Indian Journal of Science and Technology, *Vol 7(7), 905–915, July 2014*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Knowledge Discovery in Medical Data by using Rough Set Rule Induction Algorithms

P. K. Srimani[1] and Manjula Sanjay Koti[2*]

[1]Department of CS & Maths, Bangalore University, Bangalore, India
[2]Department of MCA, Dayananda Sagar College of Engineering, Bangalore, India
[2]Bharathiar University, Coimbatore, India; man2san@rediffmail.com

Abstract

In the performance of data mining and knowledge discovery activities, rough set theory has been regarded as a powerful, feasible and effective methodology. There is a need for analysis of medical data that deals with incomplete and inconsistent information with the tremendous manipulation at different levels. In this context, rough set rule induction algorithms are capable of generating decision rules which can potentially provide new medical insight and profound medical knowledge. By taking into consideration all the above aspects, the present investigation is carried out. The results clearly show that rough set approach is certainly a useful tool for medical applications. Relationships and patterns within this data could provide new medical knowledge. The genetic algorithms offer an attractive approach for solving the feature subset selection problem. The process of finding useful patterns or meaning in raw data has been called knowledge discovery in databases. The algorithms used for the present study are: Exhaustive search, Covering, LEM2 and Genetic algorithms. Rules are generated and improved in the case of the above mentioned four algorithms. Further, the generated rules are improved by applying the shortening ratio as 0.8. Some of the important results of the present investigation include maximum coverage of 100% is observed in the case of exhaustive and genetic algorithms.

Keywords: Decomposition Tree, Rough Set, Rules, Rule Induction Algorithms

1. Introduction

The great advances in information technology have made it possible to store a great quantity of data. In the late nineties, the capabilities of both generating and collecting data were increased rapidly. Millions of databases have been used in business management, government administration, scientific and engineering data management, as well as many other applications. It can be noted that the number of such databases keeps growing rapidly because of the availability of powerful database systems. This explosive growth in data and databases generated an urgent need for new techniques and tools that can intelligently and automatically transform the processed data into useful information and knowledge. In the medical domain, rough set analysis is actively utilized for the extraction of decision rules from various data sets such as diabetes mellitus, spinal cord injury[1] and hepatitis.

1.1 An Overview of Rough Set Theory (RST)

In recent years, Classical rough set theory developed[2] has made a great success in knowledge acquisition. Rough set theory is relatively a new tool that deals with vagueness and uncertainty inherent in decision making. Rough Set Theory (RST) is a useful mathematical tool to deal with imprecise and insufficient knowledge, find hidden patterns in data, and reduce the size of the dataset[3]. Also, it facilitates (i) the evaluation of the significance of the data and (ii) the easy interpretation of the results.Rough set theory has been regarded since its very inception as a powerful and feasible methodology for performing data mining and knowledge discovery activities. In Rough

*Author for correspondence

set theory, knowledge is represented in information systems and an information system is a data set represented in the form of a table, which is called as decision table4. In the area of knowledge discovery particularly in machine learning and rule extraction, Rough set theory[3] has gained much popularity. In particular, it is useful in discarding redundant information in a database, i.e. to arrive at a minimum number of attributes that would still allow each data record to be distinguished from the others[1]. This minimum set of attributes is called as a reduct. Rough set theory[5–6] can deal with uncertainty and incompleteness in data analysis and deems knowledge as a kind of discriminability. The redundant information or features are removed by the attribute reduction algorithm by selecting a feature subset that has the same discernibility as the original set of features. Rough set has the following strong special features viz., (i) a machine learning method which generates rules based on examples contained within an information table (ii) a well established theory/mechanism capable of solving, understanding and manipulating problems associated with imprecise and insufficient knowledge and (iii) finds a wide variety of applications related to artificial intelligence and (iv) from the medical point of view, rough set is an excellent tool, which aims at identifying subsets of the most important attributes influencing the treatment of patients. In fact, the main advantage of choosing the critical feature is by simplifying data description which may facilitate physicians to make a sound and prompt diagnosis. Further, collection of fewer features is certainly advantageous since the collection of data is a highly time consuming and expensive job in the medical applications. In different phases of the knowledge discovery process rough set can be used as attribute selection, attribute extraction, data reduction, decision rule generation and pattern extraction[3].

Rough set theory has found many interesting applications. The rough set approach seems to be of fundamental importance to AI and cognitive sciences, especially in the areas of machine learning, knowledge acquisition and decision analysis, knowledge discovery from databases, expert systems, inductive reasoning and pattern recognition. It seems to be of particular importance to decision support systems and data mining and has been successfully applied in many real-life problems in medicine, pharmacology, engineering, banking, financial and market analysis for various engineering applications, like diagnosis of machines using vibroacoustics symptoms (noise, vibrations) and process control. Application in linguistics, environment and databases are other important domains.

Rough set approach to data analysis has many important advantages:

- Provides efficient algorithms for finding hidden patterns in data.
- Identifies relationships that would not be found using statistical methods.
- Allows both qualitative and quantitative data.
- Finds minimal sets of data (data reduction).
- Evaluates significance of data.
- Generates sets of decision rules from data.
- It is easy to understand.
- Offers straightforward interpretation of obtained results.

Most algorithms based on the rough set theory are particularly suited for parallel processing, but in order to exploit this feature fully, a new computer organization based on rough set theory is necessary.

The Genetic Algorithms (GA) are efficient methods for function minimization. A genetic algorithm mimics the natural evolution by modeling a dynamic population of solutions. The members of the population, referred to as chromosomes, encode the selected features. By using the training data, the error of the model is quantified and serves as a fitness function. Genetic algorithm based on the Darwinian survival of the fittest theory, is an efficient and broadly applicable global optimization algorithm. In contrast to conventional search techniques, genetic algorithm starts from a group of points coded as finite length alphabet strings instead of one real parameter set. Furthermore, genetic algorithm is not a hill-climbing algorithm hence the derivative information and step size calculations are not required. The three basic operators of genetic algorithms are: selection, crossover and mutation. It selects some individuals with stronger adaptability from population according to the fitness, and then decides the copy number of individual according to the selection methods such as Backer stochastic universal sampling. It exchanges and recombines a pair of chromosome through crossover. Mutation is done to change certain point state via probability. In general, one needs to choose suitable crossover and mutation probability time and again via real problems.

