EFFECTUAL EVALUATION OF FUZZY GENETIC METHODOLOGY FOR ASSORTED APPLICATIONS

Aduthotla Ramesh[1], Dr Birjesh N Chowda[2]
[1]Research Scholar, Dept of Mathematics, JJT University, Jhunjhunu, Rajasthan, India
[2] Professor and Head, Dept of Humanities,
Jayaprakash Narayan College of Engineering, Mahabubnagar, Telangana, India

Abstract
Genetic Fuzzy frameworks are Fuzzy structures made by utilizing Genetic estimations or genetic programming, which imitate the philosophy of trademark movement, to see its structure and parameter. As to therefore perceiving and assembling a Fuzzy structure, given the strange condition of nonlinearity of the yield, standard straight streamlining contraptions have several controls. Along these lines, in the game plan of delicate enrolling, genetic estimations (GAs) and natural programming (GP) techniques have been utilized enough to see structure and parameters of Fuzzy frameworks.
Keywords: Fuzzy Genetic, Fuzzy Logic, Fuzzy Algorithms, Fuzzy Methodology

Introduction

Fuzzy structures are basic systems to address and prepare semantic data, with instruments to supervise instability and imprecision. For example, the errand of demonstrating a driver ending an auto fuses more fundamental bother in recording a traditionalist consistent model as the outline winds up being more unequivocal. Notwithstanding, the level of bother is less utilizing crucial etymological standards, which are themselves woolen. With such stunning properties, Fuzzy frameworks have been widely and enough connected with control, strategy and indicating issues (Mamdani, 1974) (Klir and Yuan, 1995) (Pedrycz and Gomide, 1998). However constrained in its course of action, the ID of a delicate structure is a really complex undertaking that joins the perceiving proof of (a) the information and yield factors, (b) the lead base (learning base), (c) as far as possible and (d) the mapping parameters.

All things considered the administer base includes a couple IF-THEN models, partner input(s) and output(s). A principal direct of a woolen controller could be:

On the off chance that (TEMPERATURE = HOT) THEN (COOLING = HIGH)

The numerical effect/which techniques for this control relies on how the intrigue parts of HOT and HIGH are formed and depicted. The change and perceiving check of a woolen framework can be distributed into (a) the structure and (b) the parameter unmistakable affirmation of a Fuzzy framework. The structure of a fluffy framework is granted by the information and yield factors and the control base, while the parameters of a Fuzzy structure are the run parameters (depicting as far as possible, the social event head and the suggestion work) and the mapping parameters identified with the mapping of another set to a Fuzzy set, and the an alternate way. (Bastian, 2000).

Much work has been done to make or alter strategies that can do really seeing a woolen framework from numerical information. Especially in the structure of delicate algorithm, imperative approaches have been proposed with the target of building Fuzzy frameworks by methods for genetic estimations (GAs) or Genetic programming (GP).

Given the anomalous condition of nonlinearity of the yield of a delicate framework, standard direct progress instruments do have their snags. Genetic algorithms have shown to be a strong and capable instrument to perform attempts, for example, the season of fluffy regulate base, change of Fuzzy control bases, time of help breaking points, and tuning of enrollment limits (Cordón et al., 2001a). Every one of these errands can be considered as streamlining or intrigue shapes inside sweeping strategy spaces (Bastian and Hayashi, 1995) (Yuan and Zhuang, 1996) (Cordón et al., 2001b).

Genetic programming for Fuzzy framework

While Genetic tallies are fit instruments to perceive the delicate help parts of a pre-depicted lead base, they have their constraint particularly when it also comes to see the information and yield components of a fluffy framework from a given strategy of information. Genetic programming has been utilized to perceive the information factors, the supervise base and besides the included collaboration segments of a fluffy model (Bastian, 2000)

Let us take support as 50%. By which each item should appear more than 50% in dataset. As we have 4 transactions 50% of 4 is 2, so each of A, B, C, D, E should appear 2 or more times. If not it is removed (pruning). In L1 item D is pruned.

Now we are left with A, B, C, E. First 2 Combinations are generated for these items and their support is checked. If any combination has a lesser support than a specified of 50%, we prune it.

If we have some combinations meeting the above criteria, we proceed to next combination (join). In L2 we have 2 combinations meeting support, so we generate 3 combinations and check their support and so on.

