Basics of Rough Set Theory and its Application in Decision Rule Generation

1 Introduction

Rough set theory was developed by Zdzislaw Pawlak in the early 1980's. It deals with the analysis of data tables. The data can be acquired from measurements or from human experts. The main goal of the rough set analysis is to synthesize approximation of concepts from the acquired data. The term *classification* concerns any context in which some decision is taken or a forecast is made on the basis of currently available knowledge or information. A *classification algorithm* is an algorithm which permits us to repeatedly make a forecast or to take a decision on the basis of accumulated knowledge in new situations. In this tutorial basic concepts of Rough Set Theory (RST) and how it can be applied to classify data patterns through decision rule generation are explained.

2 Rough Sets

2.1 Information System

A data set is represented as a table, where each row represents a case, an event 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 also be supplied by a human expert or user. This table is called an *information system*. Mathematically, an information system can be represented as,

$$T = (U, Q, V, f) \qquad (1)$$

Here, U is the finite set of objects and Q is the set of attributes. $V = \bigcup_{q \in Q} V_q$, where V_q is the domain of the values of q and f denotes decision function as, $f : U \times Q \rightarrow V$. For example, a decision table is given in Table 1. It is evident from the table that there are seven cases or *objects*, two *condition attributes* (Degree and Experience) and one *decision attribute* (Accept or Reject).

It can be easily observed that cases x3 and x4 as well as x5 and x7 are having exactly the same values of conditions, but the first pair (i.e. x3 and x4) has a different outcome (different value of the decision attribute) while the second pair (i.e. x5 and x6) has the same outcome.

One can derive a rule from the information table given in the above form as, "IF *Degree* is *MBA* and *Experience* is > *5 yrs* THEN *Accept"*. But it should be noted that among all such constructed rules, *minimality* and *consistency* of the rule set are important issues. These are illustrated in the subsequent sections.

Table 1. Example of a typical Decision Table of Hiring people

Objects	Degree	Experience	Decision
x1	MBA	> 5 yrs	Accept
x2	MBA	Nil	Reject
x3	BE	1 yr	Reject
x4	BE	1 yr	Accept
x5	ME	2-5 yrs	Reject
x6	MBA	2-5 yrs	Accept
x7	ME	2-5 yrs	Reject

2.2 Indiscernibility Relation

In RST, for different attributes, objects are called indiscernible, i.e. similar, if they are characterized by the same information. If $P \subseteq Q$ and $x_i, x_j \in U$, then x_i and x_j are indiscernible wrt the set of attributes *P*, if

$$f(x_i, q) = f(x_j, q), \forall q \in P \qquad (2)$$

An elementary set is the set of all indiscernible objects. So, for $P \subseteq Q$, an equivalence relation on *U,* called *P*-indiscernibility relation is given by,

$$I_P = \{(x_i, x_j) \in U^2 \mid \forall q \in P \; f(x_i, q) = f(x_j, q)\} \qquad (3)$$

Using Table 1 it can be illustrated that how a decision table defines an indiscernibility relation. Let, ***P***={***Degree***}. So, objects x1, x2 and x6 are indiscernible with respect to the attribute "*Degree*", because all of these three objects are having a value, "*MBA*", for attribute set ***P***. Similarly, objects x3 and x4 are also indiscernible with ***P***. In this case, both of these objects are having value "*BE*". So, the relation $\boldsymbol{I_P}$ defines three partitions of the universe, ***U*** (i.e. set of all cases),

$\boldsymbol{I_P}$ = { {x1, x2, x6} , {x3, x4} , {x5, x7} }.

It can easily be understood that, if ***P***= {***Experience***}, then,

$\boldsymbol{I_P}$ = { {x1},{x2}, { x3, x4 } , {x5, x6, x7} }.

Similarly, for ***P***= {***Degree***, ***Experience***},

$\boldsymbol{I_P}$ = { { x1 },{ x2 }, { x3, x4 } , { x5, x7 }, { x6 } }.

