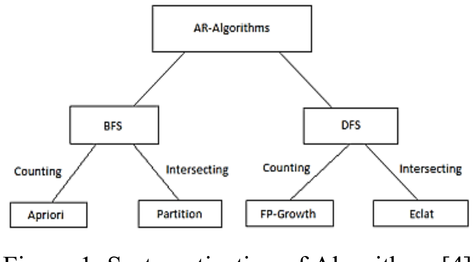


Figure 1: Systematization of Algorithms [4]

III. CLASSICAL ASSOCIATION RULE MINING AND FUZZY ASSOCIATION RULE MINING

Classical association rule mining depends on the Boolean logic to transform numerical attributes into boolean attributes by sharp partitioning of dataset. So, number of rules generated is low. It is computationally inefficient in terms of processing time, accuracy, prevention of redundant rule generation, resource requirement etc. It is inefficient in case of huge mining problems. In the classical association rule mining algorithms users have to specify the minimum support for the given dataset upon which the association rule mining algorithm will work. But it is very much possible that the user sets a wrong minimum support value which can hamper the generation of association rules. And also there is the fact that the setting of this minimum support is also not an easy task. If minimum support is set to a wrong value then there is a big possibility of combinatorial blow up of huge number of association rule within which many association rules will not be interesting.

Fuzzy association rule mining first began in the form of knowledge discovery in Fuzzy expert systems. A fuzzy expert system [8] uses a collection of fuzzy membership functions and rules, instead of Boolean logic, to reason about data [9]. The rules [10] in a fuzzy expert system are usually of a form similar to the following:

"If it is raining then put up your umbrella"

Here if part is the antecedent part and then part is the consequent part. This type of rules as a set helps in pointing towards any solution with in the solution set. But in case of Boolean logic every data attribute is measured only in terms of yes or no, in other words positive or negative. So it never allows us to have the diverse field of solutions. It always marginalizes the solutions; on the other hand fuzzy logic keeps broadways of solutions open for the users. There are many other fuzzy logic techniques which are in use in fuzzy association rule mining [38]. Wai-HO AU, Keith C.C. Chan [11] presented the first research work which introduces the fuzzy logic in the field of relational databases. After that there have been rapid developments in the field of fuzzy association rule mining. Fuzzy variants of Apriori and FP-Growth algorithms also can be classified into BFS and DFS type algorithms.

IV. Fuzzy Association Rule Mining Algorithms

In the last few decades there has been a large number of research work already done in the field of fuzzy association rule mining. The concept of fuzzy association rule mining approach generated from the necessity to efficiently mine quantitative data frequently present in databases. Algorithms for mining quantitative association rules have already been proposed in classical association rule mining. Dividing an attribute of data into sets covering certain ranges of values, engages the sharp boundary problem. To overcome this problem fuzzy logic has been introduced in association rule mining. But fuzzy association rule mining also have some problems. In these section four recent methodologies has been briefly discussed.

Ashish Mangalampalli, Vikram Pudi [12] represented the issues with classical association rule mining regarding the sharp partitioning. This is as follows,

- Use of sharp ranges creates the problem of uncertainty. More precisely loss of information happens at the boundaries of these ranges. Even at the small changes in determining these intervals may create very unfamiliar results which could be also wrong.
- These partitions do not have proper semantics attached with them.

In fuzzy association rule mining the transformation of numerical attributes into fuzzy attributes is done using the fuzzy logic concept. Attribute values are not represented by just 0 or 1. Here attribute values are represented with in a range between 0 and 1. By this way crisp binary attributes are converted to fuzzy attributes [13] and by the use of fuzzy logic we can easily resolve the above said problems. The algorithms which are already in use for fuzzy association rule mining, most of them are the fuzzy versions of Apriori algorithm. Apriori algorithm is slow and inefficient in case of large datasets. Fuzzy versions of Apriori algorithm would not be able to handle real-life huge datasets. Algorithms uses the principle of memory dependency like FP-Growth [14] and its fuzzy versions are inadequate to deal with huge datasets. But these huge data sets can be easily managed by the partial memory dependent variant algorithms like ARMOR [15] and [16].