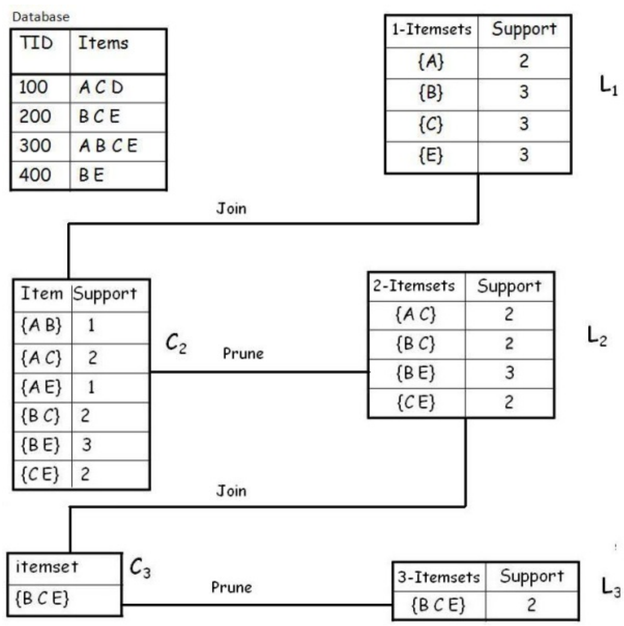

Figure 1. Rule Mining Approach

In the generated 3 combination sets only B,C,E meet the support count 2 (50%), and in the 4 combination set none meets the support , so we backtrack to retain B,C,E as frequent item set

As the name explains it's a reverse of Rule Mining algorithm where we first generate 5 combinations of A,B,C,D,E which is self set and check for its support .

It doesn't meet the support count, so we proceed to generate 4 combination sets then check for each individual support, in which nothing meets the support, so we proceed to 3 combination sets.

In 3 combination B,C,E meets the support given, so the algorithm is halted giving the result requent item set as B,C,E

Transactions	Items bought
T1	Item1, item2, item3
T2	Item1, item2
T3	Item2, item5
T4	Item1, item2, item5

Rule Mining Algorithm It is a classic algorithm used in data mining for learning association rules. It is nowhere as complex as it sounds, on the contrary it is very simple.

Suppose we have records of large number of transactions at a shopping center as follows:

Learning association rules basically means finding the items that are purchased together more frequently than others.
For example in the above table we can see Item1 and item2 are bought together frequently.
Shopping centers use association rules to place the items next to each other so that users buy more items. If you are familiar with data mining you would know about the famous beer-diapers-Wal-Mart story. Basically Wal-Mart studied their data and found that on Friday afternoon young American males who buy diapers also tend to buy beer. So Wal-Mart placed beer next to diapers and the beer-sales went up. This is famous because no one would have predicted such a result and that's the power of data mining. We can use Google for this if we are interested in further details.

Also if we are familiar with Amazon, they use association mining to recommend the items based on the current item we are browsing/buying.

Another application is the Google auto-complete, where after we type in a word it searches frequently associated words that user type after that particular word. So as I said Rule Mining is the classic and probably the most basic algorithm to do it.

Transaction ID	Items Bought
T1	{DataItem-1, DataItem-2, DataItem-3, DataItem-4, DataItem-5, DataItem-6}
T2	{DataItem-9, DataItem-2, DataItem-3, DataItem-4, DataItem-5, DataItem-6}
T3	{DataItem-1, DataItem-11, DataItem-4, DataItem-5}
T4	{DataItem-1, DataItem-7, DataItem-8, DataItem-4, DataItem-6}
T5	{DataItem-8, DataItem-2, DataItem-2, DataItem-4, DataItem-10, DataItem-5}

Now, we follow a simple golden rule: we say an item/itemset is frequently bought if it is bought at least 60% of times. So for here it should be bought at least 3 times.
For simplicity
M = DataItem-1
O = DataItem-2
And so on……
Original table:

Transaction ID	Items Bought
T1	{M, O, N, K, E, Y }
T2	{D, O, N, K, E, Y }
T3	{M, A, K, E}
T4	{M, U, C, K, Y }
T5	{C, O, O, K, I, E}

Step 1: Count the number of transactions in which each item occurs, Note 'O=DataItem-2' is bought 4 times in total, but, it occurs in just 3 transactions.

