

## Genetic Approach for Fuzzy Mining Using Modified K-Means Clustering

**Shaikh Nikhat Fatma**

Department Of Computer , Mumbai University  
Pillai's Institute Of Information Technology,  
New Panvel, Mumbai , India

**Dr. J. W Bakal**

Department Of Information Technology ,  
Mumbai University  
Mumbai , India

### Abstract

A fuzzy-genetic data-mining algorithm for extracting both association rules and membership functions from quantitative transactions is shown in this paper. It used a combination of large 1-itemsets and membership-function suitability to evaluate the fitness values of chromosomes. The calculation for large 1-itemsets could take a lot of time, especially when the database to be scanned could not totally fit into main memory. In this system, an enhanced approach, called the Modified cluster-based fuzzy-genetic mining algorithm. It divides the chromosomes in a population into clusters by the modified k-means clustering approach and evaluates each individual according to both cluster and their own information.

**Keywords—** Modified k-means Clustering, data mining, fuzzy set, genetic algorithm, Fuzzy Association Rules, Quantitative transactions

### I. INTRODUCTION

Data is very important for every supermarket. Data that was measured in gigabytes until recently, is now being measured in terabytes, and will soon approach the pentabytes range. In order to achieve our goals, we need to fully exploit this data by extracting all the useful information from it. This information can be extracted using data mining. Data mining is the use of automated data analysis techniques to uncover previously undetected relationships among data items. Data mining often involves the analysis of data stored in large database. The typical business decisions that the management of a super market has to make is, what to put on sale in what quantity. Analysis of past transaction data is a commonly used approach in order to improve the quality of such decisions. Extraction of frequent item sets is essential towards mining interesting patterns from datasets. A typical usage scenario for searching frequent patterns is the so called "market basket analysis" that involves analyzing the transactional data of a super market in order to determine which products are purchased together and how often and also examine customer purchase preferences.

Mining association rules between sets of items in large databases was first stated by Agrawal, Imelinski and Swami in 1993 and it opened brand

new family algorithms .Apriori algorithm is probably the most used algorithm in association rules mining. At present, more and more databases containing large quantities of data are available. These industrial, medical, financial and other databases make an invaluable resource of useful knowledge. The task of extraction of useful knowledge from databases is challenged by the techniques called data-mining techniques. One of the widely used data-mining techniques is association rules mining. Data Mining is commonly used in attempts to induce association rules from transaction data. Data mining is the process of extracting desirable knowledge or interesting patterns from existing databases for specific purposes. Many types of knowledge and technology have been proposed for data mining. Among them, finding association rules from transaction data is most commonly seen. Most studies have shown how binary valued transaction data may be handled. . For example, there may exist some implicitly useful knowledge in a large database containing millions of records of customers' purchase orders over the last five years. The knowledge can be found out using appropriate data-mining approaches. Data mining is most commonly used in attempts to induce association rules from transaction data. An association rule is an expression  $X \rightarrow F$ , where  $X$  is a set of items and  $Y$  is a single item. It means in the set of transactions, if all the items in  $X$  exist in a transaction, then  $Y$  is also in the transaction with a high probability. For example, assume whenever customers in a supermarket buy bread and butter, they will also buy milk. From the transactions kept in the supermarkets, an association rule such as "*Bread and Butter*  $\rightarrow$  *Milk*" will be mined out. Transactions with quantitative values are however commonly seen in real-world applications. Therefore a fuzzy mining algorithm by which each attribute used only the linguistic term with the maximum cardinality in the mining process. The number of items was thus the same as that of the original attributes, making the processing time reduced. The fuzzy association rules derived in this way are not complete. The algorithm can derive a more complete set of rules but with more computation time.[1][2]

Fuzzy set theory is being used more and more frequently in intelligent systems because of its

simplicity and similarity to human reasoning. The theory has been applied in fields such as manufacturing, engineering, diagnosis, economics. Several fuzzy learning algorithms for inducing rules from given sets of data have been designed and used to good effect with specific domains. The mined rules are expressed in linguistic terms, which are more natural and understandable for human beings.