2.3 Set Approximation

As stated earlier that an equivalence relation induces a partitioning of the universe (the set of all cases in the example). These partitions can be used to build new subsets of the universe. The equivalence classes of the partition induced by the *P*-indiscernibility relation are called *information granules*.

It may happen, however, that a concept such as “Decision” cannot be determined or defined in a crisp manner. For instance, the set of objects with a “*Decision*”= “*Accept*” cannot be defined crisply using the attributes available in Table 1. The “problematic" objects are x3 and x4. In other words, it is not possible to induce a crisp (precise) description of these objects or cases from the table, because they are having same values for the *condition attribute* but different values for *decision attribute*. Here the notion of *Rough Set* emerges. Although these cases cannot be defined crisply, but it is possible to delineate the cases that certainly have a positive decision (i.e. *Accept*), or those certainly do not have a positive decision (i.e. *Reject*) and the objects that belong to a boundary between the certain cases. If this boundary is non-empty, the set is *rough*. These notions are formally expressed as follows.

For any rough set *Y*, $\underline{P}Y$ and $\overline{P}Y$ are called *P-lower* and *P-upper approximation* of *Y* and defined as,

$$\underline{P}Y = \{x \in Y \mid I_P(x) \subseteq Y\} \qquad (4)$$

and

$$\overline{P}Y = \{x \in Y \mid I_P(x) \cap Y \neq \phi\} \qquad (5)$$

respectively. The objects in $\underline{P}Y$ can be with certainty classified as members of *Y* on the basis of knowledge in *P*, while the objects in $\overline{P}Y$ can only be classified as possible members of *Y* on the basis of knowledge in *P*. The set, $BN_P(Y) = \overline{P}Y - \underline{P}Y$, is called the *P-boundary region* of *Y*, and thus consists of those objects that cannot be decisively classified into *Y* on the basis of knowledge or information available in *P*. The set, $U - \overline{P}Y$, is called the *P-outside region* of *Y* and consists of those objects which can be classified with certainty as not belonging to *Y* (on the basis of knowledge in *P*). A set is said to be *rough* or *crisp* if the boundary region is non-empty or empty respectively.
It is worth mentioning here that, the letter *P* refers to the subset *P* of the attributes *Q* (i.e. $P \subseteq Q$). If another subset were chosen, e.g. $A \subseteq Q$, the corresponding names of the relations would have been *A-boundary region*, *A-lower-* and *A-upper approximations*.

A pictorial representation of such concepts is given in Figure 1. More precisely, for the example considered in this tutorial, the approximations of the set of “*Accepted*” objects on the basis of the two conditional attributes (i.e. *Degree* and *Experience*) are shown in Figure 2.

Here, ***P*** = {***Degree***, ***Experience***} and *Y* is the set of “*Accepted*” objects or cases. It is evident from the Table 1 that, $\underline{P}Y$, i.e. set of objects which can certainly be classified as the members of the set of Y (i.e. “*Accepted*” objects) is { x1, x6 }. Similarly, $\overline{P}Y$, i.e. set of objects which can only be classified as possible members of *Y* on the basis of knowledge in *P* is { {x1}, {x6}, {x3, x4} }. $BN_P(Y) = \overline{P}Y - \underline{P}Y$ = {{x3, x4}}. The *P-outside region* of *Y,* i.e. $U - \overline{P}Y$ = { { x2 } , {x5, x7} }.

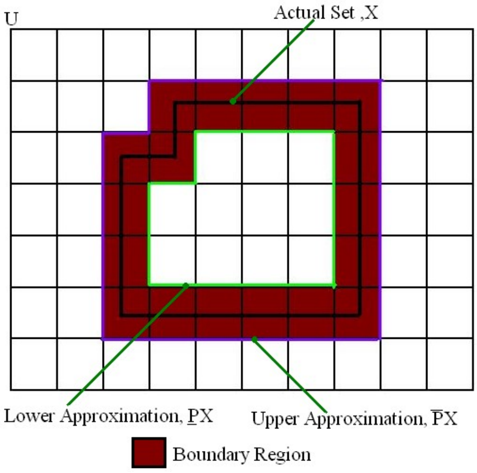

U
Actual Set ,X
Lower Approximation, $\underline{P}X$
Upper Approximation, $\overline{P}X$
🟥 Boundary Region

Figure 1. Schematic of set approximation concepts.

