A Framework for Mining Fuzzy Association Rules from Composite items

Maybin Muyeba[1], M. Sulaiman Khan[2], Frans Coenen[3]

[1]Department of Computing and Mathematics, Manchester Metropolitan University, Manchester, M1 5GD, UK
[2]Liverpool Hope University, Liverpool, L16 9JD, UK
[3] Department of Computer Science, University of Liverpool, Liverpool, L69 3BX, UK
{M.Muyeba@mmu.ac.uk, khanm@hope.ac.uk, frans@csc.liv.ac.uk}

Abstract. A novel framework is described for mining fuzzy Association Rules (ARs) relating the properties of composite attributes, i.e. attributes or items that each feature a number of values derived from a common schema. To apply fuzzy Association Rule Mining (ARM) we partition the property values into fuzzy property sets. This paper describes: (i) the process of deriving the fuzzy sets (Composite Fuzzy ARM or CFARM) and (ii) a unique property ARM algorithm founded on the correlation factor interestingness measure. The paper includes a complete analysis, demonstrating: (i) the potential of fuzzy property ARs, and (ii) that a more succinct set of property ARs (than that generated using a non-fuzzy method) can be produced using the proposed approach.

Keywords: Association rules, fuzzy association rules, composite attributes, quantitative attributes.

1 Introduction

Association Rule Mining (ARM) now a well known and established data mining topic among researchers. Mainly, ARM finds frequent items (attributes, usually binary valued) and then identifies patterns in the form of Association Rules (ARs) from large transaction data sets [5, 6, 12]. ARM has been applied to quantitative and categorical (non-binary) data [1, 13, 16]. With the latter, values can be split into linguistically labeled ranges such that each range represents a binary valued; for example “low”, “medium”, “high” etc. Values can be assigned to these attribute ranges using crisp or fuzzy boundaries. The application of the latter is referred to as fuzzy ARM (FARM) [1]. Objectively, fuzzy ARM identifies fuzzy ARs. Fuzzy ARM has been shown to produce more expressive ARs than the “crisp” methods [1, 3, 4,]. ARM (both fuzzy and standard) algorithms typically operate using the support-confidence framework, however with a number of disadvantages including (among others) the tendency to generate many and mostly redundant ARs not any more useful, expressive, succinct or significant. In contrast, the correlation measure produces a more succinct set of rules [3] and we explore this aspect.

We approach the problem differently in this paper by introducing "Composite item" Fuzzy ARM (CFARM) whose main objective is the generation of fuzzy ARs associating the "properties" linked with composite attributes [15] i.e. attributes or items composed of sets of sub-attributes or sub-items that conform to a common schema. For example, given an image mining application, we might represent different areas of each image in terms of groups of pixels such that each group is represented by the normalized summation of the RGB values of the pixels in that group. In this case the set of composite attributes (I) is the set of groups, and the set of properties (P) shared by the groups is equivalent to the RGB summation values (i.e. $P = \{R, G, B\}$). Another could be market basket analysis where I is a set of groceries, and P is a set of nutritional properties that these groceries posses i.e. P = {Pr, Fe, Ca, Cu,..} standing for protein, Iron etc. Note that the actual values (properties) associated with each element of I will be constant, unlike in the case of the image mining example. We note that there are many examples depending on application area but we limit ourselves to these given here. For quantitative attributes, we can partition them into intervals [13] and rename these with linguistic values (fuzzy sets) [1].

The contributions in this paper are :

- The framework of the concept of "Composite item" mining of property ARs
- The potential of using property ARs in many applications
- Greater accuracy using the certainty factor measure as against confidence
- Demonstration of a more succinct set of property ARs (than that generated using a non-fuzzy method) can be produced using the proposed approach.

The paper is organised as follows. In section 2 we present the background and related work to the proposed composite fuzzy ARM approach described, Section 3 presents a sequence of formal definitions for the work and section 4, the detail of the CFARM algorithm; a complete analysis of the CFARM algorithm is given in Section 5, and section 6 concludes the paper with a summary of the contribution of the work and directions for future work.

