I.J. Information Technology and Computer Science, **2012, 11, 58-66**
Published Online October 2012 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2012.11.08

Prediction of Missing Values for Decision Attribute

T. Medhat
Computer and Automatic Control Department, Faculty of Engineering, Kafrelsheikh University, 33516, Kafrelsheikh, Egypt
E-mail: tmedhatm@eng.kfs.edu.eg, tmedhatm@yahoo.com

Abstract**—** The process of determining missing values in information system is an important issue for decision making especially when the missing values are in the decision attribute. The main goal for this paper is to introduce algorithm for finding missing values of decision attribute. Our approach is depending on distance function between existing values. These values can be calculated by distance function between the conditions attributes values for the complete information system and incomplete information system. This method can deal with the repeated small distance by eliminating a condition attribute which has the smallest effect on the complete information system. This algorithm will be discussed in detail with an example of a case study.

Index Terms**—** Rough Sets, Degree of Dependency, Distance Function, Missing Values

I. Introduction

Classical rough set theory developed by Professor Z. Pawlak in 1982 has made a great success in knowledge acquisition in recent years [1,12]. In Rough set theory, knowledge is represented in information systems. An information system is a data set represented in a table, decision table [2]. Each row in the table represents an object, for example a case or an event. Each column in the table represents an attribute, for instance a variable, an observation or a property. To each object (row) some attribute values are assigned. One of the disadvantages of rough set theory is its dependence on complete information systems i.e. For a decision table to be processed, it must be complete and its all objects values must be known [3]. But in real-life applications, due to measurement errors, miscomprehension, access limitation and disoperation in register, etc, information systems with missing values often occur in knowledge acquisition. Information systems with missing data, or, in different words, the corresponding decision tables are incompletely specified, are called incomplete information systems [4]. For simplicity, incompletely specified decision tables will be called incomplete decision tables.

Often, intelligent techniques such as neural networks, decision trees, fuzzy theory, etc. [5] are based on quite strong assumptions (e.g. knowledge about dependencies, probability distributions, large number of experiments). They cannot derive conclusions from incomplete knowledge, or manage inconsistent information.

Rough set theory [6,12,13] can deal with uncertainty and incompleteness in data analysis. It deems knowledge as a kind of discriminability. The attribute reduction algorithm removes redundant information or features and selects a feature subset that has the same discernibility as the original set of features. From the medical point of view, this aims at identifying subsets of the most important attributes influencing the treatment of patients. Rough set rule induction algorithms generate decision rules [10], 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. Rough sets have been a useful tool for medical applications. Hassanien [2] applies rough set theory to breast cancer data analysis. Tsumoto [15] proposed a rough set algorithm to generate diagnostic rules based on the hierarchical structure of differential medical diagnosis. The induced rules can correctly represent experts' decision processes. Komorowski and Ohrn [6] use a rough set approach for identifying a patient group in need of a scintigraphic scan for subsequent modeling. Bazan [1] compares rough set-based methods, in particular dynamic reducts, with statistical methods, neural networks, decision trees and decision rules. He analyzes medical data, i.e. lymphography, breast cancer and primary tumors, and finds that error rates for rough sets are fully comparable as well as often significantly lower than that for other techniques. In Ref. [3,14], a rough set classification algorithm exhibits higher classification accuracy than decision tree algorithms. The generated rules are more understandable than those produced by decision tree methods.

The core of the proposed approach is how to predict the value of the decision attribute by using the distance function and degree of dependency.

The approach we used depends on determining the decision attribute values for missing values of decision attributes. By using a distance function between complete decision table and incomplete decision table, we can predict the decision of missing values.

In this paper, we apply rough sets to predict the decision of missing values. A rough set feature selection algorithm is used to select feature subsets that are more efficient (we say the feature subset is 'more efficient' because, by the rough set approach, redundant features are discarded and the selected features can describe the decisions as well as the original whole feature set, leading to better prediction accuracy. The selected features are those that influence the decision concepts, so will be helpful for cause-effect analysis). The chosen subsets are then employed within a decision rule generation process, creating descriptive rules for the classification task. The rough set rule-based method can achieve higher classification accuracy than other intelligent analysis methods.