Authors[1] focus on the extraction of minimal if-then rules from the tables of empirical data which either fully

or approximately describe the given example classifications. Author[2] describes the extension of Rough Set under Incomplete Information Systems. Author[4] discusses approaches to missing attribute values, based on "do not care" conditions and lost values using rough set methodology, including attribute-value pair blocks, characteristic sets, and characteristic relations. A rough set knowledge discovery framework[7] is formulated for the analysis of interval-valued information systems converted from real-valued raw decision tables. Authors[8] focus on the evaluation of a medical database, which shows that induced rules correctly represent experts' decision processes. Author[9] compares standard methods for extracting laws from decision tables based on rough set and boolean reasoning with the method based on dynamic reducts and dynamic rules. Authors[10] introduce a new probabilistic approach for reducing dimensions and extracting rules of information systems using expert systems. Author[11] explores how a patient group in need of a scintigraphic scan can be identified for subsequent modelling. Author[12] applies rough set theory to breast cancer data analysis. Some recent works in this direction[13–15]. Authors[13] discuss the reducts and rules generation by the rough set approach, rules generation by direct and indirect methods, and the generation of optimal rules for cost effectiveness (whether the classifier utility can be improved by altering the misclassification cost ratio which is the ratio of the false positive misclassification costs and the false negative misclassification costs) associated with the dermatology data set. Authors[14] aims at investigating the different Data mining learning models for different medical data sets and to give practical guidelines to select the most appropriate algorithm for a specific medical data set. Authors[15] focus on the theory of Rough set to find dependence relationship among data, evaluate the importance of attributes, discover the patterns of data, learn common decision-making rules, reduce the redundancies and seek the minimum subset of attributes so as to attain satisfactory classification. Authors[16] have presented a new application of rough sets to ECG recognition. Authors[17] have presented a knowledge discovery system based on rough sets and feature-oriented generalization and its application to medicine. Authors[18] have proposed a rough set algorithm to generate diagnostic rules based on the hierarchical structure of differential medical diagnosis. A rough set classification algorithm[19–20] exhibits higher classification accuracy than decision tree algorithms. The generated rules are more understandable than those produced by decision tree methods.

This paper is organized as follows: Section 1 deals with an introduction to the work together with an overview of the Rough Set theory and literature survey. Section 2 discusses the Materials and Methods. Section 4 discusses Experiments and Results and finally the Conclusion is presented in section 5.

2. Materials and Methods

We have used Pima data set for our study, which has been widely used in machine learning experiments and is currently available through the UCI repository of standard data sets. To study the positive as well as the negative aspects of the diabetes disease, Pima data set can be utilized, which contains 768 data samples. Each sample contains 8 attributes which are considered as high risk factors for the occurrence of diabetes, like Plasma glucose concentration, Diastolic blood pressure (mm Hg), Triceps skin fold thickness (mm), 2-hour serum insulin (mu U/ms), Body mass index (weight in kg/(height in m)) Diabetes pedigrees function and Age (years). All the 768 examples were randomly separated into a training set of 576 cases (378, non-diabetic and 198, diabetic) and a test set of 192 cases (122 non-diabetic and 70 diabetic cases)[14].

In this context, some important definitions may be recalled to start with: A data set is represented as a table, where each row represents a case, an event, a patient, or simply an object. Every column represents an attribute (a variable, an observation, a property, etc.) that can be measured for each object; the attribute may be also supplied by a human expert or the user. Such a table is called an information system. Formally, an information system[21,17] is a pair S = (U, A) where U is a non-empty finite set of objects called the universe and A is a non-empty finite set of attributes such that a: $U \rightarrow V_a$ called evaluation function, where V_a is called the value set of a. Elements of U could be interpreted as cases, states, patients, observations etc., for every $a \in A$. The set V_a is called the value set of a. In many applications there is an outcome of classification that is known. This posteriori knowledge is expressed by a distinguished attribute called decision attribute; the process is known as supervised learning. Information systems of this kind are called decision systems.

Indiscernibility Relation is a central concept in Rough Set Theory, and is considered as a relation between two objects or more, where all the values are identical in relation to a subset of considered attributes. Indiscernibility relation is an equivalence relation, where all identical

objects of set are considered as elementary. The starting point of rough set theory is the indiscernibility relation, generated by information concerning objects of interest. The indiscernibility relation is intended to express the fact that due to the lack of knowledge it is unable to discern some objects employing the available information. Approximation is also another important concept in Rough Set Theory, which is being associated with the meaning of the approximations in topological operations[22–23]. The lower and the upper approximations of a set are the interior and closure operations in a topology generated by the indiscernibility relation.

The process of reducing an information system such that the set of attributes of the reduced information system is independent and no attribute can be eliminated further without losing some information from the system yields the result known as reduct. A reduct is a set of attributes that preserves the basic characteristics of the original data set; therefore, the attributes that do not belong to a reduct are superfluous with regard to classification of elements of the Universe. If an attribute from the subset B∈A preserves the indiscernibility relation RA, then the attributes A-B are dispensable. Reducts are such minimal subsets that they do not contain any dispensable attributes. Therefore, the reduction should have the capacity to classify objects, without altering the form of representing the knowledge[8]. Information tables have two kinds of attributes, called condition and decision attributes. Such tables are known as decision tables. Rows of a decision table are referred to as "if...then..." decision rules, which give conditions necessary to make decisions specified by the decision attributes. The general form of a rule can be expressed as : if X then Y, where X is a conditional part made up of conditional attributes and Y is a part made up of decision attributes.

Let us have equivalence relation $R \subseteq U \times U$. The partition of set U is a set of nonempty subsets {X1,X2,…,Xk}, where $X1 \cup X2 \cup ... \cup X_k$ = U and $X_i \cap X_j$ for $i \neq j$, otherwise the partition of set in reciprocally disjunctive subsets. These subsets are called classes. If we have an equivalence relation, we can say, that items in one class are reciprocally in relation and there are no relations with items in different classes. Let us identify the partition induct by relation R and items as R(x). Let's call these subsets equivalence classes. Then, the lower approximation of X can be defined as (Figure 1):

$$R_*(X) = \bigcup_{x \in U} \{R(x) : R(x) \subseteq X\}$$

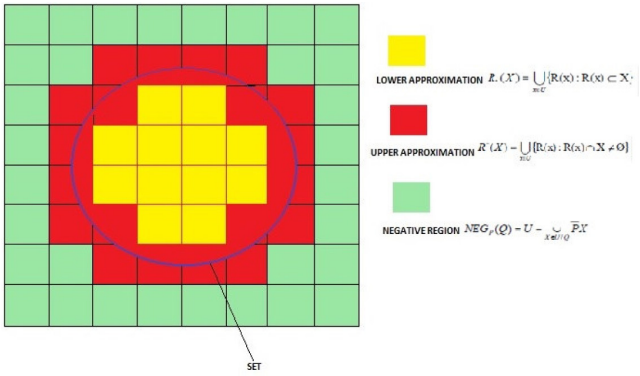

Figure 1. Illustration of the boundary region of rough set.

Upper approximation of X as:

$$R^*(X) = \bigcup_{x \in U} \{R(x) : R(x) \cap X \neq \varnothing\}$$

Boundary region of X as:

$$BN_R(X) = R^*(X) - R.(X)$$

Rough sets can be also defined by membership function

$$\mu_X^R : U \to \langle 0,1 \rangle$$

where

$$\mu_X^R(x) = \frac{cardinality(X \cap R(x))}{cardinality(R(x))}$$

2.1 Importance of Attributes

Suppose, sets C and D and attribute $a \subseteq C$, then we can set up an importance metric of attribute a, expressed as a modification of the degree of dependency.

$$\sigma_{(C,D)}(a) = 1 - \frac{\gamma(C - \{a\}, D)}{\gamma(C, D)}$$

$\sigma_{(C,D)}(\text{Reduct } C) = 1$ applies for reduct C and that means, that if we remove these attributes, we cannot make the decision certainly.

