Support and Confidence Based Methods for Data Mining On Fuzzy Weighted Association Rules

Gagan kumar[1], Ramandeep Kaur[2]

[1]Assistant Professor, Modern Institute of engineering & technology
[2]Research scholar, Modern Institute of engineering & technology

***Abstract*— The problems of mining association rules in a database are introduced. Most of association rule mining approaches aim to mine association rules considering exact matches between items in transactions. A new algorithm called "Improved Data Mining Based on Semantic Similarity to mine new Association Rules" which considers not only exact matches between items, but also the semantic similarity between them. Improved Data Mining (IDM) Based on Semantic Similarity to mine new Association Rules uses the concepts of an expert to represent the similarity degree between items, and proposes a new way of obtaining support and confidence for the association rules containing these items. An association rule is for ex: i.e. for a grocery store say "30% of transactions that contain bread also contain butter; 2% of all transactions contain both of these items". Here 30% is called the confidence of the rule, and 2% the support of the rule and this rule is represented as Bread → Butter. The problem is to find all association rules that satisfy user-specified minimum support and minimum confidence constraints. This paper then results that new rules bring more information about the database.**

Keywords— Data mining, Semantic similarity, Association Rules, Support, Confidence, Fuzzy logic

I. INTRODUCTION TO DATA MINING

Data mining (DM), also known as knowledge discovery in databases (KDD), has been recognized as a new area for database research. This positive and evolutionary cycle is now occurring in area named data mining or knowledge discovery in database for efficiently discovering interesting rules from large collections of data. Informative knowledge discovering and new valuable data finding in database are very attractive in various business scenes.[1]

Data mining (DM), also called Knowledge-Discovery in Databases (KDD) or Knowledge-Discovery and Data Mining, is the process of automatically searching large volumes of data for patterns such as association rules. Data mining has been defined as "the nontrivial extraction of implicit, previously Unknown and potentially useful information from data "and "the science of extracting useful information from large data sets or databases"[1]. It involves sorting through large amounts of data and picking out relevant information. It is usually used by businesses and other organizations, but is increasingly used in the sciences to extract information from the enormous data sets generated by modern experimentation. Although data mining is a relatively new term, the technology is not. Companies for a long time have used powerful computers to sift through volumes of data such as supermarket scanner data, and produce market research reports. Continuous innovations in computer processing power, disk storage, and statistical software are dramatically increasing the accuracy and usefulness of analysis.

II. A CONCEPTUAL MODEL OF DATA MINING

Many useful studies have been done in data mining and knowledge discovery in database. By basing on the concept [3,7]that features the process aspects of data mining, we gives attention to the interaction between a human and a machine and the purpose clarification. Figure 1. Shows the conceptual model of data mining:

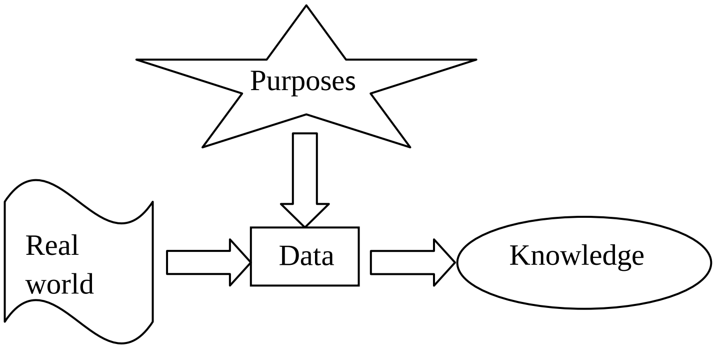

Figure1. Conceptual of data mining

A conceptual model of data mining is proposed by generalizing the actual application development process. Data mining is the process which extracts knowledge from real world environment according to a certain purpose. In this process, top-down and bottom-up approaches are performed as problem solving methods. The top-down approach clarifies purpose, defines problems to be solved, then breaks down the problems into elements until solvable level