2. Background and Related Work

Most ARM algorithms in general concentrate on performance [2, 3, 5] by first generating all large (frequent) itemsets and then find ARs from them. To limit the number of ARs generated a confidence threshold is used. However great care must be taken not to remove low support items but from which high confidence rules may be generated. In literature the term "composite item" has been used in the context of data mining. In [8, 16], a composite item is defined as a combination of several items e.g. if itemset {A, B} and {A, C} are not large then rules {B}→{A} and {C}→{A} will not be generated, but by combining B and C to make a new *composite* item {BC} which may be large, rules such as {BC}→{A} may be generated. In this paper we define composite items differently as indicated earlier, to be an item with properties (see Section 3) and also in [15], composite attributes are defined in this manner.

In ARM, quantitative attributes are usually discretised into various partitions, with each partition regarded as a binary valued attribute. One major problem in this approach is that of “sharp boundary problems”. Fuzzy ARM [3, 7, 14] has been shown to resolve this problem by mapping numeric values to membership degrees from their partitions with total individual item contributions to support counts remaining as unity value (1.0) regardless of whether an item value belongs to one or more fuzzy sets (similar to the approach in [1]). Detailed overviews of FARM are given in [1, 3, 9, 14].

To illustrate the concepts, we consider super market basket analysis (table 1) where the set of groceries (I) (or edible items) have a common set of nutritional quantitative properties.

Items/Nutrients	Protein	Fibre	Carbohydrate	Fat	…
Milk	3.1	0	4.7	.2	…
Bread	8	3.3	43.7	1.5	…
Biscuit	6.8	4.8	66.3	22.8	…
…	…	…	…	…	…

Table 1. Example composite attributes (groceries) with their associated properties (nutrients)

To illustrate the context of our problem, composite items (edible items) have common properties like Protein, Fibre, Iron etc and are defined by the same five fuzzy sets {Very Low, Low, Ideal, High, Very High}. The objective is then to identify patterns linking these properties and so derive fuzzy association rules (see next section).

3. Problem Definition

In this section a sequence of formal definitions is presented to define composite attributes, describe FARM concept, the normalization process for Fuzzy Transactions (*FT)* and interestingness measures.

3.1. Terms and definitions

Definition 1: A *Fuzzy Association Rules* [3] is an implication of the form:

$$\text{if } \langle A, X \rangle \text{ then } \langle B, Y \rangle$$

where A and B are disjoint itemsets and X and Y are fuzzy sets.

Definition 2: *Raw Dataset* (the input data) D consists of a set of transactions $T = \{t_1, t_2, t_3, \cdots, t_n\}$, composite items $I = \{i_1, i_2, i_3, \cdots, i_{|I|}\}$ and properties $P = \{p_1, p_2, p_3, \cdots, p_m\}$. Each transaction t_i is some subset of I, and each item $t_i[i_j]$ (the "j^{th}" item in the "i^{th}" transaction) is a subset of P. Thus i_j has associated sets of values in set P, i.e. $t_i[i_j] = \{v \mid v_1, v_2, v_3, \cdots, v_m\}$.

TID	Record
1	{<a,{2,4,6}>, <b,{4,5,3}>}
2	{<c,{1,2,5}>, <d,{4,2,3}>}
3	{<a,{2,4,6}>, <c,{1,2,5}>, <d,{4,2,3}>}
4	{<b,{4,5,3}>, <d,{4,2,3}>}

Table 2. Example raw dataset D.

The "k^{th}" property (categorical or quantitative) value for the "j^{th}" item in the "i^{th}" transaction is given by $t_i[i_j[v_k]]$. An example is given in Table 2 where each composite item is represented using the notation <label, value>. In the rest of this paper the term "item" is used to mean an item in an itemset as used in traditional ARM, and the term attribute is used to mean a property item (sub-item).

