Feature Selection Based On Neuro-Fuzzy-Rough System

A. Pethalakshmi
Department of Computer Science, M.V.M Government Arts College (W), Dindigul-624 001,
Tamil Nadu, India.
E-mail: pethalakshmi@yahoo.com

Abstract

Rough Sets theory provides a new mathematical tool to deal with uncertainty, inexact and vagueness of an information system. The information system may contain a certain amount of redundancy that will not aid knowledge discovery and may in fact mislead the process. The redundant attributes may be eliminated in order to reduce the complexity of the problem. This paper proposes Neuro-Fuzzy-Rough Quickreduct (NFRQ) algorithm to select the features from the information system. Neural network is used to construct the membership functions, for fuzzyfying the crisp data. The experiments are carried out on the data sets of UCI machine learning repository and the Human Immunodeficiency Virus (HIV) data set in order to achieve the efficiency of the proposed algorithm.

Keywords: Rough sets, Fuzzy sets, Neural Network, Feature selection.

1. Introduction

Feature selection is the process to choose a subset of attributes from the original attributes. Feature selection has been studied intensively in the past decades [10, 11, 12, 13]. The purpose of the feature selection identifies the significant features, eliminates the irrelevance of dispensable features to the learning task, and builds a good learning model. The benefits of feature selection are twofold: it considerably decreases the computation time of the induction algorithm, and increases the accuracy of the resulting mode.

Feature selection algorithm falls into two categories: (i) the filter approach and (ii) the wrapper approach. In the filter approach, the feature selection is performed as a preprocessing step to induction. The filter approach is ineffective if it deals with the feature redundancy. In the wrapper approach [11], the feature selection is "wrapped around" an induction algorithm, so that the bias of the operators that defines the search and that of the induction algorithm interacts mutually. Though the wrapper approach suffers less from feature interaction, nonetheless, its running time would make the wrapper approach infeasible in practice, especially if there are many features because the wrapper approach keeps running the induction algorithms on different subsets from the entire attributes set until a desirable subset is identified. The Research intends to keep the algorithm bias, as small as possible and would like to find a subset of attributes that can generate good results by applying a suite of data mining algorithms.

A decision table may have more than one reduct. Anyone of them can be used to replace the original table. Finding all the reducts from a decision table is NP-Hard. Fortunately, in many real applications it is usually not necessary to find all. A natural question arises that which reduct is the best one if there exists more than one reduct. The selection depends on the optimality criterion associated with the attributes. If it is possible to assign a cost function to attributes, then the selection can be naturally based on the combined minimum cost criteria. In the absence of an attribute cost function, the only source of information to select the reduct is the contents of the data table [13]. It is assumed that the best reduct is the one with the minimal number of attributes. Thangavel et al. proposed various feature selection algorithms and compared with the existing algorithm [19, 20].

1.1 Rough Set Theory

In 1982, Pawlak introduced the theory of Rough sets [15, 16]. This theory was initially developed for a finite universe of discourse in which the knowledge base is a partition, which is obtained by any equivalence relation defined on the universe of discourse. In rough sets theory, the data is organized in a table called decision table. Rows of the decision table correspond to objects, and columns correspond to attributes. In the data set, a class label to indicate the class to which each row belongs. The class label is called as decision attribute, the rest of the attributes are the conditional attributes. The conditional attributes are represented by C, the decision attributes are denoted by D, where $C \cap D = \Phi$, and t_j denotes the j^{th} tuple of the data table.

Rough sets theory defines three regions based on the equivalent classes induced by the attribute values: lower approximation, upper approximation, and boundary. Lower approximation contains all the objects, which are classified surely based on the data collected, and Upper approximation contains all the objects, which can be classified probably, while the boundary is the difference between the upper approximation and the lower approximation. Hu et al.[6] presented the formal definitions of rough set theory and proposed a new rough sets model and redefined the core attributes and reducts based on relational algebra to take advantage of very efficient set-oriented database operations.

Let U be any finite universe of discourse. Let R be any equivalence relation defined on U. Clearly, the equivalence relation partitions the universe U. Here, (U, R) which is the collection of all equivalence classes, is called the approximation space. Let $W_1, W_2, W_3, \ldots, W_n$ be the elements of the approximation space (U, R). This collection is known as knowledge base. Then for any subset A of U, the lower and upper approximations are defined as follows:

$$\underline{R}A = \cup\{W_i / W_i \subseteq A \quad (1)$$

$$\overline{R}A = \cup \{W_i / W_i \cap A \neq \emptyset\} \quad (2)$$

The ordered pair ($\underline{R}A$, $\overline{R}A$) is called a rough set. Once defined these approximations of A, the reference universe U is divided in three different regions: the positive region $POS_R(A)$, the negative region $NEG_R(A)$ and the boundary region $BND_R(A)$, defined as follows:

$$POS_R(A) = \underline{R}A \quad (3)$$

$$NEG_R(A) = U - \overline{R}A \quad (4)$$

$$BND_R(A) = \overline{R}A - \underline{R}A \quad (5)$$

Hence, it is trivial that if BND(A) = Φ, then A is exact. This approach provides a mathematical tool that can be used to find out all possible reducts. In this paper a novel hybrid approach viz.., Neuro-Fuzzy-Rough is proposed and some of the basic concepts of Neural Network and Fuzzy sets are discussed in the subsequent sections. A good research survey on rough sets can be found in [21].