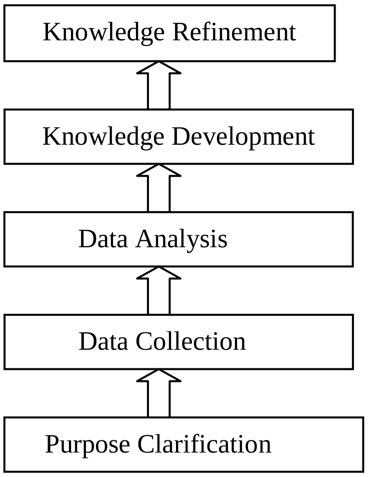

Figure2. Data Mining Process

On the other hand, the bottom-up approach collects data from the real world, analyzes them, and then integrates the findings. Both approaches are combined into data mining to find solvable goal, to select a suitable method for the goal and then to develop knowledge based on the method. The data mining process is shown in Figure 2. The steps below are the generalized data mining process. Before applying the process, we should define the benefits of developing target applications clearly to give the purpose.

1) Purpose Clarification: Clarifying the purpose, the problems to be solved, and the hypothetical goal of solution through the top-down approach.
2) Data Collection: Collecting data from the real world and visualizing them through the bottom-up approach.
3) Data Analysis: Analyzing the data collection to verify the hypothetical goal of solution through the combination of the top-down and the bottom-up approach.
4) Knowledge Development: Selecting a suitable method for the goal and developing knowledge based on the method.
5) Knowledge Refinement: Testing and refining the knowledge. If necessary, back to the previous steps.

III. RELATED WORK

Fuzzy sets theory has been applied a lot on data mining, specially on mining association rules. These association rules determine the quality of knowledge discovered, and the use of fuzzy sets. That is why so many approaches use fuzzy sets in conjunction with mining association rules. Most of these approaches are basically about enhancing the solution of two problems: mining quantitative association rules [11] and mining generalized association rules [9].Mining quantitative association rules involves specifying appropriate intervals for each attribute. Thus, instead of using intervals, some approaches employ linguistic terms. The definition of linguistic terms is based on fuzzy set theory and hence rules having these terms are called fuzzy association rules[17]. Furthermore, the intervals in quantitative association rules may not he concise and meaningful enough for human experts to obtain nontrivial knowledge. The proposed method needs user-defined fuzzy sets for describing association rules. Many algorithms have been proposed to find association rules mining in large databases. Most, such as the F-APACS and FARM are two of them, both created by Au and Chan[2]. Algorithms for mining quantitative association rules determine if a rule is interesting based on its support and confidence, which have to be greater than some user-supplied thresholds. A weakness of such approach is that many users do not have any idea what the thresholds should be. If it is set too high a user may miss some useful rules but if it is set too low, the user may be overwhelmed by many irrelevant ones. To solve this problem, F-APACS use adjusted difference analysis to determine which associations are interesting. The use of this technique has the advantage that it does not require any user-supplied thresholds. These algorithms also have the advantage that it allows us to discover both positive and negative association rules[5]. A positive association rule tells us that a record having certain attribute value will also have another attribute value whereas a negative association rule tells us that a record having certain attribute value will not have another attribute value. Gyenesei[16] introduce the problem of mining weighted quantitative association rules based on fuzzy approach.he assigns weights to the fuzzy sets to reflect their importance to the user and proposes two different definitions of weighted support: with and without normalization The authors extended the algorithm for mining generalized association rules, [4] extending the computation of support and confidence so that it could be applied in such a fuzzy context, and the algorithm was named Extended Algorithm.

IV. IMPROVED DATA MINING BASED ON FUZZY WEIGHTED ASSOCIATION RULES

Data Mining has been researched a lot due to its utility in many applications, and one of its most used tasks is Association Rule Mining. Given a set of transactions, where each transaction is a set of items, an association rule is an expression X => Y, where X and Y are sets of items (or item sets). The meaning of such a rule is that transactions which contain items in X tend to also contain items in Y. The support of the rule X => Y is the percentage of transactions that contain both X and Y. The confidence of the rule X => Y is the percentage of transactions containing X that also contain Y, an example of an association rule is “90% of transactions that contain bread also contain butter; 3% of all transactions contain both of these items.”The 90% is referred to as confidence and the 3%, the support of the rule. The problem of mining association rules is to find rules having minimum support and confidence. Many algorithms were developed to solve the problem of mining association rules. In general, new approaches were motivated by finding new ways of dealing with different attributes types or increasing computational performance. However, new approaches could address other issues. In this paper, we concern about semantics on mined data. Known algorithms only consider exact matches when mining frequent item sets, not generating some association rules which could bring important information.