2.2 Rule Induction

It is emphasized that the number of all minimal consistent decision rules for a given decision table can be exponential with respect to the size of decision table. Four heuristics have been implemented in RSES:

2.2.1 Genetic Algorithm

One can compute a predefined number of minimal consistent rules with genetic algorithm that comprises permutation encoding and special crossover operator[24–25]. Genetic algorithm is presented in Figure 2.

2.2.2 Exhaustive Algorithm

This algorithm realizes the computation of some minimal consistent decision rules for a given decision table S can be obtained from objects by reduction of redundant descriptors (Figure 3). The method is based on Boolean reasoning approach.

In the RS approach, Exhaustive (Quick) algorithm seeks the possibility of all decision rules built-up by conditional attributes C and decision attributes D.

Step 1: Discretization of quantitative attributes.
Step 2: Generating rules by quick algorithm.
Step 3: Distribution of rules among decision classes
Step 4: Calculating quality of rules
Step 5: Shortening rules with parameter 0.8
Step 6: Generation of decision rules

a) We choose the first attribute and calculate approximations of division according to D for single items.

b) If there is no boundary region for a definite item, then the decision rule can be made with 100% possibility.

c) If there are items with a boundary region, other attributes from C are added and the process is repeated. If the data is inconsistent, then we cannot unambiguously define all rules. After using all the attributes, the rules are made and they have the possibility calculated as the ratio of upper and lower approximation of sets.

The result of an algorithm is a set of decision rules that has a 100% support and a set of rules that has a lower support. For each of these, rules the support means when the rule is applicable.

```
Choose initial population
 Evaluate the fitness of each individual in the population
 Repeat
   Select best-ranking individuals to reproduce
   Breed new generation through crossover and mutation (genetic operations) and
   give birth to offspring
   Evaluate the individual fitnesses of the offspring
   Replace worst ranked part of population with offspring
        Until <terminating condition>
```

Figure 2. Genetic Algorithm.

```
exhaustive(intsol,intdepth)
{ if(issolution(sol))
printsolution(sol)
else
{ solgenerated=generatesolution()
  exhaustive(solgenerated,depth+1) }
```

Figure 3. Exhaustive algorithm.

2.2.3 Covering Algorithm

This algorithm searches for minimal (or very close to minimal) set of rules which cover the whole set of objects[26] which is presented in Figure 4.

2.2.4 LEM2 Algorithm

It is to be noted that LEM2 algorithm is another kind of covering algorithm which has been used in the present study[27] and is represented in Figure 5.

```
Input: labeled training dataset D
Output: Rule set R that covers all instances in D
Procedure:
  Initialize R as the empty set
  for each class C {
  while D is nonempty {
       Construct one rule r that correctly classifies some instances in D that  belong to class
C and does not
       incorrectly classify any non-C instances
    Add rule r to Rule set R
    Remove from D all instances correctly classified by r}}
  return R
```

Figure 4. Covering algorithm.

```
Procedure LEM2
(input: a set B, output: a single local covering 𝔉 of set B);
 begin
G := B;
𝔉 := ∅;
while G ≠ ∅
 begin
 T := ∅;
T(G) :={t|[t]∩G ≠ ∅} ;
while T = ∅ or [T] ⊆ B
begin
 select a pair t ∈ T(G) such that |[t]∩G| is maximum;
if a tie occurs, select a pair t ∈ T(G) with the smallest cardinality of [t];
 if another tie occurs, select first pair;
 T := T ∪{ t}; G := [t]∩G ;
T(G) :={t|[t]∩G ≠ ∅};
T(G) :=T(G)−T ;
End {while}
for each t ∈ T do
 if [T −{ t}] ⊆ B then T := T −{ t};
𝔉 := 𝔉 ∪{ T};
G := B −∪T∈ T [T];
end {while};
for each T ∈ 𝔉 do
if U S∈ 𝔉−{T} [S]=B then 𝔉 := 𝔉 −{ T};
end {procedure}.
```

Figure 5. LEM2 algorithm.

3. Experiments and Results

The results of the experiments conducted on the PIMA data set by using RSES system provides many interesting and valuable information in respective applications. The following important observations have been made from the present investigation: Rule generation (without reducts) (ii) Discretization and (iii) Decomposition tree.

A detailed analysis pertaining to rule generation through reducts has been done. In this context, it is emphasized that the medical experts should verify the decision rules to check whether the extracted rules provide some new knowledge in making effective decisions. Rough set rule induction algorithms generate decision rules[12–14], which may potentially reveal profound medical knowledge and provide new medical insight. These decision rules are more useful for medical experts to analyze and gain understanding into the problem at hand.

We can set several parameters that control discretization/grouping procedure:

Discretize/Discretize table refers to the discretization (group) of attributes in the data table with use of previously calculated cuts. The user sets the number of cuts to be used and the names of the objects to be stored, resulting in a discretized table. Method of choice includes choice of discretization method from either (i) Global method or (ii) Local method. The local method is slightly faster than the global method and generates much more cuts in some cases. Local method include symbolic attributes that causes algorithms to perform grouping for nominal (symbolic) attributes in parallel to discretization of numerical ones. Cuts are used to determine the names of the objects to store cuts.

Table 1 represents rule generation using the entire data set where we have applied discretization on all the four algorithms. Rule improvement for all the cases is performed by applying the shortening ratio as 0.8. Further, it is observed that the maximum coverage happens to be for Exhaustive and genetic algorithms.

Tables 2–5 represent the distribution of classes among the decision classes for Genetic, LEM2, Exhaustive and Covering algorithms using global and local discretization.

In the global method, only a part of the object is considered at every step. While in the local method, only the part of objects that is concerned with the candidate cut (i.e., which has the value of the currently considered attribute in the same range as the cut candidate) is considered. The local method produces more cuts and is faster as less objects have to be examined at every step. Another advantage of local method is that it is capable of dealing with nominal (symbolic) attributes. Further, the grouping (quantization) of nominal attribute domain with use of local method always results in two subsets of attribute values.

Table 1. Rule generation

Algorithms	Coverage (%)	Global Rules	Local Rules	Rule Shorten (Global)	Rule Shorten (Local)
Exhaustive	100	3485	9356	1730	6030
Covering	69.9	123	310	75	221
Lem2	90	286	350	159	248
Genetic	100	2772	5203	2307	4638

Table 2. Genetic: Distribution of rules among decision classes

Decision Class	Global Count	Local Count
Tested_negative	1499	3032
Tested_positive	1273	2171

Table 3. Lem2: Distribution of rules among decision classes

Decision Class	Global Count	Local Count
Tested_negative	154	197
Tested_positive	132	153

Table 4. Covering: Distribution of rules among decision classes

Decision Class	Global Count	Local Count
Tested_negative	59	181
Tested_positive	64	129

Table 5. Exhaustive: Distribution of rules among decision classes

Decision Class	Global Count	Local Count
Tested_negative	1857	5283
Tested_positive	1628	4073

3.1 Rule Improvement by RSES

After calculation of decision rules, one can select some most interesting rules for further usage. Some time we cannot do it because of the low support of calculated rules. In RSES, the support of decision rules can be increased by the following ways:

Rule generation → (Discretization (local or global) + Algorithms + Rule shortening).