1.2 Neural Network

A neural network is a technique that seeks to build an intelligent program using models that simulate the working network of the neurons in the human brain [5]. A neuron is made up of several protrusions called dendrites and long branch called the axon. A neuron is joined to the other neurons through the dendrites. The dendrites of different neurons meet to form synapses, the areas where message pass. The neurons receive the impulses via the synapses. If the total of the impulses received exceeds a certain threshold value, then the neuron sends an impulse down the axon where the axon is connected to other neurons through more synapses. The synapses may be excitatory or inhibitory in nature. An excitatory synapse adds to the total of the impulses reaching the neuron, whereas an inhibitory neuron reduces the total of the impulses reaching the neuron. In a global sense, a neuron receives a set of input pulses and sends out another pulse that is a function of the input pulses.

1.3 Fuzzy sets

The use of fuzzy set theory is one way of capturing the vagueness present in the real world, which would otherwise be difficult to use conventional set theory. There are many useful introductory resources regarding fuzzy set theory [2, 17]. In classical set theory, the elements could belong fully(i.e. have a membership of 1) or not at all(a membership of 0). Fuzzy set theory relaxes this restriction by allowing memberships to take values anywhere in the range [0, 1]. A fuzzy set can be defined as a set of ordered pairs A = $\{(x, \mu_A(x)) \mid x \in U\}$. The function $\mu_A(x)$ is called the membership function for A, mapping each element of the universe U to a membership degree in the range [0, 1]. The universe may be discrete or continuous. Any fuzzy set containing at least one element with a membership degree of 1 is called normal.

The rest of the paper is organized as follows: Section 2 describes Fuzzy-Rough attribute reduction. Section 3 deals with the construction of membership functions using a neural network. Section 4 explains the Neuro-Fuzzy-Rough Quickreduct algorithm. Section 5 describes the experimental analysis of Fuzzy-Rough Quickreduct and the Neuro-Fuzzy-Rough Quickreduct. Section 6 concludes this paper.

2. Fuzzy-Rough Quickreduct (FRQ)

The Rough Set Attribute Reduction (RSAR) operates effectively with datasets containing discrete values. Additionally, there is no way of handling noisy data. As most datasets contain real-valued features, it is necessary to perform a discretization step beforehand. To reduce this difficulty, discretization can be implemented by a standard fuzzification technique. The membership degrees of attribute values to fuzzy sets are typically not exploited in the process of dimensionality reduction. This is counterintuitive. By using fuzzy-rough sets [3, 14, 22], it is possible to use this information to guide feature selection. The fuzzy-rough method and grouping mechanism are concerned with real valued attributes with corresponding attributes.

The fuzzy lower and upper approximations are defined as

$$\mu_{\underline{p}X}(x) = \sup_{F \in U/p} \min(\mu_F(x), (\inf_{y \in U} \max\{1 - \mu_F(y), \mu_X(y)\}) \quad (6)$$

$$\mu_{\overline{p}X}(x) = \sup_{F \in U/p} \min(\mu_F(x), (\sup_{y \in U} \min\{\mu_F(y), \mu_X(y)\}) \quad (7)$$

By the extension principle, the membership of an object $x \in U$, belonging to the fuzzy positive region can be defined as

$$\mu Pos_{p(Q)}(x) = \sup_{X \in U/Q} \mu_{\underline{p}X}(x) \quad (8)$$

Using the definition of the fuzzy positive region, the new dependency function can be defined as follows:

$$\gamma_p(Q) = \sum_{x \in U} \mu Pos_{p(Q)}(x) \,/\, |U| \qquad (9)$$

In crisp rough set attribute reduction, the data set would be discretized using the non-fuzzy sets. However, in the new approach, membership degrees are used in calculating the fuzzy lower approximations and fuzzy positive regions. The pseudocode of the fuzzy-rough Quickreduct algorithm [7, 8, 9] is given below:

FRQUICKREDUCT(C, D)
C, the set of all conditional features
D, the set of all decision features

(1) $R \leftarrow \{\}$; $\gamma_{best} = 0$; $\gamma_{prev} = 0$
(2) do
(3) $T \leftarrow R$
(4) $\gamma_{prev} = \gamma_{best}$
(5) $\forall x \in (C - R)$
(6) if $\gamma_{R \cup (x)}(D) >= \gamma_T(D)$
(7) $T \leftarrow R \cup (x)$
(8) $\gamma_{best} \leftarrow \gamma_T(D)$
(9) $R \leftarrow T$
(10) until $\gamma_{best} == \gamma_{prev}$
(11) return R

When the Fuzzy-Rough Quickreduct algorithm applied to a Car data set (Table. 2), it produces the given reduct set (Weight, Door, Size}.