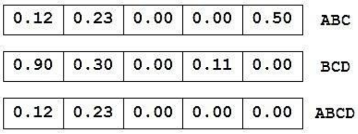
Ashish Mangalampalli, Vikram Pudi [12] proposed a new fuzzy association rule mining algorithm which will perform mining task efficiently and fast on huge datasets. Their proposed algorithm has two-step processing of dataset. But before the actual algorithm there is preprocessing of dataset by fuzzy c-means clustering [17], [18], [19]. Fuzzy partitions can be done on given data set so that every data point is a member of each and every cluster with a certain membership value. The aim of the algorithm is to minimize the Equ (1)

$$\sum_{i=1}^{N} \sum_{j=1}^{C} \mu_{ij}^{m} \left\| x_i - c_j \right\|^2 \qquad (1)$$

Where m is any real number such that 1 ≤ m < ∞, μ_{ij} is the degree of membership of $x_i$in the cluster of$j$, $x_i$is thei^{th}dimensional measured data,c_j is the d-dimensional cluster center, and$\|*\|$is any norm expressing the similarity between any measured data and the center. By this way corresponding fuzzy partitions of the dataset is generated where each value of numeric attributes are uniquely identified by their membership functions (μ). Depending upon the number of fuzzy partitions defined for an attribute, each and every existing crisp data is converted to multiple fuzzy data. This has the possibility of combinatorial explosion of generation of fuzzy records. So they have set a low threshold value for the membership function μ which is 0.1 to keep control over the generation of fuzzy records. During the fuzzy association rule mining process, the original data set is extended with attribute values within the range (0, 1) due to the large number of fuzzy partitions are being done on each of the quantitative attribute. To process this extended fuzzy dataset, some measures are needed which are based on t-norms [20], [21], [22]. In this way the fuzzy dataset E is created upon which the proposed algorithm will work. The dataset is logically divided into p disjoint horizontal partitions$P_1, P_2, \ldots., P_P$. Each partition is as large as it can fit in the available main memory. They have used the following notations,

- *E*=fuzzy dataset generated after pre-processing
- Set of partitions$P = \{P_1, P_2, \ldots., P_P\}$
- *td[it]* = tidlist of itemset *it*
- μ = fuzzy membership of any itemset
- *count[it]* = cumulative μ of itemset *it* over all partitions in which *it* has been processed
- *d* = number of partitions (for any particular itemset *it*) that have been processed since the partition in which *it* was added.

Byte-vector like data structure is used to represent fuzzy partitions of given data set. Each element of the byte-vector is nothing but the membership value (µ) of the itemset. In a transaction the byte-vector cell which does not contain any itemset, is assigned a value of 0. Initially the each byte-vector cell has the value of 0. Byte-vector representation of Tidlists is huge and could lead to incessant thrashing problem. They have used zlib compression algorithm to overcome this problem.



0.12	0.23	0.00	0.00	0.50	ABC
0.90	0.30	0.00	0.11	0.00	BCD
0.12	0.23	0.00	0.00	0.00	ABCD

Fig-2: Byte-vector representation of tidlist

In the step 1, algorithm scans every transaction present in the current partition of the dataset. And creates a tidlist for each singleton found. After getting all singletons from the current partition, algorithm keeps the d-frequent tidlists of singletons. The data structure count[s] maintains the count of each singleton s. Authors have incorporated BFS like technique to create larger itemsets. To generate a (*k*+1)-itemsetit_{k+1}, it needs to combine each *k*-itemset it_k with another *k*-itemset $it_k^{'}$, under the consideration of having a difference of just one singleton of two *k*-itemsets. The tidlist *td* $[it_{k+1}]$ for each (*k*+1)-itemset it_{k+1} is generated by intersecting the tidlists of its parent *k*-itemsets, *td* $[it_k]$ and td $[it_k^{'}]$ .If it_{k+1} is not *d*-frequent, then *td* $[it_{k+1}]$ is deleted. The data structure *count* $[it_{k+1}]$ maintains the count of each (*k*+1)-itemset. Tidlist is generated soon after the generation of an itemset. Tidlist can be deleted after the combination of it_k with all the other *k*-itemsets is done.

In the step 2, algorithm starts from the first partition and traverses each partition one-by-one. Itemsets which are enumerated in step 1, those can be discarded. Itemsets which are frequent throughout the entire dataset from the discarded itemsets, those are the result.After that algorithm identifies the singletons$S_1, S_2, \ldots, S_m$; for rest of the itemset *it*. Algorithm intersects the tidlists of each and every corresponding singletons to create the tidlist (*td[it]*) of *it*. Data structure *count [it]* maintains the count of every singleton *it*. Algorithm simultaneously generates outputs, discards itemsets and generates tidlists. This process continues up to each and every itemsets has been computed.

