International Journal of Soft Computing 3 (4): 321-325, 2008
ISSN: 1816-9503

Reduct Based Decision Tree (RDT)

[1]Ramadevi Yellasiri, [2]C.R. Rao, [1]Hari RamaKrishna and [1]T. Prathima
[1]Department of CSE, Chaitanya Bharathi Institute of Technology, Hyderabad, India
[2]DCIS, School of MCIS, University of Hyderabd, Hyderabad, India

Abstract: New approaches to compute predominant attributes (referred as reduct) are discussed in this study. Rough Sets concepts and 'val' theory are adopted in this process. Procedure to construct a decision tree using these reduct is presented. These trees are referred as Reduct based Decision Tree (RDT). Decision rules for these RDTs are generated. 'Kappa statistics' was used to prove the efficiency of this model which is supported by K-fold test.

Key words: Rough sets, predominant attributes, composite reduct, RDT and GPCR

INTRODUCTION

Classification is the problem of forecasting the decisions of new cases, based on their traditional features, by using a model learned from the already known instances. Decision Tree (DT), Rough Sets (RS), Neural Networks, SVM, Statistics, etc., are some of the popular techniques to learn such models. RS theory provided by Pawlak (1981) is an important mathematical tool for computer technology. It is involved in several decision-makings, data mining, knowledge representation, knowledge acquisition and many more applications (Radwan and Tazaki, 2004). A basic problem for many practical applications of RS is defining a method for efficient selection of the set of attributes (features) necessary for the classification of objects in the considered universe. These knowledge reduction problems are highly involved in Information Systems (IS). In general, any IS consists of several attributes. In the process, it is tedious to recall each attribute every time. So, it is necessary to avoid the redundant attributes as well as to pick up the minimal feature. This minimal feature is called a reduct, which can be computed using rough sets. By using discernibility matrices, the method of computing reducts was described by Skowron and Ruszer (1991). However, this method cannot list all possible reducts of the IS. Starzyk (1999) gave an algorithm to list all reducts of the given IS. The predominant attributes were found using val theory (Rao and Kumar, 1999), which were equivalent to reducts. The DT is constructed based on the predominant attributes.

ROUGH SETS (RS) AND DECISION TREE (DT)

Rough sets (RS): Pawlak (1982) introduced theory of RS. Pawlak (1985) derived rough dependency of attributes in IS. Some of the concepts of RS are given:

Knowledge base: In RS theory, a decision table is denoted by T = (U, A, C, D), where U is universe of discourse, A is a set of primitive features and C, D $\subset$ A are the two subsets of features that are called condition and decision features, respectively.

Let a $\in$ A, P $\subseteq$ A. A binary relation IND (P), called the indiscernibility relation, is defined as follows:

$$IND(P) = \{(x, y) \in U \times U: \forall a \in P, a(x) = a(y)\}$$

Let U/IND (P) denote the family of all equivalence classes of the relation IND(P). For simplicity of notation U/P will be written instead of U/IND(P). Equivalence classes U/IND(C) and U/IND (D) will be called condition and decision classes, respectively.

Let R$\subseteq$ C and X $\subseteq$ U, $\underline{R}X = \cup\{Y \in U/R: Y \subseteq X\}$ and $\overline{R}X = \cup\{Y \in U/R: Y \cap X \neq \Phi\}$. Here $\underline{R}X$ and $\overline{R}X$ are said to be R-lower and R-upper approximations of X and ($\underline{R}X$, $\overline{R}X$) is called R-rough set (Ganesan *et al.*, 2005; Pawlak, 1991). If X is R-definable then $\underline{R}X = \overline{R}X$ otherwise X is R-Rough. The boundary $BN_R(X)$ is defined as $BN_R(X) = \overline{R}X - \underline{R}X$. Hence, if X is R-definable, then $BN_R(X) = \Phi$. $\underline{R}X$ is also called positive region. The negative region of R is written as $NEG_R(X)$.