In our approach, besides exact matches, the semantic similarity between items is also taken on account. For example, consider the set of transactions shown in Table 1.

TID	attribute1	attribute2
1	Chair	Table
2	Sofa	Desk
3	Chair	Desk
4	Chair	Table

TABLE1. A set of transaction examples

If this set of transactions were mined by a traditional association rule mining algorithm, the following association rules would be obtained:

Chair => table (support 50%, confidence 67%)
Sofa => desk (support 25%, confidence 100%)
Chair => desk (support 25%, confidence 33%)

Thus, if a minimum support of 50% and a minimum confidence of 60% were established, the only rule generated would be chair=> table. In this situation, only strings of characters are being considered, and as they have the same characters, with the same order and the same length, the mining algorithm will recognize a match. Table and desk, for example, are totally different words, but it does not mean they are totally different items. If we semantically analyze the words table and desk, we can consider them similar (both are furniture and have similar utilities, for example). In this case, there is not an exact match, but there is a kind of “similarity match”, which can be also useful to find relevant association rules and therefore important information. That is what traditional approaches can not reveal: association rules including semantically similar items. To make it possible, in this paper we present an algorithm called IDM.

V. ALGORITHM

A. *Semantic Similarity*

The objective of data mining is to discover knowledge, and that is why so many approaches try to make rules more understandable. Analyzing the meaning of mined data (i.e., the data semantics) naturally contributes to increase the quality of information obtained through the mining process, and consequently better the decisions guided by this information will be. Database transactions have different attribute types. The attributes can be quantitative or categorical[6] i.e. during the mining process, quantitative attributes cannot be semantically analyzed, but some categorical attributes can be. Known algorithms usually deal with categorical attributes as if they were mere character strings. These strings are recognized and counted along the transactions, and associations between them are found. In this case, matches occur only when strings have exactly the same characters, in the same order, with the same length. However, different strings can represent similar meanings. Consider, for instance, the words cupboard and wardrobe. Although character strings are totally different, they represent semantically similar words. Cupboard and wardrobe are different objects, but they both have shelves and doors and are used for storing things. They are not identical, but they are similar. This semantic similarity between items is ignored by traditional algorithms, what can make them lose important information. This additional analysis considering associations between similar items may reveal other association rules, which can be also relevant. We call semantically similar data mining the mining process which also considers the semantic similarity between data items, extending the usual way of mining association rules.

In this paper, we present a new algorithm called IDM. In IDM, the semantic similarity between data is expressed by a similarity degree between items. Thus, if the value of similarity degree between items is 1 (one), this means that compared items have maximum similarity. According to the reflexive property of binary fuzzy relations, it can only occur if an item is compared to itself. Therefore, when comparing two non-identical items, the similarity degree (sim) between them must be a value greater or equal to zero and less than one ($0 \leq \text{sim} < 1$). During the mining process, if the similarity degree between items is greater than a user-defined parameter, a semantic similarity association is detected, meaning that items contained in this association are similar enough (and therefore interesting to the user). Next section shows how IDM detects these semantic similarity associations and uses them to get important association rules.

1) *Algorithm Structure*

IDM is based on Apriori and, as an association rule mining algorithm, it needs user-provided minimum support and minimum confidence parameters to run. Moreover, by using fuzzy logic concepts, IDM also needs a user provided parameter which indicates the minimum similarity degree desired, called minsim. Thus, there are the following parameters:

• minsup, which indicates the minimum support;
• minconf, which represents the mininum confidence;
• minsim, which is the minimum similarity degree necessary to consider two items similar enough, and then associate them during mining.

