International Journal of Soft Computing 1(2) : 111-117, 2006

A Novel Reduct Algorithm for Dimensionality Reduction with Missing Values Based on Rough Set Theory

[1]K. Thangavel, [2]A. Pethalakshmi and [3]P. Jaganathan
[1]Department of Computer Science, Periyar University, Salem, Tamil Nadu, India
[2]Department of Computer Science, M.V.M Government Arts College (W),
Dindigul-624 001, Tamil Nadu, India
[3]Department of Computer Science and Applications, Gandhigram Rural Institute-Deemed University,
Gandhigram-624 302, Tamil Nadu, India

Abstract: Database with missing values is a common phenomenon in data mining, statistical analysis, as well as in machine learning. Missing values in the database will affect the classification accuracy and effectiveness of classification rules. In this study, we have used four different methods such as Indiscernibility, Mean, Median and Mode for dealing with missing attribute values and proposed a Revised Quickreduct algorithm for dimensionality reduction. A comparative study is also performed with Revised and original Quickreduct algorithms based on the four different methods. The public domain datasets available in UCI machine learning repository with missing attribute values are used.

Key words: Data mining, rough set theory, reduct, missing attribute values, indiscernibility relation

INTRODUCTION

Rough Sets Theory proposed by Pawlak[1,2] in 1980's, creates a framework for handling the imprecise and incomplete data in information systems. A rough set is a mathematical tool to deal with uncertainty and vagueness of an information system. An information system can be presented as a Table with rows analogous to objects and columns analogous to attributes. Each row of the Table contains values of particular attributes representing information about an object.

Using the Rough Sets approach, one can deal with two major problems in the analysis of an information system: (i) Reducing unnecessary objects and attributes so as to get the minimum subset of attributes, ensuring a good approximation of classes and an acceptable quality of classification. (ii) representing the information system as a decision Table which shows dependencies between the minimum subset of attributes (called conditions) and particular class numbers (called decisions), without redundancy.

In real-life data, some attribute values are frequently missing, imperfect and incomplete. There are two main reasons for attribute values to be missing: either they are 'lost 'or they 'do not care' conditions. That is, the original values were not recorded at all since they were irrelevantand the decision to which concept a case belongs was taken without that information.

Missing data is a common problem in statistical analysis. In particular, missing values in a data set can affect the performance of a classifier constructed using such a data set as a training sample. Rates of less than 1% missing data are generally considered trivial, 1-5% manageable. However, 5-15% must be handled by complicated methodsand more than 15% may severely impact any kind of interpretation[3]. Conventional methods usually cannot deal directly with real-world data, because of missing or wrong values. The majority of interesting data bases are incomplete, i.e., one or more values are missing inside or some records are missing at all.

In this study, firstly we have made an attempt to fill-up the missing values in the decision Table using four different methods to reconstruct the Table. Then the revised Quick Reduct algorithm is implemented and its performance is compared with original Quick Reduct algorithm.

The problem of mining with missing values is known in data mining and a lot of work has already been reported in the literature. However, there is no one satisfactory solution to the problems.

Grzymala-Busse *et al.*,[4] proposed and compared nine different approaches to calculate missing attribute

Corresponding Author: P. Jaganathan, Department of Computer Applications, PSNA College of Engineering and Technology, Dindigul, Tamil Nadu, India

values. They have alternatively described[5] that the attribute-value pair blocks were used to construct characteristic relations, lower and upper approximations for decision Tables with missing attribute values. Yet another study[6], they have defined an incompletely specified decision Tables as a lattice. They introduced three definitions of lower and upper approximations and induced rules from incompletely specified decision Tables.

Latkowski *et al.*,[7] have presented a new approach to handling incomplete information and classifier complexity reduction. They described a method called D^{3}RJ, that performs data decomposition and decision rule joining to avoid the necessity of reasoning with missing attribute values.

Jiye *et al.*,[8] have described uncertainty measures of roughness of knowledge and rough sets by introducing rough entropy in incomplete information systems. Slowinski *et al.*,[9] have investigated the case of incomplete information systemsand presented a generalization of the rough sets approach which deals with missing and imprecise descriptors.