Definition 3: A given raw dataset D is initially transformed into a *property data set* D^p with property transactions $T^p = \{t_1^p, t_2^p, t_3^p, \cdots, t_n^p\}$ and property attributes P (instead of a set of composite items I). Thus $\forall t_i^p \subset P$. The value for each property attribute $t_i^p[p_j]$ (the "j^{th}" property attribute in the "i^{th}" property transaction) is obtained by aggregating the numeric values for all p_j in t_i (See Table 3). Thus:

$$\text{Prop Value}(t_i^p[p_j]) = \frac{\sum_{j=1}^{|t_i|} t_i[i_j[v_k]]}{|t_i|} \quad \textbf{(1)}$$

TID	X	Y	Z
1	3.0	4.5	4.5
2	3.0	2.0	4.0
3	2.3	2.3	4.7
4	4.0	3.5	3.0

Table 3: Example property data set D^p generated from raw data set given in table 2

Definition 4: Once a property data set D^p is defined, it is then transformed into a *Fuzzy Dataset* D'. A fuzzy dataset D' consists of fuzzy transactions $T' = \{t'_1, t'_2, t'_3, ..., t'_n\}$ and a set of fuzzy property attributes P' each of which has fuzzy sets with linguistic *labels* $L = \{l_1, l_2, l_3, ..., l_{|L|}\}$. Each property attribute $t_i^p[p_j]$ is associated (to some degree) with several fuzzy sets and given by a *membership degree* value in $[0..1]$ in some *fuzzy linguistic labels*. The “k^{th}” label for the “j^{th}” property attribute for the “i^{th}” fuzzy transaction is given by $t_i'[p_j[l_k]]$. The nature of the user defined fuzzy ranges is expressed in a *properties table* (see definition 6 below). The numeric values for each property attribute $t_i^p[p_j]$ are *fuzzified* (mapped) into the appropriate membership degree values using a membership function $\mu(t_i^p[p_j], l_k)$ that applies the value of $t_i^p[p_j]$ to a label $l_k \in L$, e.g. $t_i'[p_j] = \{\mu(t_i^p[p_j]], l_1), \mu(t_i^p[p_j]], l_2), \mu(t_i^p[p_j]], l_3), \cdots, \mu(t_i^p[p_j]], l_{|L|})\}$. The complete set of fuzzy property attributes P' is then given by $P \times L$. A fuzzy data (Table 4) based on the property data set (Table 3) is given.

TID	X			Y			Z		
	Small	Medium	Large	Small	Medium	Large	Small	Medium	Large
1	0.0	1.0	0.0	0.0	0.4	0.6	0.0	1.0	0.0
2	0.0	1.0	0.0	1.0	0.0	0.0	0.3	0.7	0.0
3	0.3	0.7	0.0	1.0	0.0	0.0	0.0	0.9	0.1
4	0.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	0.0

Table 4: Example Fuzzy data set ($L = \{\text{small, medium, large}\}$, μ unspecified).

Definition 5: *Composite Itemset Value (CIV)* table allows us to get property values for specific items. Note that a CIV table is not always required; the values may be included in the raw data as in the case of the example raw dataset presented in Table 2 where property values are all in the range [1..6]. The CIV table for the example raw dataset given in Table 2 is given in Table 5 below.

Item	Property attributes		
	X	Y	Z
A	2	4	6
B	4	5	3
C	1	2	5
D	4	2	3

Table 5: Composite Itemset Value Table for raw dataset given in Table 2

Definition 6: *Properties Table* is a table that maps all possible values for each property attribute $t_i^p[p_j]$ onto user defined (and overlapping) linguistic labels L. An example is given in Table 6 for the raw data set given in Table 2.

Property	Linguistic values		
	Low	Medium	High
X	$v_k \leq 2.3$	$2.0 < v_k \leq 3.7$	$3.3 < v_k$
Y	$v_k \leq 3.3$	$3.0 < v_k \leq 4.3$	$4.1 < v_k$
Z	$v_k \leq 4.0$	$3.6 < v_k \leq 5.1$	$4.7 < v_k$

Table 6: Properties Table for raw dataset given in Table 2

Definition 7: A property attribute set A, where $A \subseteq P \times L$, is a *Fuzzy Frequent Attribute Set* if its fuzzy support value is greater than or equal to a user supplied minimum support threshold (see sub-section 3.3 below).