All of these parameters are expressed by a real value in the interval [0, 1]. The steps performed by IDM are shown below

1. Data Scanning: Identifying items and their domains
2. Determining similarity degrees between items for each domain
3. Identifying similar items
4. Generating candidates
5. Calculating the weight of candidates
6. Evaluating candidates
7. Generating rules

Now, consider as an example a table containing transactions of buys from a furniture store (Table 2), where Tid is an identifier for each transaction, whereas Dom1, Dom2 and Dom3 contain items bought by the furniture store customers. Moreover, suppose henceforth that we have the following parameter values:

• minimum support (minsup) = 0.45
• minimum confidence (minconf) = 0.3
• minimum similarity (minsim) = 0.8

Tid	Dom1	Dom2	Dom3
10	Chair	Table	wardrobe
20	Sofa	Desk	cupboard
30	Seat	Table	wardrobe
40	Sofa	Desk	cupboard
50	Chair	Board	wardrobe
60	Chair	Board	cupboard
70	Chair	Desk	cupboard
80	Seat	Board	cabinet
90	Chair	Desk	Cabinet
100	Sofa	Desk	cupboard

TABLE2. Transactions of buys from a furniture store

2) *Data Scanning*

The first step is a data scanning that identifies items in the database. IDM identifies each item, generating 1-itemsets (itemsets with size one). Moreover, in this step each item is associated to a domain, which is important because they make possible to relate items according to their similarity only when is convenient — that is, if they belong to the same domain. When mining relational tables, domains can be defined by the column where the item is. Thus, considering the furniture store example, after data scanning we have items and domains identified, as shown in Table 3.

Items	Domain
sofa, chair, seat	Dom1
board, desk, table	Dom2
cabinet, cupboard, wardrobe	Dom3

Table3. Items and domains identified by data scanning

In this example, domain Dom1 contains items of furniture where one can sit, domain Dom2 contains items of furniture where one can place things on them, and domain Dom3 contains items of furniture where one can store things. Each domain contains items used in similar situations, what makes domains identification semantically coherent. The number of items belonging to domain determines its size. Thus, all domains in Table 3 have size 3.

3) *Determining Similarity Degrees*

After having items and their domains identified, it is time to determine the values of similarity relations within each domain. These values must be supplied by a domain specialist (usually the user himself). This task corresponds to one of the steps of KDD [3], prior to the step of data mining. Alternatively, it would be possible to obtain these values automatically, through a rule or method. However, to determine the similarity values between items so that the semantics is considered, it is necessary to adopt a way of reproducing, with high fidelity, the capacity of the human mind of doing this. Any rule chosen to determine these values automatically will consider non-semantic factors, decreasing the quality of the analysis realized and this way going against the objective of the semantically similar data mining, which is to enrich the analysis and consequently enrich the information obtained from the rules. In each domain, the similarity degree values are stored in a similarity matrix. In the furniture store example, 3 domains were identified, and the correspondent similarity matrices can be seen in Table 4. The values in the matrices inform the similarity degree between the items of the domain. For example, chair is 70% similar to sofa. Next subsection shows how each similarity matrix is consulted to identify similar items.

4) *Identifying Similar Items*

In this step, the similarity matrix of each domain is analyzed, thus identifying pairs of items whose similarity degree is greater than or equal to minsim. These pairs of items compose fuzzy associations of size 2. In IDM, these associations are expressed through fuzzy items,[2] which are representations where the ~ symbol is used to indicate the relation between items. Thus, supposing that the sufficiently similar items are item1 and item2, for example, a fuzzy item on the form item1~item2 represents the fuzzy association between them.

Dom1	sofa	seat	chair
Sofa	1	0.75	0.7
Seat	0.75	1	0.6
Chair	0.7	0.6	1

Dom2	desk	table	Board
desk	1	0.9	0.75
table	0.9	1	0.7
board	0.75	0.7	1

Dom3	cabinet	wardrobe	cupboard
cabinet	1	0.9	0.85
wardrobe	0.9	1	0.8
cupboard	0.85	0.8	1

Table4. Domains and their respective similarity matrices

In the furniture store example, the similarity matrices in Table 4are analyzed and, considering the minsim value (0.8), the associations shown in Table 5 are obtained.

