MODULAR APPROACH WITH ROUGH DECISION MODELS

Ahmed T. Shawky[1], Hesham A. Hefny[2] and Ashraf H. Abd Elwhab[3]

[1]Computer Sciences and Information Department,
Institute of Statistics and Research, Cairo University, Egypt
ah_taisser@yahoo.com

[2]Computer Sciences and Information Department,
Institute of Statistics and Research, Cairo University, Egypt
hehefny@ hotmail.com

[3] Computer Sciences and Systems Department,
Electronics Research Institute, Cairo, Egypt
awahab@ad.gov.eg

ABSTRACT

Decision models which adopt rough set theory have been used effectively in many real world applications. However, rough decision models suffer the high computational complexity when dealing with datasets of huge size. In this research we propose a new rough decision model that allows making decisions based on modularity mechanism. According to the proposed approach, large-size datasets can be divided into arbitrary moderate-size datasets, then a group of rough decision models can be built as separate decision modules. The overall model decision is computed as the consensus decision of all decision modules through some aggregation technique. This approach provides a flexible and a quick way for extracting decision rules of large size information tables using rough decision models.

KEYWORDS

Rough sets, Fuzzy sets, modularity, Data mining.

1. INTRODUCTION

During the last two decades, rough set theory has received much attention as a promising technique for data mining. Its ability to deal with imperfect data analysis makes it quite efficient for drawing useful decisions from datasets of various real world applications. However, rough decision models suffer the problem of high computational complexity of extracting decision rules when dealing with large-size datasets. Several researchers have proposed variuos approaches to overcome this problem. One of the recent attractive approaches, has been suggested by C. Degang and Z. Suyun [1], to improve the performance of rough decision making process, by integration of fuzzy set and rough set theories. This approach provides two benefits; the first is turning the continuous-valued conditional attributes into nominal or ordinal values which greatly simplifies the computations of finding reducts. The second benefit is the ability to deal with attributes with uncertain values which can be handled with fuzzy linguistic values that differ from decision maker to other. In this paper we make a further step for decreasing the computational cost of finding rough decision rules by introducing the approach of modularity. Modular approach to decision making uses the central idea of task-decomposition to reduce the computational cost of drawing decisions over large datasets. Modular neural networks are successful applications of such an approach [2, 3]. In this paper, we show how this approach can be adopted to reduce the computational cost of large rough decision models. This paper is organized as follows: section 2

DOI : 10.5121/ijdkp.2012.2507

presents the approach of modularity, section 3 discusses rough sets in details, the proposed modular rough decision model (MRDM) is presented in section 4, and section 5 concludes the paper.

2. THE APPROACH OF MODULARITY

Modular approach for decision making has been firstly proposed in literature in the works of R. A. Jordan et al (1991, 1994), see [4, 5]. The basic idea behind modular approach is task-decomposition, where large tasks can be divided into relatively small ones to be handled easily. Similarly, Large decision making tasks can be decomposed into small ones among group of local experts to reduce the cost of the overall decision. Modular design has been successfully used in various areas, e.g., robotics and neural networks as presented by Tseng, and Almogahed [2].

Procedure in modular design starts with decomposing the given system (task) into subsystems, i.e. modules, for simpler design, followed by aggregating the modular designs. The idea is to ignore interconnection among these subsystems in the design stage. Since the subsystems are smaller than the original system, the design effort and computation needed in each subsystem design are typically lower. The system will also be easier to debug and maintain due to a smaller system size. In many cases, appropriate decomposition of modules is a designer's issue. Figure (1) represents modular expert with k local experts with outputs O1, O2, ..Ok, while g1, g2, ..gk are the integrating weights, and Σ denotes to gating process.