Experiments were performed in two different machine configurations. One with higher configuration and the other one with lesser configuration. In [12] authors have used the USCensus1990raw dataset which has 2.5 million transactions. And their fuzzy dataset can process 10 million transactions. So, it is obvious this amount of dataset cannot be handle in the available main memory. In their dataset 8 are quantitative and 4 are binary. Results obtained on higher configuration system on USCensus1990raw dataset as the experimental dataset, using multiple support values ranging between 0.075 to 0.400. Their proposed algorithm outperforms fuzzy Apriori algorithm in terms of speed. From the results of same experiment on lesser configuration system they got that their algorithm around 10 times faster than the fuzzy Apriori algorithm. From their experimental results they have shown that their proposed algorithm is fast and efficient than the previous fuzzy ARM algorithms.

Ehsan Vejdani Mahmoudi, Vahid Aghighi, Masood Niazi Torshiz, Mehrdad Jalali, Mahdi Yaghoobi [23] also pointed out the same difficulty that association rule mining is based on the assumption that users can specify the minimum-support for mining their databases.Han et al [24] have shown that setting of the minimum support is quite subtle work, which can hamper the applications of these mining algorithms. In [23] authors are using fuzzy logic to extract knowledge by calculating minimum support value for each item. More specifically, their algorithms automatically generate actual minimum-supports according to users' mining requirements.Discovering association rules at different levels may thus provide more information than the one at a single level [25].

In previous methods a unique high value is assigned to each of the items as their minimum support. Previous researchers used only a single concept level for extracting association rules. Later the use of multiple concept level gives us more important rules. The proposed fuzzy mining algorithm first encodes items (nodes) in a given taxonomy similar to Han and Fu's approach [25] which was capable of finding level crossing association rules at multiple levels and not confined to strict, pre-arranged conceptual hierarchies. This algorithm works on multiple non-uniform minimum support values. There are reasons behind non-uniform minimum support values.The proposed algorithm calculates the minimum support for each item by the Equ (2),

$$min\_Sup(I_i) = \frac{\sum_{i=1}^{n} S_i}{T*N*P} \qquad (2)$$

Let, I = {i1, i2… im} is a set of items and D = {t1, t2… tn} is a set of transactions. N is the total number of transactions. T is the number of occurrence of an item. S_i is sum of the values of an item in D. P belongs to the interval (0, 1).After that k (the level number) is set to 1 and grouping of the items with same first k digits in each transaction $D_i$and addition of the amounts of the items in the same group in D_i is done. $v_{ij}^k$is the amount of the$j$-th group I_j^k forD_i. For each groupI_j^k, v_{ij}^k (quantitative value) is then transformed into a fuzzy set f_{ij}^k which is represented as,

$$\frac{f_{ij1}^k}{R_{j1}^k} + \frac{f_{ij2}^k}{R_{j2}^k} + \cdots + \frac{f_{ijh}^k}{R_{jh}^k}$$

using specified membership functions in the previous section. Here$h$ is the number of fuzzy regions for I_j^k where I= l to n. R_{jl}^k is the l-th fuzzy region of I_j^k , $1<l < h$, and f_{ijl}^k is v_{ij}^k 's fuzzy membership value in region R_{jl}^k . After this algorithm collects the fuzzy regions (linguistic terms) with membership values larger than zero to form the candidate set C_1^k and calculates the Scalar Cardinality ($count_{ij}^k$) of each fuzzy region R_{jl}^k in the transaction data by the following Equ (3),

$$count_{ij}^k = \sum_{\mathrm{i}=1}^{\mathrm{n}} f_{ijl}^k \qquad (3)$$

Algorithm then checks either the value $count_{ij}^k$ of each region R_{jl}^k in C_1^k is larger than or equals to the thresholdτ_j^k , which is the minimum of minimum supports of the primitive items descending from it. If R_{jl}^k satisfies the threshold, put it into the large1-itemset L_1^k for level k. That is, $L_1^k = \{R_{jl}^k \mid count_{ij}^k \geq \tau_j^k, 1 \leq j \leq m^k\}$. Here if k reaches the level number of the taxonomy, then algorithm constructs the fuzzy association rules for all large q-itemset s containing regions $(S_1, S_2, \ldots, S_q)$, $q \geq 2$ otherwise the L_1^k is a null. Then algorithm sets $k = k + 1$ where k =1 and again starts from the grouping of the items. This algorithm works in an iterative way. It then discards unpromising itemsets by the count of a fuzzy region is checked to determine whether it is larger than support threshold value or not. And then it finds all the large itemsets of the given transactions by comparing the fuzzy count of each candidate itemset with its support threshold value. Algorithm works until it finds all possible large itemsets and association rule from them. Lastly the confidence values of association rules are calculated and the ones which have confidence value equal to or greater than the predefined confidence value are the final output of the algorithm.