## II. EXISTING SYSTEM

Data mining is commonly used in attempts to induce association rules from transaction data. Most previous studies focused on binary-valued transaction data. Transactions in real-world applications, however, usually consist of quantitative values. In the past, a fuzzy-genetic data-mining algorithm for extracting both association rules and membership functions from quantitative transactions. It used a combination of large 1-itemsets and membership-function suitability to evaluate the fitness values of chromosomes. The calculation for large 1-itemsets could take a lot of time, especially when the database to be scanned could not totally fit into main memory. Therefore, the k-means clustering approach was used to gather similar chromosomes into groups. Since the number for scanning a database will decrease, the evaluation cost can thus be reduced.

The existing system maintains a population of sets of membership functions, and use the genetic algorithm to automatically derive the resulting one. It will first transform each set of membership functions into a fixed-length string.

Each chromosome will represent a set of membership functions used in fuzzy mining. Then, it will use the K-means clustering approach to gather similar chromosomes into groups. All the chromosomes in a cluster will use the number of large 1-itemsets derived from the representative chromosome in the cluster and their own suitability of membership functions to calculate their fitness values. Since the number for scanning a database will decrease, the evaluation cost can thus be reduced. The evaluation results can be utilized to choose appropriate chromosomes for mating in the next generation. The offspring membership function sets will then undergo recursive evolution until a good set of membership functions has been obtained. Finally, the derived membership functions will be used to mine fuzzy association rules.

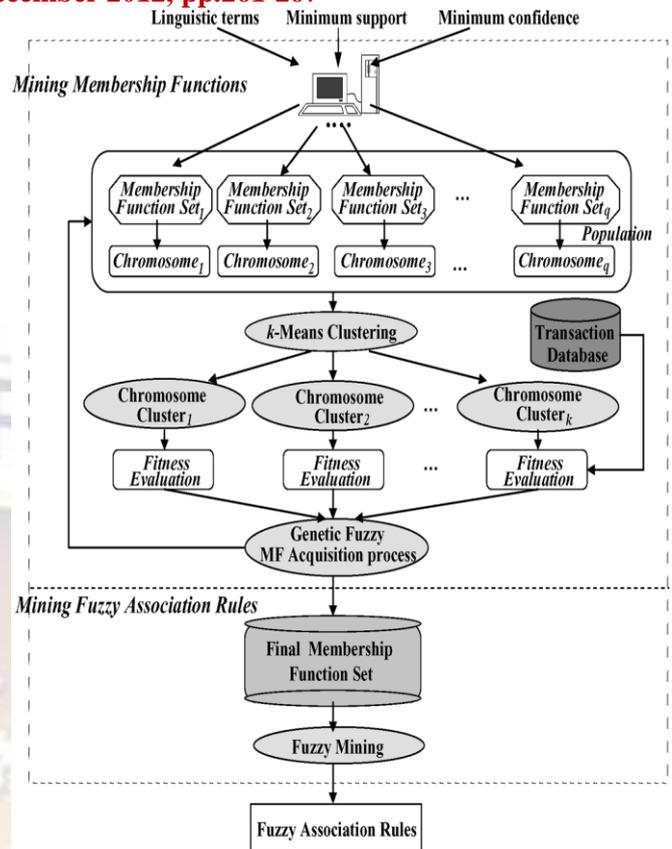


Fig 1: Framework for the existing cluster based fuzzy genetic data mining system

## III. PROPOSED SYSTEM

### A) Chromosome Representation:

Each set of membership functions is encoded. All pairs of (c.w)'s for a certain item are concatenated to represent its membership functions. Since and are both numeric values, a chromosome is thus encoded as a fixed-length real-number string rather than a bit string.[1][2]

### B) Initial Population

A genetic algorithm requires a population of feasible solutions to be initialized and updated during the evolution process. As mentioned above, each individual within the population is a set of isosceles-triangular membership functions. Each membership function corresponds to a linguistic term in a certain item. The initial set of chromosomes is randomly generated with some constraints or forming feasible membership function.[1][2]