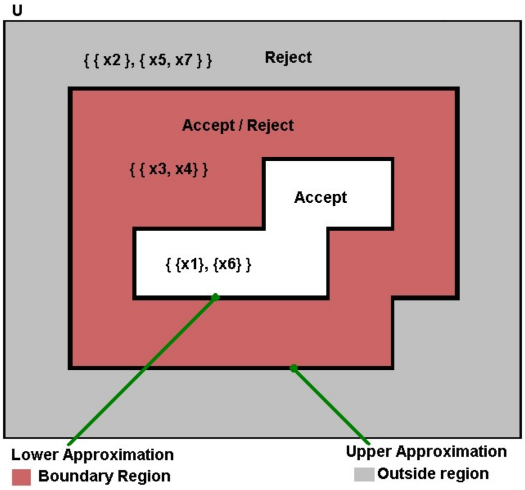


Figure 2. Set approximations of the decision table given in Table 1.

2.4 Computation of *Reduct* and *Core*

In the previous section it is observed that one way of reducing data table is to identify equivalence classes, i.e. objects that are indiscernible using the available attributes. The data table is reduced since only one element of the equivalence class is needed to represent the entire class. The other dimension in reduction is to keep only those attributes that preserve the indiscernibility relation and, consequently, set approximation. The rejected attributes are redundant since their removal does not worsen the classification. Thus *minimal* sufficient subsets of attributes which keep all the information intact and remove the superfluous attributes are called *Reduct* or $RED(P)$, where, $P \subseteq Q$. The CORE is the set of relations occurring in every *Reduct*, i.e. $CORE(P) = \bigcap RED(P)$. One of the unique aspects of the RST approach is the attribute reduction of knowledge with *Reducts* and *Core*. From the *Core* and *Reducts* one can generate the decision rules. Usually these rules are considered in "IF…THEN" formats.

To explain these aspects another decision table is considered, shown as Table 2. Here, the decision table shows fault diagnosis of distribution feeder. For a given subset $P \subseteq Q$, an attribute $q \in P$ is dispensable in P if and only if, $I_P = I_{(P-\{q\})}$; otherwise q is indispensable. If every element in P is indispensable then P is called *independent* otherwise *dependent*. Let $P \subseteq Q$ and $D \subseteq Q$ have equivalence relations in U. The *P-positive* region of D is indicated as,

$$POS_P(D) = \bigcup_{Y \in I_D} \underline{P}Y \qquad (6)$$

In other words, it denotes the set of elements that can correctly be classified into D-elementary sets obtained from I_D using the knowledge described by I_P. If $q \in P$ and

$$POS_P(D) = POS_{(P-\{q\})}(D) \qquad (7)$$

then, q is D-dispensable in P, otherwise q is D-indispensable in P. If the set of attributes G ($G \subseteq P$) is a D-independent in P and

$$POS_G(D) = POS_P(D), \qquad (8)$$

then, G is called D-reduct of P or in general *Reduct* of P.

For example, as shown in Table 2, let, P= {"weather", "cause of fault", "type of fault"}, D={"faulty equipment"}, and thus $I_{P=}${1, 4}, {2}, {3}, and {5}. Similarly, I_D={1, 2} and {3, 4, 5}, the $POS_P(D)$ = {2, 3, 5}. Removing the attribute "weather", $POS_{(P-\{weather\})}(D)$={2,5} ≠ $POS_P(D)$. So, the attribute "weather" is D-indispensable in P. Removing the attribute "cause of fault", $POS_{(P-\{cause\ of\ fault\})}(D)$={2,3,5}=$POS_P(D)$. So, the attribute "cause of fault" is D-dispensable in P. Similarly, the attribute "type of fault" is D-indispensable in P. Thus, the set {"weather", "type of fault"} is the D-reduct of P. So, the simplified form of Table 2 is shown in Table 3. '-' indicates "don't care" (i.e. dispensable) condition.