3. Construction of membership functions using a Neural Network

In this section, the construction of the membership function for fuzzification using neural network is discussed [18]. A number of input data values are selected and divided into a training data set and a testing data set. Consider a system of 8 data points (5 for training data points and 3 for testing data points) described using four conditional attributes and one decision attribute (Table 1) adopted from [6]. Then Table 1 can be normalized into Table 2.

Table.1 Car Data

Object	Weight	Door	Size	Cylinder	Mileage
1	Low	2	Com	4	High
2	Low	4	Sub	6	Low
3	Medium	4	Com	4	High
4	High	2	Com	6	Low
5	High	4	Com	4	Low
6	Low	4	Com	4	High
7	High	4	Sub	6	Low
8	Low	2	Sub	6	Low

Low=1 Medium=2 High=3 Com=1 Sub=2

Table.2 Normalized Data

Object	Weight	Door	Size	Cylinder	Mileage (1,3,6)	(2,4,5,7,8)
1	0.1667	0.3333	0.1667	0.6667	1	0
2	0.1667	0.6667	0.3333	1.0000	0	1
3	0.3333	0.6667	0.1667	0.6667	1	0
4	0.5000	0.3333	0.1667	1.0000	0	1
5	0.5000	0.6667	0.1667	0.6667	0	1
6	0.1667	0.6667	0.1667	0.6667	1	0
7	0.5000	0.6667	0.3333	1.0000	0	1
8	0.1667	0.3333	0.3333	1.0000	0	1

These data points have been placed in four fuzzy classes, R_1, R_2, R_3 and R_4 using any one of the clustering methods [4]. A neural network that can determine the membership values of any data point in the four classes is to be formed. The membership values in Table 3 can be used to train and check the performance of the neural network which has been assigned membership values of unity for the classes into which they have been originally assigned.

Table. 3 Membership values

Data points	1	2	3	4	5	6	7	8
R_1	1	0	0	0	0	0	0	0
R_2	0	0	0	1	0	0	0	1
R_3	0	0	1	0	1	1	0	0
R_4	0	1	0	0	0	0	1	0

Select a 4 X 5 X 5 X 4 neural network to simulate the relationship between the data points and their membership in the four fuzzy sets, R_1, R_2, R_3 and R_4 (Fig. 1). The attributes Weight, Door, Size and Cylinder for each data point are used as the input values and the corresponding membership values in the four fuzzy classes for each data point are the output values for the neural network.

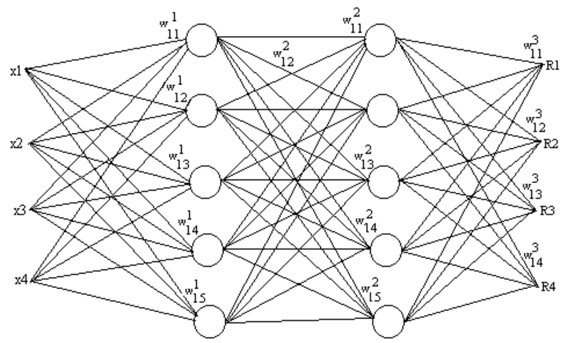

Fig. 1 4 X 5 X 5 X 4 neural network

Table 4 shows the initial random values that have been assigned to the different weights connecting the paths between the elements in the layers in the network shown in Fig. 1.

