A Combined Approach for Mining Fuzzy Frequent Itemset

R. Prabamanieswari
Department of Computer Science
Govindammal Aditanar College for Women
Tiruchendur – 628 215

ABSTRACT
Frequent Itemset Mining is an important approach for Market Basket Analysis. Earlier, the frequent itemsets are determined based on the customer transactions of binary data. Recently, fuzzy data are used to determine the frequent itemsets because it provides the nature of frequent itemset ie., it describes whether the frequent itemset consists of only highly purchased items or medium purchased items or less purchased items or combination of all these based on the fuzzy partitions correspond to quantity purchased. This paper concentrates on fuzzy frequent itemset mining in multi-dimensional aspect by combining previously used approaches. This proposed approach initially creates fuzzy partitions for numerical attributes and selects the fuzzy partitions to construct the fuzzy records and create the cluster-based fuzzy set table. Then, it uses cluster-based fuzzy set table, finds the fuzzy frequent itemset and reduces the size of the cluster-based fuzzy set table iteratively. Finally, it concludes with the large fuzzy frequent itemset. This paper also compares the proposed approach with the fuzzy Apriori approach and suggests the proposed approach is better than existing fuzzy Apriori approach.

General Terms
Data Mining, Association Rules, Frequent itemset

Keywords
Fuzzy set, FCM, Cluster-based fuzzy set table, Fuzzy frequent itemset

1. INTRODUCTION
Knowledge discovery, whose objective is to obtain useful knowledge from data stored in large repositories, is recognized as a basic necessity in many areas, especially those related to business. Since data represent a certain real-world domain, patterns that hold in data show us interesting relations that can be used to improve our understanding of that domain. Data mining is the step in the knowledge discovery process that attempts to discover novel and meaningful patterns in data.

One of the best studied models for data mining is that of association rules. In the original context of association rule mining, data are represented by a table with binary values. The main task of association rule discovery is to extract frequent itemsets from data. Apriori[1] algorithm of frequent itemset mining is a powerful method for market basket analysis. This market basket analysis system will help the managers to understand about the set of items are customers likely to purchase. It will also help the managers to propose new way of arrangement in store layouts. It will guide them to plan marketing or advertising approach.This analysis may be carried out on all the retail stores data of customer transactions. Most of the problems deal with quantitative attributes instead of binary attributes. Therefore, there is a need to change from quantitative attribute into binary attribute. In [2], mining quantitative association rules has been proposed. This algorithm finds the association rules by partitioning attribute domains and combining adjacent partitions, then transforms the problem into binary one. Although, this method can solve problem introduced by infinite domain, it causes sharp boundary problem. It either ignores or overemphasizes the elements near the boundaries in the mining process.

In dealing with the sharp boundary problem in partitioning, fuzzy sets which can deal with the boundary problem naturally have been used in the association rule mining domains. Fuzzy sets are first introduced by Lofti A. Zadeh in 1965[3]. After the introduction of fuzzy sets, Fuzzy Association Rule Mining algorithm (Fuzzy Apriori) is introduced by kuok,Fu and Wong[4]. Fuzzy Apriori is similar to classical Apriori but , it uses fuzzy data.

This paper also uses fuzzy sets for mining fuzzy frequent itemsets. The fuzzy sets can be created by using membership functions such as Triangle, Trapezoidal and Gaussian or by using clustering algorithms such as K-Means and Fuzzy C-Means. The created fuzzy sets are used to construct the fuzzy transaction records. In [5] and [6] trapezoidal shape membership function is used for finding fuzzy values. In [7] and [8-12], FCM algorithm is used for generating fuzzy sets. The fuzzy records can be clustered for reducing the execution time. The cluster-based approach is used in [11] and [13-14]. In [11] and [14], triangle function is used initially and generated fuzzy records are clustered depends on the length of the original transaction. In [13], FCM is used instead of triangle function.