In the experiment, used dataset [26] was with a total of 64 items and 10,000 transactions. The dataset contained quantitative transactions about the products sold in the chain store. The minimum confidence was set 0.5. Algorithm was tested with two previously proposed approaches: Mining Fuzzy Multiple-Level Association Rules from Quantitative Data [27] and Multi-level fuzzy mining with multiple minimum supports [28]. It remarkably produces more rules after 8000 transactions than the algorithms in [27] and [28]. From the test results it was found that their algorithm is efficient and effective.

Toshihiko Watanabe [29] has pointed out some drawbacks of fuzzy association rule mining. Those are as follows,

- Execution time for mining algorithm is high.
- Large number of redundant rule extracted as the result of mining process.

In [30] the proposed algorithm which incorporates memory based method significantly improved in terms of computational time. In contrary it consumes internal memory. In [29], author proposed an algorithm based on the utilization of specifications of output fields [31] and redundancy of the extracted rules. The proposed algorithm evaluates rules before calculating the minimum confidence.Let F= {P, Q, T} denotes the fuzzy itemset which consists of fuzzy sets in different attribute. Here P, Q, and T denote the fuzzy sets. Support of the itemset F is defined as:

$$s(F) = \frac{\sum \mu_F(x_p)}{m} \qquad (4)$$

Where x_p denotes the transaction of the database, $\mu_F(x_p)$ denotes the membership value calculated by the product operation (*t*-norm) of each item in *F*, and *m* denotes the total number of transactions in the database. From the support value, confidence of the fuzzy association rule $G \rightarrow H$ is calculated by:

$$c(G \rightarrow H) = \frac{s(G \cup H)}{s(G)} \qquad (5)$$

Where G and H are fuzzy itemsets. A fuzzy association rule is extracted when these values of a rule are more than predefined minimum confidence value. The itemset which have the value greater than the predefined threshold value is called "frequent itemset". The Apriori algorithm [6] is an effective method for finding the frequent itemsets. The key concept is that the frequent itemset should contain the subsets of frequent itemsets. Let *k*-itemset indicates *k* number of items in an itemset. Let $L_k$represent the set of frequent *k*-itemsets, and $C_k$the set of candidate *k*-itemsets. The algorithm follows the following steps,

A1. C_k is generated by joining the itemsets in L_{k-1} .

A2. The itemsets in $C_k$which have some *(k-1)*-subset that is not in L_{k-1} are deleted.

A3. The support of itemsets in C_k is calculated through database scan to decide L_k .

And after this the confidence of the derived association rules are calculated by the previous equation of confidence. Those who will be having confidence value equal or greater than predefined minimum confidence value will be the final output. Author has proposed a little modification of the Apriori algorithm after the step A3. He suggested the pruning of the redundant itemsets and then the confidence calculation. Author suggested the importance redundancy measure of extracted rules. In this paper it is done by the "Certainty Factor" [32], [33], [34]. The certainty factor (CF) of the rule $A \rightarrow C$ is defined as follows:

$$CF(A \rightarrow C) = \frac{Conf(A \rightarrow C) - supp(C)}{1 - supp(C)} \ ; \quad ifConf(A \rightarrow C) > supp(C)$$

$$CF(A \rightarrow C) = \frac{Conf(A \rightarrow C) - supp(C)}{supp(C)} \ ; \quad ifConf(A \rightarrow C) \leq supp(C)$$

$$CF(A \rightarrow C) = 1 \ ; \quad ifsupp(C) = 1$$

$$CF(A \rightarrow C) = -1 \ ; \quad ifsupp(C) = 0 \qquad (6)$$

There is also the definition of redundant rule and non-redundant rule. Let $X \rightarrow Y$ be a fuzzy association rule, where *X* and *Y* are fuzzy itemsets. Let Q be= $2^x - X - \emptyset$, where 2^x is the power set of *X* and $\emptyset$ is the empty set.

Ifmax$_{Z \in Q}\big(Conf(Z \rightarrow Y)\big) \geq Conf(X \rightarrow Y)$, then the rule $X \rightarrow Y$ is a redundant rule.