Table. 4 Random Weight

W^1_{11}= 0.4349	W^2_{11} = 0.2560	W^3_{11} = 0.5456
W^1_{12}= 0.2463	W^2_{12} = 0.8859	W^3_{12}= 0.4851
W^1_{13} = 0.5593	W^2_{13} = 0.2239	W^3_{13} = 0.2681
W^1_{14} = 0.0595	W^2_{14} = 0.3012	W^3_{14} = 0.8056
W^1_{15} = 0.6565	W^2_{15} = 0.5871	W^3_{21} = 0.4568
W^1_{21} = 0.5130	W^2_{21} = 0.8659	W^3_{22} = 0.7847
W^1_{22} = 0.2740	W^2_{22} = 0.9816	W^3_{23} = 0.4431
W^1_{23} = 0.2684	W^2_{23} = 0.6769	W^3_{24} = 0.7968
W^1_{24} = 0.4334	W^2_{24} = 0.2219	W^3_{31} = 0.0856
W^1_{25} = 0.1406	W^2_{25} = 0.4579	W^3_{32} = 0.4243
W^1_{31} = 0.1381	W^2_{31} = 0.9020	W^3_{33} = 0.2191
W^1_{32} = 0.8613	W^2_{32} = 0.6478	W^3_{34} = 0.6193
W^1_{33} = 0.5867	W^2_{33} = 0.6688	W^3_{41} = 0.1357
W^1_{34} = 0.5190	W^2_{34} = 0.9789	W^3_{42} = 0.7710
W^1_{35} = 0.1481	W^2_{35} = 0.9708	W^3_{43} = 0.0576
W^1_{41} = 0.2389	W^2_{41} = 0.6239	W^3_{44} = 0.0796
W^1_{42} = 0.7733	W^2_{42} = 0.5108	W^3_{51} = 0.8151
W^1_{43} = 0.1406	W^2_{43} = 0.6488	W^3_{52} = 0.8970
W^1_{44} = 0.5199	W^2_{44} = 0.3791	W^3_{53} = 0.1081
W^1_{45} = 0.5190	W^2_{45} = 0.3159	W^3_{54} = 0.4314
	W^2_{51} = 0.5186	
	W^2_{52} = 0.5237	
	W^2_{53} = 0.3791	
	W^2_{54} = 0.5429	
	W^2_{55} = 0.9758	

Take the first data point (0.1667, 0.3333, 0.1667, 0.6667) as the input to the neural network. Use the following equation (10) as follows:

$$O = 1 / 1 + \exp^{-(\Sigma x_i w_i - t)} \qquad (10)$$

where O – output of the threshold element computed using the sigmoidal function

x_i - inputs to the threshold element(i = 1, 2, 3,….n)

w_i - weights attached to the inputs

t - threshold for the element

First iteration: In the first iteration to train the neural network, equation (10) is used and choose t = 0.

Outputs for the second layer

$O_1{}^2$ = 1 / 1 + exp – [(0.1667*0.4349)+(0.3333*0.5130)+(0.1667*0.1381) +(0.6667*0.2389) –0.0] = 0.6049

$O_2{}^2$ = 1 / 1 + exp – [(0.1667*0.2463)+(0.3333*0.2740)+(0.1667*0.8613) +(0.6667*0.7733) –0.0] = 0.6882 and so on.

Outputs for the third layer

$O_1{}^3$=1/1+exp–

[(0.6049*0.2560)+(0.6882*0.8659)+(0.5925*0.9020)+

(0.6428*0.6239) +(0.6288*0.5186) –0.0] = 0.8821

$O_2{}^3$=1/1+exp– [(0.6049*0.8859)+(0.6882*0.9816)+(0.5925*0.6478)+

(0.6428*0.5108) +(0.6288*0.5237) –0.0] = 0.9049

and so on.

Outputs for the fourth layer

$O_1{}^4$=1/1+exp– [(0.8821*0.5456)+(0.9049*0.4568)+(0.8393*0.0856)+

(0.8176*0.1357) +(0.8872*0.8151) – 0.0] = 0.8582

$O_2{}^4$=1/1+exp– [(0.8821*0.4851)+(0.9049*0.7847)+(0.8393*0.4243)+

(0.8176*0.7710) +(0.8872*0.8970) –0.0] = 0.9488

and so on

Determining errors

Compare the outputs of the fourth layer (which is the output layer) to the correct outputs (previously known membership values listed in Table 3) to determine the final error of the neural network as follows:

$R_1 : E_1{}^4$ = $O_1{}^4$ - $O_1{}^4$. actual = 0.8582 – 1.0000

= -0.1418

$R_2 : E_2{}^4$ = $O_2{}^4$ - $O_2{}^4$. actual = 0.9488 – 0.0

= 0.9488 and so on.

The errors of the output layer have been computed for the first iteration and these errors are distributed to the other nodes in the previous layer using the equation (11)

$$E_n = O_n (1 - O_n) \Sigma w_{nj} E_j \qquad (11)$$

Assigning errors: First, assign errors to the elements in the third layer.

$E_1{}^3$ = 0.8821 *(1-0.8821)*((0.5456*(- 0.1418)) + (0.4851* 0.9488) + (0.2681* 7240)+(0.8056*0.9168)) = 0.1368

$E_2{}^3$ = 0.9049 *(1-0.9049)*((0.4568*(- 0.1418)) + (0.7847* 0.9488) + (0.4431* 7240)+(0.7968*0.9168)) = 0.1490 and so on.

and then assign errors to the elements in the second layer.