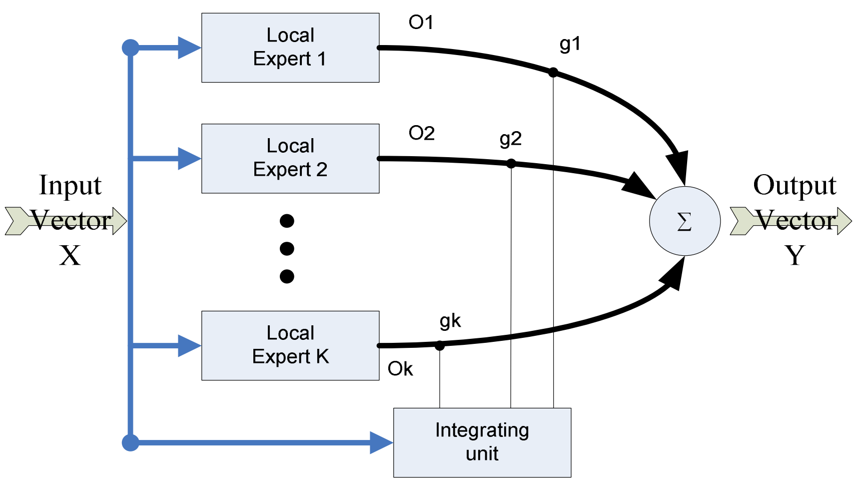


Figure 1. Modular expert architecture

2.1 ELEMENTS OF MODULARITY

When considering modular structure to solve a problem, one has to take into account the following points by P. Melin et al [3]:

- Decompose the main problem into subtasks.
- Organizing the modular architecture, taking into account the nature of each subtask.
- Communication between modules is important, not only in the input of the system but also in the response integration.

2.2 RESPONSE INTEGRATION

Integrating the response outputs of decision modules can be one of the following choices [3]:

- The method of “Winner Takes All”, for problems that require similar tasks.
- Models in series, the output of one module is the input to the following one.
- Voting methods, for example the use of the “Softmax” function.
- Linear combination of output results.
- Using discrete logic.
- Using statistical methods.
- Using fuzzy logic.

3. ROUGH SETS

Rough set theory, proposed by Z. Pawlak (1982), can serve as a mathematical tool for dealing with classification of datasets based on uncertainty concepts [6, 7]. Various successful rough decision models have been developed in several fields [8]. The following subsections gives a more detail discussion about the main characteristics of rough set theory.

3.1. Information Systems

According to Pawlak (1982), Data are usually presented in the form of decision tables, also called information systems, which consist of rows and columns. Rows of the decision table represent cases, while columns represent variables. Such variables are considered as properties of each case. The set of independent variables are called conditional attributes and a dependent variable is called a decision attribute. Table 1 shows an example of a decision table.

Table 1. Example of Decision table.

Object	Conditional Attributes			Decision Attribute
	Age	Height	Gender	Accepted
X1	Young	Tall	Male	Yes
X2	Baby	Tall	Female	Yes
X3	Young	Tall	Female	Yes
X4	Old	Medium	Female	No
X5	Baby	Short	Male	Yes
X6	Old	Medium	Male	No

An information system S, as presented by Y. Qian et al [9], is a pair (U, A), where U is a non-empty, finite set of objects and is called the universe and A is a non-empty, finite set of attributes. V is the set of all attribute values, such as $V_a: U \times A \rightarrow V$ for each $x \in U$. According to the example shown in Table 1, U = {X1, X2, X3, X4, X5, X6}, A= {Age, Height, Gender, Accepted}, and V(X1, Age)= Young.

3.2. Indiscernibility Relation

One of the fundamental ideas of rough set theory is the Indiscernibility relation. For $B \subseteq A$ and $x, y \in U$, the Indiscernibility relation IND(B) is a relation on U defined as follows [10]:

$(x, y) \in IND(B)$ if and only if $V(x,a) = V(y,a)$ for all $a \subseteq B$. The Indiscernibility relation IND(B) is an equivalence relation. Equivalence classes of IND(B) are called elementary sets and are denoted by $[x]_B$. Elementary sets may be computed by using attribute-value pair blocks. Let a ϵ A and let v be a value of a, for some case. For complete decision tables if t = (a, v) is an attribute value pair, then a block of t, denoted [t], is a set of all cases from U that for attribute a has value v. For example if t = (Gender, Male) then [t] = {X1, X5, X6}.

3.3. Reducts

For B ⊆ A, the corresponding partition on U will be denoted by B^*. B is called reduct if and only if [3]: $B^* = A^*$ and, B is minimal if: $(B\text{-}\{a\})^* \neq A^*$ for all $a \in B$

For example, from Table 1 we have: $A^* = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}\}$

Let B = {Age, Height, Gender}, and C = {Age, Gender}

Then, it is clear that: $B^* = A^*$, and $C^* = A^*$

Also $(B$ - {Height}$)^* = A^*$ this means that B is not minimal

On the other hand, C is reduct of A since

$(C - \{a\})^* \neq A^*$ for all a ϵ C.