Hong *et al.*,[10] have dealt with the problem of producing a set of certain and possible rules from incomplete data sets based on rough sets. Leung *et al.*,[11] have described the knowledge acquisition in incomplete information systems using rough set theory. They proposed similarity classes in incomplete information systems and two kinds of partitions namely lower and upper approximations for the mining of certain and association rules in incomplete decision Tables.

MATERIALS AND METHODS

Indiscernibility relation: Consider a Car data Table of 8 data points with four condition attributes and a decision attribute. This is tabulated in Table 1. This Table contains some of the values which are missing and is denoted by '*'. For indiscernibility calculation, assume this missing attribute value is the same as the previous one. In Table 1, apply the Revised Quickreduct algorithm (Section 4). Iinitially, $\Upsilon_{best} = 0$, $\Upsilon_{prev} = 0$, R ← {} and T ← R.

$$\text{Ind(W)} = \{1, 2, 6, 8\}, \{3\}, \{4, 5, 7\}.$$
$$\text{Ind(M)} = \{1, 3, 6\}, \{2, 4, 5, 7, 8\}.$$
$$\text{Pos(W)/(M)} = \text{Ind(W)} \subseteq \text{Ind(M)}$$
$$= \{3, 4, 5, 7\}$$

Therefore, Υ(W)/(M) = 4/8. Similarly, other columns are calculated as follows:

Table 1: Car data set

Object	Weight	Door	Size	Cylinder	Mileage
1	1	2	1	4	3
2	1	4	2	6	1
3	2	4	*	4	3
4	3	2	1	6	1
5	3	4	1	4	1
6	*	4	*	4	3
7	3	4	2	*	1
8	1	*	2	6	1

Υ(D)/(M) = 0/8, Υ(S)/(M) = 3/8, Υ(C)/(M) = 3/8. The maximum degree of dependency value is taken. In this case, 'WEIGHT' is the highest dependency. Therefore R ← {WEIGHT} and T ← R, Υ_{best} = 4/8, Until $\Upsilon_{best} = \Upsilon_{prev}$, this condition proves false, therefore, go to step 'e'. Take the next combination and find out the degree of dependency as follows:

$$\text{Ind(W, D)} = \{1, 8\}, \{2, 6\}, \{3\}, \{4\}, \{5, 7\}$$
$$\text{Pos(W, D)} = \{3, 4, 5, 7\}$$

Υ(W, D)/(M) = 4/8 , Similarly, other columns are calculated as follows:

Υ(W, S)/(M) = 8/8, Υ(W, C)/(M) = 8/8, Take the maximum degree of dependency value. If this degree of dependency has the same values then take any one value. This process is repeatedand the final reduct set for the above Table is {WEIGHT, SIZE}. Similarly apply the Quickreduct algorithm (Section 3), the reduct set is again {WEIGHT, SIZE}

Mean imputation: In this, we propose two different methods for calculating the mean value.

Mean imputation 1: In this method, find out the mean value of each column, that is, mean = $\in X_i/n$. Using Table 1, the mean of 'WEIGHT' column is calculated. In this column, the 6th row value is missing, so mean value of 'WEIGHT' is equal to the summation of all row value except 6th row divided by the total number of rows except the 6th row. Therefore, Mean (WEIGHT) = 14/7 = 2. This mean value is substituted in the missing row position. Similarly the other columns are calculated and the decision Table is again reconstructed and is shown in Table 2.

The reduced attribute set of Quickreduct algorithm is {WEIGHT, DOOR} and Revised Quickreduct algorithm is {WEIGHT, DOOR}.

Mean imputation 2: In this method, missing value is calculated as follows:

Table 2: Mean imputation1

Object	Weight	Door	Size	Cylinder	Mileage
1	1	2	1	4	3
2	1	4	2	6	1
3	2	4	2	4	3
4	3	2	1	6	1
5	3	4	1	4	1
6	2	4	2	4	3
7	3	4	2	5	1
8	1	3	2	6	1

Table 3: Mean imputation 2

Object	Weight	Door	Size	Cylinder	Mileage
1	1	2	1	4	3
2	1	4	2	6	1
3	2	4	2	4	3
4	3	2	1	6	1
5	3	4	1	4	1
6	3	4	2	4	3
7	3	4	2	5	1
8	1	2	2	6	1

Table 4: Median imputation

Object	Weight	Door	Size	Cylinder	Mileage
1	1	2	1	4	3
2	1	4	2	6	1
3	2	4	2	4	3
4	3	2	1	6	1
5	3	4	1	4	1
6	2	4	2	4	3
7	3	4	2	4	1
8	1	4	2	6	1

- Find out the missing row denoted by 'n'.
- Mean = ((n – 1) +(n + 1))/2 , where (n – 1) is the previous row value and (n + 1) is the next row value.