Table 2. Decision table for fault diagnosis of distribution feeder

Objects	**Condition Attributes**			**Decision Attribute**
	Weather	**Cause of Fault**	**Type of Fault**	**Faulty Equipment**
1	0	1	1	0
2	1	0	0	0
3	1	1	1	1
4	0	1	1	1
5	1	0	1	1
	"0" = Rain "1" = Cloudy	"0" =Collided by outside object "1" = Natural degradation	"0"= Breakdown type of outage "1"= Burn-out type of outage caused by faulty current	"0"=Cable outage "1"= Outage of feeder fuse switch

Table 3. Reduced form of Table 2

Objects	**Condition Attributes**			**Decision Attribute**
	Weather	**Cause of Fault**	**Type of Fault**	**Faulty Equipment**
1	0	-	1	0
2	1	-	0	0
3	1	-	1	1
4	0	-	1	1
5	1	-	1	1

Furthermore, a *reduct* can be transformed into a decision rule in which partial information of the indispensable condition attributes is used to derive specific knowledge of the output. In particular, objects 3 and 5 belong to the same elementary set with respect to the attributes "weather" and "type of fault" and their decision attribute values are the same (i.e. the value is 1). Thus, objects 3 and 5 can be precisely classified as an "outage of the feeder fuse switch" using the attributes "weather" and "type of fault". In other words, it can be said that, attribute values, (Weather=1 $\wedge$ Type of Fault=1) are the characteristic for decision class value =1 (i.e. "outage of the feeder fuse switch"). '$\wedge$' and '$\vee$' are logical "AND" and "OR" operators respectively. This is called a *reduct*. The value of the attribute "cause of fault", is not included in the *reduct* (i.e. it is dispensable). Intersections of these *reduct* values for each of the decision class (i.e. Cause of fault =1 or 0) will give the *Core* for the respective class. Here *Core* and *Reduct* values are same, because the condition attribute values for case 3 and 4 are same. Case 4 is not considered here, due to the reason explained later.

Then, the derived rules can be transformed into "IF *condition is satisfied* THEN *outcome is this*" formats. For example, the above *Reduct* and *Core* can be transformed into a decision rule:

IF
the value of attribute "weather"= 1 (i.e. cloudy)
AND
the value of attribute "type of fault"= 1 (i.e. burn-out type of outage caused by fault current)
THEN
the value of decision attribute "faulty equipment" =1 (i.e. outage of feeder fuse switch)

However, objects 1 and 4 belong to the same elementary set with respect to the attributes "weather" and "type of fault", yet their decision values (i.e. faulty equipment) are not the same. Thus, given the attributes "weather" and "type of fault", objects 1 and 4 cannot be properly classified as cable outage or outage of the feeder fuse switch. Objects 1 and 4 cannot be further classified without additional information.

To evaluate the relative goodness of the decision rules, two parameters are used – ***strength*** and ***coverage.*** *Strength* of a rule may be described as the fraction of total cases satisfying the rule considering all decision classes and *coverage* is the fraction of total cases satisfying the rule considering each decision class. Higher the value for *strength* and *coverage* higher will be the goodness or weightage of the rule.

3 Discretization of Decision Table

If a decision table is having a large number of attribute values i.e. *card(V_a)* is very high for some $a \in Q$, then there is a very low chance that a new object will be properly classified by matching its attribute value vector with the rows of the table. Here, *card()* means *cardinality* operator, which means "*number of elements of a set*". Therefore, discretization of the decision table is required for large real-valued decision table to achieve higher quality of classification. Discretization of a data table indicates some partitioning of the attribute values.