The present paper is organized as follows. In section 2, the main concepts of rough sets are introduced. The proposed new method for giving a decision for missing values is demonstrated in Section 3. The algorithm and classification method are described in Section 4 by an example. Section 5 is conclusion.

II. Basic Concepts

Let $I = (U, A \cup \{d\})$ be an information system, where U is the universe with a non-empty set of finite objects. A is a nonempty finite set of condition attributes, and d is the decision attribute (such a table is also called decision table). $\forall a \in A$ there is a corresponding function $f_a: U \rightarrow V_a$, where V_a is the set of values of a. If $P \subseteq A$, there is an associated equivalence relation [7,10,11]:

$$IND(P) = \{(x, y) \in U \times U \mid \forall a \in P, f_a(x) = f_a(y)\} \quad (1)$$

The partition of U, generated by IND(P) is denoted U/P. If $(x,y) \in$ IND(P), then x and y are indiscernible by attributes from P.

The equivalence classes of the P-indiscernibility relation are denoted $[x]_P$. Let $X \subseteq U$, the P-lower approximation $\underline{P}X$ and P-upper approximation $\overline{P}X$ of set X can be defined as :

$$\underline{P}X = \{x \in U \mid [x]_P \subseteq X\} \quad (2)$$

$$\overline{P}X = \{x \in U \mid [x]_P \cap X \neq \phi\} \quad (3)$$

Let $P, Q \subseteq A$ be equivalence relations over U, then the positive, negative and boundary regions can be defined as:

$$POS_P(Q) = \bigcup_{X \in U/Q} \underline{P}X \quad (4)$$

$$NEG_P(Q) = U - \bigcup_{X \in U/Q} \overline{P}X \quad (5)$$

$$BND_P(Q) = \bigcup_{X \in U/Q} \overline{P}X - \bigcup_{X \in U/Q} \underline{P}X \quad (6)$$

The positive region of the partition U/D with respect to C, $POS_c(D)$, is the set of all objects of U that can be certainly classified to blocks of the partition U/D by means of C. D depends on C in a degree k ($0 \le k \le 1$) denoted $C \Rightarrow_k D$

$$k = \gamma_c(D) = \frac{|POS_c(D)|}{|U|} \quad (7)$$

If k=1, D depends totally on C, if 0<k<1, D depends partially on C, and if k=0 then D does not depend on C. When C is a set of condition attributes and D is the decision, $\gamma_C(D)$ is the quality of classification [4,7,13].

III. The New Method

We will introduce a method (depending on the distance function) to detect the decision for missing values. This will be done by calculating the distance function between complete decision table and incomplete decision table. If the small distance is repeated with more than one object, then we must eliminate one of the condition attribute which has a small effect on the information system, by using the quality of classification, and then calculating the distance function again. The decision for missing values equals the decision of object which the smallest distance will be found between them.

3.1 Distance Function

The distance between the complete decision table and incomplete decision table can be calculated by the following function:

$$dis(O_{incomp.\_i}, O_{comp.\_i}) = \sqrt{\sum_{i=1}^{N} [c_i(O_{incomp.}) - c_i(O_{comp.})]^2} \quad (8)$$

$$\forall O_{comp.\_i}, O_{incomp.\_i} \in U,$$
$$O_{incomp.\_i} \textit{ is an incomplete decision object},$$
$$O_{comp.\_i} \textit{ is a complete decision object},$$
$$c_k \in C,\ N = \|C\|;\ \textit{number of condition attributes}$$

where c($O_{comp.\_i}$) is the value of condition attribute c with respect to the object $O_{comp.\_i}$.

3.2 New method

Calculate the distance function.

Put all of distance functions in a new array A(m).

Compute the smallest number and its order of the array A(m).

The smallest number means that the decision for missing values of the incomplete decision table equals the decision value of the objects which its order is the order of the smallest number.