1. Discretization of real value attributes: two discretization methods called "local" and "global" methods have been implemented in RSES. Both the methods are based on Boolean reasoning approach. The local method is built on decision tree and it usually products more cuts than global method.
2. Rule shortening: Removing some descriptors from a given decision rule can increase its support, but it decreases its confidence. In RSES, we can determine the "shortening ratio", which is a minimal acceptable threshold for confidences of decision rules in shortening process.
3. Generalization of rules: By generalized decision rules we denote the implications of the form: $r =_{def} (a_{i1} \in S_1) \wedge \ldots \wedge (a_{im} \in S_m) \Rightarrow (dec = k)$ where $S_j \subset V_{aij}$. RSES can construct generalized decision rules by merging those rules containing some common descriptors.

Table 6 presents the results pertaining to rule generation through covering algorithm using global rule criteria which generates the strongest decision rules for the class tested_positive, i.e., group of patients who will have problem with diabetic. The last column of the table reveals the number of matches associated with that particular rule. For example the number of matches for rule 2 happens to be 10 and so on.

Table 7 presents the results pertaining to rule generation through exhaustive method using global rule criteria which generates the strongest decision rules for the class tested_positive, i.e., group of patients who will have problem with diabetic. The last column of the above table reveals the number of matches associated with that particular rule. For example the number of matches for rule 1 happens to be 98 and so on.

Table 8 presents the results pertaining to rule generation through exhaustive method using global rule criteria which generates the strongest decision rules for the class tested_negative, i.e., group of patients who do not have problem with diabetic. The last column of the table reveals the number of matches associated with that

Table 7. Exhaustive algorithm-global method

(1-173...	Decision rules	Match
1	('plas'='(154.5,Inf)')=>('class'={tested_positive[98]})	98
2	('preg'='(6.5,Inf)')&('skin'='(28.5,Inf)')&('pedi'='(0.5605,Inf)')=>('class'={tested_positive[24]})	24
3	('preg'='(6.5,Inf)')&('insu'='(137.5,Inf)')=>('class'={tested_positive[30]})	30
4	('skin'='(28.5,Inf)')&('pedi'='(0.5605,Inf)')&('age'='(41.5,Inf)')=>('class'={tested_positive[19]})	19
5	('preg'='(-Inf,1.5)')&('mass'='(33.85,38.25)')&('age'='(30.5,41.5)')=>('class'={tested_positive[8]})	8
6	('preg'='(2.5,6.5)')&('plas'='(154.5,Inf)')&('age'='(22.5,30.5)')=>('class'={tested_positive[12]})	12
7	('skin'='(-Inf,28.5)')&('insu'='(137.5,Inf)')&('age'='(41.5,Inf)')=>('class'={tested_positive[17]})	17
8	('pres'='(85.5,Inf)')&('insu'='(137.5,Inf)')&('age'='(41.5,Inf)')=>('class'={tested_positive[9]})	9
9	('preg'='(6.5,Inf)')&('mass'='(38.25,Inf)')&('pedi'='(0.5605,Inf)')=>('class'={tested_positive[11]})	11
10	('pres'='(73.0,85.5)')&('insu'='(137.5,Inf)')&('age'='(41.5,Inf)')=>('class'={tested_positive[15]})	15
11	('preg'='(6.5,Inf)')&('pres'='(-Inf,73.0)')&('pedi'='(0.5605,Inf)')&('age'='(30.5,41.5)')=>('class'={teste...	10
12	('preg'='(-Inf,1.5)')&('plas'='(123.5,154.5)')&('age'='(30.5,41.5)')=>('class'={tested_positive[6]})	6
13	('pres'='(-Inf,73.0)')&('mass'='(38.25,Inf)')&('age'='(30.5,41.5)')=>('class'={tested_positive[9]})	9
14	('mass'='(33.85,38.25)')&('pedi'='(0.377,0.5605)')&('age'='(30.5,41.5)')=>('class'={tested_positive[6...	6
15	('preg'='(6.5,Inf)')&('insu'='(-Inf,20.0)')&('mass'='(38.25,Inf)')&('pedi'='(0.208,0.377)')=>('class'={tes...	6
16	('preg'='(6.5,Inf)')&('pres'='(85.5,Inf)')&('pedi'='(0.5605,Inf)')=>('class'={tested_positive[8]})	8
17	('pres'='(85.5,Inf)')&('mass'='(29.95,32.65)')&('age'='(41.5,Inf)')=>('class'={tested_positive[6]})	6
18	('preg'='(-Inf,1.5)')&('pedi'='(0.377,0.5605)')&('age'='(30.5,41.5)')=>('class'={tested_positive[5]})	5
19	('pres'='(85.5,Inf)')&('skin'='(28.5,Inf)')&('insu'='(137.5,Inf)')&('mass'='(33.85,38.25)')=>('class'={te...	5
20	('pres'='(-Inf,73.0)')&('insu'='(-Inf,20.0)')&('mass'='(32.65,33.85)')&('pedi'='(0.208,0.377)')=>('class'...	5
21	('mass'='(29.95,32.65)')&('pedi'='(0.5605,Inf)')&('age'='(30.5,41.5)')=>('class'={tested_positive[7]})	7
22	('plas'='(123.5,154.5)')&('pres'='(-Inf,73.0)')&('mass'='(38.25,Inf)')=>('class'={tested_positive[12]})	12
23	('plas'='(123.5,154.5)')&('pedi'='(0.377,0.5605)')&('age'='(41.5,Inf)')=>('class'={tested_positive[8]})	8
24	('mass'='(29.95,32.65)')&('pedi'='(0.208,0.377)')&('age'='(41.5,Inf)')=>('class'={tested_positive[5]})	5
25	('pres'='(-Inf,73.0)')&('insu'='(-Inf,20.0)')&('mass'='(33.85,38.25)')&('pedi'='(0.5605,Inf)')=>('class'={...	5
26	('preg'='(6.5,Inf)')&('pres'='(-Inf,73.0)')&('mass'='(38.25,Inf)')=>('class'={tested_positive[7]})	7
27	('plas'='(123.5,154.5)')&('pedi'='(0.5605,Inf)')&('age'='(30.5,41.5)')=>('class'={tested_positive[12]})	12
28	('preg'='(6.5,Inf)')&('pres'='(-Inf,73.0)')&('pedi'='(0.377,0.5605)')&('age'='(30.5,41.5)')=>('class'={te...	4
29	('pres'='(73.0,85.5)')&('insu'='(137.5,Inf)')&('mass'='(38.25,Inf)')&('pedi'='(0.377,0.5605)')=>('class'...	4
30	('pres'='(85.5,Inf)')&('insu'='(-Inf,20.0)')&('mass'='(38.25,Inf)')&('age'='(30.5,41.5)')=>('class'={test...	4
31	('insu'='(137.5,Inf)')&('pedi'='(0.208,0.377)')&('age'='(41.5,Inf)')=>('class'={tested_positive[5]})	5