Consider Table 1. The 6th row value is missing corresponding to 'WEIGHT' column. Then, mean (WEIGHT) = (5th row value + 7th row value)/2. Therefore, mean (WEIGHT) = 3. Similarly, in the 'DOOR' attribute column, 8th row value is missing. Therefore mean (DOOR) = (4 + 0)/2 = 2, because in this column, there is no 9th row value. Therefore assume that the 9th row value = 0. Thus we can calculate the other missing row values. Then the decision Table is again reconstructed and is tabulated as in Table 3.

Then the Quickreduct and Revised Quickreduct algorithms are applied to find out the reduct set. The reduced attribute set of Quickreduct algorithm is {WEIGHT, DOOR, SIZE, CYLINDER} and Revised Quickreduct algorithm is {WEIGHT, SIZE, CYLINDER}.

Median imputation: In this method, missing value is calculated as follows:

- Except the missing value each column is arranged in an ascending order.
- If the number of rows is an odd number, then median = (n+1)/2.
- If the number of rows is an even number, then median = (mid+(mid+1))/2.

Table 5: Weight attribute column

Value	No. of times occurred
1	3
2	1
3	3

Table 6: Mode imputation

Object	Weight	Door	Size	Cylinder	Mileage
1	1	2	1	4	3
2	1	4	2	6	1
3	2	4	2	4	3
4	3	2	1	6	1
5	3	4	1	4	1
6	2	4	2	4	3
7	3	4	2	4	1
8	1	4	2	6	1

In Table 1, find the ascending order of 'WEIGHT' column. The values are { 1, 1, 1, 2, 3, 3, 3}. It is an odd number, then median (WEIGHT) = (7+1)/2 = 4. Therefore median value of 4th row = 2. This value is substituted as the missing row value. Similarly find the ascending order of 'SIZE' column. The values are {1, 1, 1, 2, 2, 2}. It is an even number. So, median (SIZE) = (3rd value + 4th row value)/2, i.e. median(SIZE) = (1+2)/2 = 2. This value is substituted as in the 3rd row. Similarly, we can calculate the other missing row values and the Table 1 is reconstructed and is presented as in Table 4.

Then the Quickreduct and Revised Quickreduct algorithms are applied to find out the reduct set. The reduced attribute set of Quickreduct algorithm is {WEIGHT, DOOR} and Revised Quickreduct algorithm is {WEIGHT, DOOR}.

Mode imputation: In this method, missing value is calculated as follows:

- For each attribute, find out the number of occurrences of each value except the missing value.
- Find out which value has occurred maximum number of times. This value is Substituted in the place of missing row.
- If more than one value occurs same number of times, find out the average of corresponding values.

By considering Table 1, the mode of 'WEIGHT' attribute column is calculated (Table 5) as follows:

In this column, value 1 and 3 occur same number of times, hence the average (WEIGHT) = (1+3)/2 = 2. This value is substituted as the missing row value. Similarly calculate the other missing row values. Then the decision Table is again reconstructed and is tabulated as in Table 6.

Then the Quickreduct and Revised Quickreduct algorithms are applied to find out the reduct set. The reduced attribute set of Quickreduct algorithm is {WEIGHT, DOOR} and Revised Quickreduct algorithm is {WEIGHT, DOOR}.

QUICKREDUCT ALGORITHM

The reduction of attributes is achieved by comparing equivalence relations generated by sets of attributes. Attributes are removed so that the reduced set provides the same predictive capability of the decision feature as the original. A reduct is defined as a subset of minimal cardinality R_{min} of the conditional attribute set C such that

$$\Upsilon_R(D) = \Upsilon_C(D).$$
$$R = \{X : X \subseteq C; \Upsilon_X(D) = \Upsilon_C(D)\}$$
$$R_{min} = \{X : X \in R; \forall Y \in R; |X| \leq |Y| \}$$

The intersection of all the sets in R_{min} is called the core, the elements of which are those attributes that cannot be eliminated without introducing more contradictions to the dataset. In this method a subset with minimum cardinality is searched for.