We can make this as follows:

- If incomplete object $O_{incomp.\_i}$ has smallest distance with only one complete object $O_{comp.\_i}$, then the decision of the incomplete object $O_{incomp.\_i}$ equals the decision of the complete object $O_{comp.\_i}$.
- If the incomplete object $O_{incomp.\_i}$ has the smallest distance with more than one complete objects $O_{comp.\_i}$,…. and $O_{comp.\_k}$, where the decision of these complete objects equals, then the decision of incomplete object $O_{incomp.\_i}$ equals the same decision of these complete objects.
- If the incomplete object $O_{incomp.\_i}$ has the smallest distance with more than one complete objects $O_{comp.\_i}$,…. and $O_{comp.\_k}$, but its decision is different, then we can't give the decision of the incomplete object $O_{incomp.\_i}$. To give the decision of incomplete object $O_{incomp.\_i}$, we need to eliminate the attribute which has small effects on the information table by using the quality of classification $\gamma_C(D)$. After deleting the attribute which have small effects, we determine the distance between the incomplete object $O_{incomp.\_i}$ and complete objects $O_{comp.\_i}$,…. and $O_{comp.\_k}$, not all completed objects. According to the new small distance, we can put the decision of missing values as mentioned in the previous steps.

IV. The New Algorithm

We need to make an algorithm that can be used for detecting the value of missing values in the incomplete decision table, according to the given complete decision table.

The algorithm depends on 6 main steps:

a) Reading complete decision table and incomplete decision table.

b) Calculating the distance between incomplete objects and complete objects.

c) Arranging the values of the small distance.

d) Detecting repeated small distance.

e) Putting the decision of missing values.

f) Determining the decision of incomplete object which has decision value x(non).

These steps will be shown in more details as shown below:

a) Read complete decision table and incomplete decision table:

- Read complete objects condition data "condition attributes values of complete objects", put it in an array a(i,j)
- Read complete objects decision data " decision attribute values of complete objects", put it in an array oldd(i)
- Read incomplete objects condition data "condition attributes values of incomplete objects", put it in an array nd(I,j)
- Let m = the number of complete objects, n = the number of condition attributes
- Let mm = the number of incomplete objects.

b) Calculate the distance between incomplete objects and complete objects.

- Put the value of distance function in array dis(I,j)
- Put the value of incomplete object number , complete object number, small distance and decision of complete object in array new_dis(I,j)

```
For i = 1 To mm
  For j = 1 To m
      d = 0
      For k = 1 To n
         d = d + (nd(i,k) - a(j,k))^2
     Next k
     dis(i,j) = d^0.5
     new_dis(j + i * m - m, 1) = i, new_dis(j + i * m -
m, 2) = j
```

```
    new_dis(j + i * m - m, 3) = dis(i, j),   new_dis(j + i * m - m, 4) = oldd(j)
  Next j
Next i
```

c) **Arrange the values of array new_dis(i,j) as ascending order:**

```
For k = 0 To mm - 1
    For i = 1 To m - 1
        For j = 1 + i To m
            If (new_dis(i + k * m, 3) > new_dis(j + k * m, 3)) Then
            x = new_dis(i + k * m, 3), new_dis(i + k * m, 3) = new_dis(j + k * m, 3)
            new_dis(j + k * m, 3) = x,
            x = new_dis(i + k * m, 1), new_dis(i + k * m, 1) = new_dis(j + k * m, 1)
            new_dis(j + k * m, 1) = x
            x = new_dis(i + k * m, 2), new_dis(i + k * m, 2) = new_dis(j + k * m, 2)
            new_dis(j + k * m, 2) = x
            x = new_dis(i + k * m, 4), new_dis(i + k * m, 4) = new_dis(j + k * m, 4)
            new_dis(j + k * m, 4) = x
            End If
        Next j
    Next i
Next k
```

d) **Putting the values of array new_dis(i,j) into array new_decision(i, j) when finding repeated small distance:**

```
Let k = 0,  c = 0
 For i = 1 To mm * m
   k = k + 1
   If (new_dis(i, 3) ≠ new_dis(i + 1, 3)) Then
         new_decision(k, 1) = new_dis(i, 1), new_decision(k,2) = new_dis(i,2)
         new_decision(k, 3) = new_dis(i, 3), new_decision(k,4) = new_dis(i,4)
   i = i + m - c - 1
   c = 0
   Else
         new_decision(k, 1) = new_dis(i, 1), new_decision(k,2) = new_dis(i,2)
         new_decision(k, 3) = new_dis(i, 3), new_decision(k,4) = new_dis(i,4)
   c = c + 1
   End If
 Next i
```