Item	No of transactions
M	3
O	3
N	2
K	5
E	4
Y	3
D	1
A	1
U	1
C	2
I	1

Step 2: Now remember we said the item is said frequently bought if it is bought at least 3 times. So in this step we remove all the items that are bought less than 3 times from the above table and we are left with

Item	Number of transactions
M	3
O	3
K	5
E	4
Y	3

This is the single items that are bought frequently. Now let's say we want to find a pair of items that are bought frequently. We continue from the above table (Table in step 2)

Step 3: We start making pairs from the first item, like MO,MK,ME,MY and then we start with the second item like OK,OE,OY. We did not do OM because we already did MO when we were making pairs with M and buying a DataItem-1 and DataItem-2 together is same as buying DataItem-2 and DataItem-1 together. After making all the pairs we get,

Item pairs
MO
MK
ME
MY
OK
OE
OY
KE
KY
EY

Step 4: Now we count how many times each pair is bought together. For example M and O is just bought together in {M,O,N,K,E,Y}

While M and K is bought together 3 times in {M,O,N,K,E,Y}, {M,A,K,E} AND {M,U,C, K, Y}

After doing that for all the pairs we get

Item Pairs	Number of transactions
MO	1
MK	3
ME	2
MY	2
OK	3
OE	3
OY	2
KE	4
KY	3
EY	2

Step 5: Golden rule to the rescue. Remove all the item pairs with number of transactions less than three and we are left with

Item Pairs	Number of transactions
MK	3
OK	3
OE	3
KE	4
KY	3

These are the pairs of items frequently bought together.
Now let's say we want to find a set of three items that are brought together.
We use the above table (table in step 5) and make a set of 3 items.

Step 6: To make the set of three items we need one more rule (it's termed as self-join),

It simply means, from the Item pairs in the above table, we find two pairs with the same first Alphabet, so we get

- OK and OE, this gives OKE
- KE and KY, this gives KEY

Then we find how many times O,K,E are bought together in the original table and same for K,E,Y and we get the following table

Item Set	Number of transactions
OKE	3
KEY	2

While we are on this, suppose you have sets of 3 items say ABC, ABD, ACD, ACE, BCD and you want to generate item sets of 4 items you look for two sets having the same first two alphabets.

- ABC and ABD -> ABCD
- ACD and ACE -> ACDE

And so on … In general we have to look for sets having just the last alphabet/item different.

Step 7: So we again apply the golden rule, that is, the item set must be bought together at least 3 times which leaves us with just OKE, Since KEY are bought together just two times.

Thus the set of three items that are bought together most frequently are O,K,E.

MinMetric: 0.5
=== Run information ===

Scheme: weka.associations.Rule Mining -N 10 -T 0 -C 0.5 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1
Relation: Query Result
Instances: 500

Attributes: 6
p1
p2
p3
p4
p5
p6
=== Associator model (full training set) ===

Rule Mining
=======
Minimum support: 0.25 (125 instances)
Minimum metric <confidence>: 0.5
Number of cycles performed: 15

Generated sets of large itemsets:

Size of set of large itemsets L(1): 13

Size of set of large itemsets L(2): 6

Best rules found:
1. p4=Coffee 244 ==> p6=Bread 133 conf:(0.55)
2. p6=Fruit 243 ==> p4=Butter 132 conf:(0.54)
3. p5=Chocolate 250 ==> p6=Bread 135 conf:(0.54)
4. p5=Chocolate 250 ==> p4=Butter 134 conf:(0.54)
5. p6=Fruit 243 ==> p5=Milk 128 conf:(0.53)
6. p6=Bread 257 ==> p5=Chocolate 135 conf:(0.53)
7. p4=Coffee 244 ==> p5=Milk 128 conf:(0.52)
8. p4=Butter 256 ==> p5=Chocolate 134 conf:(0.52)
9. p6=Bread 257 ==> p4=Coffee 133 conf:(0.52)
10. p4=Butter 256 ==> p6=Fruit 132 conf:(0.52)

Rules Fetched

----

bread=fruit 230 ==> coffee=butter 122 conf :(0.53)

coffee=butter 238 ==> bread=fruit 122 conf :(0.51)

milk=chocolate 233 ==> coffee=butter 118 conf :(0.51)