The problem of finding a reduct of an information system has been the subject of much research in[12,13]. The most basic solution to locate such a subset is to simply generate all possible subsets and retrieve those with a maximum rough set dependency degree. Obviously, this is an expensive solution to the problem and is only practical for very simple datasets. Most of the time only one reduct is required as, typically, only one subset of features is used to reduce a dataset, so all the calculations involved in discovering the rest are pointless. Jensen *et al.*,[14-16] have developed the Quickreduct algorithm to compute a minimal reduct without exhaustively generating all possible subsets and also they developed Fuzzy-Rough attribute reduction with application to web categorization. K. Thangavel *et al.*,[17,18] applied Rough Sets for feature selection in Medical databases like Mammograms, HIV etc.

To improve the performance of the above method, an element of pruning can be introduced. By noting the cardinality of any pre-discovered reducts, the current possible subset can be ignored if it contains more elements. However, a better approach is needed - one that will avoid wasted computational effort. The pseudo code of the Quickreduct is given below:

```
QUICKREDUCT(C,D)
C, the set of all conditional features;
D, the set of decision features.
(a)        $R \leftarrow \{\}$
(b)        Do
(c)        $T \leftarrow R$
(d)        $\forall x \in (C-R)$
(e)        if $\Upsilon_{R\cup\{x\}}(D) > \Upsilon_T(D)$
```

where $\Upsilon_R(D) = card(POS_R(D))/card(U)$

```
(f)        $T \leftarrow R\cup\{x\}$
(g)        $R \leftarrow T$
(h)        until $\Upsilon_R(D) = = \Upsilon_C(D)$
(I)        return R
```

REVISED QUICKREDUCT ALGORITHM

The Quickreduct algorithm attempts to calculate a reduct without exhaustively generating all possible subsets. It starts off with an empty set and adds in turn, one at a time, those attributes that result in the greatest increase in the rough set dependency metric, until this produces its maximum possible value for the dataset. According to the Quickreduct algorithm, the dependency of each attribute is calculatedand the best candidate chosen. This, however, is not guaranteed to find a minimal subset as has been shown in[19]. Using the dependency function to discriminate between candidates may lead the search down a non-minimal path. It is impossible to predict which combinations of attributes will lead to an optimal reduct based on changes in dependency with the addition or deletion of single attributes. It does result in a close-to-minimal subset, though, which is still useful in greatly reducing dataset dimensionality. In[19], a potential solution to this problem has been proposed whereby the Quickreduct algorithm is altered, making it into an n-lookahead approach. However, even this cannot guarantee a reduct unless n is equal to the original number of attributes, but this reverts back to generate-and-test. It still suffers from the same problem as the original Quickreduct, i.e. it is impossible to mention at any stage whether the current path will be the shortest to a reduct.

The Quickreduct algorithm is improved herein and the pseudo code of the Revised Quickreduct Algorithm is given below:

```
Revised Quickreduct (C,D)
C, the set of all conditional features;
D, the set of decision features.
(a)  $R \leftarrow \{\}$
(b)  $\Upsilon_{best} = 0, \Upsilon_{prev} = 0$
```

```
(c)  Do
(d)  $T \leftarrow R$
(e) $\Upsilon_{prev} = \Upsilon_{best}$
(f)  $\forall x \in C$
(g)  if $\max(\Upsilon_{R \cup \{x\}}(D) > \Upsilon_{prev}$
```

Where $\Upsilon_R(D)=\text{card}(POS_R(D))/\text{card}(U)$

```
$POS_R(D) = \underline{R}\, x$
(h) $T \leftarrow R \cup \{x\}$
(I) $\Upsilon_{best} = \Upsilon_T(D)$
(j) $R \leftarrow T$
(k) until $\Upsilon_{best} = \Upsilon_{prev}$
(l)  return R
```