Table 6. Covering algorithm-global method

(1-75)	Decision rules	Match
2	('pedi'='(0.5605,Inf)')&('mass'='(33.85,38.25)')&('pres'='(73.0,85.5)')=>('class'={tested_positive[10]})	10
3	('plas'='(123.5,154.5)')&('pedi'='(0.377,0.5605)')&('mass'='(38.25,Inf)')=>('class'={tested_positive[7]})	7
4	('plas'='(123.5,154.5)')&('mass'='(38.25,Inf)')&('pres'='(-Inf,73.0)')=>('class'={tested_positive[12]})	12
5	('pedi'='(0.5605,Inf)')&('preg'='(6.5,Inf)')&('mass'='(38.25,Inf)')=>('class'={tested_positive[11]})	11
6	('plas'='(123.5,154.5)')&('pedi'='(0.208,0.377)')&('preg'='(2.5,6.5)')&('mass'='(33.85,38.25)')&('pres'='(73.0,85....	2
7	('plas'='(123.5,154.5)')&('pedi'='(0.208,0.377)')&('preg'='(1.5,2.5)')&('mass'='(-Inf,29.4)')&('pres'='(-Inf,73.0)')=...	2
8	('plas'='(123.5,154.5)')&('pedi'='(0.377,0.5605)')&('preg'='(2.5,6.5)')&('mass'='(29.95,32.65)')=>('class'={tested...	4
9	('plas'='(-Inf,105.5)')&('pedi'='(0.377,0.5605)')&('preg'='(6.5,Inf)')&('mass'='(29.95,32.65)')&('pres'='(-Inf,73.0)'''...	2
10	('plas'='(105.5,123.5)')&('pedi'='(0.5605,Inf)')&('preg'='(2.5,6.5)')&('mass'='(33.85,38.25)')&('pres'='(-Inf,73.0)')...	2
11	('plas'='(123.5,154.5)')&('pedi'='(0.5605,Inf)')&('preg'='(2.5,6.5)')&('mass'='(29.95,32.65)')=>('class'={tested_p...	2
12	('plas'='(123.5,154.5)')&('pedi'='(0.208,0.377)')&('preg'='(2.5,6.5)')&('mass'='(38.25,Inf)')=>('class'={tested_po...	2
13	('plas'='(123.5,154.5)')&('pedi'='(-Inf,0.208)')&('preg'='(6.5,Inf)')&('mass'='(33.85,38.25)')=>('class'={tested_po...	2
14	('plas'='(-Inf,105.5)')&('pedi'='(0.5605,Inf)')&('preg'='(1.5,2.5)')&('mass'='(32.65,33.85)')=>('class'={tested_posi...	2
15	('plas'='(154.5,Inf)')&('pedi'='(-Inf,0.208)')&('preg'='(2.5,6.5)')=>('class'={tested_positive[5]})	5
16	('plas'='(154.5,Inf)')&('pedi'='(0.208,0.377)')&('mass'='(29.4,29.95)')=>('class'={tested_positive[2]})	2
17	('plas'='(123.5,154.5)')&('pedi'='(0.5605,Inf)')&('preg'='(2.5,6.5)')&('pres'='(-Inf,73.0)')=>('class'={tested_positiv...	4
18	('plas'='(123.5,154.5)')&('preg'='(-Inf,1.5)')&('mass'='(33.85,38.25)')&('pres'='(-Inf,73.0)')=>('class'={tested_po...	4
19	('plas'='(123.5,154.5)')&('pedi'='(0.208,0.377)')&('preg'='(6.5,Inf)')&('mass'='(38.25,Inf)')&('pres'='(73.0,85.5)')...	1
20	('plas'='(123.5,154.5)')&('pedi'='(0.377,0.5605)')&('preg'='(2.5,6.5)')&('mass'='(33.85,38.25)')&('pres'='(-Inf,73....	1
21	('plas'='(123.5,154.5)')&('pedi'='(0.377,0.5605)')&('preg'='(6.5,Inf)')&('mass'='(29.95,32.65)')&('pres'='(73.0,85....	1
22	('plas'='(-Inf,105.5)')&('pedi'='(0.5605,Inf)')&('preg'='(2.5,6.5)')&('mass'='(-Inf,29.4)')&('pres'='(73.0,85.5)')=>('...	1
23	('plas'='(-Inf,105.5)')&('pedi'='(0.208,0.377)')&('preg'='(-Inf,1.5)')&('mass'='(38.25,Inf)')&('pres'='(73.0,85.5)')=...	1
24	('plas'='(-Inf,105.5)')&('pedi'='(-Inf,0.208)')&('preg'='(2.5,6.5)')&('mass'='(29.95,32.65)')&('pres'='(73.0,85.5)')=...	1
25	('plas'='(-Inf,105.5)')&('pedi'='(0.377,0.5605)')&('preg'='(2.5,6.5)')&('mass'='(33.85,38.25)')&('pres'='(73.0,85.5)...	1
26	('plas'='(-Inf,105.5)')&('pedi'='(0.377,0.5605)')&('preg'='(-Inf,1.5)')&('mass'='(32.65,33.85)')&('pres'='(-Inf,73.0)...	1
27	('plas'='(154.5,Inf)')&('pedi'='(-Inf,0.208)')&('preg'='(6.5,Inf)')&('mass'='(-Inf,29.4)')&('pres'='(-Inf,73.0)')=>('cla...	1
28	('pedi'='(0.208,0.377)')&('preg'='(2.5,6.5)')&('mass'='(29.95,32.65)')&('pres'='(73.0,85.5)')=>('class'={tested_p...	4
29	('pedi'='(0.208,0.377)')&('preg'='(-Inf,1.5)')&('mass'='(38.25,Inf)')&('pres'='(85.5,Inf)')=>('class'={tested_positiv...	5
30	('plas'='(105.5,123.5)')&('pedi'='(-Inf,0.208)')&('preg'='(6.5,Inf)')&('mass'='(-Inf,29.4)')&('pres'='(-Inf,73.0)')=>('...	1
31	('plas'='(123.5,154.5)')&('pedi'='(0.5605,Inf)')&('mass'='(32.65,33.85)')=>('class'={tested_positive[4]})	4
32	('pedi'='(0.5605,Inf)')&('preg'='(6.5,Inf)')&('mass'='(32.65,33.85)')=>('class'={tested_positive[5]})	5