Ifmax$_{Z \in Q}\big(Conf(Z \rightarrow Y)\big) < Conf(X \rightarrow Y)$, then the rule $X \rightarrow Y$ is a non-redundant rule. (7)

From the above definition it also can be derived that,

Ifmax$_{Z \in Q}\big(Conf(Z \rightarrow Y)\big) \geq Conf(X \rightarrow Y)$, then max$_{Z \in Q}\big(CF(Z \rightarrow Y)\big) \geq CF(X \rightarrow Y)$

Non-redundant rules have always higher value of certainty factor than the redundant rules among the subset fuzzy association rules. To get the strong redundant rule we have another definition which is, let $X \rightarrow Y$ be a fuzzy association rule, where *X* and *Y* are fuzzy itemsets. Let Q be $Q = 2^x - X - \emptyset$

If min$_{Z \in Q}\big(Conf(Z \rightarrow Y)\big) \geq Conf(X \rightarrow Y)$, then the rule $X \rightarrow Y$ is a "strong redundant rule". (8)

Where, min$_{Z \in Q}\big(CF(Z \rightarrow Y)\big) \geq CF(X \rightarrow Y)$

And let *Z* be a fuzzy itemset that has *k* items (*k* > 2).When all the *k*-rules generated from *Z* are strong redundant, then the itemset *Z* is a "strong redundant itemset". Pruning uses heuristics to tell which combination of items is a redundant itemset. Both the extraction processes and the redundant rule pruning have been improved by specifying the output fields in advance.In addition to the basic Apriori algorithm, the following procedures are employed after the step A3 for *k*>1 as:

A4. The association rules are decided by calculating the confidence of *k*-rule based on $L_k$and L_{k-1} .

A5. The strong redundant *k*-itemsets in L_k are deleted.

The concept of this procedure is that the confidence value of the rule should increase by increasing the number of antecedent items. Expectation was the reduction of execution time and redundant rules pruning by the additional procedures.Method was tested on "abalone data" present in the UCI Machine Learning Repository [35]. Range of the fuzzy partition was set wider than the actual data distribution for the evaluation of performance. Authors have used two type of fuzzy membership function-one, Triangular MF; second, Trapezoidal MF. The benchmark data "abalone" has been used by the author to evaluate the performance of the proposed algorithm [29].

Toshihiko Watanabe, Ryosuke Fujioka [36] has tried to overcome the limitations of the [29]. This method is basically the follow up work of the [29].They has used the same extraction method of fuzzy association rule, same concept of measure of redundancy through the certainty factor [32], [33], [34]. They have used the same Apriori algorithm with a little modification in it at the end. To have better performance they have introduced an equivalence concept of fuzzy association rules. They described two definition of equivalence concept. Those are as follows,

Let *F*= $\{B_1, B_2, \ldots, B_m\}$ be a fuzzy itemset, where B is a fuzzy item (label) defined on a different attribute and m is the number of items (*m*> 1). Assume that the parameter q is defined larger than the minimal confidence and set in advance.

$$\text{If } Conf(\cup_{i \neq s} B_i \rightarrow B_s) \geq q \ , \ \forall s \in \{1,2, \ldots ., m\} \qquad (9)$$

Then we call the generated rules equivalent rules, the fuzzy itemset F equivalence itemset, and q the equivalence threshold.

Let *F*= $\{B_1, B_2, \ldots, B_m\}$ be an equivalence fuzzy itemset, $G = \{B_1, B_2, \ldots, B_n\}$ be a fuzzy itemset ($G \supset F$),where B is a fuzzy item (label) defined on a different attribute and m and $n$are the number of items (*m* >1, *n* >1, *n*>*m*). Assume that the parameter q is an equivalence threshold. Let R_F and R_G be generated association rules from *F* and *G* respectively as:

$$R_F: \cup_{i \neq s}^{m} B_i \rightarrow B_s, \exists s \in \{1,2, \ldots, m\}, R_G: \cup_{i \neq s}^{n} B_i \rightarrow B_s (10)$$

We define if $Conf(R_F) \geq q, Conf(R_G) \geq q$ then $R_G$is termed as “apparent rule”.Hence, the rules named as redundant rule, apparent rule, and omissible rule if they are generated from equivalent itemset. So, they are deleted in mining process. They delete the apparent rules at the level when k=2 to increase computational efficiency. After that they incorporated the equivalence concept in existing Apriori algorithm which leads to the performance improvement.

The algorithm was applied to the “abalone data” available at the UCI Machine Learning Repository [35]. This time also authors have used two different fuzzy membership functions. Many redundant rules are extracted along with the necessary non-redundant rules. Their objective was to prevent the generation of the redundant rules.The redundant rules or apparent rules are successfully deleted by the proposed algorithm based on the equivalence concept for fuzzy association rules mining.