Domain	Value	Similarity relation1	Equivalent fuzzy item
Dom2	0.9	sim(desk, table)	desk~table
Dom3	0.9	sim(cabinet, wardrobe)	cabinet~wardrobe
Dom3	0.85	sim(cabinet, cupboard)	cabinet~cupboard
Dom3	0.8	sim(cupboard, wardrobe)	cupboard~wardrobe

Table5. Similarity relations that satisfy minsim

After obtaining the set of fuzzy associations of size 2, the existence of similarity cycles is verified. A similarity cycle is a fuzzy association of size greater than 2 that only exists if all of its items are, in pairs, sufficiently similar. That is, according to the intersection operation in fuzzy set theory, the minimum value among the similarity degrees involved in the cycle must be greater than or equal to minsim. It is what occurs, in the furniture store example, with the cycle cabinet~wardrobe~cupboard, shown in Figure 3. In this figure, arrows represent the similarity relations between items, and near them are the values that express the relation values. Thus, it is possible to see that in this example all the items are, in pairs, sufficiently similar (0.9 ≥ 0.8, 0.85 ≥ 0.8 and 0.8 ≥ 0.8). Or else, it can be verified that the minimum value among the similarity degrees involved is greater than or equal to minsim (min(0.9, 0.85, 0.8) ≥ 0.8). Whereas the minimum size of a similarity cycle is 3, the maximum size is equal to the size of the analyzed domain.

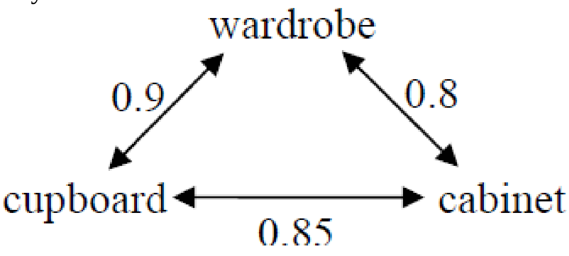

Figure3. Similarity cycle

```
A2 = { Set of fuzzy associations of size 2}
for (k = 3; k < size(Dn) ; k++)
    compare each pair of fuzzy associations in Ak-1
        if (prefix(ai) = prefix(aj), i ≠ j)
            //if suffixes union is sufficiently similar
            if ((suffix(ai) U suffix(aj)) ∈A2)
                ak = ai U suffix(aj)
            end if
        end if
    end compare
    Ak = {set of all ak}
end for
Sn = Group of all Ak
```

Figure4. Algorithm to find similarity cycles

Ak	Set of fuzzy associations of size *k*
Dn	Each one of the *n* domains analyzed
Ak	Each fuzzy association in *Ak* set
Sn	Set of similar items on domain *n*
size(*Dn*)	*Dn* size
prefix(*ak*)	*ak* fuzzy association prefix
suffix(*ak*)	*ak* fuzzy association suffix

Table6. Notation used in figure 4

In the furniture store example, *Dom3* size is 3, and that is why no fuzzy association of size greater than 3 can be obtained. However, for bigger domains, there can be cycles of bigger sizes. That is why IDM checks for the existence of similarity cycles iteratively on each domain, where the fuzzy association sets of size k-1 are analyzed on each step k (k ∈ N, k ≥ 3) to obtain fuzzy associations of size k. The notation sim(item1,item2) represents the similarity relation between item1 and item2.

A fuzzy association *ak* is of the form {*s1*, *s2*, ..., *sk-1*, *sk*}, where *s1*, *s2*, ..., *sk-1*, *sk* are the *k* items which composes it. Its suffix is on the form {*sk*}, whereas its prefix is on the form {*s1*, *s2*, ...,*sk-1*}. Every obtained *A k* in this step are grouped in *Sn*. This is how the step of identifying similar items ends, and then another iterative part of the algorithm begins. In this part, for each step *k* (*k* ∈ N), the *k*-item set candidates are generated from the frequent item sets obtained on previous step (*k*-1). Also, weights of *k* item set candidates are calculated and candidates are evaluated.