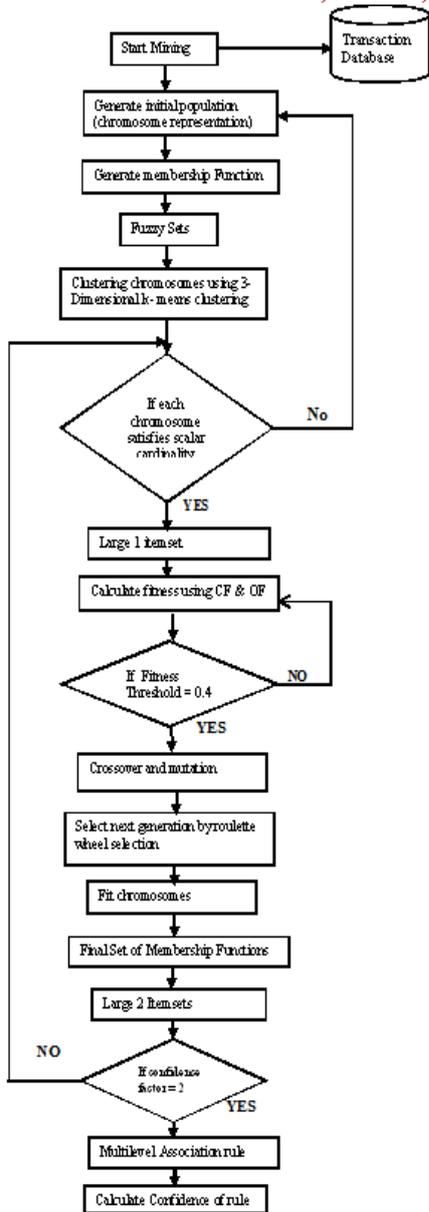


Fig 2: Flowchart for the proposed cluster based fuzzy genetic data mining system

### C) Fitness and Selection

In order to develop a good set of membership functions from an initial population, the genetic algorithm selects *parent* membership function sets with its probability values for mating. An evaluation function is then used to qualify the derived membership function sets. It is shown as follows:

$$f(C_q) = \frac{|L_{1q}|}{\text{Suitability}(C_q)}$$

Where  $|L_{1q}|$  is the number of large 1-itemsets obtained by using the set of membership functions in chromosome  $C_q$  and suitability of  $(C_q)$  represents the shape suitability of  $(C_q)$ . Suitability of  $(C_q)$  is defined as:

$$\sum_{j=1}^m [\text{overlap\_factor}(C_{qj}) + \text{coverage\_factor}(C_{qj})]$$

Where  $m$  is the number of item.  $\text{Overlap\_factor}(C_{qj})$  represents the overlap factor of the membership functions for an item  $I_j$  in the chromosome  $C_q$  and is defined as

$$\text{overlap\_factor}(C_{qj}) = \sum_{k \neq i} \left[ \max \left( \left( \frac{\text{overlap}(R_{jk}, R_{ji})}{\min(w_{jk}, w_{ji})} \right), 1 \right) - 1 \right]$$

Where  $(R_{j1}, R_{j2}, \dots, R_{ji})$  is the coverage range of the membership functions  $I_j$ , is the number of membership functions for  $I_j$  and  $\max(I_j)$  is the maximum quantity of in the transactions.

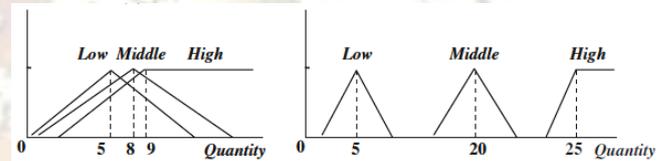


Fig 3: Two bad membership functions

### D) Clustering Chromosomes

The coverage factors and overlap factors and the average of both of all the chromosomes are used to form appropriate clusters. The Modified k-means clustering approach is adopted here to cluster chromosomes. Since the chromosomes with similar coverage factors and overlap factors will form a cluster, they will have nearly the same shape of membership functions and induce about the same number of large 1-itemsets. For each cluster, the chromosome which is the nearest to the cluster center is thus chosen to derive its number of large 1-itemsets. All chromosomes in the same cluster then use the number of large 1-itemsets derived from the representative chromosome as their own. Finally, each chromosome is evaluated by this number of large 1-itemsets divided by its own suitability value.