$POS_R(X) = \underline{R}X$, $BN_R(X) = \overline{R}X - \underline{R}X$,
$NEG_R(X) = U - \overline{R}X$.

Corresponding Author: Ramadevi Yellasiri, Department of CSE, Chaitanya Bharathi Institute of Technology, Hyderabad, India

Example 1: Consider the universe of discourse U = {a, b, c, d, e, f} and R be any equivalence relation in IND (K) (where K is knowledge base consisting of U and R) which partitions U into {{a, b, d}, {c, f}, {e}}. Then for any subset X = {a, b, c, d} of U, $\underline{R}X$ = {a, b, d} and $\overline{R}X$ = {a, b, c, d, f}. Hence, $BN_R(X)$ = {c, f}. Hence, $POS_R(X)$ is {a, b, d} and the $NEG_R(X)$ is {e}.

On the other hand, consider a subset Y = {c, e, f}. Here, $\underline{R}Y$ = {c, e, f} and $\overline{R}Y$ = {c, e, f}. Therefore, $BN_R(Y) = \Phi$. Hence, Y is said to be R-definable.

Dispensable and indispensable features: Let $c \in C$. A feature 'c' is dispensable in T, if $POS_{(C-\{c\})}(D) = POS_C(D)$; otherwise feature c is indispensable in T. 'c' is independent if all $c \in C$ are indispensable.

Reduct and CORE: A set of features $R \subseteq C$ is called a Reduct of C, if T' = {U, A, R, D} is independent and $POS_R(D)$. In other words, a reduct is the minimal feature subset preserving the above condition.

CORE (C) denotes the set of all features indispensable in C. We have CORE (C) = ∩ RED(C), where RED(C) is the set of all reducts of C.

Discernibility matrix: If the decision attributes of a pair of instances differ, then the matrix entry for that pair will be the list of attributes in which they differ.

Example 2: The discernibility matrix corresponding to the sample database shown in Table 1 with U = {X_1, X_2,.., X_7} C = {a, b, c, d}, D = {E} is shown in Table 2.

$M_{(X1, X3)}$ = (b, c, d) as X_1 and X_3 have different decision value, they differ in b, c and d attributes.

The Reduct for data of Table 1 are {b, c} and {b, d}. CORE = {b}.

Example 3: Consider sunburn dataset
U = {X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8}
A = {Hair, Height, Weight, Lotion, Sunburn}
For R = {Weight} ⊆ A, U/IND(R) = {{X_1, X_8}, {X_2, X_3, X_4}, {X_5,X_6, X_7}}
Let X = {X_1, X_4, X_5}. Then with R = {Weight} for instances with Sunburn = Yes,
$\underline{R}X = \Phi$ and $\overline{R}X$ = {X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8}.

Selecting an optimal reduct R from all subsets of features is not an easy work. Considering the combinations among N features, the number of possible reducts can be 2^N shown in Table 3. Hence, selecting the optimal reduct from all of possible reduct is NP-hard.

Decision tree construction: A DT is typically constructed recursively in a top-down manner by splitting the given set of examples. The test defines a partition of the instances according to the outcome of the test as applied to each instance. A branch is created for each block of the partition and for each block, a DT is constructed recursively. A DT can be pruned, i.e., restricting the growth of the tree before it occurs. If the test at the node is done based on just one variable, it is called univariate test, otherwise it is multivariate test.

The best test at the internal node is selected based on heuristic function, which includes Information gain, Gain ratio, GINI and Kolmogorov-Smirnoff distance. It is quite possible that a tree will over fit the data, therefore post-pruning methods are available that reduce the size of the tree after it has been grown.

Traversing the tree from root to different leaf nodes will generate different decision rules. The path from root to each leaf is one decision rule. The DT and decision rules generated for the weather dataset is shown in Table 4.