5) *Generating Candidates*

The way candidates are generated is very similar to the way it is done in Apriori. However, in IDM, besides items identified during the data scanning step, fuzzy items — which represent fuzzy associations obtained in the step of identifying similar items — also integrate the generated candidates. At the end of this step, we have the set of *k*-item set candidates, which is submitted to the step of calculating the weight of candidates.

6) *Calculating The Weight Of Candidates*

In this step, the weight of each item set candidate is calculated. The weight of an item set corresponds to the number of its occurrences in the database. In IDM, differently from what happens in A priori, an item set can have fuzzy items, hence called fuzzy item set. The notation item1~item2, has the following meaning: if item1 and item2 are very similar, they can be considered as being practically identical; thus, if occurrences of item1 or item2 are found in the database, they will be associated and, together with the similarity degree between items, they will compose a fuzzy occurrence of item1~item2. Therefore, we need to know if the item set is fuzzy or not, before calculating its weight: if the item set is not fuzzy, we calculate its weight in the conventional way, counting its exact occurrences; if the item set is fuzzy, we shall consider its fuzzy occurrences to obtain its weight. To understand how fuzzy occurrences happen, suppose that the similarity degree between item1 and item2 is 0.8. In this case, each occurrence of item2 in the database can be considered equal to 80% of item1 occurrence. Consequently, for each item1 occurrence we sum one item1 occurrence (of course), and for each item2 occurrence we sum 0.8 item1 occurrence (Table 7– situation A).

The problem can also be seen in the contrary manner, summing one item2 occurrence for each item2 occurrence and 0.8 for each item1 occurrence (Table 7– situation B). Notice that, for situation A, the fuzzy occurrences totalize the value of 2.8 (1.0 + 1.0 + 0.8), whereas for situation B fuzzy occurrences totalize the value of 2.6 (0.8 + 0.8 + 1.0).

Tid	Dom1	
10	item1	1.0
20	item1	1.0
30	item2	0.8

SITUATION A

Tid	Dom1	
10	item1	0.8
20	item1	0.8
30	item2	1.0

SITUATION B

Table7. Fuzzy Occurrences

Hence, depending on situation, the result obtained for the same similar items could be different. To avoid this distortion, it is necessary to balance this counting. To do that, consider weight(item1) as the number of item1 occurrences, weight(item2) as the number of item2 occurrences; and sim(item1,item2) as the similarity degree between item1 and item2. Thus, for situation A in Table 7, the number of occurrences is given by the expression.

$$\text{weight}(\text{item}_1\ ) + \text{weight}(\ \text{item}_{2)}\ \times \text{sim}(\ \text{item}_{1,}\ \text{item}_2)$$

In the same way, for situation B in Table 7, the number of occurrences is given by the expression.

$$\text{weight}(\text{item}_1\ ) \times \text{sim}(\ \text{item}_{1,}\ \text{item}_2) + \text{weight}(\ \text{item}_2)$$

We adopt the arithmetic average between situations A and B to balance the two situations, getting the fuzzy weight of item1~item2 through the Equation 1.

$$\text{Fuzzy Weight} = \frac{[\text{weight}(\text{item1}) + \text{weight}(\text{item2})][1 + \text{sim1}(\text{item1}, \text{item2})]}{2}$$

Equation1. Fuzzy weight for two similar items

Equation 1 is useful to calculate the weight of fuzzy items formed by an association of only two similar items. After this, itemset candidates are evaluated in the next step of IDM.

