J. Ind. Soc. Agril. Statist.
59(2), 2005 : 161–168

Dynamic RDT Model for Mining Rules from Real Data

Rajni Jain and Sonajharia Minz[1]
National Centre for Agricultural Economics and Policy Research, New Delhi

(Received : July, 2005)

SUMMARY

Dynamic Rough Set based Decision Tree Induction (RDT) model is proposed to deal with the noise present in the real time dataset. The paper explores the variants of RDT models for learning classification rules. The required set of classification rules is aimed to help in identification of the households which are vulnerable to food shortage. The classification rules are desired to be as simple as possible. In this paper, classical rough set method, C4.5 algorithm, the hybrid algorithm RDT and its variants as well as dynamic RDT model are used for mining rules from a real dataset. The experimental results are compared graphically with that of the base algorithms based on the performance parameters classification accuracy, complexity, number of rules and the CS score for the resulting classifier. The performance parameter accuracy as obtained by using Linear Discriminant Analysis is used as a benchmark for comparing accuracy of the proposed model called dynamic RDT. The performance of the proposed model is observed to be better for the real dataset.

Key words : Rough set, Data mining, Dynamic reduct, Classification, Dynamic RDT model, Rule extraction, LDA.

1. INTRODUCTION

The aim of the paper is to present a basic framework of dynamic Rough Set based Decision Tree Induction (RDT) model for learning classification rules from real data. RDT model is an integration of the classical Rough Set (RS) approach and the Decision Tree (DT) induction (Minz and Jain (2003a)). The set of decision rules obtained using all the conditional attributes can be too large to be suitable for the classification of the unseen object(s). It is also difficult for human users to comprehend and use them directly without the use of a computer. The problem of learning simpler classification rules is analyzed in this paper using the RDT model. The suitability of the RDT model for learning simple and accurate classification rules from small datasets and from the repository of datasets from University of California (UCI) available at http://www.ics.uci.edu/~mlearn is examined in Jain and Minz (2003a); and Minz and Jain (2003b) based on the experimental results. In this paper, we propose a novel variant of RDT namely dynamic RDT which involves computation of dynamic reducts (Bazan *et al.* (1994)). Some other variations of basic RDT approach are also presented for the sake of completeness. All these approaches are compared by applying each to a real data, Nutrition dataset. A study for the nutritional security of rural households in India using other methods has already been carried out in Adhiguru and Ramasamy (2003).

2. REVIEW OF BASIC CONCEPTS

2.1 Noise in Real Dataset

Real world data is almost always characterized by incomplete data as well as imperfect values of the attributes which require due attention in designing a learning algorithm. Data is incomplete because attributes of interest may not always be available or the data may not be included because it was not considered important at the time of entry. The data is imperfect if it is noisy and inconsistent (e.g., containing discrepancies in the codes used to categorize items or counter examples in the training data) and/or contains only aggregated values. Noise is a random error or variance in a measured variable (Han and Kamber (2001)). There may be many possible reasons for noisy data. The data collection instruments used may be faulty. There may have been human or computer errors occurring at the time of data entry. Zhu *et al.* (2003) defines two types of noise: (a) attribute noise; and (b) class noise. The former is the

[1] *School of Computers and Systems Sciences, Jawaharlal Nehru University, New Delhi-110067*

error that is introduced in the attribute values of the instances while the latter is either due to contradictory examples or because of misclassifications. Parsons (1998) surveyed methods for representing and reasoning with imperfect information. He also classified different types of imperfection that may pervade data, and discussed the sources of such imperfections in detail. Some of the popular techniques to cope up with the noisy datasets are as follows.

2.1.1 Removal of erroneous training examples

Induction is performed using only representative training examples, either selected by the expert or automatically, as was done by the ESEL system (Clark and Niblett (1986)) for the task of Soybean diagnosis. However, selection of noise based on expert advice is not practically feasible for data mining because of cost constraints, non-availability of experts and difficulties in identification of a noisy example from the large datasets. In an automatic approach, erroneous training examples can be detected by clustering, where similar values are organized into clusters. Intuitively, values that fall outside the set of clusters may be due to the noise and hence can be removed (Han and Kamber (2001)).