e) **Putting the decision of missing values:**

- Put data in final decision file as final.txt

```
For i = 1 To k
 If (new_decision(i,1) = new_decision(i + 1, 1)) Then
   If (new_decision(i,4) = new_decision(i + 1, 4)) Then
    put in final.txt, new_decision(i, 1); new_decision(i, 2); new_decision(i,3); new_decision(i,4)
     i = i + 1
   Else
    put in final.txt, new_decision(i, 1); new_decision(i, 2); new_decision(i,3); "x"
    put in final.txt, new_decision(i + 1, 1); new_decision(i + 1, 2); new_decision(i + 1,3); "x"
    i = i + 1
  End If
Else
 put in final.txt, new_decision(i, 1); new_decision(i, 2); new_decision(i,3); new_decision(i,4)
End If
Next i
```

f) **Determine the decision of incomplete object which has decision value x(non).**

Eliminate the attribute which has a small effect on the system, and try the algorithm again. This will be done by calculating the quality of classification of data for each condition attribute as in equation (7):

Example 1

The optician's decisions data set concerns an optician's decisions as to whether or not a patient is suited to contact lens use. The set of all possible decisions is listed in Table 1.

Experimental Results

By converting the data in table 1 as follows,

Converting condition attributes as follows:

a- Age

Young ⇒10
Pre-presbyopic ⇒20
Presbyopic ⇒30

b- Spectacle

Myope ⇒10
Hypermetrope ⇒20

c- Astigmatic

No ⇒10
Yes ⇒20

d- Tear production rate

Normal ⇒10
Reduced ⇒20

And converting the decision attribute as follows:

D- Optician's decisions

hard contact lenses ⇒10,

soft contact lenses ⇒20,

no contact lenses ⇒30

Table 1: The optician's decisions data set

U/A	Condition attributes				Decision attribute (Optician's decision)
	Age	Spectacle	Astigmatic	Tear production rate	
P1	Young	Hypermetrope	No	Reduced	?
P2	Young	Hypermetrope	No	Normal	soft contact lenses
P3	Pre-presbyopic	Hypermetrope	No	Reduced	no contact lenses
P4	Pre-presbyopic	Hypermetrope	No	Normal	soft contact lenses
P5	Presbyopic	Hypermetrope	No	Reduced	no contact lenses
P6	Presbyopic	Hypermetrope	No	Normal	soft contact lenses
P7	Young	Hypermetrope	Yes	Reduced	?
P8	Young	Hypermetrope	Yes	Normal	hard contact lenses
P9	Pre-presbyopic	Hypermetrope	Yes	Reduced	no contact lenses
P10	Pre-presbyopic	Hypermetrope	Yes	Normal	no contact lenses
P11	Presbyopic	Hypermetrope	Yes	Reduced	no contact lenses
P12	Presbyopic	Hypermetrope	Yes	Normal	no contact lenses
P13	Young	Myope	No	Reduced	?
P14	Young	Myope	No	Normal	?
P15	Pre-presbyopic	Myope	No	Reduced	no contact lenses
P16	Pre-presbyopic	Myope	No	Normal	soft contact lenses
P17	Presbyopic	Myope	No	Reduced	no contact lenses
P18	Presbyopic	Myope	No	Normal	no contact lenses
P19	Young	Myope	Yes	Reduced	?
P20	Young	Myope	Yes	Normal	?
P21	Pre-presbyopic	Myope	Yes	Reduced	no contact lenses
P22	Pre-presbyopic	Myope	Yes	Normal	hard contact lenses
P23	Presbyopic	Myope	Yes	Reduced	no contact lenses
P24	Presbyopic	Myope	Yes	Normal	hard contact lenses

Then, we get the following table (Table 2) which can be converted into two; Table 3 of complete information table and Table 4 of incomplete information table as follows:

Table 2: The optician's decisions data set after converting attribute values into numbers