Let us assume a decision table in the form,

$$T = (U, A \cup \{d\}) \tag{9}$$

Here, ***U*** **= { x1, x2, x3,…, xn }**, i.e. set of all objects; ***A*** is the set of condition attributes and ***{ d }*** the set of decision attribute. It is assumed that $V_a = [l_a, r_a) \subset \Re$ for any $a \in A$. $\Re$ is the set of all real numbers. Let R_a a partition on V_a for $a \in A$ into subintervals as,

$$R_a = \{[c_0^a, c_1^a), [c_1^a, c_2^a), \ldots, [c_{k_a}^a, c_{k_a+1}^a)\} \tag{10}$$

for some integer κ, where,

$$l_a = c_0^a < c_1^a < c_2^a < \ldots < c_{k_a}^a < c_{k_a+1}^a = r_a \tag{11}$$

and

$$V_a = [c_0^a, c_1^a) \cup [c_1^a, c_2^a) \cup \ldots \cup [c_{k_a}^a, c_{k_a+1}^a) \tag{12}$$

Any R_a is uniquely defined by the set $C_a = \{c_1^a, c_2^a, c_3^a, \ldots, c_{k_a}^a\}$ called *set of cuts* on V_a **.** The *set of cuts* is empty if *card(R_a)=1.* Then any global family of cuts ***R*** can be defined as,

$$\boldsymbol{R} = \bigcup_{a \in A} \{a\} \times C_a \tag{13}$$

Any pair $(a, c) \in \boldsymbol{R}$ is called a cut on V_a **.** To illustrate this, Figure 3 shows discretization of a real line with a set of cuts. In essence, after discretization of a decision table with real valued attributes, a modified decision table is obtained whose attribute values are discrete numbers (integers) such as, 0, 1, 2, 3, … etc. depending on the *set of cuts* used for the discretization.

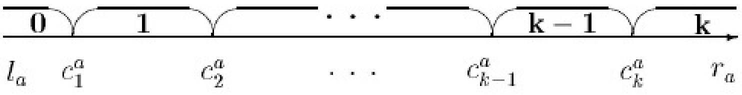
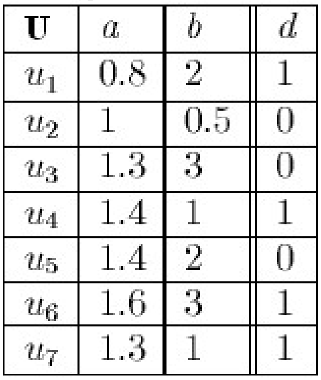


Figure 3 The partition on real line using the set of cuts $C_a = \{c_1^a, c_2^a, c_3^a, ..., c_{k_a}^a\}$.

Selecting the optimal set of cuts is a difficult task. In the present case of study Maximal Discernible (MD) heuristic is followed which is discussed in details in different papers. In the following paragraphs this MD heuristics is explained step by step with the help of an example shown in decision table given in Table 4. Here, a and b are condition attributes and d is decision attribute.

Table 4. A typical real-valued decision table



U	a	b	d
u_1	0.8	2	1
u_2	1	0.5	0
u_3	1.3	3	0
u_4	1.4	1	1
u_5	1.4	2	0
u_6	1.6	3	1
u_7	1.3	1	1

Step 1.

The observed values of an attribute a is sorted such that $v_1^a < v_2^a < v_3^a < ... < v_{k_a}^a$, where, $\{v_1^a, v_2^a, v_3^a, ..., v_{k_a}^a\} = \{a(x) : x \in U\}$, i.e. values of the attribute a. Then the *set of cuts* for attribute a is defined as,

$$C_a = \{\frac{v_1^a + v_2^a}{2}, \frac{v_2^a + v_3^a}{2}, ..., \frac{v_{n_a - 1}^a + v_{n_a}^a}{2}\} \qquad (14)$$