Table 8. LEM2 algorithm-global Method

(1-159)	Decision rules	Match
96	('plas'="(-Inf,105.5)")=>('class'={tested_negative[232]})	232
97	('age'="(-Inf,22.5)")=>('class'={tested_negative[119]})	119
98	('mass'="(-Inf,29.4)")=>('class'={tested_negative[229]})	229
99	('pres'="(73.0,85.5)")&('plas'="(123.5,154.5)")&('preg'="(-Inf,1.5)")=>('class'={tested_negative[18]})	18
100	('insu'="(20.0,137.5)")=>('class'={tested_negative[174]})	174
101	('preg'="(1.5,2.5)")=>('class'={tested_negative[84]})	84
102	('preg'="(2.5,6.5)")&('pedi'="(0.208,0.377)")&('age'="(22.5,30.5)")&('plas'="(105.5,123.5)")=>('clas...	7
103	('skin'="(28.5,Inf)")&('preg'="(2.5,6.5)")&('pres'="(85.5,Inf)")=>('class'={tested_negative[9]})	9
104	('preg'="(2.5,6.5)")&('plas'="(105.5,123.5)")&('pres'="(85.5,Inf)")=>('class'={tested_negative[9]})	9
105	('age'="(41.5,Inf)")&('plas'="(105.5,123.5)")&('pres'="(85.5,Inf)")=>('class'={tested_negative[5]})	5
106	('pres'="(73.0,85.5)")&('plas'="(123.5,154.5)")&('mass'="(-Inf,29.4)")=>('class'={tested_negative[2...	20
107	('pedi'="(0.208,0.377)")&('age'="(30.5,41.5)")&('preg'="(6.5,Inf)")&('mass'="(-Inf,29.4)")=>('class'=...	3
108	('age'="(22.5,30.5)")&('plas'="(105.5,123.5)")=>('class'={tested_negative[68]})	68
109	('age'="(22.5,30.5)")&('plas'="(123.5,154.5)")&('insu'="(137.5,Inf)")&('mass'="(33.85,38.25)")=>('cl...	6
110	('pres'="(-Inf,73.0)")&('preg'="(2.5,6.5)")&('plas'="(123.5,154.5)")&('age'="(30.5,41.5)")=>('class'={...	5
111	('insu'="(137.5,Inf)")&('mass'="(38.25,Inf)")&('pres'="(85.5,Inf)")&('plas'="(123.5,154.5)")=>('class'...	3
112	('pedi'="(-Inf,0.208)")&('plas'="(105.5,123.5)")=>('class'={tested_negative[33]})	33
113	('insu'="(-Inf,20.0)")&('preg'="(2.5,6.5)")&('age'="(22.5,30.5)")&('mass'="(33.85,38.25)")&('pedi'="(...	2
114	('pres'="(-Inf,73.0)")&('plas'="(105.5,123.5)")&('preg'="(6.5,Inf)")&('mass'="(33.85,38.25)")=>('clas...	3
115	('pres'="(73.0,85.5)")&('mass'="(-Inf,29.4)")&('age'="(41.5,Inf)")&('preg'="(6.5,Inf)")&('plas'="(154....	2
116	('age'="(22.5,30.5)")&('preg'="(-Inf,1.5)")&('pedi'="(0.377,0.5605)")=>('class'={tested_negative[19]})	19
117	('pres'="(73.0,85.5)")&('plas'="(123.5,154.5)")&('age'="(22.5,30.5)")=>('class'={tested_negative[21...	21
118	('insu'="(-Inf,20.0)")&('preg'="(2.5,6.5)")&('pedi'="(0.5605,Inf)")&('plas'="(105.5,123.5)")=>('class'=...	4
119	('pres'="(73.0,85.5)")&('plas'="(105.5,123.5)")&('preg'="(-Inf,1.5)")=>('class'={tested_negative[10]})	10
120	('pres'="(-Inf,73.0)")&('insu'="(137.5,Inf)")&('plas'="(123.5,154.5)")&('mass'="(33.85,38.25)")=>('cl...	4
121	('mass'="(38.25,Inf)")&('insu'="(137.5,Inf)")&('pres'="(85.5,Inf)")&('pedi'="(0.377,0.5605)")=>('clas...	2
122	('pres'="(73.0,85.5)")&('age'="(22.5,30.5)")&('preg'="(1.5,2.5)")=>('class'={tested_negative[13]})	13
123	('pres'="(73.0,85.5)")&('pedi'="(0.208,0.377)")&('preg'="(2.5,6.5)")&('plas'="(105.5,123.5)")=>('clas...	4
124	('pres'="(-Inf,73.0)")&('plas'="(123.5,154.5)")&('mass'="(-Inf,29.4)")&('preg'="(6.5,Inf)")=>('class'={...	3
125	('skin'="(28.5,Inf)")&('pres'="(-Inf,73.0)")&('insu'="(137.5,Inf)")&('preg'="(2.5,6.5)")&('plas'="(123.5...	2
126	('pres'="(73.0,85.5)")&('plas'="(-Inf,105.5)")&('age'="(30.5,41.5)")&('preg'="(6.5,Inf)")=>('class'={te...	7

Table 9. Genetic algorithm- global Method

(1-230...	Decision rules	Match
1213	('plas'="(-Inf,105.5)")&('age'="(-Inf,22.5)")=>('class'={tested_negative[69]})	69
1214	('plas'="(-Inf,105.5)")&('mass'="(-Inf,29.4)")=>('class'={tested_negative[115]})	115
1215	('preg'="(-Inf,1.5)")&('plas'="(-Inf,105.5)")=>('class'={tested_negative[92]})	92
1216	('mass'="(-Inf,29.4)")&('age'="(-Inf,22.5)")=>('class'={tested_negative[75]})	75
1217	('skin'="(-Inf,28.5)")&('age'="(-Inf,22.5)")=>('class'={tested_negative[97]})	97
1218	('insu'="(20.0,137.5)")&('age'="(-Inf,22.5)")=>('class'={tested_negative[49]})	49
1219	('plas'="(-Inf,105.5)")&('pres'="(-Inf,73.0)")=>('class'={tested_negative[159]})	159
1220	('preg'="(-Inf,1.5)")&('mass'="(-Inf,29.4)")=>('class'={tested_negative[88]})	88
1221	('plas'="(-Inf,105.5)")&('age'="(22.5,30.5)")=>('class'={tested_negative[96]})	96
1222	('mass'="(-Inf,29.4)")&('pedi'="(-Inf,0.208)")=>('class'={tested_negative[61]})	61
1223	('preg'="(1.5,2.5)")&('age'="(-Inf,22.5)")=>('class'={tested_negative[31]})	31
1224	('pedi'="(-Inf,0.208)")&('age'="(-Inf,22.5)")=>('class'={tested_negative[29]})	29
1225	('pres'="(-Inf,73.0)")&('age'="(-Inf,22.5)")=>('class'={tested_negative[91]})	91
1226	('plas'="(-Inf,105.5)")&('skin'="(-Inf,28.5)")=>('class'={tested_negative[164]})	164
1227	('plas'="(-Inf,105.5)")&('insu'="(20.0,137.5)")=>('class'={tested_negative[109]})	109
1228	('preg'="(1.5,2.5)")&('plas'="(105.5,123.5)")=>('class'={tested_negative[27]})	27
1229	('plas'="(105.5,123.5)")&('insu'="(20.0,137.5)")&('pedi'="(0.208,0.377)")=>('class'={...	16
1230	('plas'="(-Inf,105.5)")&('pedi'="(-Inf,0.208)")=>('class'={tested_negative[51]})	51
1231	('preg'="(1.5,2.5)")&('mass'="(-Inf,29.4)")&('pedi'="(0.5605,Inf)")=>('class'={tested...	15
1232	('plas'="(-Inf,105.5)")&('skin'="(28.5,Inf)")&('mass'="(33.85,38.25)")&('pedi'="(0.20...	13
1233	('preg'="(1.5,2.5)")&('pedi'="(-Inf,0.208)")=>('class'={tested_negative[19]})	19
1234	('skin'="(-Inf,28.5)")&('insu'="(20.0,137.5)")&('mass'="(-Inf,29.4)")=>('class'={teste...	69
1235	('plas'="(-Inf,105.5)")&('pedi'="(0.208,0.377)")=>('class'={tested_negative[76]})	76
1236	('preg'="(-Inf,1.5)")&('skin'="(-Inf,28.5)")&('insu'="(20.0,137.5)")=>('class'={tested_...	48
1237	('mass'="(-Inf,29.4)")&('pedi'="(0.377,0.5605)")=>('class'={tested_negative[37]})	37
1238	('preg'="(1.5,2.5)")&('insu'="(20.0,137.5)")&('mass'="(-Inf,29.4)")=>('class'={tested...	15
1239	('preg'="(1.5,2.5)")&('insu'="(20.0,137.5)")&('pedi'="(0.208,0.377)")=>('class'={test...	11
1240	('pres'="(73.0,85.5)")&('insu'="(20.0,137.5)")&('pedi'="(-Inf,0.208)")=>('class'={test...	11
1241	('preg'="(-Inf,1.5)")&('plas'="(123.5,154.5)")&('pres'="(73.0,85.5)")=>('class'={teste...	18
1242	('preg'="(2.5,6.5)")&('mass'="(-Inf,29.4)")&('age'="(22.5,30.5)")=>('class'={tested_...	30
1243	('plas'="(105.5,123.5)")&('pres'="(73.0,85.5)")&('insu'="(20.0,137.5)")=>('class'={te...	11
1244	('plas'="(123.5,154.5)")&('pres'="(73.0,85.5)")&('insu'="(-Inf,20.0)")&('mass'="(-Inf	10