In this paper, initially fuzzy partitions are created for each numerical attribute using FCM [15] algorithm. Then, one of the fuzzy partitions for each numerical attribute is selected by finding the sum of each partition separately and considering the maximum value among them. Afterwards, fuzzy records are constructed by considering selected partition for each attribute and joining them. It is similar to [9], but it considers only fuzzy values for numerical attributes. Now, fuzzy frequent 1-itemset is determined by using the created fuzzy records and simultaneously cluster-based fuzzy set table with count >1 is created by eliminating the fuzzy records whose number of fuzzy values satisfying threshold equal to1. Similarly fuzzy frequent 2-itemset is determined by using the created cluster-based fuzzy set table with count >1 and simultaneously cluster-based fuzzy set table with count >2 is created by eliminating the fuzzy records with count equal to 2. Iteratively, the size of the cluster-based fuzzy set table is reduced and fuzzy frequent large itemset is determined from the reduced table.

Section 2 illustrates Fuzzy Apriori and FCM. Section 3 describes the proposed algorithm for determining fuzzy

frequent itemsets. Section 4 demonstrates the algorithm with an example. Section 5 discusses the experimental results. Finally, conclusion is given in Section 6.

2. FUZZY APRIORI AND FCM

Fuzzy Apriori is similar to classical Apriori but , it uses fuzzy data. In recent years, there have been many attempts to improve the classical approach. Fuzzy sets can generally be viewed as an extension of the classical crisp sets. They have been first introduced by Lofti A. Zadeh in 1965[3]. A fuzzy set *A* on a universe *X* is characterized by $X - [0, 1]$ mapping, also called the membership function of *A*. Therefore, it is necessary to convert the crisp dataset into fuzzy dataset.

For converting crisp dataset into fuzzy dataset, first fuzzy partitions are created for each numerical attribute then, multiple fuzzy records are created for each record in the crisp dataset. There is no well-defined and coordinated fuzzy-oriented method to create fuzzy partitions. But, FCM[15] is a very popular and established algorithm for fuzzy clustering in various domains. It helps in the fuzzy partitions of the dataset, where every data point belongs to every cluster to a certain degree μ in the range [0, 1]. Thus, each piece of data can belong to two or more clusters. The algorithm[15] tries to minimize the objective function: $\sum_{i=1}^{N}\sum_{j=1}^{c}\mu_{ij}^{m}\left\|x_i - c_j\right\|^2$ (1) where m is any real number such that $1 \leq m < \infty$, μ_{ij} is the degree of membership of x_i in the cluster j, x_i is the i^{th} d-dimensional measured data, c_j is the d-dimension center of the cluster, and ||*|| is any norm expressing the similarity between any measured data and the center. The fuzziness parameter m is an arbitrary real number ($m > 1$). The higher the value of m, the fuzzier is the resulting partitioning. The fuzzy partitions generated by FCM are normalized such that for each data point the sum of the membership degrees for each cluster is 1 ($\sum_{i=1}^{c}\mu_i = 1$, where *C* is the total number of one dimensional clusters for that particular attribute). Fuzzy partitioning is carried out through an iterative optimization of the objective function shown above, with the update of membership μ_{ij} and the cluster centers c_j by:

$$\mu_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{\left\|x_i - c_j\right\|}{\left\|x_i - c_k\right\|}\right)^{\frac{2}{m-1}}} \quad (2)$$

where $c_j = \frac{\sum_{i=1}^{N}\mu_{ij}^{m} x_i}{\sum_{i=1}^{N}\mu_{ij}^{m}}$ (3)

Based on the fuzzy partitions correspond to each numerical attribute, each record in the original dataset is converted into multiple fuzzy records in the fuzzy dataset.

Both the algorithms Apriori and Fuzzy Apriori count the support of each itemset for finding frequent itemset but, the only difference is that Fuzzy Apriori calculates sum of the membership function μ corresponding to each record where the itemset exists. Thus, the support for any itemset is the sum of membership functions over the whole fuzzy dataset. This calculation is done with the help of a suitable t-norm.

3. PROPOSED APPROACH

The proposed approach considers only the numerical attributes. The following steps are used in this approach.

Step1: Fuzzy partitions are generated using FCM algorithm. Here, the single dimensional FCM is used.