Rules generated are of the form
1. If Outlook = Sunny and Humidity = Normal,Play = Yes.
2. If Outlook = Sunny and Humidity is = High,Play = No.
3. If Outlook = Overcast, Play = Yes.
4. If Outlook = Rain and Wind = Weak, Play = Yes.
5. If Outlook = Rain and Wind = Strong, Play = No.

Table 1: A sample database

ID	a	b	c	d	e
X_1	1	0	2	1	1
X_2	1	0	2	0	1
X_3	1	2	0	0	2
X_4	1	2	2	1	0
X_5	2	1	0	0	2
X_6	2	1	1	0	2
X_7	2	1	2	1	1

Table 2: Discernibility matrix for data in Table 1

	X_1	X_2	X_3	X_4	X_5	X_6
X_2	-					
X_3	b, c, d	b, c				
X_4	b	b, d	c,d			
X_5	a,b,c,d	a,b,c	-	a,b,c,d		
X_6	a,b,c,d	a,b,c	-	a,b,c,d	-	
X_7	-	-	a,b,c,d	a,b	c,d	c,d

Table 3: Sunburn dataset

ID	Hair	Height	Weight	Lotion	Sunburn
X_1	Blonde	Average	Light	No	Yes
X_2	Blonde	Tall	Average	Yes	No
X_3	Brown	Short	Average	Yes	No
X_4	Blonde	Short	Average	No	Yes
X_5	Red	Average	Heavy	No	Yes
X_6	Brown	Tall	Heavy	No	No
X_7	Brown	Average	Heavy	No	No
X_8	Blonde	Short	Light	Yes	No

Table 4: Weather dataset

Id	Outlook	Temp	Humid	Wind	Play
X1	Sunny	Hot	High	Weak	No
X2	Sunny	Hot	High	Strong	No
X3	Overcast	Hot	High	Weak	Yes
X4	Rain	Mild	High	Weak	Yes
X5	Rain	Cool	Normal	Weak	Yes
X6	Rain	Cool	Normal	Strong	No
X7	Overcast	Cool	Normal	Strong	Yes
X8	Sunny	Mild	High	Weak	No
X9	Sunny	Cool	Normal	Weak	Yes
X10	Rain	Mild	Normal	Weak	Yes
X11	Sunny	Mild	Normal	Strong	Yes
X12	Overcast	Mild	High	Strong	Yes
X13	Overcast	Hot	Normal	Weak	Yes
X14	Rain	Mild	High	Strong	No

REDUCT BASED DECISION TREE (RDT)

RDT algorithm mainly consists of two important steps i.e., Reduct Computation and DT Construction. RDT combines the merits of both RS theory and DT induction algorithm, thus improving efficiency, simplicity and generalization capability of both the base algorithms. Datasets can be discrete or continuous, in our work we experimented with discrete type; therefore continuous attributes are discretized using any available discretization algorithms (Han and Kamber, 2001; Ramadevi and Rao, 2007) like B Rorthogonal Scalar, Boolean reasoning (Hoa and Son, 1996; Ohrn, 1999) etc. In our work, Threshold value was used to discretize the data.

The RDT Algorithm: In the Reduct Computation Algorithm (RCA), the decision table is given as input and predominant attributes called reduct is obtained as output. If the data is large, vertical fragmentation is performed. Decision attribute is appended to each fragment and RCA is applied. The predominant attributes for all fragments are obtained and they are grouped together with fragment information and decision attribute. To this RCA is once again applied giving rise to, a new set of attributes called composite reduct.