V. COMPARATIVE ANALYSIS

The methodologies which have been described here, was chosen on the basis of various parameters like, mining strategy, algorithmic complexity, experimental statistics etc. All the algorithms discuss in this paper are different not only in terms of their nature but also in terms of working logic, execution time, accuracy etc. Experimental datasets are also different. All the test results are promising from the statistical point of view. But there are limitations. In [12], authors found that if the number of partitions for a dataset is increased then it leads to increase of the overall time required for processing. There is a chance of generating redundant association rules and useless association rules. In case of [23], though the execution time in the interval of 6000 to 8000 transactions is good than the other algorithms but the execution time is higher for the proposed method in the interval of 8000 to 10000 transactions as the number of association rules increases. Proposed algorithm generates fewer rules with in the interval of 6000 to 8000 transactions. There is also a chance of generating redundant association rules and useless association rules. In [29], author observed that along with redundant rules few non-redundant rules are also deleted for having their lower confidence value. Itemset extraction may be reduced due to multiple field specifications. The proposed algorithm shows minute improvement over its expectation in terms of computational time and efficiency. The test dataset was standard but not so huge. So it is not known that how the algorithm will work with the huge dataset where huge data mining process has to be done. In case of [36], the value of parameter q (equivalence threshold) is set to 0.8 by them so that they get this type of results. Any variation in this value can produce different type of results. In their test dataset there was only 8 quantitative attribute but now a days databases are getting bigger and bigger where the number of quantitative attributes can be very high. So, it is not clear that how the proposed algorithm will work in case of huge mining problems.

VI. Conclusion And Future Work

Knowledge extraction in databases is a way of extracting knowledge in the form of interesting rules. These rules are domain specific. These rules reveal the association relationship among different data’s, that how a particular data item is related to another data item. So, we call these rules as association rule. These rules are heuristic in nature. The process of extracting and managing these rules is known as association rule mining. Association rule mining is an important process in intelligent systems like Expert system. Because these intelligent systems solves domain specific problems. And this requires domain specific knowledge. Association rule mining is basically of two types. One is classical or crisp association rule mining and the other one is fuzzy association rule mining. Classical association rule mining uses Boolean logic to convert numerical attributes into binary attributes by the help of sharp crisp partitions. But the use of sharp partitions creates the problem of uncertainty. In which valuable data may become inconsistent over these sharp partitions. Another problem with classical association rule mining is, here a user has to provide a minimum support value for the mining purpose. And as we know that we humans are error prone. Any wrong setting of minimum support could end up in erroneous results. This can even cause the generation of huge number of redundant rules as well as useless rules. So, it is very a difficult task of setting an accurate minimum support value manually. That is why classical association rule mining is time consuming and less accurate process. Fuzzy association rule mining is relatively a newer concept. This uses the concepts of fuzzy set theory [13] for mining job. This survey paper represents a review of some of the existing fuzzy association rule mining methodologies [12], [23], [29], [36]. Algorithms are based on different concepts like, Fuzzy c-means clustering, multiple minimum supports, output field specification, certainty factor, equivalence redundancy of items etc. Paper [12] introduces the technique of fuzzy c-means clustering [17], [18], [19] for preprocessing. And the proposed algorithm uses the byte vector representation of datasets for the actual mining task. Algorithm works in two steps. Paper [23] represents the concept of calculating minimum supports for each item of given dataset. Their proposed algorithm works iteratively and works until it finds all possible interesting association rules. Paper [29] represents the fact that there can be redundant rule present in the generated association rules by the previous algorithms. So, the authors introduced the concept of Certainty Factor [32], [33], [34] for the redundant rule pruning. Their proposed algorithm is similar to the Apriori algorithm [5]. Just there is this pruning mechanism attached at the end of Apriori algorithm. Paper [36] is the follow up work of the paper [29]. Here authors introduced equivalence concept for the redundant rule pruning after the Apriori algorithm.

In case of paper [12], experimental results show that if the number of partitions are increased then there is the possibility of increase in execution time due to the increased number of itemsets. In case of paper [23] we can see that performance of the algorithm is good within a certain range of transactions. At the higher number of transactions algorithm does not works as efficiently as the authors expected. In case both the paper [12], [23] there is the chance of redundant rule generation. In case of paper [29], algorithm successfully rejects the redundant rules. But algorithm also has deleted some non-redundant rules. This is a big problem because some of those non-redundant could be some interesting rules. Paper [36] seems to be a little more efficient in terms of performance, accuracy and speed than the other three papers [12], [23], and [29]. But still it is unclear that how these algorithms will perform in case of large mining tasks. Day by day our mining tasks are becoming bigger and more complex. Either we have to improve the existing algorithms or we have to look for the alternatives like Rough set [39]. Through which we can perform association rule mining efficiently. Basic objective is to design an efficient algorithm which will execute huge mining task in suitable system configuration with less execution time and with higher accuracy. It is also a point of interest to research whether any heuristic search algorithm can be implemented in such cases. So, we believe this survey paper will reflect the current research trends in field of association rule mining.