Step2: Fuzzy records are created by considering one of the selected partitions of each numerical attribute and joining them. The partition can be selected by finding the sum of each partition separately and choosing the maximum among them. Then, fuzzy frequent 1-itemset is determined by using the created fuzzy records and simultaneously cluster-based fuzzy set table with count >1 is created by eliminating the fuzzy records whose number of fuzzy value satisfying threshold equal to1.

Step3: Frequent large itemset is determined by following the modified Apriori algorithm[16] along with cluster-based fuzzy set table.

This approach follows [15] for performing Step1. Step2 is carried out by following [9] and [11]. In [9], both numerical and categorical attributes are considered and in [11], the length of original transaction is considered for clustering in to corresponding table and triangle function is used for generating membership value, but in this approach, FCM is used for generating membership value and number of satisfied membership values of each record(each membership value should be ≥ specified threshold membership value) is considered for clustering into cluster-based fuzzy set table. Here, the same cluster-based fuzzy set table is used successively by eliminating the fuzzy records whose number of membership values is not satisfied. Initially, the records whose no of membership =1 are removed, then the records whose no of membership =2 are removed and so on till the large fuzzy frequent itemset is detemined instead of clustering into cluster-table(k),cluster-table(k+1),cluster table(k+2)….. in [11]. Step 3 is performed by modifying candidate generation process of an Apriori algorithm[1]. This proposed algorithm finds the subsets of a candidate itemset partially. It also follows the same procedure used in [16] along with cluster-based fuzzy set table and finds frequent k-itemset. In [11], frequent k-itemset is determined with reference to the cluster-table(k), then it is contacted with the cluster-table(k+1), cluster table(k+2)…….but here, the same cluster-based fuzzy set table is used. Therefore, the number of times of scanning the database is reduced. Hence, the proposed approach is faster than existing fuzzy Apriori algorithm. The proposed algorithm is given below:

Proposed Algorithm

Input: database D, min-support, threshold -membership value

Output: fuzzy frequent large itemset

Main()
begin
1. Call FCM1
2. Call Modified Apriori()

End

Procedure FCM1

```
{
    D1= Null;
    candidate= { set of numerical items}
    for candidate's numerical attribute= 1 to n do
    {
      For each transaction T ∈ D do
      {
        1 create fuzzy partitions using FCM
        2. sum each partition separately
      }
     select any one partition with greater sum
```

```
      D1=D1+selected partition
      }
//create fuzzy_cltable

Fuzz_cltable={}
for each transaction T ∈ D1 do
{
  candidate.supp is calculated based on threshold-membership value
 if (no of membership of  T >1)
     add T into  fuzzy_cltable
}
L1={candidate | candidate.supp  ≥ min-support}
D1=fuzzy_cltable
}

Procedure Modified Apriori()
{
   number of items in a combination (c1) = 1
    k:=2
while (Lk-1 != φ)
{
  call_generate_new (c1)
  c1++
 k++
}
}
Procedure call_generate_new(c1)
{
generate candidates (Lk-1)
 fuzzy_cltable={}
for each  generated candidates c ∈ Ck
{
   if (c1=1) call frequent(c)
   else
   {
      get  all partial subset( c)
      if (all partial subset(c) ∈ Lk-1 )
           call frequent(c)
   }
D1=fuzzy_cltable
}
}

Procedure frequent(c)
{
 c.supp=0
for each  fuzzy transaction T ∈ D1
{
find fuzzy intersection(c)   // during fuzzy intersection, check each value against threshold -membership value; if value is < threshold -membership value, then  skip that transaction
    if  ( fuzzy intersection(c) ≥ threshold -membership value )
{
      c.supp++

}
      add no of membership of T >k to fuzzy_cltable

 }
    Lk = {c | c.supp  ≥ min-support}
}
```

4. EXAMPLE

Let D= {T_1, T_2……T_n} be a set of transaction and I= {i_1,i_2…….i_n} be a set of attributes. Each attribute i_k associates with several fuzzy sets such as F_i={f_1,f_2…….f_n}. Table 1 shows sample database with quantitative attributes (quantity corresponds to items).