3.4. Approximation space

In a completely specified decision table, any finite union of elementary sets, associated with B, will be called B-definable set [x]B. The concept of "approximate space" has arisen due to the fact that not any subset of X is B-definable [10, 11]. In this case, the concepts of lower and upper approximations are defined on the basis of the Indiscernibility relation. Let X be any subset of the set U of all cases. The set X is called a "concept" and is usually defined as the set of all cases defined by a specific value of the decision attribute. In general, X is not a B-definable set. However, set X may be approximated by two B-definable sets; the B-lower approximation of X, denoted by $\underline{B}X$ and the B-upper approximation of X, denoted by $\overline{B}X$ which are defined as follows:

$\underline{B}X = \{ x \in U \mid [x]_B \subseteq X \}$, and

$\overline{B}X = \{ x \in U \mid [x]_B \cap X \neq \phi \}$

The above shown way of computing lower and upper approximations, by constructing these approximations from singletons x, will be called the first method. The B-lower approximation of X is the greatest B-definable set, contained in X. The B-upper approximation of X is the smallest B-definable set containing X. This concept can be illustrated from Table 1 as follows:

let U = {X1, X2, X3, X4, X5, X6} and A = {Age, Height, Gender, Accepted}

If B ⊆ A and B = {Height}, then B* = {{X1, X2, X3}, {X5}, {X4, X6}}

Suppose we have a new concept X = {X2, X3, X5} , in this case, we have
Lower approximation $\underline{B}X$ = {X5}

Upper approximation $\overline{B}X$ = {X1, X2, X3, X5}

Once the concepts of lower and upper approximations are adopted, then we can distinguish three regions in approximation space as follows:

- The positive region POS(BX) = $\underline{B}X$
- The boundary region BND (BX) = $\overline{B}X$ - $\underline{B}X$
- The negative region NEG (BX) = U - $\overline{B}X$

3.5. Rule Induction

For the inconsistent input data, Y. Qian et al [11] and J. W. Grzymala-Busse and S. Siddhaye [8] explained that the rules induced from the lower approximation of the concept certainly describe the concept, so they are called certain. On the other hand, rules induced from the upper approximation of the concept describe the concept only possibly (or plausibly), so they are called possible.

For example: As a certain we can say:

If (Age, old) and (Height, medium) then (accepted, no)

As a possible we can say:

If (Gender, Female) then (accepted, yes) with $\alpha = 0.67$

α is called a confidence factor and can be defined as the percentage of the number of elements that are in the elementary set and satisfy the concept for the rule from the total number of elements in the elementary set (upper approximation) in this example

B = {Gender} then B* = {{X1, X5, X6}, {X2, X3, X4}}

X = {X2, X3} P|X| = 2

$\overline{B}X$ = {X2, X3, X4} P| $\overline{B}X$ | = 3

$$\alpha = \frac{P \mid X \mid}{P \mid \overline{B}X \mid} \qquad \alpha = \frac{2}{3} = 0.67$$

4. MODULAR ROUGH DECISION MODEL (MRDM)

The proposed MRDM aims to simplify the process of drawing decisions using rough decision model over a large information system. This is can be performed through what we call "Grid Modular" approach, through which the given large information system is split into a group of moderate size sub-information systems. Then, each sub-information system is treated as a separate rough decision module. Through the MRDM proposed model, we can control the number of modules created from the main information system. The final decision of the overall decision model is computed through a gating technique of the output decisions of all decision modules. In this paper, we choose voting as the gating technique of the proposed MRDM. Figure 2 represents the structure of proposed model.

The main adopted algorithms within through the proposed MRDM are as follows:

- Algorithms 1: Which returns the confidence degree of the considered association rule based on the information table as a whole.

- Algorithm 2: Which splits the original information system into n-sequential module.
- Algorithm 3: Which splits the original information system into n-random module.
- Algorithm 4: which performs the gating of the output decisions of n-decision modules.