Definition 8: *Fuzzy Normalisation* is the process of finding the contribution to the fuzzy support value, m', for individual property attributes ($t_i^p[p_j[l_k]]$) such that a partition of unity is guaranteed. This is given by the equation (where μ is the membership function):

$$t_i'[p_j[l_k]] = \frac{\mu(t_i^p[p_j[l_k]])}{\sum_{x=1}^{|L|} \mu(t_i^p[p_j[l_x]])} \qquad \textbf{(2)}$$

Without normalisation, the sum of the support contributions of individual fuzzy sets associated with an attribute in a single transaction may no longer be unity. This is illustrated in Tables 7 and 8. In the tables, the possible values for the item "Proteins" have been ranged into five fuzzy sets labelled: "Very Low" (VL), "Low" (L), "Ideal", "High" (H) and "Very High" (VH). Table 7 shows a set of raw membership degree values, while Table 8 shows the normalised equivalents. The normalisation process ensures membership values for each property attribute are consistent and are not affected by boundary values.

TID	**Proteins**					…
	VL	**L**	**Ideal**	**H**	**VH**	…
1	0.0	0.0	0.0	1.0	0.32	…
2	0.83	0.38	0.0	0.0	0.0	…
3	…	…	…	…	…	…

Table 7: Fragment data set without normalization

TID	**Proteins**					
	VL	**L**	**Ideal**	**H**	**VH**	…
1	0.0	0.0	0.0	0.76	0.24	…
2	0.69	0.31	0.0	0.0	0.0	…
3	…	…	…	…	…	…

Table 8: Fragment data set with normalization

3.2. Fuzzy Support and Confidence

The support-confidence framework can also be applied to fuzzy association rule mining through fuzzy support (significance) values. Fuzzy Support (FS) is typically calculated as follows [1]:

$$FS(A) = \frac{\text{Sum of votes satisfying A}}{\text{Number of records in } T}$$

where $A = \{a_1, a_2, a_3, ..., a_{|A|}\}$ is a set of property attribute-fuzzy set (label) pairs such that $A \subseteq P \times L$. A record t'_i "satisfies" A if $A \subseteq t'_i$. The individual vote per record is found by multiplying the membership degree with an attribute-fuzzy set pair $[i[l]] \in A$:

$$\text{vote for } t_i \text{ satisfying } A = \prod_{\forall [i[l]] \in A} t'_i[i[l]] \tag{3}$$

So we have,

$$\text{FS}(A) = \frac{\sum_{i=1}^{i=n} \prod_{\forall [i[l]] \in A} t'_i[i[l]]}{n} \tag{4}$$

Frequent attribute sets with fuzzy support above the specified threshold are used to generate all possible rules. A fuzzy AR derived from a fuzzy frequent attribute set C is of the form:

$$A \rightarrow B$$

where A and B are disjoint subsets of the set $P \times L$ such that $A \cup B = C$. Fuzzy Confidence *(FC)* is calculated in the same manner that confidence is calculated in classical ARM:

$$FC(A \rightarrow B) = \frac{FS(A \cup B)}{FS(A)} \tag{5}$$

3.3. Fuzzy Correlation

The Fuzzy Confidence measure (FC) described does not use $FS(B)$ but the fuzzy correlation measure (F_{CORR}) addresses this. The correlation measure is a statistical measure founded on the concepts of *covariance* (Cov) and *variance* (Var) and is calculated as follows:

$$F_{CORR}(A \rightarrow B) = \frac{Cov(A, B)}{\sqrt{Var(A) \times Vat(B)}} \tag{6}$$

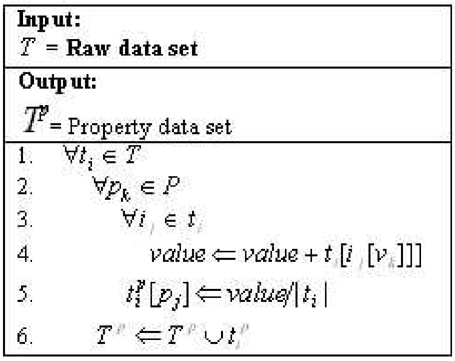
In statistics covariance is calculated by subtracting the product of the individual expected values for A and B from the expected value of C where $C = A \cup B$. The value of correlation ranges from -1 to +1. Value -1 means no correlation and +1 means maximum correlation. Thus we are only interested in rules that have a correlation value that is greater than 0. As the certainty value increases from 0 to 1, the more related the attributes are and consequently the more interesting the rule.