### E) Genetic Operators

Two genetic operators, *crossover* proposed in and the *mutation* are used. Assume there are two parent chromosomes

$$C_u^{rt} = (c_1, \dots, c_h, \dots, c_z) \text{ and } C_w^{rt} = (c_1, \dots, c_h, \dots, c_z)$$

The crossover operator will generate the following four candidate chromosomes from them:

- 1)  $C_1^{t+1} = (c_{11}^{t+1}, \dots, c_{1h}^{t+1}, \dots, c_{1Z}^{t+1})$ , where  $c_{1h}^{t+1} = dc_h + (1-d)c'_h$ ;
- 2)  $C_2^{t+1} = (c_{21}^{t+1}, \dots, c_{2h}^{t+1}, \dots, c_{2Z}^{t+1})$ , where  $c_{2h}^{t+1} = dc'_h + (1-d)c_h$ ;
- 3)  $C_3^{t+1} = (c_{31}^{t+1}, \dots, c_{3h}^{t+1}, \dots, c_{3Z}^{t+1})$ , where  $c_{3h}^{t+1} = \min\{c_h, c'_h\}$ ;
- 4)  $C_4^{t+1} = (c_{41}^{t+1}, \dots, c_{4h}^{t+1}, \dots, c_{4Z}^{t+1})$ , where  $c_{4h}^{t+1} = \max\{c_h, c'_h\}$ .

Where the parameter  $d_i$  is either a constant or a variable whose value depends on the age of the population.

#### IV. ALGORITHM

##### INPUT:

A body of  $n$  quantitative transactions, a set of  $m$  items, each with a number of linguistic terms, a parameter  $k$  for Modified  $k$ -means clustering, a population size  $P$ , a crossover rate  $P_c$ , a mutation rate  $P_m$ , a support threshold  $\alpha$ , and a confidence threshold  $\lambda$  and a fitness threshold.

##### OUTPUT:

A set of multilevel fuzzy association rules with its associated set of membership functions and its confidence.

**Step 1:** Randomly generate a population of  $P$  individuals, each individual is a set of membership functions for all the  $m$  items.

**Step 2:** Encode each set of membership functions into a string representation according to the schema explained above.

**Step 3:** Calculate the coverage\_factor and the overlap\_factor of each chromosome using the formulas defined above.

**Step 4:** Divide the chromosomes into  $K$  clusters by the Modified  $k$ -means clustering approach based on the three attributes (coverage factors and overlap factors and average of both) Find out the representative chromosome in each cluster, which is the nearest to the center.

**Step 5:** Calculate the number of large 1-itemsets for each representative chromosome.

**Step 6:** Calculate the fitness value of each chromosome using the number of large 1-itemsets of its representative chromosome and the suitability value of its own according to the formula defined above.

**Step 7:** Execute the crossover operation on the population.

**Step 8:** Execute the mutation operation on the population.

**Step 9:** Use the Roulette-wheel selection operation to choose appropriate individuals for the next generation.

**Step 10:** If the termination criterion is not satisfied, go to Step 6; otherwise, do the next step.

**Step 11:** Get the set of membership functions with the highest fitness value.

**Step 12:** This set of membership functions are then used to mine fuzzy association rules from the given database.

**To mine fuzzy association rules from the above membership functions the algorithm works as follows:**

**Step 1:** Transform the quantitative values of each transaction, for each attribute into a fuzzy set using above found membership functions.