REFERENCES

[1] Fayyad, Usama; Piatetsky-Shapiro, Gregory; Smyth, Padhraic: The KDD Process for Extracting Useful Knowledge from Volumes of Data. *Communications of the ACM, Volume 39, Issue 11,* Page(s): 27 – 34, 1996.

[2] http://www.umsl.edu/~joshik/msis480/chapt11.htm

[3] J. Han and M. Kamber, *Data Mining: Concepts and Techniques: The Morgan Kaufmann Series,* 2001.

[4] Hipp, Jochen; Guentzer, Ulrich; Nakhaeizadeh, Gholamreza: Algorithms for Association Rule Mining - A General Survey and Comparison. *ACM SIGKDD Explorations Newsletter, Volume 2, Issue 1,* 2000.

[5] Agrawal, Rakesh; Imielinski, Tomasz; Swami, Arun: Mining Association Rules between Sets of Items in Large Databases. *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data,* 1993.

[6] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases", *Proc. of the 20th International Conference on Very Large Data Bases, VLDB,* Page(s): 487-499, 1994.

[7] Han, J, Pei, J, Yin, Y: Mining Frequent Patterns without Candidate Generation. In: *SIGMOD Conference, ACM Press,* Page(s): 1-12, 2000.

[8] Türks¸en, I.B. and Tian Y. 1993. Combination of rules and their consequences in fuzzy expert systems, Fuzzy Sets and Systems, No. 58,3-40, 1993.

[9] http://www.cs.cmu.edu/Groups/AI/html/faqs/ai/fuzzy/part1/faq-doc-4.html

[10] L.A.Zadeh.Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on System, Man, and Cybernetics, Volume 3,* Pages(s):28-44, January, 1973.

[11] Wai-HO AU, Keith C.C. Chan: An Effective Algorithm for Discovering Fuzzy Rules in Relational Databases, *Fuzzy Systems Proceedings, IEEE World Congress on Computational Intelligence. Volume 2.*ISSN: 1098-7584, Print ISBN: 0-7803-4863-X, Page(s):1314 – 1319, 1998.

[12] Ashish Mangalampalli, Vikram Pudi: Fuzzy Association Rule Mining Algorithm for Fast and Efficient Performance on Very Large Datasets. *FUZZ-IEEE 2009,* Korea, ISSN: 1098-7584, E-ISBN: 978-1-4244-3597-5, Page(s): 1163 – 1168, August 20-24, 2009.

[13] Zadeh, L. A.: Fuzzy sets. Inf. Control, 8, Page(s): 338–358, 1965.

[14] Borgelt, Christian: An Implementation of the FP-growth Algorithm. *ACM Press,* New York, NY, USA, 2005.

[15] Pudi, V., Haritsa, J.: ARMOR: Association Rule Mining based on Oracle. *CEUR Workshop Proceedings,* 90, 2003.

[16] Savasere, A., Omiecinski, E., Navathe, S.B.: An Efficient Algorithm for Mining Association Rules in Large Databases. In: *VLDB, Morgan Kaufmann,* Page(s): 432-444, 1995.

[17] Dunn, J. C.: A Fuzzy Relative of the ISODATA Process and its Use in Detecting Compact Well Separated Clusters. J. *Cybernetics and Systems,Volume 3,*Page(s):32-57, 1974.

[18] Bezdek, J. C.: Pattern Recognition with Fuzzy Objective Function Algorithms. *Kluwer Academic Publishers, Norwell, MA,* 1981.

[19] Hoppner, F., Klawonn, F., Kruse, R, Runkler, T.: Fuzzy Cluster Analysis, Methods for Classification, Data Analysis and Image Recognition. *Wiley,* New York, 1999.

[20] De Cock, M., Cornelis, C., Kerre, E.E.: Fuzzy Association Rules: A Two-Sided Approach. In: *FIP,* Page(s): 385-390, 2003.

[21] Yan, P., Chen, G., Cornelis, C., De Cock, M., Kerre, E.E.: Mining Positive and Negative Fuzzy Association Rules. In: *KES, Springer,* Page(s): 270-276, 2004.

[22] Verlinde, H., De Cock, M., Boute, R.: Fuzzy Versus Quantitative Association Rules: A Fair Data-Driven Comparison. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics, Volume 36,* Page(s): 679-683, 2006.