Table1. A Sample Transactional Database

Trans-id	Item with quantity
1	i_1: 31,i_2:27,i_3:39,i_4:39,i_6:39
2	i_2:39,i_4:27,i_6:29
3	i_1:21,i_3:27,i_5:29
4	i_1:37,i_2:27,i_5:35
5	i_2:35,i_4:20,i_6:21
6	i_1:20,i_2:27,i_3:31,i_6:23
7	i_3:25,i_5:29
8	i_4:25,i_5:38,i_6:36
9	i_5:39,i_6:31
10	i_3:27,i_4:37,i_6:34

The fuzzy sets for each attribute is determined by using FCM algorithm with the clusters' centers such as c_1=15, c_2=30, and c_3=40. Therefore, three fuzzy partitions are created for each quantitative attribute and one of the fuzzy partitions is selected based on the maximum cumulative value among them. Then, fuzzy records are created by joining all the selected partitions and removing the fuzzy value 0. Table 2 shows fuzzy transactional database.

Table2. A fuzzy Transactional Database/ cluster-based fuzzy set table (1).

Trans-id	Item with fuzzy value
1	$i_{1.med}$: 0.853;$i_{2.med}$:0.896;$i_{4.med}$: 0.008;$i_{6.med}$: 0.012
2	$i_{2.med}$: 0.012;$i_{4.med}$: 0.99;$i_{6.med}$: 0.987
3	$i_{1.med}$: 0.465;$i_{3.med}$: 0.956;$i_{5.med}$: 0.987
4	$i_{1.med}$: 0.047;$i_{2.med}$: 0.896;$i_{5.med}$: 0.485
5	$i_{2.med}$: 0.485;$i_{4.med}$: 0.463;$i_{6.med}$: 0.288
6	$i_{1.med}$: 0.338;$i_{2.med}$: 0.896;$i_{3.med}$: 0.931;$i_{6.med}$: 0.517
7	$i_{3.med}$: 0.85;$i_{5.med}$: 0.987
8	$i_{4.med}$:0.915;$i_{5.med}$: 0.05;$i_{6.med}$: 0.3
9	$i_{5.med}$: 0.012;$i_{6.med}$: 0.984
10	$i_{3.med}$: 0.956;$i_{4.med}$: 0.099;$i_{6.med}$: 0.672

Initially, cluster-based fuzzy set table is same as constructed fuzzy records. Then its size is reduced successively. Here, each fuzzy record's each membership value is checked against the threshold membership value and the satisfied value is considered for finding the count value. Thereafter, the fuzzy record is stored in a cluster-based fuzzy set table depends on the count value for further iteration. It is shown as below:

Table3. Cluster-based fuzzy set table (2)

Trans-id	Item with fuzzy value
1	$i_{1.med}$: 0.853;$i_{2.med}$:0.896;$i_{4.med}$: 0.008;$i_{6.med}$: 0.012
2	$i_{1.med}$: 0.465;$i_{3.med}$: 0.956;$i_{5.med}$: 0.987
3	$i_{1.med}$: 0.047;$i_{2.med}$: 0.896;$i_{5.med}$: 0.485

4	$i_{2.med}$: 0.485;$i_{4.med}$: 0.463;$i_{6.med}$: 0.288
5	$i_{1.med}$: 0.338;$i_{2.med}$: 0.896;$i_{3.med}$: 0.931;$i_{6.med}$: 0.517

Table4. Cluster-based fuzzy set table (3)

Trans-id	Item with fuzzy value
1	$i_{1.med}$: 0.465;$i_{3.med}$: 0.956;$i_{5.med}$: 0.987
2	$i_{2.med}$: 0.485;$i_{4.med}$: 0.463;$i_{6.med}$: 0.288
2	$i_{1.med}$: 0.338;$i_{2.med}$: 0.896;$i_{3.med}$: 0.931;$i_{6.med}$: 0.517

Table5. Cluster-based fuzzy set table (4)

Trans-id	Item with fuzzy value
1	$i_{1.med}$: 0.338;$i_{2.med}$: 0.896;$i_{3.med}$: 0.931;$i_{6.med}$: 0.517

Assume that the threshold membership value is 0.05 and min-support is 0.05. The fuzzy support values of candidate 1-itemsets are calculated as:

$i_{1.med}$ = 0.853+0.0465+0.047+0.338
= 1.703

$i_{2.med}$ = 0.896+0.896+0.485+0.896
=3.173

$i_{3.med}$ =0.956+0931+0.85+0.956
=3.693

$i_{4.med}$ =6.99+0.463+0.915+0.099
=8.467

$i_{5.med}$ =0.987+0.485+0.987+0.058
=2.517

$i_{6.med}$ =0.987+0.288+0.517+0.3+0.672
=2.764

The candidate 1-itemset $\geq$ 0.05 are $\{\{i_{1.med}\},\{i_{2.med}\},\{i_{3.med}\},\{i_{4.med}\},\{i_{5.med}\},\{i_{6.med}\}\}$.
Therefore, $L_1 = \{\{i_{1.med}\}, \{i_{2.med}\}, \{i_{3.med}\}, \{i_{4.med}\}, \{i_{5.med}\},\{i_{6.med}\}\}$.

Then candidate 2-itemsets are generated and frequent 2-itemsets are determined from cluster-based fuzzy-sets table with count >1 .i.e., cluster-based fuzzy-sets table (2). For example, the fuzzy support value of candidate 2-itemset $\{i_{1.med}, i_{2.med}\}$ is calculated as

$\{i_{1.med}, i_{2.med}\}$ = min (0.853, 0.896) + min (0.047, 0.896)
+min (0.338, 0.896)
=0.853+0.047+0.338
=1.238

Similarly, the fuzzy support values of further candidate2-itmsets are calculated. Therefore,
$L_2=\{\{i_{1.med},i_{2.med}\},\{i_{1.med},i_{3.med}\},\{i_{1.med},i_{5.med}\},\{i_{1.med},i_{6.med}\},\{i_{2.med},i_{3.med}\},\{i_{2.med},i_{4.med}\},\{i_{2.med},i_{5.med}\},\{i_{2.med},i_{6.med}\},\{i_{3.med},i_{4.med}\},\{i_{3.med},i_{5.med}\},\{i_{3.med},i_{6.med}\},\{i_{4.med},i_{5.med}\},\{i_{4.med},i_{6.med}\},\{i_{5.med},i_{6.med}\}\}$

Now, the candidate 3-itemsets are generated and frequent 3-itemsets are determined from cluster-based fuzzy-sets table with count >2 as,
$L_3=\{\{i_{1.med},i_{2.med},i_{3.med}\},\{i_{1.med},i_{2.med},i_{6.med}\},\{i_{1.med},i_{3.med},i_{5.med}\},\{i_{1.med},i_{3.med},i_{6.med}\}\{i_{2.med},i_{3.med},i_{6.med}\},\{i_{2.med},i_{4.med},i_{6.med}\}\}$

In the similar way, large itemsets L_n are discovered. In this example, the large itemset determined is $\{i_{1.med}, i_{2.med}, i_{3.med}, i_{6.med}\}$.

5.EXPERIMENTAL RESULTS

To evaluate the efficiency of the proposed method, it is implemented along with fuzzy Apriori algorithm and experimented with the mushroom dataset. The mushroom dataset contains the characteristics of various species of mushrooms and is originally obtained from the UCI repository of machine learning databases. It has 119 items and 8124 transactions. The minimum, maximum and average length of its transaction is 23.

Both algorithms are implemented using C# language and carried on the computer with the configuration such as Intel(R) Core(TM) i3CPU, 3 GB RAM, 2.53 GHz Speed and Windows 7 Operating System. Fig 1 shows the performance of both algorithms

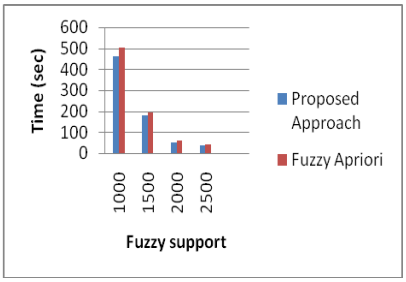

Fig1. Performance of Fuzzy Apriori and Proposed Approach

Here, the quantities are randomly added with the mushroom dataset and fuzzy transactions are created. The time taken for creating fuzzy transactions is 54.5 sec and for finding fuzzy frequent itemset is given in the figure 1. The experiment is done by varying fuzzy support values such as 1000.0, 1500.0, 2000.0 and 2500.0 and getting frequent 5-itemset, frequent 4-itemset, frequent 3-itmset and frequent 3-itemset correspondingly. From the figure 1, it is clear that the proposed algorithm takes lesser time comparing to Fuzzy Apriori algorithm.