**Step 2:** Calculate the count of each attribute region (linguistic term)  $R_{jk}$  in the transaction data.

**Step 3:** Collect each attribute region (linguistic term) to form the candidate set  $C1$ .

**Step 4:** Check whether  $count_{jk}$  of each  $R_{jk}$  is larger than or equal to the predefined minimum support value  $\alpha$ . If  $R_{jk}$  satisfies the above condition, put it in the set of large 1-itemsets ( $L1$ ).

**Step 5:** IF  $L1$  is not null, then do the next step; otherwise, exit the algorithm.

**Step 6:** Set  $r=1$ , where  $r$  is used to represent the number of items kept in the current large itemsets.

**Step 7:** Join the large itemsets  $Lr$  to generate the candidate set  $Cr+1$  in a way similar to that in the apriori algorithm, except that two regions (linguistic terms) belonging to the same attribute cannot simultaneously exist in an itemset in

$Cr+1$ . Restated, the algorithm first joins  $Lr$  and  $Lr$  under the condition that  $r-1$  items in the two itemsets are the same and the other one is different. It then keeps in  $Cr+1$  the itemsets which have all their sub-itemsets of  $r$  items existing in  $Lr$  and do not have any two items  $R_{jp}$  and  $R_{jq}$  ( $p \neq q$ ) of the same attribute  $R_j$ .

**STEP 8:** Do the following substeps for each newly formed candidate itemset:

(a) Calculate the fuzzy membership value of each transaction datum. Here, the minimum operator is used for the intersection.

(b) Calculate the scalar cardinality (count) of each candidate 2-itemset in the transaction data.

(c) Check whether these counts are larger than or equal to the predefined minimum support value  $\lambda$ , put  $s$  in  $Lr+1$ .

**Step 9:** If  $Lr+1$  is null, then do the next step; otherwise, set  $r=r+1$  and repeat Steps 6 to 8.

**Step 10:** Collect the large itemsets together.

**Step 11 :** Construct association rules for each large  $q$ -itemset  $s$  with items(  $s1, s2, \dots, sq$ ),  $q \geq 2$ , using the following substeps:

(a) Form each possible association rule as follows:

$s1 \wedge \dots \wedge sk-1 \wedge sk+1 \wedge \dots \wedge sq \rightarrow sk$ ,  $k = 1$  to  $q$ .

(b) Calculate the confidence factors for the above association rules.

**Step 12:** Output the association rules with confidence values larger than or equal to the predefined confidence threshold  $\lambda$ .

**V. EXAMPLE**

In this section, a simple example is given to illustrate the proposed Modified cluster-based fuzzy-genetic mining algorithm. Assume there are ten items in a transaction database: milk, bread, cookies, beverage, chocolate, icecream, coldrink, curd, fruit, butter. Since the item *milk* has three possible linguistic terms, *Low*, *Middle* and *High*, the membership functions for *milk* are thus encoded as (2, 11, 13, 11, 24, 11) for chromosome C2.

**1. Randomly generate chromosomes**

**C0:** 1 5 6 5 11 5, 1 10 11 10 21 10, 1 8 9 8 17 8, 0 6 6 6 12 6, 0 3 3 3 6 3, 0 3 3 3 6 3, 2 7 9 7 16 7, 2 3 5 3 8 3, 0 4 4 4 8 4, 1 3 4 3 7 3

**C1:** 1 9 10 9 19 9, 0 7 7 7 14 7, 2 11 13 11 24 11, 1 4 5 4 9 4, 2 3 5 3 8 3, 1 3 4 3 7 3, 2 4 6 4 10 4, 0 6 6 6 12 6, 1 3 4 3 7 3, 0 7 7 7 14 7

**C2:** 2 11 13 11 24 11, 2 14 16 14 30 14, 1 8 9 8 17 8, 1 5 6 5 11 5, 0 5 5 5 10 5, 0 2 2 2 4 2, 0 3 3 3 6 3, 1 4 5 4 9 4, 0 4 4 4 8 4, 0 7 7 7 14 7

**C3:** 2 11 13 11 24 11, 2 5 7 5 12 5, 1 9 10 9 19 9, 1 5 6 5 11 5, 0 3 3 3 6 3, 1 2 3 2 5 2, 0 5 5 5 10 5, 0 7 7 7 14 7, 1 4 5 4 9 4, 0 5 5 5 10 5