U/A	C				D
	a	B	c	d	
p1	10	20	10	20	?
p2	10	20	10	10	20
p3	20	20	10	20	30
p4	20	20	10	10	20
p5	30	20	10	20	30
p6	30	20	10	10	20
p7	10	20	20	20	?
p8	10	20	20	10	10
p9	20	20	20	20	30
p10	20	20	20	10	30
p11	30	20	20	20	30
p12	30	20	20	10	30
p13	10	10	10	20	?
p14	10	10	10	10	?
p15	20	10	10	20	30
p16	20	10	10	10	20
p17	30	10	10	20	30
p18	30	10	10	10	30
p19	10	10	20	20	?
p20	10	10	20	10	?
p21	20	10	20	20	30
p22	20	10	20	10	10
p23	30	10	20	20	30
p24	30	10	20	10	10

Table 3: Complete decision system

U/A	C				D
	a	**b**	**c**	**D**	
p2	10	20	10	10	20
p3	20	20	10	20	30
p4	20	20	10	10	20
p5	30	20	10	20	30
p6	30	20	10	10	20
p8	10	20	20	10	10
p9	20	20	20	20	30
p10	20	20	20	10	30
p11	30	20	20	20	30
p12	30	20	20	10	30
p15	20	10	10	20	30
p16	20	10	10	10	20
p17	30	10	10	20	30
p18	30	10	10	10	30
p21	20	10	20	20	30
p22	20	10	20	10	10
p23	30	10	20	20	30
p24	30	10	20	10	10

Table 4: Incomplete decision system

U/A	C				D
	a	b	c	D	
p1	10	20	10	20	?
p7	10	20	20	20	?
p13	10	10	10	20	?
p14	10	10	10	10	?
p19	10	10	20	20	?
p20	10	10	20	10	?

By calculating the distance function according to the new method and algorithm; we get the results in Table 5

In Table 5; we see that:

Object P13 has the smallest distance with only one object P15, so the decision of object P13 has the decision of object P15 (which be 30).

Also, the decision of object P19 has the decision of object P21 (which be 30)

But the object P14 has the smallest distance with two objects P2 and P16, where the decision of P2 and P16 are equal (which be 20), so the decision of object P14 is also equal 20.

In Addition, the decision of P20 equal 10.

But the object P1 has the smallest distance with objects P2 and P3, where its decision is different (20 and 30).

So , we can't give the decision of the object P1 or P7.

Table 5: Decision Table of Missing Values of Some Objects

Objects with no decision	**Objects with decision**	**Small distance**	**Old decision**	**New decision**
P1	P2	1	20	?
	P3	1	30	?
P7	P8	1	10	?
	P9	1	30	?
P13	P15	1	30	30
P14	P2	1	20	20
	P16	1	20	
P19	P21	1	30	30
P20	P8	1	10	10
	P22	1	10	

To give the decision of objects P1 and P7, we need to eliminate the attribute which has small effects on the information table according to the degree of dependency, as shown below:

U/IND(D)={
{P8,P22,P24},
{P2,P4,P6,P16},
{P3,P5,P9,P10,P11,P12,P15,P17,P18,P21,P23}
}

U/IND({a})={
{P2, P8},
{P3, P4, P9, P10, P15, P16, P21, P22},
{P5, P6, P11, P12, P17, P18, P23, P24}
}

U/IND({b})={
{P15, P16, P17, P18, P21, P22, P23, P24},
{P2, P3, P4, P5, P6, P8, P9, P10, P11, P12}
}

U/IND({c})={
{P2, P3, P4, P5, P6, P15, P16, P17, P18},
{P8, P9, P10, P11, P12, P21, P22, P23, P24}
}

U/IND({d})={
{P3, P5, P9, P11, P15, P17, P21, P23},
{P2, P4, P6, P8, P10, P12, P16, P18, P22, P24}
}

U/IND(C)={
{P2},{P3},{P4},{P5},{P6},{P8},{P9},
{P10},{P11},{P12},{P15},{P16},{P17},
{P18},{P21},{P22},{P23},{P24}
}

$POS_C(D)$ ={P2,P3,P4,P4,P5,P6,P8,P9,P10,P11,P12,P15,P16,P17,P18,P21,P22,P23,P24}

$$k=\gamma_C(D)=\frac{|POS_C(D)|}{|U|}=1$$

U/IND(C-{a})={
{P15, P17}, {P16, P18}, {P21, P23},
{P22, P24}, {P3, P5}, {P2, P4, P6},
{P9, P11}, {P8, P10, P12}
}

U/IND(C-{b})={
{P2}, {P6}, {P3, P15}, {P4, P16},
{P9, P21}, {P10, P22}, {P5, P17},
{P6, P18}, {P11, P23}, {P12, P24}
}

U/IND(C-{c})={
{P2, P8}, {P15, P21}, {P16, P22},
{P3, P9}, {P4, P10}, {P17, P23},
{P18, P24}, {P5, P11}, {P6, P12}
}

U/IND(C-{d})={
{P2}, {P6}, {P15, P16}, {P21, P22},
{P3, P4}, {P9, P10}, {P17, P18},
{P23, P24}, {P5, P6}, {P11, P12}
}

$$k=\gamma_{C-\{a\}}(D)=\frac{|POS_{C-\{a\}}(D)|}{|U|}=\frac{13}{18}=0.722$$

$$k=\gamma_{C-\{b\}}(D)=\frac{|POS_{C-\{b\}}(D)|}{|U|}=\frac{12}{18}=0.666$$

$$k=\gamma_{C-\{c\}}(D)=\frac{|POS_{C-\{c\}}(D)|}{|U|}=\frac{8}{18}=0.444$$

$$k=\gamma_{C-\{d\}}(D)=\frac{|POS_{C-\{d\}}(D)|}{|U|}=\frac{8}{18}=0.444$$

We see that: we can eliminate attribute a, which has small effects.

Note:

After deleting the attribute which has small effects, we determine the distance between the objects which has no decision and objects which have decision for only the objects which have the same small distance.

See table 5:

Table 6: Decision of Two Objects P1 and P7 after Elimination of Attribute which has a Small Effect

Objects with no decision	Objects with decision	small distance	Old decision	New decision
P1	P2	1	20	30
	P3	0	30	
P7	P8	1	10	30
	P9	0	30	

The following table gives the decision of all objects:

Table 7: Complete Decision Table for All Objects

U/A	a	b	c	d	D
p1	10	20	10	20	30
p2	10	20	10	10	20
p3	20	20	10	20	30
p4	20	20	10	10	20
p5	30	20	10	20	30
p6	30	20	10	10	20
p7	10	20	20	20	30
p8	10	20	20	10	10
p9	20	20	20	20	30
p10	20	20	20	10	30
p11	30	20	20	20	30
p12	30	20	20	10	30
p13	10	10	10	20	30
p14	10	10	10	10	20
p15	20	10	10	20	30
p16	20	10	10	10	20
p17	30	10	10	20	30
p18	30	10	10	10	30
p19	10	10	20	20	30
p20	10	10	20	10	10
p21	20	10	20	20	30
p22	20	10	20	10	10
p23	30	10	20	20	30
p24	30	10	20	10	10

Table 8: The Optician's Decisions Data Set after Converting Attribute values into another Numerical Coding

U/A	C				D
	a	b	c	d	
p1	150	100	150	150	?
p2	150	100	150	100	100
p3	100	100	150	150	50
p4	100	100	150	100	100
p5	50	100	150	150	50
p6	50	100	150	100	100
p7	150	100	100	150	?
p8	150	100	100	100	150
p9	100	100	100	150	50
p10	100	100	100	100	50
p11	50	100	100	150	50
p12	50	100	100	100	50
p13	150	150	150	150	?
p14	150	150	150	100	?
p15	100	150	150	150	50
p16	100	150	150	100	100
p17	50	150	150	150	50
p18	50	150	150	100	50
p19	150	150	100	150	?
p20	150	150	100	100	?
p21	100	150	100	150	50
p22	100	150	100	100	150
p23	50	150	100	150	50
p24	50	150	100	100	150

If the condition attribute values are symbols, then we must convert them into integers according to the order of symbols. If there are three values as high, medium and low, then we can convert them into 3, 2 and 1 respectively.

Remark (1):

By converting the information table "Table 1" into another numerical coding as in Table 8, we get the same result of prediction. This mean that our method is independent to numerical assumptions "coding". i.e. if you make many numerical coding to the information system table, then you will get the same prediction result.

V. Conclusion

By calculating the distance function between complete decision table and incomplete decision table, we can put a decision for missing values according to the algorithm which is explained in section 4. When a small distance is repeated with more than one object, we make an elimination of a condition attribute which has a small effect on the information system, and then we calculate the distance function again, and apply the algorithm.