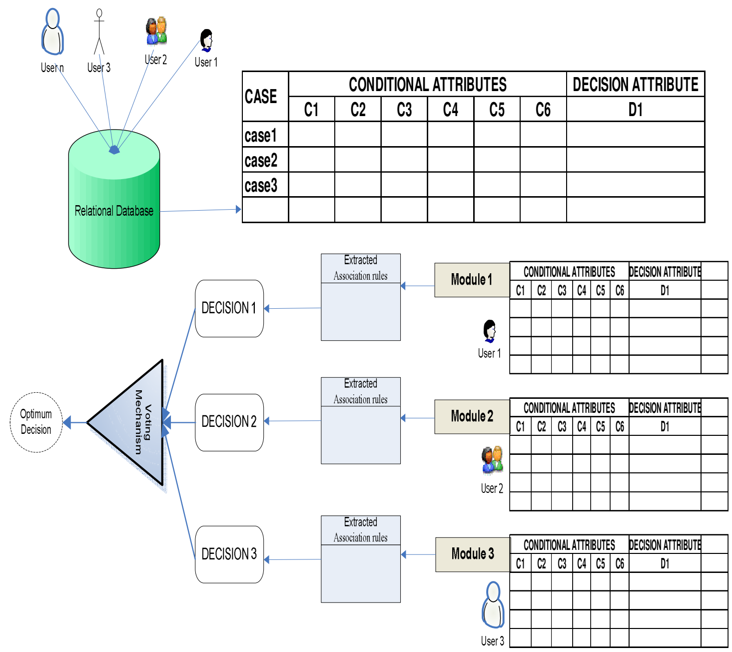


Figure 2. Structure of the Proposed MRDM

Algorithm (1) Returning a decision from the Information Table as a whole

```
// Input: rough.table(U,A) is an information table
// Output: chosen decision, greatest α degree

    Read  Rule, nofdecisions
    Define BX, TX, k, l, X[], alfa[], Descision[]
    BX = |cases check with Rule|
      For i = 0 to nofdecisions
       Input Descision[i]
       X[i]=|cases check with Rule ∧ cases which satisfy Descision[i]|
           alfa[i] = X[i] / BX
           k = i
       Next
        For s = 1 To k
            If alfa[s] < alfa[s-1] Then
                alfa[s] = alfa[s - 1]
                X[s] = X[s - 1]
                Descision[s] = Descision[s - 1]
            End If
            l = s
        Next
        return Descision[l]
        return alfa[l]
```

Algorithm (2) Splitting The Information Table Into N Sequential Sub-modules

```
// Input: rough.table(U,A) is an information table
// Output: devide the main rough model into "N" modules

      Read nofmodule
      Define total, n, d, d2, case
         total = rough.Tables(0).Rows.count
         n = int(total / nofmodule)
         d = total / nofmodule
         d2 = (d - n) * nofmodule
             For i = 1 To nofmodule
                 If i = nofmodule Then
                     n = n + d2
                  End If
                  For j = 1 To n
                      case = select min(cases) from rough.table(0)
                      R(case) = rough.Tables(0).Rows(case)
                      Rough.tables(i).rows(case) = R(case)
                    rough.Tables(0).Rows(case).clear
                  Next
              Next
```

Algorithm (3) Splitting The Information Table Into N Random Sub-modules

```
// Input: rough.table(U,A) is an information table
// Output: devide the main rough model into "N" modules

1     Read nofmodule   // number of modules
2     Define total, n, d, d2, case
3        total = rough.Tables(0).Rows.count
4        n = int(total / nofmodule)
5        d = total / nofmodule
6        d2 = (d - n) * nofmodule
7            For i = 1 To nofmodule
8                If i = nofmodule Then
9                    n = n + d2
10                End If
11                For j = 1 To n
12                 case = Int((total * Rnd()) + 1)
13                  If Rough.tables(i).Rows(case).Rows.count > 0 Then
14                      GOTO 12
15                  Else
16                      R(case) = rough.Tables(0).Rows(case)
17                      Rough.tables(i).Rows(case) = R(case)
18                      rough.Tables(0).Rows(case).clear
19                  End if
20                Next
21            Next
```