**C4:** 0 10 10 10 20 10, 2 12 14 12 26 12, 0 8 8 8 16 8, 1 7 8 7 15 7, 0 4 4 4 8 4, 0 3 3 3 6 3, 2 5 7 5 12 5, 0 4 4 4 8 4, 1 2 3 2 5 2, 1 6 7 6 13 6

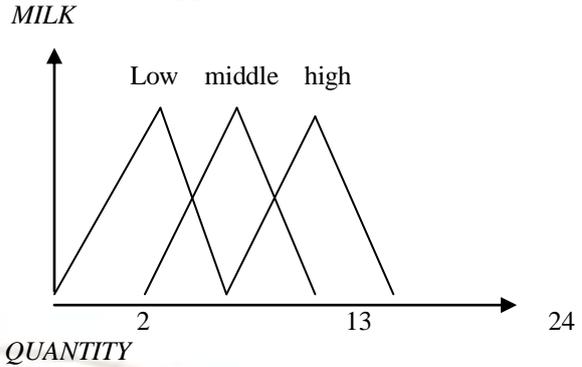
**C5:** 1 7 8 7 15 7, 0 5 5 5 10 5, 2 4 6 4 10 4, 2 10 12 10 22 10, 0 4 4 4 8 4, 0 2 2 2 4 2, 1 7 8 7 15 7, 1 7 8 7 15 7, 1 4 5 4 9 4, 0 5 5 5 10 5

**C6:** 2 3 5 3 8 3, 4 13 17 13 30 13, 1 9 10 9 19 9, 0 6 6 6 12 6, 1 6 7 6 13 6, 1 3 4 3 7 3, 1 7 8 7 15 7, 0 3 3 3 6 3, 1 3 4 3 7 3, 0 7 7 7 14 7

**C7:** 2 7 9 7 16 7, 0 14 14 14 28 14, 0 5 5 5 10 5, 0 11 11 11 22 11, 2 3 5 3 8 3, 1 3 4 3 7 3, 1 3 4 3 7 3, 1 4 5 4 9 4, 0 3 3 3 6 3, 0 8 8 8 16 8

**C8:** 1 4 5 4 9 4, 0 12 12 12 24 12, 2 6 8 6 14 6, 1 9 10 9 19 9, 2 5 7 5 12 5, 0 3 3 3 6 3, 1 4 5 4 9 4, 2 4 6 4 10 4, 1 4 5 4 9 4, 0 7 7 7 14 7

**C9:** 0 5 5 5 10 5, 2 10 12 10 22 10, 2 11 13 11 24 11, 2 11 13 11 24 11, 1 5 6 5 11 5, 0 2 2 2 4 2, 1 7 8 7 15 7, 2 3 5 3 8 3, 1 3 4 3 7 3, 0 5 5 5 10 5



**Fig No.6:** The membership functions for milk in C2

**2. Calculation of overlap factor, coverage factor and both**

The overlap factor in suitable( $Cq$ ) is designed for avoiding the first bad case, and the coverage factor is for the second one.[2]

The dimensions used for clustering are overlap factor, coverage factor and average of both. Therefore, the calculation of minimum distance is as follows

$$\sqrt{(x1 - x2)^2 + (y1 - y2)^2 + (z1 - z2)^2}$$

**Table No.1 :** Calculation of CF & OF & Average

CHROMOSOME	OVERLAP FACTOR	COVERAGE FACTOR	AVERAGE OF OF & CF
C0	0.0	0.9478684	1.4369881
C1	0.9478684	0.0	0.48911962
C2	1.4369881	0.48911962	0.0
C3	0.7182525	0.22961593	0.7187355
C4	1.6091698	0.66130143	0.17218181
C5	0.66150403	0.28636444	0.775484
C6	0.69528997	0.25257847	0.7416981
C7	0.996007	0.048138615	0.440981
C8	1.2431476	0.29527926	0.19384035
C9	1.3238536	0.37598512	0.113134496