4. The CFARM Algorithm

Fuzzy ARM can use standard ARM algorithms and few works report on their efficient implementations [7]. Fuzzy ARM do a significant amount of processing (filtration, conversions, normalization) to prepare the raw data prior to mining it.

The proposed Composite Fuzzy ARM (CFARM) algorithm (similar to Apriori [5]), belongs to the *breadth first traversal* family of ARM algorithms, developed using tree data structures [6]. The CFARM algorithm consists of four major steps:

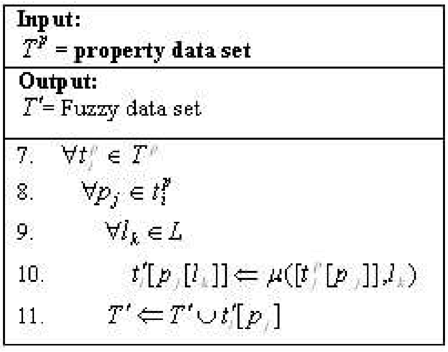
1. Transformation of ordinary transactional data set (T) into a property data set (T^p).
2. Transformation of property data set (T^p) into a fuzzy data set T'.
3. Apply an Apriori style fuzzy association rule mining algorithm to T'using fuzzy support, confidence and correlation measures of the form described above to produce a set of frequent item sets F .
4. Process F and generate a set of fuzzy ARs R such that $\forall r \in R$ the certainty factor (either confidence or correlation as desired by the end user) is above some user specified threshold.



Input:
T **= Raw data set**

Output:
T^p= Property data set

```
$\forall t_i \in T$
    $\forall p_k \in P$
        $\forall i_j \in t_i$
            $value \Leftarrow value + t_i[i_j[v_k]]$
        $t_i^p[p_j] \Leftarrow value/|t_i|$
    $T^p \Leftarrow T^p \cup t_i^p$
```



Input:
T^p **= property data set**

Output:
T'= Fuzzy data set

```
$\forall t_i^p \in T^p$
    $\forall p_j \in t_i^p$
        $\forall l_k \in L$
            $t'_j[p_j[l_k]] \Leftarrow \mu([t_j^p[p_j]], l_k)$
        $T' \Leftarrow T' \cup t'_j[p_j]$
```

Table 9: rawToPropertyDataSetConverter(T) **Table 10 : p**ropertToFuzzyDataSetConverter(T^p)

The algorithms for steps 1and 2 are presented in Tables 9 and 10. To illustrate steps 1 and 2 a fragment of a raw data set (T) is given in Table 11(a). This raw data is then cast into a properties data set (T^P) by averaging the property values for each transaction (see definition 3 and table 3). For example, assuming the CIV table given in table 5 and considering transaction $t_1 = \{a, b\}$, from Table 5, a has property values {2, 4, 6} and b has property values {4, 5, 3}. Thus $t_1^P = \{(2+4)/2, (4+5)/2, (6+3)/2\} = \{3.0, 4.5, 4.5\}$, assuming the properties table of the form presented in Table 4 where $L = \{S\text{mall}, \text{Medium}, \text{Large}\}$. The result is as shown in Table 11(b) which is then cast into a fuzzy data set T' as shown in Table 11(c).

(a) Raw data (T)

TID	Items
1	a, b
2	c
3	a, b, d
4	…

(b) Property data set (T^P)

TID	X	Y	Z
1	3.0	4.5	4.5
2	1	2	5
3	3.3	3.3	4.0
4	…	…	…

(c) Fuzzy data set (T')

TID	X			Y			Z		
	S	M	L	S	M	L	S	M	L
1	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.7	0.3
2	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.2	0.8
3	0.0	0.7	0.3	0.4	0.6	0.0	0.0	1.0	0.0
4	…	…	…	…	…	…	…	…	…