Reduct computation algorithm (RCA)
Algorithm RCA
(Input: decision table; output: reduct):

- Read the decision table T1.
- Sort the rows in ascending order of the decision attribute.
- Initialize the set of predominant attributes SPA to Null
- Construct a Boolean Matrix (BM), as explained in step 5, by generating a row for each pair of rows having different values for the decision attribute.
- Construct a row with 1's and 0's. Assign a '1' to an element if the corresponding independent attribute values are different, otherwise assign '0'.
- Repeat the following steps 7 through 8 until the sum of rows in the BM are zeroes or null matrix.
- Pick up the attribute 'a', which has the maximum sum and append it to the SPA.
- Remove all the rows from BM for which the elements are '1' corresponding to 'a'.
- If BM is non-null, then print, "The SPA roughly explains about the decision attribute".
- Assign SPA to reduct.

Decision tree constructs by taking reduct for splitting
Algorithm RDT
(Input: training dataset T1; output: decision rules):

- Input the training dataset T1.
- Discretize the continuous attributes if any and label the new dataset as T2.
- Compute reduct of T2, say R using RCA.
- Reduce T2 based on reduct R and label-reduced dataset as T3.
- Construct decision tree on T3 with reduct R, taking one attribute at a time and using it for splitting in breadth first manner (all nodes at the same level).
- Generate the decision rules by traversing all the paths from the root to the leaf node in the decision tree.

Complexity of RDT: RDT consists of two steps mainly Reduct Computation and Decision tree Construction. Complexity of the RDT depends on the complexity of RCA and DT construction algorithms.

If the training data consists of 'n' instances and 'm' attributes, then the problem of computing all possible minimal length reducts is NP-Hard (Hoa and Son, 1996; Ohrn, 1999). RCA consists of preprocessing the data for reduct computation. The computational cost of preprocessing the training data and sorting the data is $O(n^2)$ sets of length O(m). C(n,2) comparisons are required and if it is with 'm' attributes the complexity will be of order $O(mn^2)$. The complexity of a decision tree depends upon the splitting attribute values.

IMPLEMENTATION OF RDT

Data sets: Proteins are made up of 20 Amino Acids (AA). They consist of long sequences of AA, each AA is treated as one character for performing spatial analysis. GPCR is one such protein family; it consists of 3896 sequences, which are divided into 5 classes. A sample

Table 5: Results of Five-fold test of GPCR dataset

	Correctly classified		Misclassified	
Sets	RDT (%)	ID3 (%)	RDT (%)	ID3 (%)
Set 1	80	78	20	22
Set 2	80	81	20	19
Set 3	82	80	18	20
Set 4	84	84	16	16
Set 5	82	80	18	20

Table 6: Comparison of different classification techniques using Kappa Statistics

	Kappa statistics		
Classification technique	Sunburn	Weather	GPCR
Bayes Network	0	0	0.306
ComplementNaiveBayes	0.142	0	0.29
NaiveBayesMultinomial	0	0	0.013
Logistic	0	0.588	0.362
RBFNetwork	0	0.256	0.416
SimpleLogistic	0	0	0.359
SMO	0	0	0.338
BFTree	0	0	0.574
J48	0	0.143	0.582
J48graft	0	0.143	0.585
LMT	0	0	0.556
NBTree	0	0	0.445
RandomForest	0	0.429	0.652
RandomTree	0	0.378	0.595
SimpleCart	0	0	0.572
RDT	0.5	0.58	0.62

GPCR dataset was considered for the implementation of RDT. Spatial Analysis with different AA as center is performed on the dataset and then discretized based on threshold (T). The potential for discriminating the features is lost when all the entries of the matrix is 1. The decision attribute is added as the first column for the resultant Binary Association Matrix and RCA is executed. Reducts are generated taking into consideration different centers for the same dataset. These reducts of different centers are used for the generation of the corresponding decision trees. The less frequent occurring character is taken as center and spatial analysis is performed in our experiment.

In addition to the above data the popular datasets that are frequently used in machine learning are considered for demonstrative purpose. Here the data is organized into nominal form by appropriate transformations in the present study.