2.1.2 Combined computer and human inspection

Outliers may be identified through a combination of computer and human inspection (Han and Kamber (2001)). In one application for example, an information–theoretic measure was used to help in identification of outlier patterns in a handwritten character database for classification. The value of the measure reflected the surprise content of the predicted character label. Patterns whose surprise content is observed as above a threshold value are output to a list. Outlier patterns may be informative (useful data exceptions) or garbage (mislabelled). Subsequently, a human can sort through the patterns in the list to identify the actual garbage. Then the garbage patterns can be excluded from use in subsequent data mining.

2.1.3 Binning

Binning methods may be used as a technique for data smoothing (Han and Kamber (2001)). Binning methods smooth a sorted data value by consulting the values around it. In smoothing by bin means/ medians, each value in a bin is replaced by the mean/ median value of the bin. In smoothing by bin boundaries, the minimum and maximum values in a given bin are identified as the bin boundaries. Then each bin value is replaced by the closest boundary value.

2.1.4 Flexible rule application

In this technique, interpretation of the rule involves the use of weights and probabilities instead of Boolean values alone, thus exploiting the maximum information contained in the training data. Misclassification by an erroneous rule may be overridden by other rules, whose conditions are nearly satisfied and which have higher weights attached to them.

2.1.5 Consideration of rules with counter examples

This method involves relaxing of the constraint that the induced rules should be completely consistent (or consistent to the maximum extent if complete consistency is impossible) with the training data. The pruning of decision trees is an example of this technique. Quinlan (1986) presents a detailed empirical study of the effect of tree pruning in noisy domains.

2.1.6 The use of domain knowledge

Additional domain knowledge can be used to reduce the problems of description language and noise (Michalski (1986)). For example, if the raw data contain some coded information then the noisy attributes can be appropriately tackled if the domain knowledge exists. Domain knowledge can also help to reject some unimportant attributes out rightly. However, this method will not be useful if no domain expert is available.

2.1.7 Partition the dataset for eliminating class noise

Zhu *et al.* (2003) proposed a novel approach for identifying and eliminating mislabelled instances in large datasets (class noise). They first partitioned the dataset into subsets, constructed rules from each subset, and then used the rules to evaluate the whole dataset. For a given instance they also defined the two types of error count variables to count the number of times the instance has been identified as noise by all the subsets. The instances with higher error values are assigned a higher probability of being a mislabelled example.

2.2 Rough Sets

2.2.1 Basic concepts

Rough Set theory was introduced in early 1980s by Z. Pawlak and since has come into focus as alternative to the more widely used methods of machine learning and statistical data analysis (Pawlak (1991), Bazan *et al.* (1994), Witten and Frank (2000)). In RS, an information system S is a 4-tuple, S = (U, Q, V, f) where, U a non-empty, finite set of objects is called the universe; Q a finite set of attributes; $V = \cup Vq, \forall q \in Q$ and Vq being the domain of the attribute q; and f: U × Q ⟶ V, be the information function assigning values from the universe U to each of the attributes q for every object in the set of examples. A decision table is an information system where $Q = (C \cup D)$. C is the set of categorical attributes and D is the set of decision attributes. In RS, the decision table represents either a full or partial dependency occurring in data. For $P \subseteq Q$, a subset of attributes of an information system S, an indiscernibility relation denoted by IND is defined as

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

If (x, y) ∈ INDs(P) then objects x and y are called indiscernible with respect to P. The subscript s may be omitted if information system is implied from the context. IND(P) is an equivalence relation that partitions U into equivalence classes, the sets of objects indiscernible with respect to P. Set of such partitions are denoted by U/IND(P). Let $X \subseteq U$ be a subset of the universe. The description of X is defined in terms of P-lower approximation (denoted as $\underline{P}$) and P-upper approximation (denoted as $\overline{P}$) where for P ⊆ Q, then

$$\underline{P}X = \cup\{Y \in U/IND(P) : Y \subseteq X\}$$

$$\overline{P}X = \cup\{Y \in U/IND(P) : Y \cap X \neq \phi\}$$

A set X for which $\underline{P}X = \overline{P}X$ is called an exact set otherwise it is called rough set with respect to P.