6. CONCLUSION

The proposed approach determines the fuzzy frequent large itemsets based on the approaches such as FCM, FPREP, FCBAR and modified Apriori algorithm. It illustrates the fast performance of the proposed approach with the comparison to Fuzzy Apriori approach. It considers only numerical attributes for constructing fuzzy records. In future, it will consider categorical attributes along with the numerical attributes for creating fuzzy records and finding fuzzy frequent itemset.

7. REFERENCES

[1] Agrawal. R., Imielimski. T and Swami. A.N., "Mining Association Rules between sets of items in large database", Proceedings of ACM SIGMOD International Conference Management of Data, CM Press, pp. 207-216, 1993.

[2] R. Srikant and R. Agrawal, "Mining quantitative association rules inlarge relational tables," in Proc.

ACM SIGMOD Conf. Management Data, Montréal, QC, Canada,pp. 1–12, 1996.

[3] Zadeh L.A., "Fuzzy Sets and systems", International Journal of General Systems, Vol. 17, pp. 129-138, 1990.

[4] C. M. Kuok, A. W.-C. Fu, and M. H. Wong, "Mining fuzzy association rules in databases," ACM SIGMOD Rec., vol. 27, no. 1, pp. 41–46, Mar.1998.

[5] Weining Zhang, "Mining Fuzzy Quantitative Association Rules", in Proc. ICTAI, pp. 99-102, 1999.

[6] Keon-Myung Lee, "Mining Generalized Fuzzy Quantitative Rules with Fuzzy Generalization Hierarchies", IFSA World Congress and 20th NAFIPS International Conference, 2001, vol. 5, pp.2977-2982, Digital Object Identifier : 10.1109/NAFIPS.2001.943701.

[7] H.Verlinde, M. De Cock, and R. Boute, "Fuzzy versus quantitative association rules: A fair data driven comparison," IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 36, no. 3, pp. 679–684, Jun. 2006.Man, Cybern. B, Cybern., vol. 36, no. 3, pp. 679–684, Jun. 2006.

[8] E.Hüllermeier, Y. Yi, "In Defense of Fuzzy Association Analysis", IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics, vol. 37, no.4,pp. 1039-1043, August.2007.

[9] Ashish Mangalampalli and Vikram Pudi, "FPrep: Fuzzy Clustering driven Efficient Automated Pre-processing for Fuzzy Association Rule Mining", IEEE Intl Conference on Fuzzy Systems (FUZZ-IEEE) Barcelona, Spain

[10] Ashish Mangalampalli and Vikram Pudi, "Fuzzy Association Rule Mining Algorithm for Fast and Efficient Performance on Very Large Datasets", FUZZ-IEEE 2009, Korea, pp.1163-1168, August 20-24, 2009.

[11] Reza Sheibani and Amir Ebrahimzadeh, "An Algorithm For Mining Fuzzy Association Rules", Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 Vol I, IMECS 2008, 19-21 March, 2008, Hong Kong.

[12] Toshihiko WATANABE, "A Fast Fuzzy Association Rules Mining Algorithm Utilizing Output Field Specification", Biomedical Soft Computing and Human Sciences, vol. `16, no. 2, pp. 69-76.

[13] Amir Ebrahimzadeh and Reza Sheibani, "Two Efficient Algorithms for Mining Fuzzy Association Rules", International Journal of Machine Learning and Computing, Vol. 1, No. 5, December 2011.

[14] Hung-Pin Chiu et.al., "Applying cluster-based fuzzy association rules mining framework into EC environment", Applied Soft Computing, Article in Press 2012.

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

[16] D.Ashok Kumar and R.Prabamanieswari, "A Modified Algorithm for Generating Single Dimensional Fuzzy itemset Mining", International Journal of Computational Intelligence and Informatics, Vol. 1, N0. 3, pp. 162-166, ISSN: 2231-0258, October-December 2011.