For example, in Table 4, the set of values of a and b on objects from U are given by,

$$a(U) = \{0.8, 1, 1.3, 1.4, 1.6\} ;$$

$$b(U) = \{0.5, 1, 2, 3\} ,$$

So, the set of cuts for attribute a is,

$$C_a = \{\frac{0.8+1}{2}, \frac{1+1.3}{2}, \frac{1.3+1.4}{2}, \frac{1.4+1.6}{2}\}$$

$$\text{or, } C_a = \{0.9, 1.15, 1.35, 1.5\}$$

Now, if any value of attribute a is $v_i^a < 0.9$ then in the discretized table it will be replaced by '0'. Similarly, if $0.9 \leq v_i^a < 1.15$ then in the discretized table it will be replaced by '1' and so on. In a similar way cuts for attribute b can also be chosen as,

$$C_b = \{0.75, 1.5, 2.5\}$$

A discretized form of Table 4 is shown in Table 5. Here, *set of cuts* is chosen as, $R = \{(a,0.9),(a,1.5),(b,0.75),(b,1.5)\}$. This cut is chosen for demonstration only and is not the optimal set of cuts. Pictorial representation of the cut is shown in Figure 4.

Table 5. Discretized form of Table 4 using set of cuts $R = \{(a,0.9),(a,1.5),(b,0.75),(b,1.5)\}$.

U	a	b	d
u_1	0	2	1
u_2	1	0	0
u_3	1	2	0
u_4	1	1	1
u_5	1	2	0
u_6	2	2	1
u_7	1	1	1

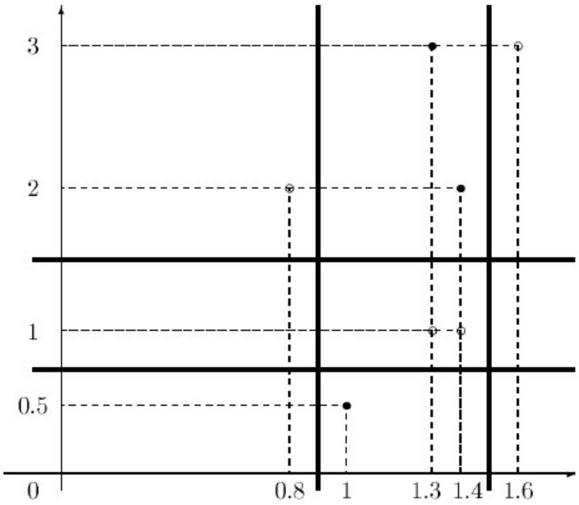


Figure 4. Graphical representation of the cuts on Table 4 using $R = \{(a,0.9),(a,1.5),(b,0.75),(b,1.5)\}$.

Step 2.

To obtain the optimal set of cuts a new table $T^* = (U^*, A^*)$ is derived from the Table 4 using C_a and C_b such that,

$$U^* = \{(u_i, u_j) \in U \times U : (i < j) \wedge (d(u_i) \neq d(u_j))\} \quad (15)$$

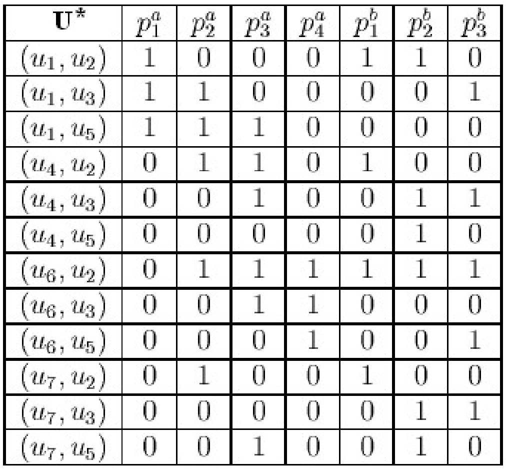
and

$$A^* = \{p_s^a : a \in A \text{ and } s \text{ corresponding to the } s^{th} \text{ interval} [v_s^a, v_{s+1}^a) \text{ for } a \quad (16)$$

The table is shown in Table 6. Here objects are all pairs of objects from A with different decision values, and all object pairs are now to be discerned using the values of the attributes modified after applying the cuts C_a and C_b . The set of condition attributes in the new decision system is equal to the set of all attributes defined by all cuts. These attributes are binary. The value of the new attribute corresponding to a cut (a,c) on the pair (u_i, u_j) is equal to 1 if this cut is discerning objects (u_i, u_j) and 0 otherwise.