Evaluation methods and measures: Cross-validation is used for predicting the accuracy of the techniques. In this, the dataset is randomly partitioned into predefined number of folds say, 'K'. Then each fold is taken for testing in turn and remaining K-1 folds are used for training. Five-fold test was performed on the datasets (Skowron and Rauszer, 1991) and results are shown in Table 5. A performance evaluation with various classification algorithms is made and results are shown in Table 6.

Table 7: Reduct for GPCR Data with LFC of 5 training sets

SET	LFC	Reduct
Set 1	H	E K Y G N C D R Q M F H P T V I A S
	M	E N Y D R K P G C Q I M W H F T S A V
	Q	Q K D N E R F P Y G I M C V T H A
	W	D R E P N C K M F H A Y V I Q W T S G
Set 2	H	E C Y R Q N G D K M H V P T F A S I
	M	E R K N D Y Q G H M C W F P V I T S
	Q	Q D Y N M K I P G E F V R C W A H T
	W	D E N R C P K W M Q H F A T I Y G S
Set 3	H	E Q C G N D R K P M F H I V Y S A T W
	M	E N K R Y D P M W C Q G H F I T V S
	Q	D N Y Q K P F E R H I M V C W A T G
	W	D R E C N K P W Q M I H A F T V Y S G
Set 4	H	E C Q Y K D N G H M I P F R V T A W
	M	E N K R D Y P Q C W H I F G V T M S
	Q	Q N Y K D P C I G E F V M R A T W H
	W	E D C R P N K W M H Q A F I T Y S G
Set 5	H	E Y C Q D K N R G M H V P I F S A T W
	M	E K R P D Y N Q M C W F G H I T S V
	Q	D Q N Y K C P F R G I E A V W H M T
	W	D E R C N P K W M H Q I T Y A V F S

EXPERIMENTAL RESULTS

RDT combines the efficiency of RS and DT induction. Predominant attributes are used for reducing the data size. Composite reducts are obtained by vertically fragmenting the data. RDT application on discrete data and five-fold test shows that RDT is better than existing DM classification algorithms in terms of efficiency and complexity. RDT can be applied on continuous or categorical data after discretization. Noise effects and their elimination have to be studied.

GPCR: G-Protein Coupled Receptor (GPCR) is a protein data (www.rcsb.org). The data is extracted from the standard database. It consists of 3896 sequences belonging to 5 classes (class A, B, C, D and E). Spatial analysis is performed on the data (Ramadevi and Rao, 2007) and the data is transformed into nominal data (using threshold). The data is divided into five sets and the Least Frequent occurring Character (LFC) is taken as center and the reduct is computed. Various set and corresponding reduct (for LFC) is shown in Table 7.

Sunburn data set: Sunburn data set consists of 5 attributes (Hair, Height, Weight, Lotion and Sunburn). Here the sunburn is the decision attribute (Table 3). The data is categorical and it is transformed into nominal. The reduct is computed to be {Height, Hair, Lotion}. The proposed RDT is demonstrated on it and compared with other classification tools (WEKA tool is used for experimentation of other methods and RDT was constructed according to the procedure). Kappa Statistics is used for measuring the efficiency.

The efficiency of RDT over other algorithms is shown in Fig. 1. In case of large dataset (GPCR), RDT and Random Forest are equally efficient.