RESULTS AND DISCUSSION

The Quickreduct and Revised Quickreduct algorithm have been implemented using MATLAB for databases available in the UCI data repository[20]. In our experiments, no decision value is unknown. The Comparative Analysis of Quickreduct and Revised Quickreduct algorithm for four different methods is shown in Table 7, 8, 9, 10 and 11

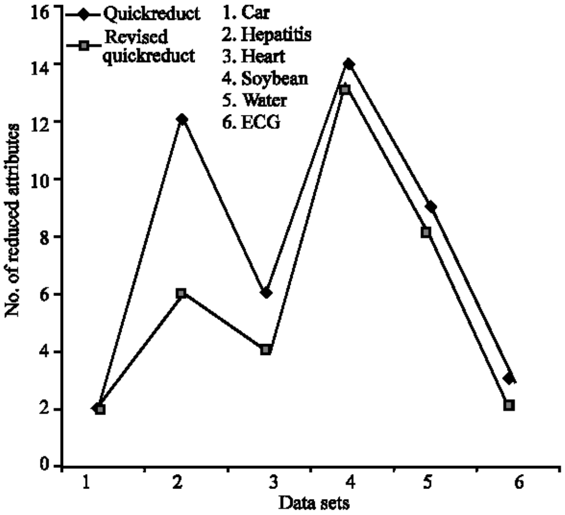

Fig. 1: Performance Analysis of the Quickreduct and the Revised Quickreduct(Indiscernibility)

From the above Tables, it is evident that Revised Quickreduct algorithm produces minimal reduct for large data sets with more number of attributes. The performance analysis of the Quickreduct and the revised Quickreduct

Table 7: Comparative analysis of indiscernibility relation method

Data set	Instances	No. of attributes	No. of missing values	Quickreduct	Revised quickreduct
Car	8	4	5	2	2
Hepatitis	155	19	167	12	6
Heart (Switzerland)	123	13	273	6	4
Soybean (Large)	307	35	705	14	13
Water-treatment-data	527	38	591	9	8
Echocardiogram	74	13	132	3	2

Table 8: Comparative analysis of mean imputation 1

Data set	Instances	No. of attributes	No. of missing values	Quickreduct	Revised quickreduct
Car	8	4	5	2	2
Hepatitis	155	19	167	12	5
Heart (Switzerland)	123	13	273	6	5
Soybean (Large)	307	35	705	12	11
Water-treatment-data	527	38	591	7	6
Echocardiogram	74	13	132	3	2

Table 9: Comparative analysis of mean imputation 2

Data set	Instances	No. of attributes	No. of missing values	Quickreduct	Revised quickreduct
Car	8	4	5	4	3
Hepatitis	155	19	167	12	6
Heart (Switzerland)	123	13	273	6	5
Soybean (Large)	307	35	705	15	12
Water-treatment-data	527	38	591	7	6
Echocardiogram	74	13	132	2	2

Table 10: Comparative analysis of median imputation

Data set	Instances	No. of attributes	No. of missing values	Quickreduct	Quickreduct revised
Car	8	4	5	2	2
Hepatitis	155	19	167	11	5
Heart (Switzerland)	123	13	273	6	5
Soybean (Large)	307	35	705	13	12
Water-treatment-data	527	38	591	7	6
Echocardiogram	74	13	132	3	2

Table 11: Comparative analysis of mode imputation

Data set	Instances	No. of attributes	No. of missing values	Quickreduct	Revised quickreduct
Car	8	4	5	2	2
Hepatitis	155	19	167	12	6
Heart (Switzerland)	123	13	273	6	5
Soybean (Large)	307	35	705	12	10
Water-treatment-data	527	38	591	8	7
Echocardiogram	74	13	132	2	2

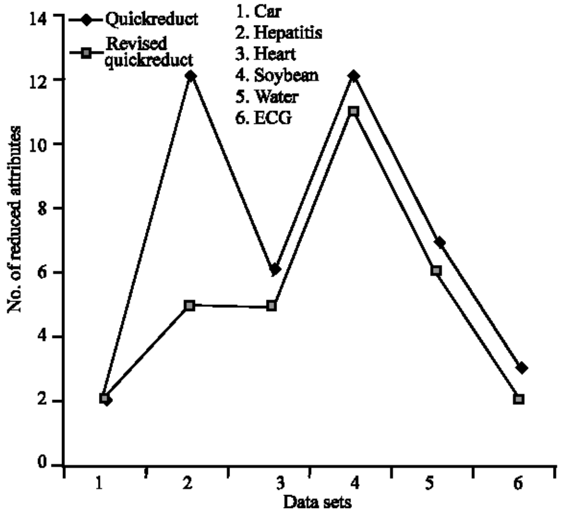

Fig. 2: Performance analysis of the Quickreduct and the Revised Quickreduct(Mean Imputation1)