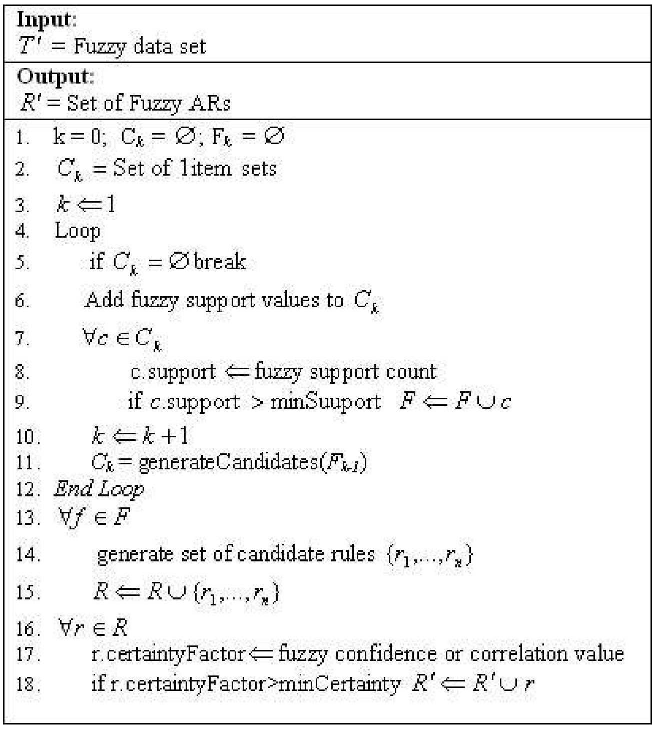
Table 11 Some example data sets (raw, property, conventional)

An alternative approach is to discretise the data. For example, again assuming no overlapping (say) $\text{Small} < 2$, $2 < \text{Medium} < 4$ and $4 < L\text{arge}$, then the values in Table 12(b) can be discretised into the set of attributes $\{X_{\text{Small}}, X_{\text{Medium}}, X_{\text{Large}}, Y_{\text{Small}}, Y_{\text{Medium}}, Y_{\text{Large}}, Z_{\text{Small}}, Z_{\text{Medium}}, Z_{\text{Large}}\}$ and then assigned to a sequence $\{1,2,3,4,5,6,7,8,9\}$. In that case the property data set in Table 11(b) could be represented in conventional ARM terms, which can then be mined using a conventional ARM algorithm. The significance is that we shall use an example property dataset cast into this format for evaluation purposes in Section 5.

The final part of the CFARM algorithm is given in Table 12. In the Table: C_k is the set of candidate itemsets of cardinality k, F is the set of frequent item sets, R is the set of potential rules and R' is the final set of generated fuzzy ARs. Note that the certainty factor can be confidence or a correlation or some other certainty measure.

5. Experimental Results

To demonstrate the effectiveness of the approach, we performed several experiments using a T10I4N0.6KD100k data set generated using IBM Quest data generator [11]. The data is a transactional database containing 100K records. For the purpose of the experiment we mapped the 600 item numbers onto 600 products in a real RDA table.



Input:
T' = Fuzzy data set

Output:
R' = Set of Fuzzy ARs

```
k = 0;  $C_k = \varnothing$; $F_k = \varnothing$
$C_k$ = Set of 1item sets
$k \Leftarrow 1$
Loop
    if $C_k = \varnothing$ break
    Add fuzzy support values to $C_k$
    $\forall c \in C_k$
        c.support $\Leftarrow$ fuzzy support count
        if $c$.support > minSuuport  $F \Leftarrow F \cup c$
    $k \Leftarrow k+1$
    $C_k$ = generateCandidates($F_{k-1}$)
End Loop
$\forall f \in F$
    generate set of candidate rules $\{r_1,\ldots,r_n\}$
    $R \Leftarrow R \cup \{r_1,\ldots,r_n\}$
$\forall r \in R$
    r.certaintyFactor $\Leftarrow$ fuzzy confidence or correlation value
    if r.certaintyFactor>minCertainty  $R' \Leftarrow R' \cup r$
```

Table 12: fuzzyDataSetToFuzzyARs(T')

5.1. Experiment One: (Quality Measures)

Our experiment in the first instance compares CFARM, with and without normalisation, against standard (discrete) ARM with respect to: (i) the number of frequent sets generated and (ii) the number of rules generated (using both the confidence and the correlation measure). Figure 1 shows the results and demonstrates the difference between the number of frequent itemsets generated using (i)Standard ARM using discrete intervals, (ii)CFARM with fuzzy partitions without normalization (CFARM1), and (iii)Fuzzy ARM with fuzzy partitions with normalization (CFARM2).