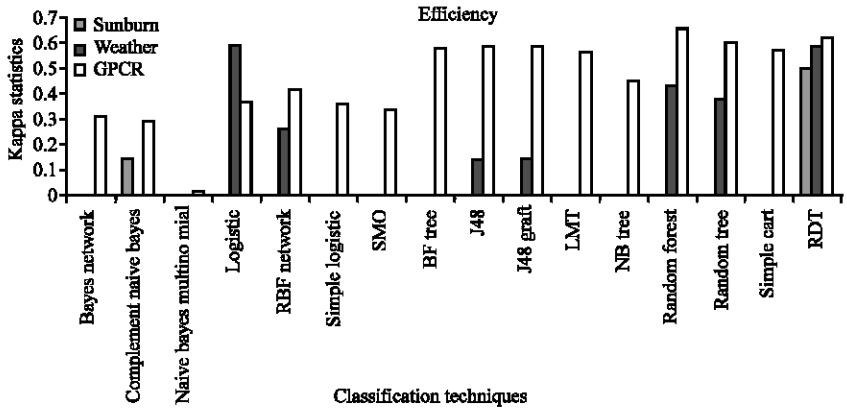

Fig. 1: Graph showing the efficiency using Kappa Statistics

CONCLUSION

Reduct computation using predominant methods is presented in this study. The Kappa Statistics shows that, for small and large data sets with nominal data, RDT is efficient than other algorithms. Five-fold test indicates that when compared to other data mining classification algorithms, RDT was more consistent in correctly classifying the data and less consistent in misclassification. Preprocessing the data (elimination of redundant attributes) facilitates less memory and accelerates the process of classification.

REFERENCES

Ganesan, G., C. Raghavendra Rao and D. Latha, 2005. An overview of rough sets. Proceedings of the National Conference on the emerging trends in Pure and Applied Mathematics, Palayamkottai, India, pp: 70-76.

Han, J. and M. Kamber, 2001. Data Mining: Concepts and Techniques. Morgan Kaufmann, pp: 279-325.

Hoa, N.S. and N.H. Son, 1996. Some Efficient Algorithms for Rough Set Methods. Proceedings of International Conference IPMU-96, Spain.

Minz, S. and R. Jain, 2005. Refining decision tree classifiers using rough set tools. Int. J. Hybrid Intelligent Sys., 2: 133-148.

Ohrn, 1999. Discernibility and Rough Sets in Medicine: Tools and Applications. Ph.D Thesis, Norwegian University of Science and Technology.

Pawlak, Z., 1981. Information systems-theoretical foundations. Inform. Sys., 6: 205-218.

Pawlak, Z., 1982. Rough sets. Int. J. Comput. Inform. Sci., 11: 341-356.

Pawlak, Z., 1985. On rough dependency of attributes in information systems. Bull. Polish Acad. Sci. Tech., 33: 481-485.

Pawlak, Z., 1991. Rough Sets-Theoretical Aspects and Reasoning about Data. Kluwer Academic Publications.

Rao, C.R. and P.V. Kumar, 1999. Functional Dependencies through Val, ICCMSC, India. TMH Publications, pp: 116-123.

Radwan, E. and E. Tazaki, 2004. Rough Sets and Genetic Algorithms inLearning Cellular Neural Networks Cloning Template For Decision Making System. Int. J. Neural Sys., 14 (1): 57-68.

Ramadevi, Y. and C.R. Rao, 2006. Knowledge Extraction Using Rough Sets-GPCR-Classification. International Conference on Bioinformatics and Diabetes Mellitus, India.

Ramadevi, Y. and C.R. Rao, 2006. Feature Extraction from large database using Spatial Analysis-Concept Lattice. ICORG, Hyderabad, India.

Ramadevi, Y. and C.R. Rao, 2007. Decision tree induction using rough set theory-comparative study. Journal of Theoretical and Applied Information Technology, Vol. 3.

Skowron, A. and C. Rauszer, 1991. The discernibility matrices and functions in information systems. Fundamenta Informaticae, 15 (2): 331-362.

Son, H.N. and A. Skowron, 1997. Quantization of Real Value Attributes Rough Set and Boolean Reasoning Approach. Bulletin of International Rough Set Society 1.

Sree Hari Rao, V., C.R. Rao and K.Y. Vikram, 2006. A Novel technique to evaluate fluctuations of mood: Implications for evaluating course and treatment effects in bipolar/affective disorders. Bipolar Disorders, 8: 453-466.

Starzyk, J., D.E. Nelson and K. Sturtz, 1999. Reduct Generation in Information Systems. Bulletin of International Rough Set Society, 3 (1/2).