Algorithm (4) Gating a decision from N Rough Decision modules

```
// Input: N modules of rough.table(U,A)
// Output: chosen decision, greatest α degree

      Read rule, nofdecisions, nofmodules
      Define K, netdecision, netalfa, Descision[], C[], alfa[],
      Define poss[],avgalfa[], avgposs[],decval[]
      For I = 0 to nofdecisions
      Input Descision[i]
            For N = 1 to nofmodules
            Run Algorithm 1. For rough.Tables(N)
      If rough.Tables(N).getdecision(decision) =  Descision[i] then
      {
            C[i] = C[i] + 1
            Alfa[i] = alfa[i] + rough.Tables(N).getdecision(alfa)
            Poss[i] = poss[i] + rough.Tables(N).getdecision(poss)
      }
      End if
            Next
            Avgalfa[i] = alfa[i]/C[i]
            Avgposs[i] = poss[i]/C[i]
            Decval[i] = Avgalfa[i] * Avgposs[i]
            K = i
     Next
          For s = 1 To k
              If decval[s] < decval[s-1] Then
                 Descision[s] = Descision[s - 1]
              End If
              l = s
          Next
        Netdecision  = Descision[l]
        Netalfa = decval[l]
        return Netdecision
        return netalfa
```

5. Case Study

As a case study used to explain the implementation of MRDM proposed model, we prepared a set of data collected in excel file as in table (2) and throw MRDM proposed model the data in this file have been cached and arranged into some sort of attributes. The data represented in rough information table as previous in two types of attributes (Conditional attributes and Decision attributes).

Table (2) information system for the given case study

Case	Temperature	Hypertension	Headache	Cough	Flue
1	39	120	Yes	Yes	Yes
2	42	180	Yes	No	Yes
3	39	130	No	No	No
4	38	200	Yes	Yes	Yes
5	37	170	Yes	No	No
6	37	180	No	Yes	No
7	40	190	Yes	No	No
8	40	200	Yes	Yes	Yes
9	38	200	Yes	Yes	Yes
10	37	170	Yes	No	No
11	37	180	No	Yes	Yes
12	37	120	No	No	No
13	42	130	Yes	Yes	Yes
14	37	220	Yes	No	No
15	41	180	Yes	No	No
16	39	130	No	Yes	Yes
17	40	200	Yes	Yes	Yes
18	38	130	No	No	No
19	42	220	Yes	Yes	Yes
20	37	120	Yes	Yes	Yes

The steps of algorithm (1) proceed as follows:

Step 1: define inference rules which are given to take a decision.
Step 2: which is done by (MRDM) proposed model is determining upper and lower approximation for all possible cases to choose the optimum decision.
Step 3: calculating α degree for the different decisions according to the given rules.
Step 4: chose the optimum decision which has the greatest α degree.

In our example given rules is
Headache = 'Yes' and Temperature > = 38
The possible decisions Flue = {'Yes', 'No'}
B = {Headache, Temperature}
B* = {{1, 2, 4, 7, 8, 9, 13, 15, 17, 19}, {3, 16}, {5, 10, 14, 20}, {6, 11, 12}}
X = {x | if headache = 'Yes' and temperature >= 38 then flue = 'Yes'}
X = {1, 2, 4, 8, 9, 13, 17, 19} P|X| = 8
$\overline{B}X$ = {1, 2, 4, 7, 8, 9, 13, 15, 17, 19} P|$\overline{B}X$| = 10

$$\alpha = \frac{P\,|\,X\,|}{P\,|\,\overline{B}X\,|} \qquad \alpha = \frac{8}{10} \qquad \alpha = 0.8 \qquad (1)$$

Y = {x | if headache = 'Yes' and temperature >= 38 then flue = 'No'}
Y = {7, 15} P|Y| = 2

$$\alpha = \frac{P\,|\,Y\,|}{P\,|\,\overline{B}X\,|} \qquad \alpha = \frac{2}{10} \qquad \alpha = 0.2 \qquad (2)$$

From 1, 2 α (X) > α (Y)
The optimum decision is Flue = 'Yes' with α = 0.8
From 1, 2 α (X) is the greatest

The optimum decision is VaseLf = 'B' with $\alpha = 1$

<u>The steps of algorithm (2) proceed as follows</u>:

Step 1: determining the number of module needed to create from the main module, the user ofMRDM do this step.
Step 2: calculate number of cases in each module (N cases)
Step 3: select the first N cases from the main information system and insert them into module 1, this step is repeated as the number of modules determined in previous step.