For example, u_1 and u_2 are discernible using $C_a(1) = 0.9$, because using this cut on attribute a the values $a(u_1) = 0$ and $a(u_2) = 1$ as shown in Table 5. So, $p_1^a = 1$ for object pair (u_1, u_2).

Table 6. New table T^* obtained from decision table T given in Table 4 using MD heuristics



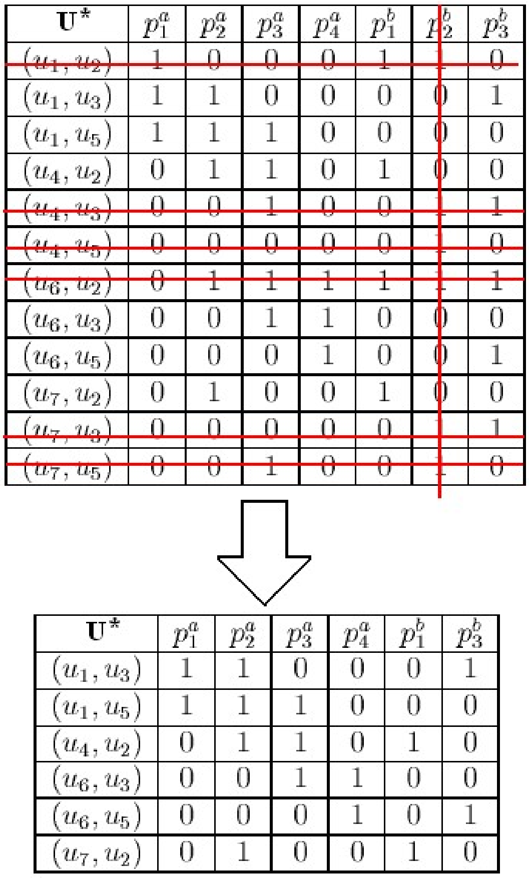
U*	p_1^a	p_2^a	p_3^a	p_4^a	p_1^b	p_2^b	p_3^b
(u_1, u_2)	1	0	0	0	1	1	0
(u_1, u_3)	1	1	0	0	0	0	1
(u_1, u_5)	1	1	1	0	0	0	0
(u_4, u_2)	0	1	1	0	1	0	0
(u_4, u_3)	0	0	1	0	0	1	1
(u_4, u_5)	0	0	0	0	0	1	0
(u_6, u_2)	0	1	1	1	1	1	1
(u_6, u_3)	0	0	1	1	0	0	0
(u_6, u_5)	0	0	0	1	0	0	1
(u_7, u_2)	0	1	0	0	1	0	0
(u_7, u_3)	0	0	0	0	0	1	1
(u_7, u_5)	0	0	1	0	0	1	0

Step 3.

Now a column with maximum number of occurrences of 1's is to be chosen from table T^* . This column is to be deleted from the table and also the rows that contain 1's in this column. This procedure is continued on the new modified table until table T^* becomes empty.

In this example p_2^b column is chosen first because, it contains six 1's. The table obtained after deleting the column and rows having value 1, is given in Figure 5.

Then column p_2^a and finally p_4^a is chosen because the table becomes empty thereafter. So, the resultant set of cuts is $R = \{(a,1.15),(a,1.5),(b,1.5)\}$ **.** The set of cuts is marked in Figure 6 with bold line. The final of the dicretized form of Table 4 is Table 7 using MD heuristics.