$E_1{}^2$ = 0.6049 *(1- 0.6049) * ((0.2560*0.1368) + 0.8859*0.1490) + (0.2239*0.1506) +(0.3012*0.1233)*(0.5871*0.1210))

= 0.0486

$E_2{}^2$= 0.6882 *(1-0.6882) * ((0.8659*0.1368) + (0.9816*0.1490) + (0.6769*0.1506) +(0.2219*0.1233)*(0.4579*0.1210)) = 0.0790 and so on.

Now the errors associated with each element in the network are known, the weights associated with these elements can be updated so that the network approximates the output more closely. To update the weights the following equation (12) can be used:

$$w^{i}_{jk}\,(new) = w^{i}_{jk}\,(old) + \alpha\, E_{k}^{(i+1)}\, x_{jk} \qquad (12)$$

where w^{i}_{jk} is the weight associated with path connecting the j^{th} element of the i^{th} layer to the k^{th} element of the $(i + 1)^{th}$ layer , α is the learning constant from 0 to 1. Here it is assumed as 0.4 for this example. $E_k^{(i+1)}$ is the error associated with the k^{th} element of the $(i + 1)^{th}$ layer and x_{jk} is the input from the j^{th} element in the i^{th} layer to the k^{th} element in the $(i + 1)^{th}$ layer (O^{i}_{j}).

Updating Weights

The weights connecting elements in the third and fourth layers will be updated thus:

W^3_{11} = 0.5456+(0.4*(-0.1418)*0.8821) = 0.4956

W^3_{21} =0.4568+(0.4*(-0.1418)*0.9049) = 0.4055 and so on.

Then weights connecting elements in the second and the third layers are updated thus:

W^2_{11} = 0.2560+(0.4*(0.1368)*0.6049) = 0.2891

W^2_{21} =0.8659+(0.4*(0.1368)*0.6882) = 0.9036 and so on.

And then finally, weights connecting elements in the first and the second layers are updated thus:

W^1_{11} = 0.4349+(0.4*(0.0486)*0.1667) = 0.4381

W^1_{21}= 0.5130+(0.4*(0.0486)*0.3333) = 0.5195 and so on.

All the weights in the neural network have been updated and the first input data point is again passed through the neural network. The errors in approximating the output are computed again and redistributed as before. This process is continued until the errors are within acceptable limits. Next, the second data point and the corresponding membership values are used to train the neural network. This process is continued until all the data points in the training data set are used. The performance of the neural network (how closely it can predict the value of the membership of the data point) is then checked using the data points in the testing data set. Once the neural network is trained and verified to be performing satisfactorily, it can be used to find the membership of any other data points in the four fuzzy classes. The membership function values obtained from the neural network are tabulated in Table 5.

Table. 5 After Applying the Membership Function

Object	Weight	Door	Size	Cylinder		Mileage (1,3,6)	(2,4,5,7,8)
1	0.8512	0.9849	0.9900	0.9704	1	1	0
2	0.8584	0.9868	0.9914	0.9735	0	0	1
3	0.8566	0.9863	0.9910	0.9727		1	0
4	0.8569	0.9864	0.9911	0.9729		0	1
5	0.8579	0.9866	0.9912	0.9733		0	1
6	0.8552	0.9859	0.9907	0.9721		1	0
7	0.8606	0.9873	0.9917	0.9744		0	1
8	0.8551	0.9860	0.9908	0.9722		0	1

4. Neuro-Fuzzy-Rough Quickreduct (NFRQ)

In the Fuzzy-Rough Quickreduct algorithm, each and every conditional attribute requires the membership functions to fuzzify the crisp values while the extended principle of the fuzzy set is being used. In this approach all the conditional attributes should be mapped into two regions namely N_a and Z_a. This is a time consuming and tedious process. To overcome this disadvantage, neural network has been used to construct the membership values and suitably the Fuzzy-Rough Quickreduct has been modified to construct the reduct set.

In [3], the fuzzy p-lower and p-upper approximations are defined as

$$\underline{\mu p}X(F_i) = \inf_x \max\{1 - \mu F_i(x), \mu X(x)\}\ \forall i \qquad (13)$$

$$\overline{\mu p}X(F_i) = \sup_x \min\{\mu F_i(x), \mu X(x)\}\ \forall i \qquad (14)$$

where F_i denotes a fuzzy equivalence class belonging to U / P. As the universe of discourse in feature selection is finite, the use of sup and inf are to be altered. As a result of this, the fuzzy lower and upper approximations are herein redefined as:

$$\underline{\mu p}X(x) = \min(\mu F(x),\ (\inf_{y\in U} \max\{1 - \mu F(y), \mu X(y)\}) \qquad (15)$$

$$\overline{\mu p}X(x) = \min(\mu F(x),\ (\sup_{y\in U} \min\{\mu F(y), \mu X(y)\}) \qquad (16)$$