**3. Clustering Chromosomes using 3-dimensional clustering**

K=3

- Chromosome no 0 In Cluster No. 0
- Chromosome no 1 In Cluster No. 1
- Chromosome no 2 In Cluster No. 2
- Chromosome no 3 In Cluster No. 1
- Chromosome no 4 In Cluster No. 2
- Chromosome no 5 In Cluster No. 1
- Chromosome no 6 In Cluster No. 1
- Chromosome no 7 In Cluster No. 1
- Chromosome no 8 In Cluster No. 2
- Chromosome no 9 In Cluster No. 2

**4. Suitability of Chromosomes**

The suitability factor used in the fitness function can reduce the occurrence of the two bad

kinds of membership functions shown in Fig. 3, where the first one is too redundant, and the second one is too separate.

**Table No.2 : Suitability of Chromosomes**

Chromosome No.	Suitability
0	9.663095
1	8.815296
2	8.377814
3	9.020671
4	8.22381
5	9.071428
6	9.041209
7	8.77224
8	8.55119
9	8.479005

**5. Fitness of Chromosomes**

The fitness of each set of membership functions is evaluated by the number of large 1-itemsets generated by executing part of the previously proposed fuzzy mining algorithm. Using the number of large 1-itemsets can achieve a trade-off between execution time and rule interestingness. Usually, a larger number of 1-itemsets will result in a larger number of all itemsets with a higher probability, which will thus usually imply more interesting association rules. The evaluation by 1-itemsets is, however, faster than that by all itemsets or interesting association rules.[2]

**Table No. 3: Fitness of Chromosomes**

Chromosome No.	Fitness
0	2.897622
1	2.8359797
2	3.3421605
3	2.7714126
4	3.4047477
5	2.7559056
6	2.765117
7	2.8498993
8	3.2743979
9	3.3022742

**6. Multilevel Association Rule and Confidence**

**IF MILKL AND BREADL THEN COOKIESL 30.0 %**  
**IF MILKL AND BREADL AND COOKIESL THEN ChoclatesL 40.0 %**

**VI. COMPARISON OF THE EXISTING AND PROPOSED SYSTEM**

For clustering the parameter k was taken as 3 and comparison was done between the k-means algorithm and the Modified k-means clustering and we got the following results.

**Table No. 4: Comparison of clusters formed**

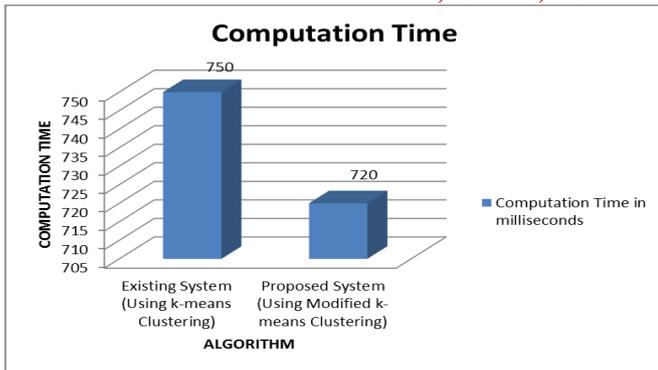
CLUSTERS FORMED BY 2-DIMENSIONAL K-MEANS ALGORITHM	CLUSTERS FORMED BY MODIFIED K-MEANS CLUSTERING
Chromosome no 0 In Cluster No. 0	Chromosome no 0 In Cluster No. 0
Chromosome no 1 In Cluster No. 1	Chromosome no 1 In Cluster No. 1
Chromosome no 2 In Cluster No. 2	Chromosome no 2 In Cluster No. 2
Chromosome no 3 In Cluster No. 1	Chromosome no 3 In Cluster No. 1
Chromosome no 4 In Cluster No. 2	Chromosome no 4 In Cluster No. 1
Chromosome no 5 In Cluster No. 1	Chromosome no 5 In Cluster No. 2
Chromosome no 6 In Cluster No. 1	Chromosome no 6 In Cluster No. 1
Chromosome no 7 In Cluster No. 1	Chromosome no 7 In Cluster No. 0
Chromosome no 8 In Cluster No. 2	Chromosome no 8 In Cluster No. 1
Chromosome no 9 In Cluster No. 2	Chromosome no 9 In Cluster No. 0

In order to check performance of the Association algorithms, we have applied the algorithm to item dataset. Comparison is done by considering the Computation Time.