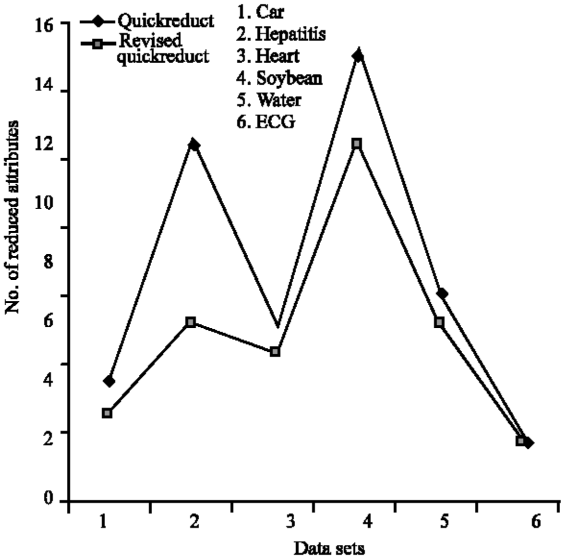

Fig. 3: Performance analysis of the Quickreduct and the Revised Quickreduct(Mean Imputation2)

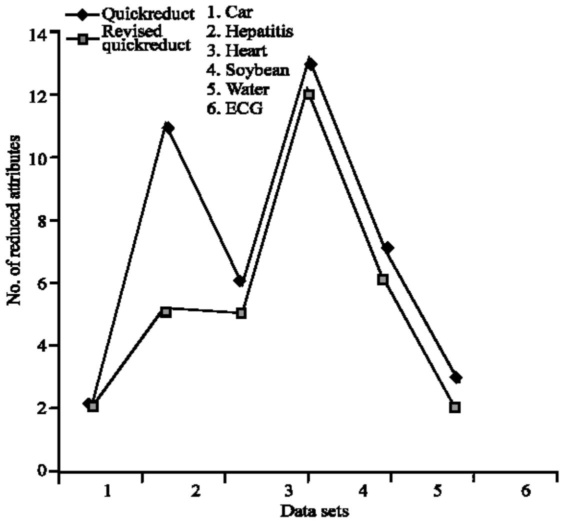

Fig. 4: Performance analysis of the Quickreduct and the Revised Quickreduct(Median)

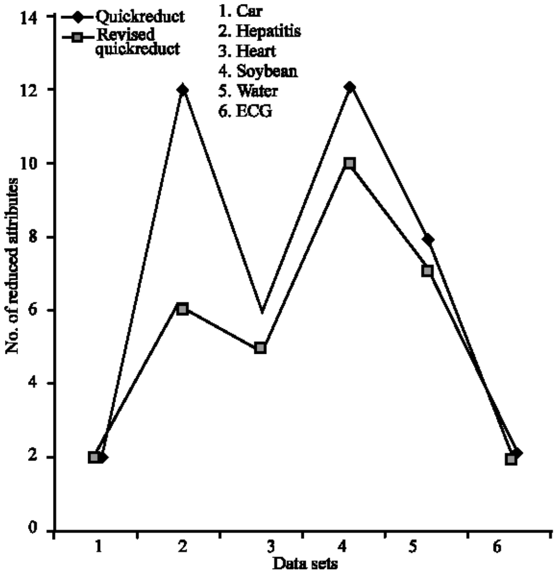

Fig. 5: Performance analysis of the Quickreduct and the Revised Quickreduct(Mode)

for four different methods is also depicted in Fig. 1, 2, 3, 4 and 5, respectively. It is observed from the graph that we could see that the minimal reduct is produced for the data sets using Mode imputation method. In the case of Car data set, the same number of reducts is obtained in both the algorithms except the Mean Imputation 2. Similarly in the case of Echocardiogram data set, same number of reducts is obtained in mean Imputation 2 and mode imputation, perhaps the reason may be the Car and the Echocardiogram information system consists of small data.