Fuzzy-Rough set based feature selection builds on the notion of fuzzy lower approximation to enable reduction of data sets containing real valued features. The fuzzy positive region can be defined as

$$\mu Pos_{p(Q)}(x) = \sup_{X\in U/Q} \underline{\mu p}X(x) \qquad (17)$$

Using the definition of the fuzzy positive region, the new dependency function can be defined as follows:

$$\gamma_p(Q) = |\mu Pos_{p(Q)}(x)| / |U|$$

$$= \sum_{x\in U} \mu Pos_{p(Q)}(x) / |U| \qquad (18)$$

The Neuro-Fuzzy-Rough Quickreduct Algorithm (NFRQA) is given here under.

NFRQUICKREDUCT(C, D)
C, the set of all conditional features
Q, the set of all decision features

(1) C ← the set of all membership values generated by Neural Network
(2) RED ← {}
(3) Do
(4) TEMP ← RED
(5) For $x \in C$
(6) if $\min\{\gamma_{RED\cup(x)}(Q) > \gamma_{TEMP}(Q)\}$
(7) TEMP ← RED ∪ min(x)
(8) RED ← TEMP
(9) until $\gamma_{REd}(Q) == \gamma_C(Q)$
(10) **return RED**

Worked Example

Using the Table. 5, (for all conditional and decision attributes), and setting A = {Weight}, B = {Door}, C = {Size}, D = {Cylinder} and Q = {Mileage}. First the lower approximations of A, B, C and D can be calculated using the equation. For simplicity, only 'A' will be considered here; that is 'A' is to approximate 'Q'. For the first decision equivalence class X = {1, 3, 6}, $\mu_{A\{1,3,6\}}(x)$ needs to be calculated:

$$\mu_{\underline{A}\{1,3,6\}}(x) = \min(\mu F(x), (\inf_{y\in U} \max\{1 - \mu F(y), \mu\{1,3,6\}(y)\})$$

For object 1 this can be calculated as follows:

max (1 - μ a(1), μ{1, 3, 6}(1)) = max(0.1488, 1.0) = 1.0
max (1 - μ a(2), μ{1, 3, 6}(2)) = max(0.1416, 0.0) = 0.1416
max (1 - μ a(3), μ{1, 3, 6}(3)) = max(0.1434, 1.0) = 1.0
max (1 - μ a(4), μ{1, 3, 6}(4)) = max(0.1431, 0.0) = 0.1431
max (1 - μ a(5), μ{1, 3, 6}(5)) = max(0.1421, 0.0) = 0.1421
max (1 - μ a(6), μ{1, 3, 6}(6)) = max(0.1448, 1.0) = 1.0
max (1 - μ a(7), μ{1, 3, 6}(7)) = max(0.1394, 0.0) = 0.1394
max (1 - μ a(8), μ{1, 3, 6}(8)) = max(0.1449, 0.0) = 0.1449

Therefore, $\min(\mu F(x), (\inf_{y\in U} \max\{1 - \mu F(y), \mu\{1,3,6\}(y)\})$

= min(0.8512, inf{1.0, 0.1416, 1.0, 0.1431, 0.1421, 1.0, 0.1394, 0.1449})
= min(0.8512, 0.1394) = 0.1394

Thus, $\mu_{\underline{A}\{1,3,6\}}(1) = 0.1394$. Similar calculation can be made for other objects. Then the corresponding values for X = {2, 4, 5, 7, 8} can also be determined. For object1, $\mu_{\underline{A}\{2,4,5,7,8\}}(1) = 0.1434$. Similar calculation can be made for other objects. Using these values, the fuzzy positive region for each object can be calculated using

$$\mu Pos_{p(Q)}(x) = \sup_{X\in U/Q} \mu_{\underline{AX}}(x)$$

For object1,
$\mu Pos_{p(Q)}(1) = \sup(0.1394, 0.1434) = \mathbf{0}.1434$. Similar calculation can be made for other objects. The next step is to determine the degree of dependency of Q on A:

$$\gamma_A(Q) = \sum_{x\in U} \mu Pos_{p(Q)}(x) / |U|$$

$$= 1.1472 / 8 = 0.1434$$

Similarly, Calculating for B, C and D as:

$\gamma_B(Q) = 0.0137$
$\gamma_C(Q) = 0.00090$
$\gamma_D(Q) = 0.0273$

From this it can be seen that the attribute 'C' will cause the smallest in dependency degree. The attribute is chosen and added to the potential reduct. Thus the derived value is compared with all conditional attribute values as. $\gamma_{\{A,B,C,D\}}(Q)$.