For standard ARM, the Apriori-TFP algorithm was used [6] with a range of support thresholds. As expected the number of frequent itemsets increases as the minimum support decreases. From the results, it is clear that standard ARM produces more frequent itemsets (and consequently rules) than fuzzy ARM (figure 1).

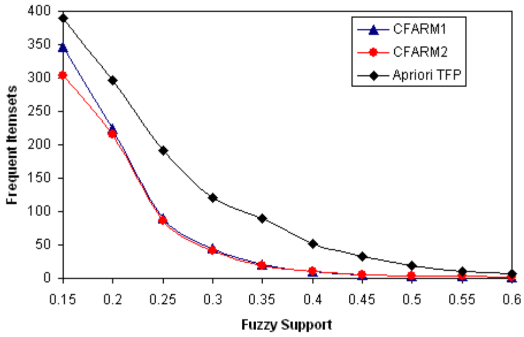


Fig. 1. Number of frequent Itemsets

This is because the frequent itemsets generated more accurately reflect the true patterns in the data set than the numerous artificial patterns resulting from the use of crisp boundaries in standard ARM. At low support threshold levels, the approach with normalization (CFARM2) starts to produce less frequent itemsets than the approach without normalization (CFARM1). This is because the average contribution to support counts per transaction is greater without using normalization than with normalization.

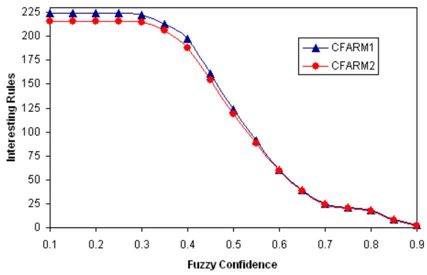


Fig. 2. No. of Interesting Rules using confidence

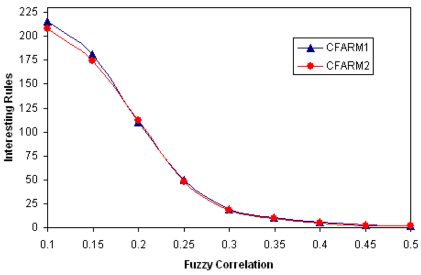


Fig. 3. No. of Interesting Rules using Correlation

Figures 2 and 3 shows the comparison of number of interesting rules generated using user specified fuzzy confidence and fuzzy correlation values respectively. In both cases, the number of interesting rules is less as using CFARM2; this is a direct consequence of the fact that CFARM 2 generates fewer frequent itemsets. Note that fewer, but arguably better, rules are generated using the correlation measure (Figure 3) than the confidence measure (Figure 2). The experiments show that using the proposed fuzzy normalization process less fuzzy ARs are generated. In addition, the novelty of the approach is its ability to analyse datasets comprised of composite items e.g. nutritional properties. Some example fuzzy ARs generated has the form:

IF *Protein* intake is *Low* THEN *Vitamin A* intake is *High.*

IF *Protein* intake is *High* AND *Vitamin A* intake is *Low* THEN *Fat* intake is *High.*

These rules would be useful in analysing customer buying patterns concerning their nutrition.

5.2. Experiment Two: (Performance Measures)

Experiment two investigated the effect on execution time by varying the number of attributes and the size of data (number of records) with and without normalization using a support threshold of 0.3, confidence 0.5 and correlation value to 0.25. Figure 4 shows the effect of increasing the number of records partitioned into 10 equal partitions 10K, 20K,..,100K with all 27 nutrients (properties) used. Both algorithms have similar timings with execution time scales linearly with the number of records. Figure 5 shows the effect on execution time using different numbers of attributes each with 5 fuzzy sets and thus uses 135 columns (27x5).