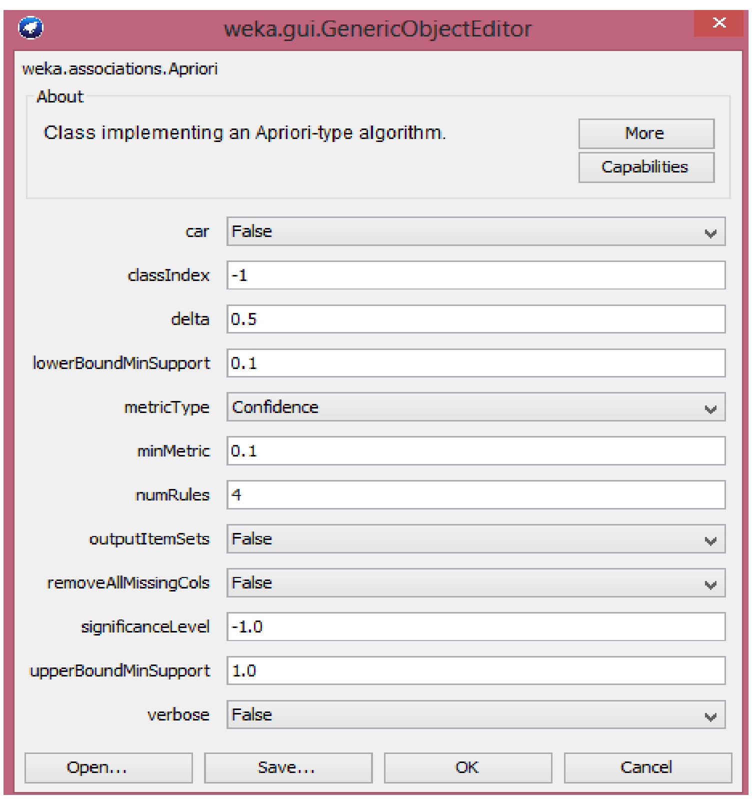

Figure 2. WEKA Parameters and Settings

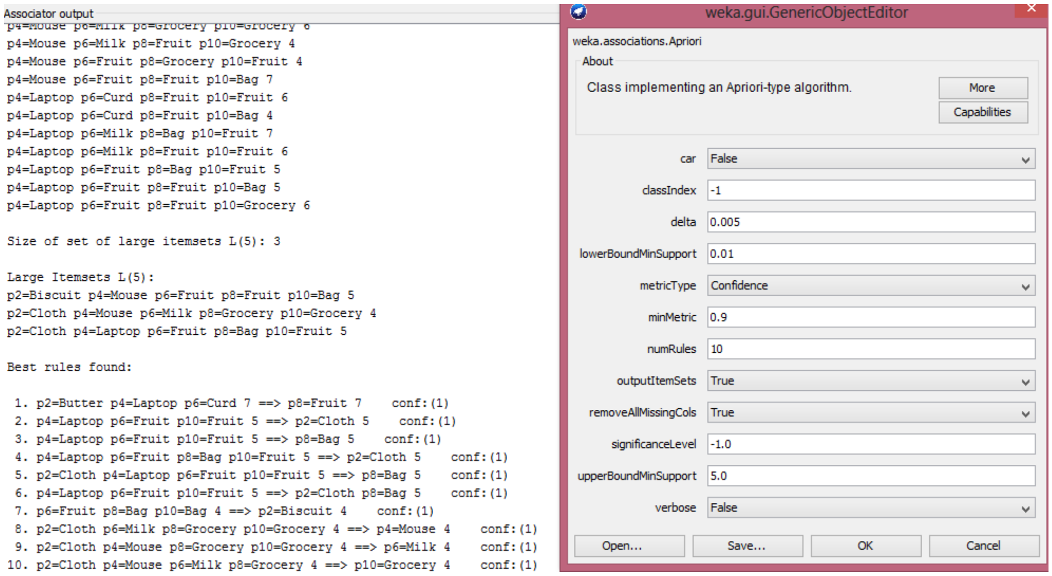

Figure 3. WEKA Results Sets from DataSet obtained from JDBC

STEPS OF OPTIMIZATION OF RECORDS IN MATLAB

1. Perform Implementation of Association Rule Mining in WEKA
   - Preparation of DataSet in MySQL Database Engine
   - Activate WEKA for JDBC Connection
   - Perform and Deploy JDBC with MySQL
   - Show All DataSets
   - Implement Rule Mining Algorithm
   - Get Best Rules from WEKA
2. Allocation of the Rules in form of an individual vector
3. Assign the bit 1 to each value of product participating in the rules
4. Assign the Value 2 to each master rule of every individual vector
5. Create a new matrix having all the vectors
6. A new sparse matrix generation
7. Master Matrix formation based on all 1's and 2's
8. Count number of 1's in each vector
9. Count number 0's in each vector
10. Apply the minimum threshold to results optimization
11. Minimum values are cut down from the main matrix
12. Generate new set of rules from MATLAB after removing minimum valued candidates

IMPLEMENTATION SCENARIOS

MinMetric : 0.5

=== Run information ===