$\gamma_{\{A,B,C,D\}}(Q) = 0.1434$

If the values are equal, the reduct attributes are obtained. Otherwise the smallest attribute can be combined with the other attributes. By taking the minimum values (min(C, A), min(C, B), (min(C, D)) of original attributes can be obtained from using the Table. 5.

The process iterates and the two dependency degrees are calculated as,

$\gamma_{\{C,A\}}(Q) = 0.1434$
$\gamma_{\{C,B\}}(Q) = 0.0137$
$\gamma_{\{C,D\}}(Q) = 0.0273$

In the above combinations, again the minimum dependency value can be taken. Therefore, $\gamma_{\{C,B\}}(Q)$ is minimum. The attribute is chosen and added to the potential reduct. This value needs to be checked with all conditional attribute values i.e. $\gamma_{\{A,B,C,D\}}(Q)$. If the values are equal, the reduct attributes can be returned. If not the smallest attribute set can be combined with the other attributes. It states that the minimum values (min(C, B, A), min(C, B, D)) of original attributes can be taken using the Table 3.5. This process iterates and the three dependency degrees are calculated as,

$\gamma_{\{C, A, B\}}(Q) = 0.1434$

$\gamma_{\{C, B, D\}}(Q) = 0.1434$

From this it can be seen that both sets have the same degree of dependency. In this case, the attributes can be taken priority-wise and added to the potential reduct. This value can be checked with all conditional attribute values as $\gamma_{\{A, B, C, D\}}(Q)$.

$\gamma_{\{A, B, C, D\}}(Q) = 0.1434$

Hence, the reduct attributes such as {A, B, C} or {B, C, D} is returned, since all the values are equal.

5. Experimental Analysis

The Fuzzy-Rough Quickreduct (FRQ) and the proposed Neuro-Fuzzy-Rough Quickreduct (NFRQ) algorithm have been implemented for the data sets available in the UCI machine learning repository [1] viz., Bupa, New-Thyroid, Pima, Hepatitis, Iris, Dermatology, Postoperative and Ecoli.

The HIV database consists of information collected from the HIV patients at the Voluntary Counseling and Testing Center (VCTC) of the Government Headquarters Hospital, Dindigul District, Tamilnadu, India, a well-known center for diagnosis and treatment of HIV. It contains the records of 2200 patients with 21 conditional attributes and a decision attribute.

The proposed algorithm has been compared with the existing Fuzzy-Rough Quickreduct algorithm. The Comparative Analysis is tabulated in Table. 6. It is observed that the proposed algorithm Neuro-Fuzzy-Rough Quickreduct produced minimal reducts for the data sets Bupa, New-Thyroid, Hepatitis, Iris, Dermatology, Ecoli and HIV than the existing algorithm Fuzzy-Rough Quickreduct. In the case of Car, Pima and Postoperative data sets, the same number of reducts are obtained in both the algorithms. The performance analysis of the Fuzzy-Rough Quickreduct (FRQ) and the Neuro-Fuzzy-Rough Quickreduct (NFRQ) algorithm is also depicted in Fig. 2.

Table. 6 Comparative Analysis

Data Set	Instances	Original Attributes	FRQ	N FRQ
Car	8	4	3	3
Bupa	345	6	4	3
New-Thyroid	215	5	4	3
Pima	768	8	3	3
Hepatitis	80	19	8	7
Iris	150	4	3	2
Dermatology	358	34	18	16
Postoperative	90	8	5	5
Ecoli	336	7	5	3
HIV	2200	21	18	15

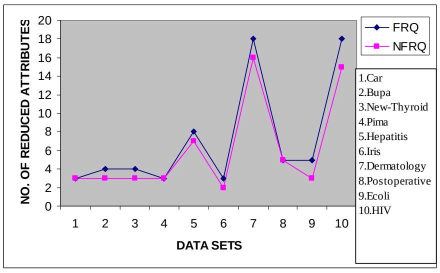

Fig. 2 Performance Analysis of FRQ with NFRQ

6. Conclusion

If the data in the information system is redundant, it will not aid in the effective knowledge discovery. This in turn will mislead the process. The redundant attributes may be eliminated in order to generate the reduct set (i.e., reduced set of necessary attributes) or to construct the core set of attributes. The disadvantage of fuzzy rough quick reduct algorithm is that, each and every conditional attributes which requires the membership functions to fuzzify the crisp values. Meanwhile the extended principle of the fuzzy set has been addressed and to overcome the disadvantages, the neural network has been used to construct the membership values of each and every value. The efficiency of the proposed Neuro-Fuzzy-Rough Quickreduct Algorithm (NFRQ) has been achieved.