7) Evaluating Candidates

This is the step of IDM where the support of itemset candidates is evaluated, similar to what is done in Apriori. The support corresponds to the weight divided by the number of rows (or total of transactions) in the database (Equation 2). If the itemset candidate is fuzzy, its weight is also fuzzy, and then when its weight is divided by the number of rows, the result is its fuzzy support. Thus, generically, the support of each item set is calculated from its weight[14,15], and it is verified if its support is greater than or equal to minsup. In negative case, the item set is considered not frequent, and is therefore discarded. In positive case, the item set is stored in the set of frequent item sets.

$$\text{Support} = \frac{\text{weight(itemsets)}}{\text{number of rows in the database}}$$

Equation2. Support of the item set

The end of this step is also the end of the iterative part of IDM. At this time, all frequent item sets are grouped in a set, from which it is possible to start the step of generating rules.

8) Generating Rules

Association rules have antecedents (items left of arrow) and consequents (items right of arrow), as shown in Figure 5.

Antecedent →Consequent

Figure5. Antecedent and consequent of the rule

In this last step, for each item set from the set of frequent item sets, IDM generates all the possibilities of antecedents and consequents for the rules, using the confidence as choice criteria. If confidence, given by Equation 3, is greater than or equal to minconf, then rule is valid.

$$\text{Confidence} = \frac{\text{Support (rule)}}{\text{Support (anteecedent)}}$$

Equation3. Rule confidence

Regardless of supports being fuzzy or not, confidence is obtained in the same way. When IDM is concluded, all valid rules are exhibited, showing antecedent, consequent, support and confidence of each rule, in the format shown in Figure 6.

Anteecedent→ Consequent sup = < support value > conf = <confident value>

Figure6. Association Rule Format

In IDM, antecedents and consequents of the rule can contain fuzzy items, and the values of support and confidence reflect the influence of the similarity degree between items in their calculations.

VI. TESTS

We realized some tests to compare the results obtained with IDM and Apriori, using real data about furniture store. We started testing our first set of data, named FURNITURE STORE, containing transactions with the following attributes. There are semantic similarities in the domain and the similarity degrees between its items are shown in Table 8.These similarity values are manually decided.

Item1	Item2	Similarity
Chair	Sofa	70
Sofa	Seat	75
Desk	Table	90
Desk	Board	75
Table	Board	70
Cupboard	Wardrobe	90
Cupboard	Cabinet	85
Wardrobe	Cabinet	80

Table8. Similarity degrees for furniture store

We mined FURNITURE STORE using Apriori with parameters minsup = 40 and minconf = 40, obtaining the rules shown in Figure 8. We also mined FURNITURE STORE using IDM with the parameters minsup = 40, minconf = 40 and minsim = 80, obtaining the rules shown in Table 9.

Test with Apriori over the set FURNITURE STORE, with minsup = 40 and minconf = 40,Itemsets pair above minimum support and minimum confidence rule:
Rules generated Chair → Sofa sup= 50% conf= 66.6% Sofa → Chair sup= 50% conf= 66.6%

Table9. Test with Apriori over the set FURNITURE STORE

Test with IDM over the set FURNITURE STORE, with minsup = 40 and minconf = 40,Itemsets pair above minimum support and minimum confidence rule:
Rules generated Chair~sofa → table sup= 50% conf= 100% Table→ chair~sofa sup= 50% conf= 100%
Chair → Sofa sup= 50% conf= 66.6% Sofa → Chair sup= 50% conf= 66.6% Chair~sofa → table sup= 50% conf= 100% Table→ chair ~ sofa sup= 50% conf= 100%

Table10. Test with IDM over the set FURNITURE STORE

In Table 10, the underlined rules are those ones which are obtained by IDM, but are not obtained by Apriori. The additional rules bring more information, which can be useful for decision making. When the association rule contains fuzzy items, its support and confidence values are calculated considering the semantic similarity between items. Association rules obtained by IDM contain fuzzy items like chair~sofa (chair and sofa can be considered similar) and which represents interesting semantic similarities not revealed by Apriori. Analyzing the additional rules obtained by IDM, we can show that IDM generates more association rules than Apriori does, with the same support and confidence parameters. As expected, the computation performance of Apriori is better than the computational performance of IDM, because IDM has a more complex structure to find semantically similarity items

VII. CONCLUSION AND FUTURE WORK

We have discussed the data mining algorithms and techniques, which have been used by the researchers to implement the data mining for very large data. With the creation and application of IDM, it has become possible to discover association rules that reflect the semantic similarity among data. The use of fuzzy logic concepts in IDM contributed to make information representation and manipulation closer to the human language, making them more understandable. The better the comprehension of the obtained knowledge, the bigger the knowledge utility. We have also discussed the data mining challenges, in which the researches are required for developing efficient and uniform data mining algorithms, software tools and techniques for very large, high dimensional and complex data.