Scheme: weka.associations.Rule Mining -N 10 -T 0 -C 0.5 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1

Relation: QueryResult

Instances: 500

Attributes: 6

p1
p2
p3
p4
p5
p6

=== Associator model (full training set) ===

Rule Mining

=======

Minimum support: 0.25 (125 instances)

Minimum metric <confidence>: 0.5

Number of cycles performed: 15

Generated sets of large itemsets:

Size of set of large itemsets L(1): 13

Size of set of large itemsets L(2): 6

Best rules found:

1. p4=Coffee 244 ==> p6=Bread 133 conf:(0.55)
2. p6=Fruit 243 ==> p4=Butter 132 conf:(0.54)

3. p5=Chocolate 250 ==> p6=Bread 135 conf:(0.54)
4. p5=Chocolate 250 ==> p4=Butter 134 conf:(0.54)
5. p6=Fruit 243 ==> p5=Milk 128 conf:(0.53)
6. p6=Bread 257 ==> p5=Chocolate 135 conf:(0.53)
7. p4=Coffee 244 ==> p5=Milk 128 conf:(0.52)
8. p4=Butter 256 ==> p5=Chocolate 134 conf:(0.52)
9. p6=Bread 257 ==> p4=Coffee 133 conf:(0.52)
10. p4=Butter 256 ==> p6=Fruit 132 conf:(0.52)

BEST RULES FETCHED FROM WEKA

----

bread=fruit 230 ==> coffee=butter 122 conf:(0.53)

coffee=butter 238 ==> bread=fruit 122 conf:(0.51)

milk=chocolate 233 ==> coffee=butter 118 conf:(0.51)