**Table No.5 : Comparison of Computation Time**

SR NO.	ALGORITHM	COMPUTATION TIME
1	Existing system (using k-means clustering)	750 milliseconds
2	Proposed system (using modified k-means clustering)	720 milliseconds

Following graph depicts comparison of Computational time of existing system and proposed system with respect to Computational time for database of 10 items and 20 transactions .X axis in Graph denotes algorithm and y axis denotes Computational time. Computational time is measured in terms of milliseconds.



**Graph 1 . Comparison of Algorithm using K-means and Modified K-means Clustering**

## VII. CONCLUSION

This project presents an optimization method developed to deal with a data mining problem. Decision making in business sector is considered as one of the critical tasks. The objective is to provide a tool to help experts to find associations between the items bought by a customer in supermarket. An association rule may be more interesting if it reveals relationship among some useful concepts such as quantity of items bought by customer rather than which item is bought by the customer that is by using fuzzy mining. The favorite things of customers change all the time that is the membership functions should be adjusted dynamically, which can be done by genetic algorithm. To deal with all these aspects the fuzzy genetic association rule mining using modified k-means clustering is used. To achieve this, the algorithm has been developed in two parts. The first part finds out fit membership functions using genetic algorithm and modified k-means algorithm. The second part finds association rule (second level) using these fit membership functions and the confidence of that rule. This algorithm is tested on 10 items and 20 transactions.

## REFERENCES

[1] Chun-Hao Chen, Tzung-Pei Hong, "Cluster-Based Evaluation in Fuzzy-Genetic Data Mining", IEEE transactions on fuzzy

systems, Vol. 16, No. 1, February 2008 249, pp. 249-262

- [2] T. P. Hong, C. H. Chen, Y. L. Wu, and Y. C. Lee, "A GA-based fuzzy mining approach to achieve a trade-off between number of rules and suitability of membership functions," *Soft Computing*, vol. 10, no. 11, pp. 1091–1101, 2006.
- [3] Hung-Pin Chiu, Yi-Tsung Tang, "A Cluster-Based Mining Approach for Mining Fuzzy Association Rules in Two Databases", *Electronic Commerce Studies*, Vol. 4, No.1, Spring 2006, Page 57-74.
- [4] Tzung-Pei Hong, Chan-Sheng Kuo, Sheng-Chai Chi, "Trade-Off Between computation time and number of rules for fuzzy mining from quantitative data", *International Journal of Uncertainty, Fuzziness and Knowledge-Based systems*, Vol. 9, No. 5, 2001, page 587- 604.
- [5] M. Sulaiman Khan, Maybin Mueyba, Frans Coenen, "Fuzzy Weighted Association Rule Mining with Weighted Support and Confidence Framework", *The University of Liverpool, Department of Computer Science, Liverpool, UK*.
- [6] H. Ishibuchi and T. Yamamoto, "Rule weight specification in fuzzy rule-based classification systems," *IEEE Trans. on Fuzzy Systems*, Vol. 13, No. 4, pp. 428-435, August 2005.
- [7] Miguel Delgado, Nicolás Marín, Daniel Sánchez, and María-Amparo Vila, "Fuzzy Association Rules: General Model and Applications", *IEEE transactions on fuzzy systems*, vol. 11, no. 2, April 2003
- [8] Tzung-Pei Hong, Li-Huei Tseng and Been-Chian Chien, "Learning Fuzzy Rules from Incomplete Quantitative Data by Rough Sets",
- [9] H. J. Zimmermann, "Fuzzy set theory and its applications", *Kluwer Academic Publisher*, Boston, 1991.