CONCLUSION

The performance of the Revised Quickreduct algorithm for dimensionality reduction with four different methods such as Indiscernibility, Mean, Median and Mode for dealing with missing attribute values is studied. A comparative study is also performed with Revised Quickreduct algorithm against the original Quickreduct algorithm. It is found that the Revised Quick Reduct algorithm outperforms the original Quick Reduct algorithm when there are large data sets with more number of attributes. Also we have focused on the effectiveness of different methods used for dealing with the missing attributes values.

REFERENCES

1. Pawlak, Z., 1982. Rough sets, Intl. J. Computer and Information Sci., pp: 341-356.
2. Pawlak, Z., 1991. Rough sets: Theoritical aspects and Reasoning about data, Kluwer Academic Publishers.
3. Pyle, D., 1999. Data Preparation for Data Mining, Morgan Kauffman Publishers.
4. Grzymala-Busse, J.W. and M. Hu, 2001. A Comparison of Several Approaches to Missing Attribute Values in Data Mining, W.Ziarko and Y. Yao (Eds.): RSCTS 2000, LNAI 2005, pp: 378-385.
5. Grzymala-Busse, J.W. and S. Siddhaye, 2004. Rough set approaches to rule induction from incomplete data, Proc. of the IPMU, pp: 923-930.
6. Grzymala-Busse, J.W., 2003. Rough set strategies to data with missing attribute values, Proc. of the Workshop on Foundations and New Directions in Data Mining, associated with the third IEEE International Conference on Data Mining, pp: 56-63.
7. Latkowski, R. and M. Mikolajczyk, 2004. Data Decomposition and Decision Rule Joining for Classification of Data with Missing Values, J.F. Peters *et al.*, (Eds.): Transactions on Rough Sets I, LNCS 3100, pp: 299-320.
8. Jiye, L. and X. Zongben, 2000. Uncertainty measures of roughness of knowledge and rough sets in incomplete information systems, Proc. of the 3rd World Congress on Intelligent Control and Automation, pp: 2526-2529.
9. Slowinski, R. and J. Stefanowski, 1989. Rough Classification in Incomplete information systems, Mathematical Computer Modelling, pp: 1347-1357.
10. Hong, T.P., L.H. Tseng and S.L. Wang, 2002. Learning rules from incomplete training examples by rough sets, Expert Systems with Applications, pp: 285-293.
11. Leung, Y., W.Z. Wu and W.X. Zhang, 2004. Knowledge acquisition in incomplete information systems: A rough set approach, European J. Operational Res., pp: 164-180.
12. Alpigini, J.J., J.F. Peters, J. Skowron and N. Zhong, 2002. Rough sets and current trends in computing, Third International Conference, USA.
13. Swiniarski, R.W. and A. Skowron, 2003. Rough Set methods in feature selection and recognition, Pattern Recognition Lett., 24: 833-849.
14. Jensen, R., 2005. Combining rough and fuzzy sets for feature selection, Ph.D Thesis, School of Informatics, University of Edinburgh.
15. Jensen, R. and Q. Shen, 2004. Fuzzy-rough attribute reduction with application to web categorization, Fuzzy Sets and Systems, pp: 469-485.
16. Jensen, R. and Q. Shen, 2004. Semantics-preserving dimensionality reduction: Rough and fuzzy-rough-based approaches, IEEE Transactions on Knowledge and Data Engineering.
17. Thangavel, K. and A. Pethalakshmi, 2005. Feature selection for medical database using rough system, Intl. J. Artificial Intelligence and Machine Learning.
18. Thangavel, K., M. Karnan and A. Pethalakshmi, 2005. Performance analysis of rough reduct algorithms in mammograms, Intl. J. Graphics Vision and Image Processing, pp: 13-21.
19. Chouchoulas, J., Halliwell and Q. Shen, 2002. On the implementation of rough set attribute reduction, Proceedings of the UK Workshop on Computational Intelligence, pp: 18-23.
20. Blake, C.L. and C.J. Merz, 1998. UCI Repository of machine learning databases. Irvine, University of California, http://www.ics.uci.edu/~mlearn/.