References

[1]. C. L. Blake, and C. J. Merz. "UCI Repository of Machine L earning Databases", Irvine, University of California, 1998, http://www.ics.uci.edu/~mlearn/.

[2]. E.Cox. "The Fuzzy Systems Handbook: a Practioner's Guide to Building, Using and Maintaining Fuzzy Systems", *Academic Press, Inc.*, ISBN 0-121-94270-8, 1999.

[3]. D.Dubois and H.Prade. "Putting Rough Sets and Fuzzy Sets Together", *Intelligent Decision Support*, pp. 203-232, 1992.

[4]. J.Han and M.Kamber, "Data Mining: Concepts and Techniques", *Morgan Kaufmann Publishers*, 1995.

[5]. J.Hopfield, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities", in Proc. National Academy of Sciences, USA, 1982, pp. 2554 – 2558.

[6]. X. Hu, T. Y. Lin and J. Jianchao. "A New Rough Sets Model Based on Database Systems",*Fundamenta Informaticae*, pp. 1-18, 2004.

[7]. R.Jensen and Qiang Shen. "Fuzzy-Rough Attribute Reduction with Application to Web Categorization", *Fuzzy Sets and Systems*, vol.141, no.3, pp. 469-485, 2004.

[8]. R.Jensen and Qiang Shen. "Semantics Preserving Dimensionality Reduction: Rough and Fuzzy-Rough–Based Approaches", *IEEE Transactions on Knowledge and Data Engineering*, vol.16, no.12, 2004.

[9]. R.Jensen. "Combining Rough and Fuzzy Sets for Feature Selection", Ph.D.,thesis, School of Informatics, University of Edinburgh, 2005.

[10]. G. John, R. Kohavi and K. Pfleger."Irrelevant Features and the Subset Selection Problem",in Proc. ICML, 1994,pp. 121-129.

[11]. K. Kira and L. A. Rendell. "The Feature Selection Problem: Traditional Methods and a New Algorithm", in Proc. AAAI, MIT Press, 1992, pp. 129-134.

[12]. H.Liu and H. Motoda (Eds.)."Feature Extraction Construction and Selection: A Data Mining Perspective", *Kluwer Academic Publisher*, 1998.

[13]. M. Modrzejewski. "Feature Selection Using Rough Sets Theory", in Proc. ECML, 1993,pp. 213-226.

[14]. S.K.Pal and A.Skowron."Rough-Fuzzy Hybridization: A New Trend in Decision Making", *Springer Verlag*, 1999.

[15]. Z. Pawlak., "Rough Sets", *International Journal of Computer and Information Sciences*, vol. 11, no. 5, pp. 341-356, 1982.

[16]. Z. Pawlak, "Rough Sets: Theoritical Aspects and Reasoning about Data", *Kluwer Academic Publisher*, 1991.

[17]. W.Pedrycz and F.Gomide, "An Introduction to Fuzzy Sets: Analysis and Design", *The MIT Press*, 1998.

[18]. T.J.Ross. "Fuzzy Logic with Engineering Applications", *McGraw-Hill Inc*., 1995.

[19]. K. Thangavel, Q. Shen and A. Pethalakshmi. "Application of Clustering for Feature Selection based on Rough Set Theory Approach", *International Journal on Artificial Intelligence and Machine Learning*, vol. 6, no. 1, pp. 19–27, 2005.

[20]. K. Thangavel, A. Pethalakshmi and P. Jaganathan. "A Comparative Analysis of Feature Selection Algorithms based on Rough Set Theory", *International Journal of Soft Computing*, vol. 1, no. 4, pp. 288–294, 2006.

[21]. K. Thangavel and A. Pethalakshmi."Dimensionality Reduction based on Rough Set Theory: A Review", *International Journal on Applied Soft Computing*, vol. 9, issue. 1, pp. 1-12, 2009.

[22]. H.Thiele, "Fuzzy Rough Sets versus Rough Fuzzy Sets- An Interpretation and a Comparative Study Using Concepts of Modal Logics", *Technical Report No.CI-30/98*, University of Dortmund, 1998.

A. Pethalakshmi was born in 1965 at Karaikudi, Tamilnadu, India. She received her Master of Computer Science from Alagappa University, Tamil Nadu, India in 1988 and the Master of Philosophy in Computer Science from Mother Teresa Women's University, Kodaikanal,Tamil Nadu, India in 2000. She obtained her Ph.D. Degree from Mother Teresa Women's University, Kodaikanal, Tamil Nadu, India in 2008. Currently she is working as Associate Professor and Head, Department of Computer Science, M.V.M. Govt. Arts College (w), Dindigul, Tamil Nadu, India. Her area of interests includes Data Mining, Rough Set, Fuzzy set, and Neural Network.