particular rule. For example the number of matches for rule 96 happens to be 232 and so on.

Table 9 presents the results pertaining to rule generation through exhaustive method using global rule criteria which generates the strongest decision rules for the class tested_negative, i.e., group of patients who do not have problem with diabetic. The last column of the table reveals the number of matches associated with that particular rule. For example, the number of matches for rule 1213 happens to be 69 and so on.

3.2 Decomposition Tree

This names the object in the application which is used to store the resulting decomposition tree. The parameters that may be set by the user when starting decomposition tree algorithm:

- Maximal size of leaf – maximal number of objects (rows) in the subtable corresponding to the leaf in decomposition tree. If this number is not exceeded in any of tree nodes, the algorithm ceases to work.
- Discretization in leafs – by activating this option the user orders the algorithm to perform on-the-fly discretization of attributes during decomposition. The decision on discretization of particular attribute is taken automatically by the algorithm but, the user has control over some of discretization parameters.
- Shortening ratio – The ratio of shortening applied to rules that are calculated by algorithm for subtables corresponding to decomposition tree nodes. For the value of 1:0, no shortening occurs. The smaller this ratio the more "aggressive" is the shortening.

The process of constructing a decomposition tree is fully automated, in the sense that the user has to decide only about the maximal size of subtable corresponding to the leaf. In the present study, the maximal size considered is 65 for Node 12. At subsequent levels of the tree, conditions are generated by the algorithm sequentially and these are formulated as constraints for the respective attribute values. In this way, each node in the tree has an associated template along with it's subset of training sample. Even in the case of data with numerical attributes, decomposition trees could be generated in which case discretization is performed during the selection of conditions at the tree nodes. This is presented in Table 10.

Table 11 presents the information in detail with regard to the size and the rules generated at a particular leaf node.

It is important to note that for the splitting of data set into fragments, decomposition tree is used where the split fragments are within the predefined size. After the decomposition, the fragments are represented as leaves which are supposed to be more uniform and easier to handle (Figure 6).

Table 11. Decomposition tree - Node Details

Node No.	Size of leaf	No. of rules
8	64	34
9	64	26
10	64	27
11	63	14
12	65	26
13	64	26
14	64	17
15	64	10

Table 10. Node information in Decomposition Tree

Node No.	Node Info.	Node No.	Node Info.
2	(DBP>=72)	9	(DBP>=72)&(PG>=123) &(TRICEPS<25)
3	(DBP<72)	10	(DBP>=72)&(PG<123) &(PG>=103)
4	(DBP>=72)&(PG>=123	11	(DBP>=72)&(PG<123) &(PG<103)
5	(DBP>=72)&(PG<123)	12	(DBP<72)&(PG>=112) &(BMI>=32)
6	(DBP<72)&(PG>=112)	13	(DBP<72)&(PG>=112) &(BMI<32)
7	(DBP<72)&(PG<112)	14	(DBP<72)&(PG<112) &(SERUM>=44)
8	(DBP>=72)&(PG>=123) &(TRICEPS>=25)	15	(DBP<72)&(PG<112) &(SERUM<44)

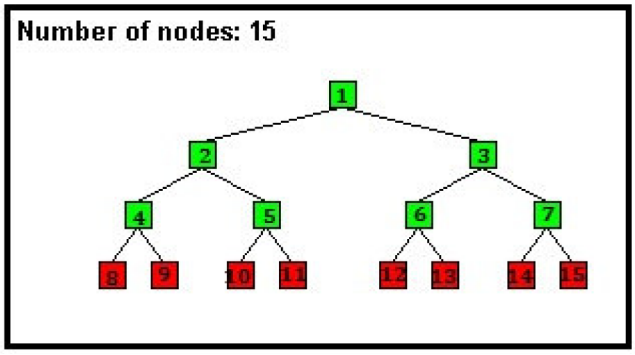

Figure 6. Decomposition Tree.

4. Conclusion

The process of finding useful patterns or meaning in raw data has been called knowledge discovery in databases. The starting point of rough set theory is the indiscernibility relation generated by information concerning objects of interest. Approximation is also another important concept in Rough Set Theory, which is being associated with the meaning of the approximations in topological operations. The lower and the upper approximations of a set are the interior and closure operations in a topology generated by the indiscernibility relation. In recent years Classical rough set theory has made a great success in the field of knowledge acquisition. We have used Pima data set for our study, which has been widely used in machine learning experiments and is currently available through the UCI repository of standard data sets. The success of machine learning associated with medical data sets is strongly affected by many factors and one such factor is the quality of the data which depends on irrelevant, redundant and noisy data. Thus, when the data quality is not excellent, the prediction of knowledge discovery during the training process becomes an arduous task. The existing intelligent techniques of medical data analysis are concerned with (i) Treatment of incomplete knowledge (ii) Management of inconsistent pieces of information and (iii) Manipulation of various levels of representation of data. This difficulty is minimized by feature selection which identifies and removes the irrelevant and redundant features in the data to a great extent. Medical data mining has great potential for exploring the hidden patterns in the data sets of the medical domain and these patterns can be utilized for clinical diagnosis. However, the available raw medical data are widely distributed, heterogeneous in nature and voluminous. These data need to be collected in an organized form and made available to a Hospital Information System (HIS). Actually, medical databases have accumulated large quantities of information about patients and their medical conditions. Relationships and patterns within this data could provide new medical knowledge.