As future work, here in this paper because a human expert knowledge is used reason that it is easy for human to recognize objects which are existing in a database or to understand the meanings from just short conversion with using their background knowledge. Thus in near future we are thinking to enhance our system in such a way that their should not be a requirement to have an expert for finding similarity between items.

REFERENCES

[1] W.H. Au, K.C.C. Chan, An effective algorithm for discovering fuzzy rules in relational databases, in: Proc. IEEE Internat. Conf. Fuzzy Systems, vol. II, 1998, pp. 1314–1319.

[2] W.H. Au, K.C.C. Chan, FARM: a data mining system for discovering fuzzy association rules, in: Proc. FUZZ-IEEE’99, vol. 3, 1999, pp. 22–25.

[3] Han, J. and Kamber, M. (2001) "Data Mining - Concepts and Techniques", 1st Edition.Nova York: Morgan Kaufmann.

[4] Chen, G. and Wei, Q. (2002) "Fuzzy association rules and the extended mining algorithms",

[5] Fuzzy Sets and Systems, v. 147, n. 1-4, p. 201-228 X.Wu, C.Zhang, and S.Zhang, Mining both Positive and Negative Association Rules, Proc. Of 19th Int. Conf. on Data Machine Learning, pp.658-665,2002.

[6] Zhang, S., Zhang, C., Yan X., "Post-mining: maintenance of association rules by weighting", Information System 28 (2003) pp.691-707.

[7] Han. J and M. Kamber (2004), “Data Mining Concepts and Techniques”: San Franscisco, CA:. Morgan Kaufmann Publishers.

[8] Daly O, Taniar D. Exception Rules Mining Based on Negative Association Rules, Leture Notes in Computer Science, Vol.3046,2004,pp543-552.

[9] Chen G, Zhu YQ, Yang HB, Study of Some Key Techniques in Mining Association Rule, Journal of Computer Research and Development, 2005, 42 (10): 1785-1789.

[10] Ji GL, Yang M. Song YQ. Sun ZH. Fast Updating Maximum Frequent Item sets. Journal of Computers, 2005, 28(1):128-135.

[11] T. P. Hong, K. Y. Lin and S. L. Wang, “Mining fuzzy association rules from quantitative transactions", Soft Computing, Vol. 10, No.10, pp. 925-932, 2006.

[12] Yun, U(2007)., "Efficient mining of weighted interesting patterns with a strong weight and/or support affinity", Information Science 177 (2007) pp.3477-3499.

[13] Preetham Kumar and Ananthanarayana V.S(2008) " Discovery of frequent itemsets using Weighted Tree method" IJCSNS, Vol. 8 No. 8 pp. 195-200

[14] Liewean Cheng, Su-Chuan Chen, and Jashen Chen (2009) "Applying Weighted Association Rules with the Consideration of Product Item Relevancy", 978-1-4244-3662-0/09, 2009 IEEE

[15] Preetham Kumar and Ananthanarayana V.S(2009) "Discovery of Frequent Itemsets Based on Minimum Quantity and Support" Computer Science Journal of Malaysia, Vol 3, Issue 3, June 2009.

[16] Gyenesei A, "Mining weighted association rules for fuzzy quantitative items" TUCS Technical Report No. 346, 2000, pp 1-12.

[17] M. Kaya, R. Alhajj, F. Polat and A. Arslan, "Efficient Automated Mining of Fuzzy Association Rules," Proc. Of aDEXA, 2002.