[23] Ehsan Vejdani Mahmoudi, Vahid Aghighi, Masood Niazi Torshiz, Mehrdad Jalali, Mahdi Yaghoobi: Mining generalized fuzzy association rules via determining minimum supports ,*IEEE Iranian Conference on Electrical Engineering (ICEE)2011,* E-ISBN :978-964-463-428-4 ,Print ISBN:978-1-4577-0730-8,Page(s):1 – 6, 2011.

[24] J. Han, et al., "Mining top-k frequent closed patterns without minimum support, " *In Proceedings of the 2002 IEEE international conference on data mining,* Page(s): 211- 218, 2002.

[25] J. Han and Y. Fu, "Discovery of multiple-level association rules from large databases," in the *international conference on very large databases,* Zurich, Switzerland, Page(s): 420- 431, 1995.

[26] T. P. Hong, et al., "An ACS-based framework for fuzzy data mining," *Expert Systems with Applications, Volume 36,* Page(s): 11844-11852, Nov, 2009.

[27] T. P. Hong, et al., "Mining fuzzy multiple-level association rules from quantitative data," *Applied Intelligence, Volume 18,* Page(s): 79-90, Jan-Feb, 2003.

[28] Y. C. Lee, et al., "Multi-level fuzzy mining with multiple minimum supports," *Expert Systems with Applications, Volume 34,*Page(s): 459-468, Jan, 2008.

[29] Toshihiko Watanabe: Fuzzy Association Rules Mining Algorithm Based on Output Specification and Redundancy of Rules, *IEEE International Conference on Systems, Man, and Cybernetics (SMC) 2011,* ISSN: 1062-922X, Print ISBN: 978-1-4577-0652-3, Page(s):283 – 289, 2011.

[30] Y. C. Lee, T. P. Hong, and T. C. Wang, “Mining Fuzzy Multiple-level Association Rules under Multiple Minimum Supports,” *Proc. of the 2006 IEEE International Conference on Systems, Man, and Cybernetics,* Page(s): 4112-4117, 2006.

[31] T. Watanabe: “An Improvement of Fuzzy Association Rules Mining Algorithm Based on Redundancy of Rules,” *Proc. of the* 2^{nd} *International Symposium on Aware Computing,* Page(s): 68-73, 2010.

[32] M. Delgado, N. Marin, M. J. Martin-Bautista, D. Sanchez, and M.-A.Vila, “Mining Fuzzy Association Rules: An Overview,” *Studies in Fuzziness and Soft Computing, Springer, Volume 164/2005,* Page(s): 351-373, 2006.

[33] M. Delgado, N. Marin, D. Sanchez, and M.-A. Vila, “Fuzzy Association Rules: General Model and Applications,” *IEEE Trans. on Fuzzy Systems, Volume 11, No.2,* Page(s): 214-225, 2003.

[34] Y. Xu, Y. Li, and G. Shaw, “Concise Representations for Approximate Association Rules,” *Proc. of the 2008 IEEE International Conference on Systems, Man, and Cybernetics,* Page(s): 94-101, 2008.

[35] UCI Machine Learning Repository: http://www.ics.uci.edu/~mlearn/MLRepository.html

[36] Toshihiko WATANABE, Ryosuke Fujioka: Fuzzy Association Rules Mining Algorithm Based on Equivalence Redundancy of Items, *IEEE International Conference on Systems, Man, and Cybernetics (SMC),* 2012, E-ISBN: 978-1-4673-1712-2, Print ISBN: 978-1-4673-1713-9, Page(s):1960 – 1965, 2012.

[37] Frawley, William J.; Piatetsky-Shapiro, Gregory; Matheus, Christopher J.: Knowledge Discovery in Databases: an Overview. *AAAI/MIT Press,* 1992.

[38] Delgado, Miguel: Fuzzy Association Rules: an Overview. *BISC Conference,* 2003.

[39] Pawlak. Z. Rough Sets *International Journal of Computer and Information Sciences,* Page(s):341-356, 1982.

Mr. Aritra Roy is an M.Tech (CSE) student in KIIT University, Bhubaneswar, India. He received the B.Tech (IT) degree from Biju Patnaik University of Technology. His areas of interest include Applied Soft Computing, Informatics and Intelligent Systems.

Mr. Rajdeep Chatterjee is Assistant Professor in KIIT University, Bhubaneswar, India. He received the B.E (CSE) degree from University of Burdwan and M.Tech (CSE) from KIIT University. His research areas include Applied Soft Computing, Informatics and Intelligent Systems.