Screenshots of Fuzzy Implementation

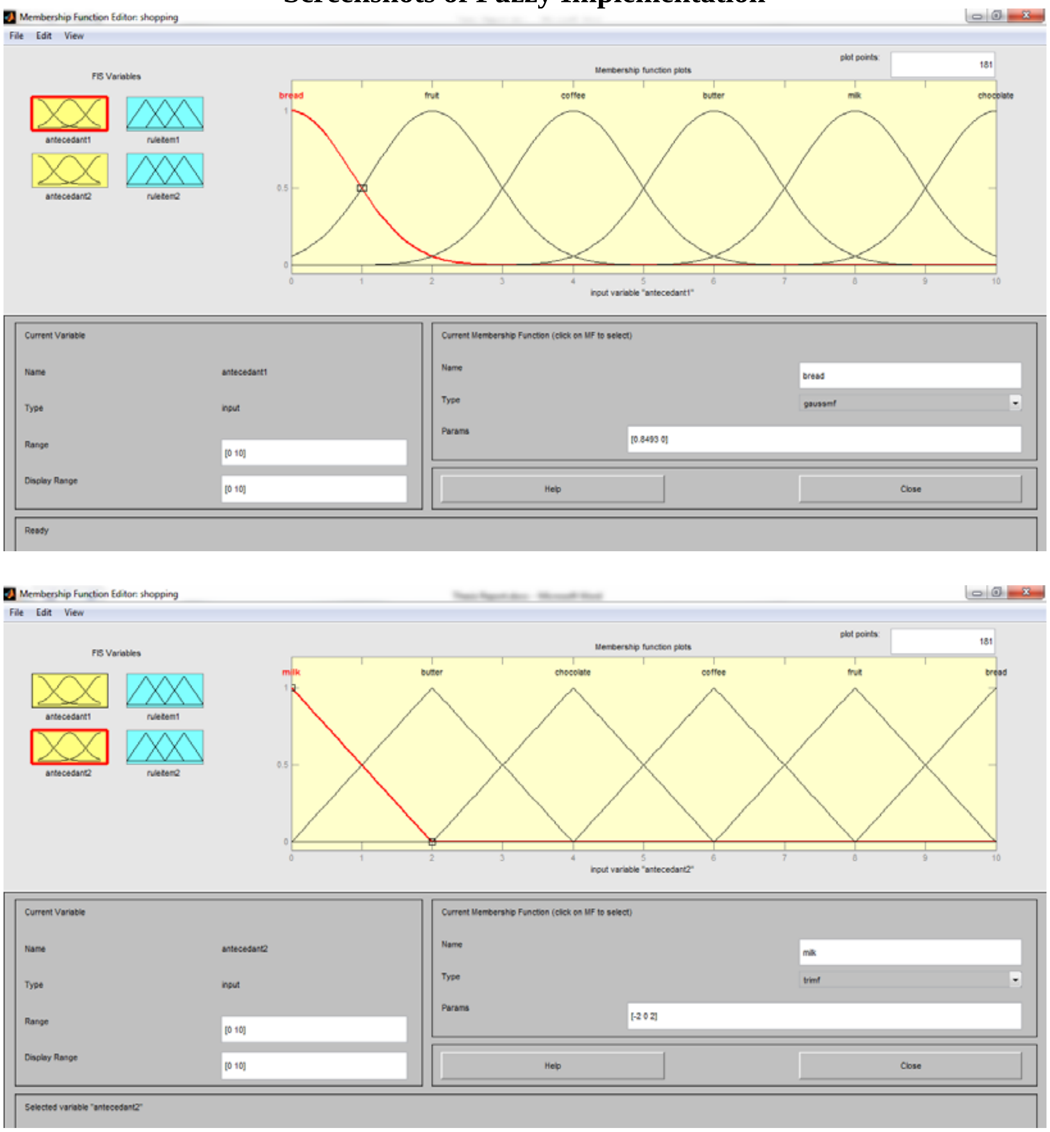

Figure 4. Fuzzy Logic Implementation in MATLAB

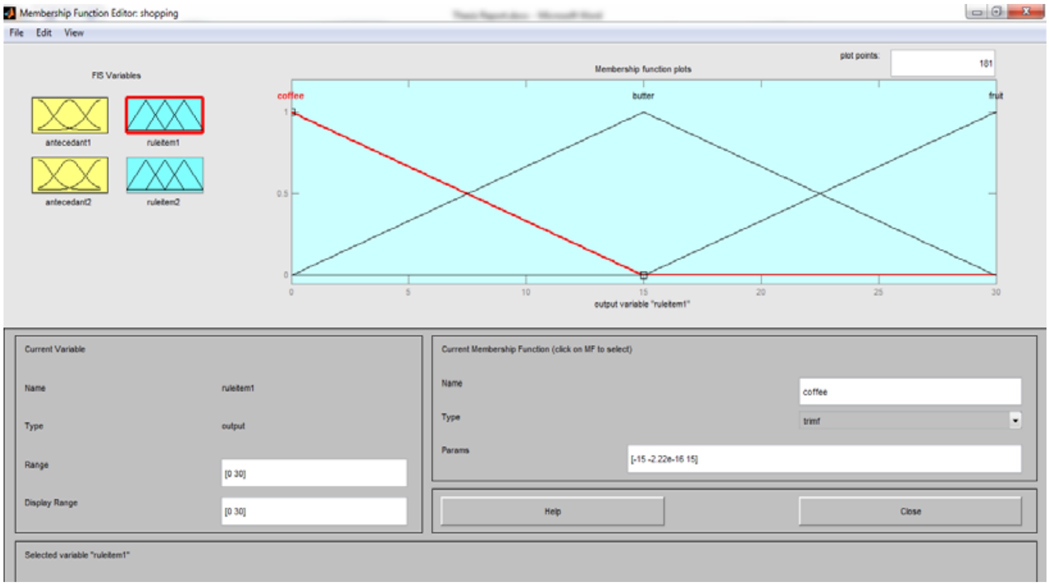

Figure 5. Fuzzy Logic Implementation in MATLAB

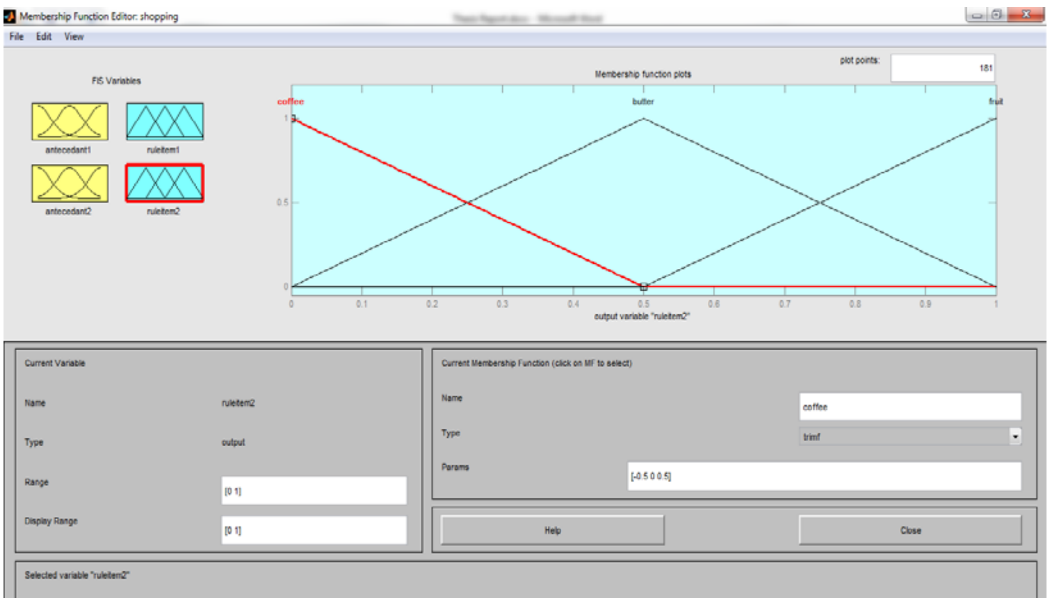

Figure 6. Fuzzy Logic Implementation in MATLAB

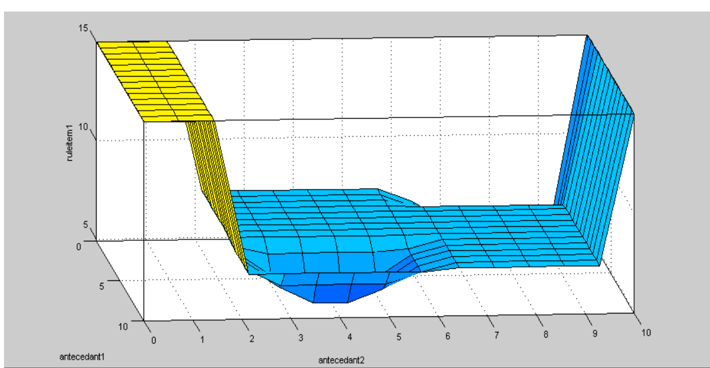