<u>The steps of algorithm (3) proceed as follows</u>:

Step 1: determining the number of module needed to create from the main module, the user of MRDM do this step.
Step 2: calculate number of cases in each module (N cases)
Step 3: select one case randomly from the main information system and insert it into module 1, this step is repeated until it is full of the module.
Step 4: the previous step is repeated as the number of modules determined in step 1.

After creating modules from the main rough information system, MRDM proposed model allows user to define rules, which are needed to get a decision. Note that the same example used in taking decision from main rough information system represented in table (2), is used to represent using MRDM proposed model to take a decision through modularity, and also given rules are used. In our example, the main information system has been converted to four models using the random mechanism. Tables (3, 4, 5, and 6) represent data of the four modules, and our given rule is the same rule used above.
Headache = 'Yes' and Temperature > = 38

TABLE (3) MODULE 1 OF THE MAIN INFORMATION SYSTEM

Case	Temperature	Hypertension	Headache	Cough	Flue
15	No	Yes	180	41	No
11	Yes	No	180	37	Yes
12	No	No	120	37	No
6	Yes	No	180	37	No
7	No	Yes	190	40	No

TABLE (4) MODULE 2 OF THE MAIN INFORMATION SYSTEM

Case	Temperature	Hypertension	Headache	Cough	Flue
16	Yes	No	130	39	Yes
1	Yes	Yes	120	39	Yes
17	Yes	Yes	200	40	Yes
9	Yes	Yes	200	38	Yes
18	No	No	130	38	No

TABLE (5) MODULE 3 OF THE MAIN INFORMATION SYSTEM

Case	Temperature	Hypertension	Headache	Cough	Flue
8	Yes	Yes	200	40	Yes
20	Yes	Yes	120	37	Yes
2	No	Yes	180	42	Yes
19	Yes	Yes	220	42	Yes
10	No	Yes	170	37	No

TABLE (6) MODULE 4 OF THE MAIN INFORMATION SYSTEM

Case	Temperature	Hypertension	Headache	Cough	Flue
13	Yes	Yes	130	42	Yes
5	No	Yes	170	37	No
14	No	Yes	220	37	No
3	No	No	130	39	No
4	Yes	Yes	200	38	Yes

After defining rules, one decision is taken from each module with α degree. Final step in taking decision through MRDM proposed model using modularity approach is gating process. In MRDM proposed model voting technique is used as a gating process, this done by making vote between the decisions taken by the modules. The voting process is taking into account two factors which are α degree and possibility degree. Possibility degree is calculated in MRDM proposed model as percentage between the numbers of cases achieve the given rule to number of cases achieve chosen decision

<u>The steps of algorithm (4) proceed as follows</u>:

Step 1: Using algorithm (1) to get a decision from each module

Step 2: Calculate α degree and possibility degree for the optimum decision of each module as following:

$\alpha = \frac{P \mid X \mid}{P \mid \overline{B}X \mid}$ where X = {x | number of cases achieve the optimum decision according to the given rule} and $\overline{B}X$ is the upper approximation of the given rules

Possibility = $\frac{P \mid X \mid}{P \mid D \mid}$ where D = {x | number of cases achieve the optimum decision}

Step 3: apply voting technique to select the optimum decision of the model among the decisions achieved in each module, according to α degree and Possibility degree

The vote of each decision is calculated as a summation of [α * possibility] for each module satisfies the decision, figure (3) explains how to implement the voting process.