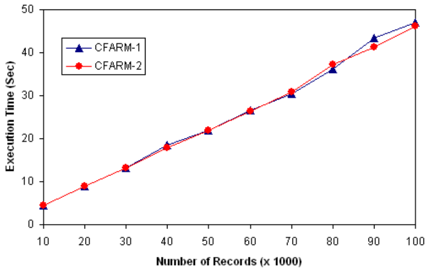


Fig. 4 : Execution time: No. of Records

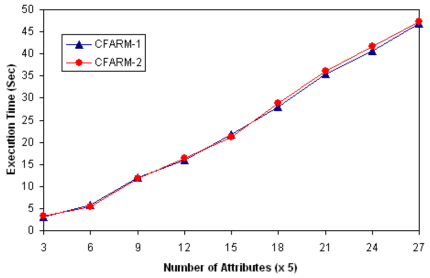


Fig. 5 : Execution time: No. of Attributes

6. Conclusion and future work

In this paper, we have presented a novel framework for extracting fuzzy ARs from “composite items” with quantitative properties (sub itemsets) using derived fuzzy sets. The CFARM algorithm produces a more succinct set of fuzzy association rules using fuzzy measures and correlation as the interestingness (certainty) measure and thus presents a new way for extracting association rules from items with properties. This is different from normal quantitative ARM. We also showed the experimental results with market basket data where edible items were used with nutritional content as properties. Largely, CFARM offers potential to apply this framework in varied applications with composite items.

References

1. Gyenesei, A.: A Fuzzy Approach for Mining Quantitative Association Rules, Acta Cybernetical, Vol. 15, (2) (2001) 305 – 320
2. Lee, C. H., Chen, M. S., Lin, C. R.: Progressive Partition Miner, an Efficient Algorithm for Mining General Temporal Association Rules, IEEE Trans. on Knowledge and Data Engineering, Vol. 15, (4) (2003) 1004-1017
3. Kuok, C., Fu, A., Wong, H.: Mining Fuzzy Association Rules in Databases, ACM SIGMOD Record Vol. 27, (1), (1998) 41-46
4. Dubois, D. E. Hüllermeier and H. Prade, A Systematic Approach to the Assessment of Fuzzy Association Rules, DM and Knowledge Discovery Journal, Vol. 13(2), (2006) 167-192
5. Bodon, F.: A Fast Apriori implementation, in: Proc. (FIMI'03), IEEE ICDM Workshop on Frequent Itemset Mining Implementations, Vol. 90, Florida, USA, (2003)
6. Coenen, F., Leng, P., Goulbourne, G.: Tree Structures for Mining Association Rules, Data Mining and Knowledge Discovery, Vol. 8, No. 1, (2004) 25 – 51
7. G. Chen and Q. Wei, Fuzzy Association Rules and the Extended Mining Algorithms, Information Sciences, Vol. 147, (1-4) (2002) 201– 28.
8. K. Wang, J. K. Liu and W. Ma, Mining the Most Reliable Association Rules with Composite Items, in: Proc. ICDMW'06, (2006), 749-754
9. M. Delgado, N. Marin, D. Sanchez and M. A. Vila, Fuzzy Association Rules, General Model and Applications, IEEE Transactions on Fuzzy Systems, 11(2) (2003) 214–225
10. M. Muyeba, M. Sulaiman, Z. Malik, and C. Tjortjis, Towards Healthy Association Rule Mining (HARM), A Fuzzy Quantitative Approach, Proc. IDEAL’06, LNCS, Vol. 4224, (2006) 1014-1022
11. R. Agrawal and R. Srikant, Quest Synthetic Data Generator. IBM Almaden Research Center.
12. R. Agrawal, T. Imielinski and A. Swami, Mining Association Rules Between Sets of Items in Large Databases, Proc. ACM SIGMOD Int. Conf. on Management of Data, Washington, D.C.(1993) 207-216
13. R. Srikant, and R. Agrawal, Mining Quantitative Association Rules in Large Relational Tables, Proc. ACM SIGMOD Conf. on Management of Data. ACM Press, Montreal, Quebec, (1996) 1 - 12
14. W. H. Au and K. Chan, Farm, A Data Mining System for Discovering Fuzzy Association Rules, Proc. 8th IEEE Int’l Conf. on Fuzzy Systems, Seoul, Korea, (1999) 1217-1222
15. W. Kim, E. Bertino and J. Garza, Composite objects revisited, ACM SIGMOD Record, Vol. 18, (2) (1989) 337-347
16. X. Ye and J. A. Keane, Mining Composite Items in Association Rules, Proc. IEEE Int. Conf. on Systems, Man and Cybernetics, (1997) 1367-1372