Acknowledgement

The author would like to thank Prof. Dr. A. M. Kozae, for his encouragement and support, and sincerely thank the anonymous reviewers whose comments have greatly helped clarify and improve this paper.

References

[1] Bazan J., "A Comparison of dynamic and nondynamic rough set methods for extracting laws from decision tables", Rough Sets in Knowledge Discovery, Physica Verlag, 1998.

[2] Hala S. Own, Aboul Ella Hassanien, "Rough Wavelet Hybrid Image Classification Scheme", JCIT, Vol. 3, No. 4, pp. 65 ~ 75, 2008.

[3] Hu K.Y., Lu Y.C., Shi C.Y., "Feature ranking in rough sets", AI Commun. 16 (1), 41~50, 2003.

[4] Hu, X., Cercone N., Han, J., Ziarko, W, "GRS: A Generalized Rough Sets Model", in Data Mining, Data Mining, Rough Sets and Granular Computing, T.Y. Lin, Y.Y.Yao and L. Zadeh (eds), Physica-Verlag, 447~ 460, 2002.

[5] Jin-Cherng Lin and Kuo-Chiang Wu, "Using Rough Set and Fuzzy Method to Discover the Effects of Acid Rain on the Plant Growth", JCIT, Vol. 2, No. 1, pp. pp ~ 48, 2007.

[6] Komorowski J., Ohrn A., "Modelling prognostic power of cardiac tests using rough sets", Artif. Intell. Med. 15, 167~191, 1999.

[7] Lashin E.F, Kozae A.M., Abo Khadra A.A., and Medhat T., "Rough set theory for topological spaces", International Journal of Approximate Reasoning, Vol. 40, No. 1-2, 35~43, 2005.

[8] Li G.Z., Yang J., Ye C.Z., Geng D.Y., "Degree prediction of malignancy in brain glioma using support vector machines", Comput. Biol. Med. Vol. 36, No. 3, 313~325, 2006.

[9] Lin T.Y., "Granular computing on binary relations I: data mining and neighborhood systems, II: rough set representations and belief functions", In: Rough Sets in Knowledge Discovery , Lin T.Y., Polkowski L., Skowron A., (Eds.). Physica-Verlag, Heidelberg ,107~140, 1998.

[10] Lin T.Y., Yao Y.Y., Zadeh L.A., (Eds.) " Rough Sets, Granular Computing and Data Mining", Physica-Verlag, Heidelberg, 2002.

[11] Medhat T., "Missing Values Via Covering Rough Sets", IJMIA: International Journal on Data Mining and Intelligent Information Technology Applications, Vol. 2, No. 1, pp. 10 ~ 17, 2012.

[12] Pawlak Z., "Rough set approach to multi-attribute decision analysis", European Journal of Operational Research, Vol. 72, No. 3, 443~459, 1994.

[13] Pawlak Z., "Rough Sets - Theoretical Aspects of Reasoning about data.", Kluwer Academic Publishers, Dordrecht, Boston, London, 1991.

[14] Shifei Ding, Yu Zhang, Li Xu, Jun Qian, "A Feature Selection Algorithm Based on Tolerant Granule", JCIT, Vol. 6, No. 1, pp. 191 ~ 195, 2011.

[15] Tsumoto S., "Mining diagnostic rules from clinical databases using rough sets and medical diagnostic model", Inform. Sci. Vol. 162, 65~80, 2004.

Author's Profile

Tamer Medhat, born in Kafrelsheikh, Egypt, in July 13th, 1974, and received his Ph.D degree in Faculty of Engineering, Tanta University, Tanta, Egypt in 2007. He is currently an assistant professor in the Faculty of Engineering, Kafrelsheikh University, Egypt. His current research interests include Information Systems, Augmented Reality, Decision Making, Computer Science, and Rough Set Theory Applications.

How to cite this paper: T. Medhat,"Prediction of Missing Values for Decision Attribute", International Journal of Information Technology and Computer Science(IJITCS), vol.4, no.11, pp.58-66, 2012. DOI: 10.5815/ijitcs.2012.11.08