The genetic algorithms offer an attractive approach for solving the feature subset selection problem. The algorithms used for the present study are: Exhaustive search, Covering, LEM2 and Genetic algorithms. Rules are generated and improved in the case of the above mentioned four algorithms. Further, the generated rules are improved by applying the shortening ratio as 0.8. Some of the important results of the present investigation are: (i) Maximum coverage of 100% is observed in the case of exhaustive and genetic algorithms (ii) Table 6 presents the results pertaining to rule generation through Covering method using global rule criteria which generates the strongest decision rules for the class tested_positive, i.e., group of patients who will have problem with diabetic. The last column of the table reveals the number of matches associated with that particular rule. For example the number of matches for rule 2 happens to be 10 and so on (iii) Table 7 presents the results pertaining to rule generation through exhaustive method using global rule criteria which generates the strongest decision rules for the class tested_negative, i.e., group of patients who will have problem with diabetic. The last column of the above table reveals the number of matches associated with that particular rule. For example the number of matches for rule 1 happens to be 98 and so on (iii) Table 8 presents the results pertaining to rule generation through exhaustive method using global rule criteria which generates the strongest decision rules for the class tested_negative, i.e., group of patients who do not have problem with diabetic. The last column of the table reveals the number of matches associated with that particular rule. For example the number of matches for rule 96 happens to be 232 and so on (iv) Table 9 presents the results pertaining to rule generation through exhaustive method using global rule criteria which generates the strongest decision rules for the class tested_negative, i.e., group of patients who do not have problem with diabetic. The last column of the table reveals the number of matches associated with that particular rule. For example, the number of matches for rule 1213 happens to be 69 and so on.

The work is first of its kind and the results provide an excellent platform for efficient medical diagnosis and decision.

5. Acknowledgement

Mrs. Manjula Sanjay Koti is grateful to Dayananda Sagar College of Engineering, Bangalore and Bharathiar University, Tamil Nadu for providing the facilities to carry out the research work.

6. References

1. Øhrn A, Rowland T. Rough sets: a knowledge discovery technique for multifactorial medical outcomes. American Journal of Physical Medicine and Rehabilitation. 2000; 79(1):100–08.
2. Wang G. A new extension of rough set under incomplete information. Rough Sets and Knowledge Technology-Lecture Notes in Computer Science. 2006; 4062:141–46.
3. Komorowski J, Pawlak Z, Polkowski L, Skowron A. Rough sets: A tutorial. Rough Fuzzy Hybridization: A New Trend in Decision Making. Pal SK, Skowron A, editors. Springer; 1999.
4. Grzymala-Busse JW. Three approaches to missing attribute values - a rough set perspective. Data Mining: Foundations and Practice. 2008; 139–52.
5. Komorowski J, Ohrn A. Modelling prognostic power of cardiac tests using rough sets. Artificial Intelligence Med. 1999; 15:167–91.
6. Pawlak Z, Ziarko W. Rough sets: probabilistic versus deterministic approach. Int J Man Mach Stud. 1988; 29(1):81–95.
7. Leung Y, Fischer MM, Wu W-Z, Mi J-S. A rough set approach for the discovery of classification rules in interval-valued information systems. International Journal of Approximate Reasoning. 2008; 47(2):233–46.
8. Bazan J. A Comparison of dynamic and non-dynamic rough set methods for extracting laws from decision tables. Rough Sets in Knowledge Discovery, PhysicaVerlag. (Chapter 17); 1998.
9. Komorowski J, Ohrn A. Modelling prognostic power of cardiac tests using rough sets. Artificial Intelligence Med; 1999; 15:167–91.
10. Hossam A, Nabwey. A probabilistic rough set approach to rule discovery. Int J Adv Comput Sci Tech. 2011; 30: 25–34.
11. Matel O. Applying rough sets algorithm for radiography diagnosis. Proceedings of 9th International Conference on Development and Application Systems; 2008. Suceava, Romania. p. 272–77.
12. Hassanien AE, Ali JMH. Rough set approach for generation of Classification rules of breast cancer data. Informatica. 2004; 15(1):23–38.
13. Srimani PK, Koti MS. Cost sensitivity analysis and the prediction of optimal rules for medical data by using rough set theory. International Journal of Industrial and Manufacturing Engineering. 2012:74–80.
14. Srimani PK, Koti MS. A Comparison of different learning models used in data mining for medical data. The Smithsonian Astrophysics Data System, American Institute of Physics Conf. Proceedings 1414. 51–55. doi: 10.1063/1.3669930
15. Srimani PK, Koti MS. Rough set approach for novel decision making in medical data for rule generation and cost sensitiveness. ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India. 2013. doi: 10.1007/978-3-319-03095-1_33, 303-311
16. Srimani PK, Koti MS. The dynamic allocation management of emergency cases in hospital information system using data clustering approach. Eur J Sci Res. 2013; 107(3):59–67.
17. Huang XM, Zhang YH. A new application of rough set to ECG recognition. International Conference on Machine Learning and Cybernetics, IEEE Explore. 2003; 3:1729–34.
18. Tsumoto S. Mining diagnostic rules from clinical databases using rough sets and medical diagnostic model. Int J Inform Sci. 2004; 162(2):65–80.
19. Lashin EF, Kozae AM, Abo Khadra AA, Medhat, T. Rough set theory for topological spaces. International Journal of Approximate Reasoning. 2005; 40(1–2):35–43.
20. Ohsaki M, Abe H, Tsumoto S, Yokoi H, Yamaguchi T. (2007) Evaluation of rule interestingness measures in medical knowledge discovery in databases. Artif Intell Med. 2007; 41(3):177–96.
21. Hu KY, Lu YC, Shi CY. Feature ranking in rough sets. Artificial Intelligence Communications. 2003; 16(1):41–50.
22. Ding S, Zhang Y, Xu L, Qian J. A feature selection algorithm based on tolerant granule. Journal of Convergence Information Technology. 2011; 6(1):191–95.
23. Lin TY. Granular computing on binary relations I: Data mining and neighborhood systems, II: rough set representations and belief functions. Rough Sets in Knowledge Discovery,(Eds.). Physica-Verlag, Heidelberg; 1998.
24. Lin TY. From rough sets and neighborhood systems to information granulation and computing in words. Proceedings of European Congress on Intelligent Techniques and Soft Computing. 1997; 1602–07.
25. Lin TY, Yao YY, Zadeh LA. Rough sets, granular computing and data mining. Physica-Verlag, Heidelberg; 2002.
26. Brill F, Brown D, Martin W. Fast Genetic Selection of Features for Neural Network Classifiers. IEEE Trans Neural Network. 1992; 3(2):324–28.
27. Chan CC, Grzymala-Busse JW. On the attribute redundancy and the learning programs ID3, PRISM, and LEM2. (Chapter 3); 1991.