$\mathbf{U}^*$	p_1^a	p_2^a	p_3^a	p_4^a	p_1^b	p_2^b	p_3^b
~~(u_1,u_2)~~	~~1~~	~~0~~	~~0~~	~~0~~	~~1~~	~~1~~	~~0~~
(u_1,u_3)	1	1	0	0	0	0	1
(u_1,u_5)	1	1	1	0	0	0	0
(u_4,u_2)	0	1	1	0	1	0	0
~~(u_4,u_3)~~	~~0~~	~~0~~	~~1~~	~~0~~	~~0~~	~~1~~	~~1~~
~~(u_4,u_5)~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~1~~	~~0~~
~~(u_6,u_2)~~	~~0~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~
(u_6,u_3)	0	0	1	1	0	0	0
(u_6,u_5)	0	0	0	1	0	0	1
(u_7,u_2)	0	1	0	0	1	0	0
~~(u_7,u_3)~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~1~~	~~1~~
~~(u_7,u_5)~~	~~0~~	~~0~~	~~1~~	~~0~~	~~0~~	~~1~~	~~0~~

$\mathbf{U}^*$	p_1^a	p_2^a	p_3^a	p_4^a	p_1^b	p_3^b
(u_1,u_3)	1	1	0	0	0	1
(u_1,u_5)	1	1	1	0	0	0
(u_4,u_2)	0	1	1	0	1	0
(u_6,u_3)	0	0	1	1	0	0
(u_6,u_5)	0	0	0	1	0	1
(u_7,u_2)	0	1	0	0	1	0

Figure 5. Reduction of column and rows of table T^* according to MD heuristics and reduced form of T^* .

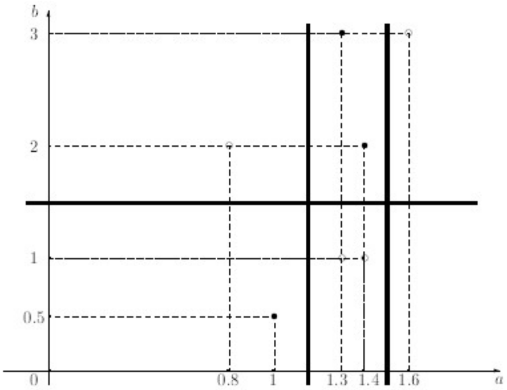
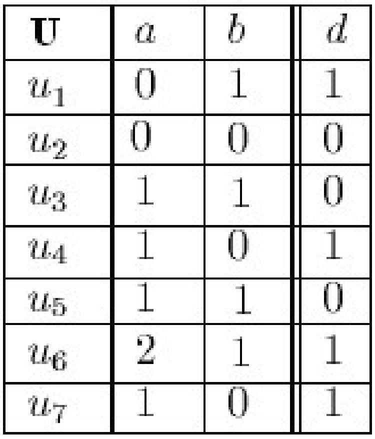


Figure 6. Graphical representation of optimal set of cuts obtained from MD heuristics.

Table 7. Final Discretized form of Table 4 using MD heuristics



U	a	b	d
u_1	0	1	1
u_2	0	0	0
u_3	1	1	0
u_4	1	0	1
u_5	1	1	0
u_6	2	1	1
u_7	1	0	1

4 Conclusions

This tutorial gives an elaborate view of Rough Set theory (RST). The discussion emphasizes upon how RST can be applied for classification of data by simplifying a decision table. In this context RST based rule generation has been explained. It shows that RST is an effective methodology for classification of data patterns where the information system contains imprecise, superfluous and inconsistent data. It is known that prior to classification of data, some features are extracted from the data so that a classification algorithm can classify the data patterns efficiently. Selection of proper features for efficient classification is not an easy task. For example, in many problems significant features may be extracted from the cross-wavelet spectrum, but the difficulty of choosing appropriate features is also a problem there. Improper selection of feature vector may make the data table inconsistent or it may contain superfluous data requiring unnecessarily higher time of processing. In this context RST based classification is very effective. That is why Rough Set Theory (RST) has successfully been used for condition monitoring of distribution feeder, for fraud detection in electrical energy consumers, in data-mining for semiconductor manufacturing and also in case generation.