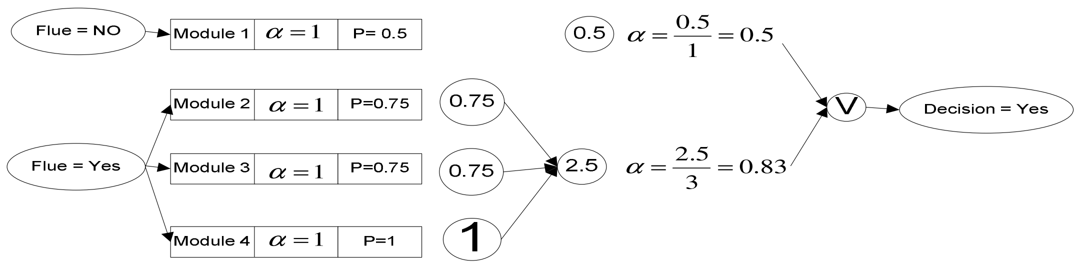


Figure. 3 voting process implementation

6. CONCLUSION

Modular approach reduces the computation cost of drawing decision of large rough decision models. A gating technique can be proposed by the user to aggregate the output decisions of the various decision modules so as to get the overall output decision. In this paper we adopt a voting technique as the gating method. The illustrative example shows that the design effort and computation needed in each subsystem design are typically lower. Also, the system will be easier to debug and maintain. As a future work.

As a future work, we plan to propose a "user modular: approach of the above MRDM through which fuzzy terms are used to replace the continuous real values in the given information system. This wil result in a further reduction of the computation cost for drawing the final decision output of the rough decision model.

REFERENCES

[1] C. Degang, Z. Suyun, "Local reduction of decision system with fuzzy rough sets", Fuzzy Sets and Systems 161 (2010) 1871–1883

[2] H. C. Tseng, B. Almogahed, "Modular neural networks with applications to pattern profiling problems", Neurocomputing, 2008.10.020.

[3] P. Melin, C. Gonzalez, D. Bravo, F. Gonzalez and G. Martinez, "Modular Neural Networks and Fuzzy Sugeno Integral for Pattern Recognition", Studies in Fuzziness and Soft Computing, Vol 208 (2007), 311-326.

[4] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts", Neural Computation, 3:79:87, 1991.

[5] M. I. Jordan, R. A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm", Neural Computation, 6:181-214,1994.

[6] Z. Pawlak, "Rough sets", International Journal of Computer and Information Sciences 11 (1982) 341–356.

[7] Z. Pawlak, A. Skowron, Rudiments of rough sets, Information Sciences, Information Sciences 177 (2007) 3–27.

[8] K. Thangavel , A. Pethalakshmi, " Dimensionality reduction based on rough set theory ", Applied Soft Computing 9 (2009) 1–12

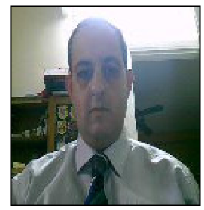
[9] Y. Qian , J. Liang, D. Li , H. Zhang, C. Dang, "Measures for evaluating the decision performance of a decision table in rough set theory", Information Sciences 178 (2008) 181–202.

[10] J. W. Grzymala-Busse, S. Siddhaye," Rough Set Approaches to Rule Induction from Incomplete Data", the 10th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Perugia, Italy, July 4–9, 2004, vol. 2, 923–930

[11] Y. Qian , J. Liang, D. Li , F. Wang, N. Ma, "Approximation reduction in inconsistent incomplete decision tables", Knowledge-Based Systems 23 (2010) 427-433.

Authors

Mr. Ahmed Taisser Shawky is IT spicialist in Health Insurance Organization in Cairo, and a Ph.D. student in Computer Sciences Department at the Institute of Statistical Studies and Research, Cairo University. His Ph.D. in the filed of Computational Intelligence. His research interests include Rough set theory, Fuzzy set theory , Modularity, Rough decision models



Dr. Hesham Ahmed Hefny is an Associate Professor and the head of Computer & Information Sciences Department at the Institute of Statistical Studies and research, Cairo University. His research interests include Artificial Neural Networks, Fuzzy Systems, Genetic Algorithms, Swarm Intelligence, Pattern Recognition, and Data Mining. Dr. Hesham has published over 35 peer refereed papers in academic journals and conferences on topics within Artificial Intelligence and related areas.

Prof. Ashraf Hassan Abdelwahab is a professor of computer engineering, Electronics Research Institute, Cairo, Egypt. He received his M. Sc. in 1988, Faculty of Engineering, Cairo University in the area of Artificial Intelligence, and in 1992 he received his Ph.D. in Machine Learning and Evolutionary Algorithms. He has published over 60 technical papers in National, and International journals and conferences in the areas of Evolutionary Algorithms, Machine Learning, and Data Mining.