Figure 7. Fuzzy Logic Implementation in MATLAB

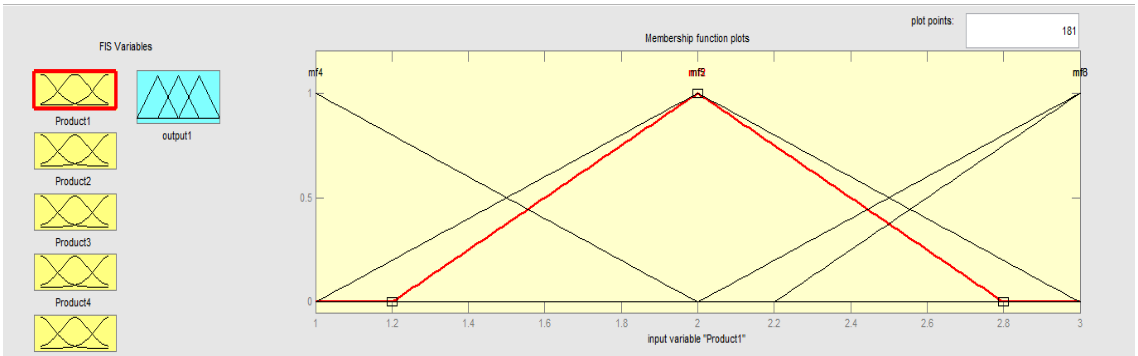

Figure 8. Rules Set and Fuzzy Implementation in MATLAB

MAMDANI FUZZY LOGIC SET OF RULES

[System]
Name='shopping'
Type='mamdani'
Version=2.0
NumInputs=2
NumOutputs=2
NumRules=2
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='centroid'

[Input1]
Name='antecedant1'
Range=[0 10]
NumMFs=6
MF1='bread':'gaussmf',[0.8493 0]
MF2='fruit':'gaussmf',[0.8493 2]
MF3='coffee':'gaussmf',[0.8493 4]
MF4='butter':'gaussmf',[0.8493 6]
MF5='milk':'gaussmf',[0.8493 8]
MF6='chocolate':'gaussmf',[0.8493 10]

[Input2]
Name='antecedant2'
Range=[0 10]
NumMFs=6
MF1='milk':'trimf',[-2 0 2]
MF2='butter':'trimf',[0 2 4]
MF3='chocolate':'trimf',[2 4 6]
MF4='coffee':'trimf',[4 6 8]
MF5='fruit':'trimf',[6 8 10]
MF6='bread':'trimf',[8 10 12]

[Output1]
Name='ruleitem1'
Range=[0 30]
NumMFs=3
MF1='coffee':'trimf',[-15 -2.22e-16 15]
MF2='butter':'trimf',[0 15 30]
MF3='fruit':'trimf',[15 30 45]

[Output2]
Name='ruleitem2'
Range=[0 1]
NumMFs=3
MF1='coffee':'trimf',[-0.5 0 0.5]
MF2='butter':'trimf',[0 0.5 1]
MF3='fruit':'trimf',[0.5 1 1.5]

[Rules]
1 5, 1 2 (1) : 1
5 3, 1 2 (1) : 1

Conclusion

In the most recent decade multi-target progress of Fuzzy direct based frameworks has pulled in wide vitality inside the examination get-together and experts. It depends upon the utilization of stochastic calculations for Multi-target move up to look for the Pareto ability in a different goals situation. For example, the targets to in the meantime refresh can be exactness and multifaceted nature, or accuracy and interpretability. A present audit of the field is given in the work of Fazzolari et al.. In like way, assorted work gives an in the present style and continually making once-finished of references concerning the issue.

References

[1] Pena-Reyes CA, Sipper M. A fuzzy-genetic approach to breast cancer diagnosis. Artificial intelligence in medicine. 1999 Oct 31;17(2):131-55.
[2] Russo M. FuGeNeSys-a fuzzy genetic neural system for fuzzy modeling. IEEE

transactions on Fuzzy Systems. 1998 Aug;6(3):373-88.

[3] González A, Pérez R. Selection of relevant features in a fuzzy genetic learning algorithm. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). 2001 Jun;31(3):417-25.

[4] Li J, Kwan RS. A fuzzy genetic algorithm for driver scheduling. European Journal of Operational Research. 2003 Jun 1;147(2):334-44.

[5] Hagras H, Callaghan V, Colley M. Learning and adaptation of an intelligent mobile robot navigator operating in unstructured environment based on a novel online Fuzzy–Genetic system. Fuzzy Sets and Systems. 2004 Jan 1;141(1):107-60.

[6] Martin-Bautista M, Vila MA, Larsen HL. A fuzzy genetic algorithm approach to an adaptive information retrieval agent. Journal of the Association for Information Science and Technology. 1999 Jul 1;50(9):760.

[7] Hagras H, Callaghan V, Colley M, Clarke G. A hierarchical fuzzy–genetic multi-agent architecture for intelligent buildings online learning, adaptation and control. Information Sciences. 2003 Mar 31;150(1):33-57.

[8] Hagras H, Callaghan V, Colley M, Clarke G. A hierarchical fuzzy–genetic multi-agent architecture for intelligent buildings online learning, adaptation and control. Information Sciences. 2003 Mar 31;150(1):33-57.

[9] Pena-Reyes CA, Sipper M. Designing breast cancer diagnostic systems via a hybrid fuzzy-genetic methodology. InFuzzy Systems Conference Proceedings, 1999. FUZZ-IEEE'99. 1999 IEEE International 1999 Aug 22 (Vol. 1, pp. 135-139). IEEE.