The minimum set of attributes that preserves the indiscernibility relation is called reduct. RED(S) denotes a set of all computed reducts for an information system S. The problem of computation of all minimal reducts is observed to be an NP hard problem but many algorithms with heuristics have shown that a single relative reduct can be computed in linear time. Genetic algorithms are also used for simultaneous computation of multiple reducts (Jain and Minz (2003a), Ohrn (1999)). Tracing the attribute values from the reduced decision table produces the classification rules.

2.2.2 Rough sets for noise handling

Classical theory of RS is useful when the classification in the given decision table is fully correct or certain. The classification with a controlled degree of uncertainty or with a controlled misclassification error is outside the realm of the classical rough set approach. For example, when dealing with empirical data such as market survey data, it may not be possible to identify the non-empty lower approximation of the target category e.g. category of buyers of a service. Similarly, it is often not possible to identify non-trivial upper approximation of the target category such that it would not extend over the whole domain. These limitations are the natural consequences of the fact that classification problems are often inherently non-deterministic.

The rough set method as explained above in Section 2.2.1 is not sufficient for mining rules from real datasets. Bazan (1994) developed an idea of dynamic reducts as a tool to find relevant reducts for the decision rule generation. He explained that the underlying idea of dynamic reducts stems from the observation that reducts generated from the real life information system are unstable as they are sensitive to changes introduced by removing a randomly chosen set of objects from the information system. Dynamic reducts are the set of conditional attributes appearing *sufficiently often* as reducts of samples from the original decision table. The attributes belonging to *most* of the dynamic reducts are defined as relevant. The value thresholds for *sufficiently often* and *most* need to be tuned for a given dataset. The process of computing dynamic reducts (Bazan (1994)) from an information system S can be seen as combining normal reduct computation with resampling techniques. The reducts that occur most often in the outlined procedure are believed to be the most *stable*, and reveal more general relationships in S than an arbitrary RED(S) does. The set of ε dynamic reducts of an information system S with respect to a family of sampled subsystems F is denoted by DRED(S, ε, F), and consists of those attribute subsets that occur frequently enough as reducts, as determined by the parameter ε. To achieve dynamic reducts, the term $B \subseteq S$ in the following equation can be substituted by $B \subseteq RED(S)$.

$$DRED(S,\varepsilon,F) = \left\{ B \subseteq S / \frac{|\{S_i \in F \mid B \in RED(S_i)\}|}{|F|} \geq 1 - \varepsilon \right\} \tag{1}$$

The dynamic reducts have shown their utility in various experiments with datasets having noise. The result from the experiments using dynamic reducts also shows that these reducts can be treated as relevant features (Bazan (1994)). The quality of the classification of the set of unseen objects improves by use of the decision rules obtained from dynamic reducts for noisy datasets.

3. DYNAMIC RDT

RDT model as proposed by Minz and Jain (2003a), integrates the merits of both RS and DT induction algorithm. The issues related to the greediness of the Decision Tree algorithms and the complexity of rules in RS approach, are addressed by the RDT model as proposed by Minz and Jain (2003b). The cumulative performance evaluation of RDT and some variants of RDT namely DJU, DJP, RJU, RJP (refer Table 2 for details) based on the experiments with benchmarking datasets is presented in Jain and Minz (2003b). The real dataset containing noise requires a specialized variant of RDT to suit the corresponding challenges. The proposed dynamic RDT is expected to handle noisy domains as well.

Experimental results of Bazan (1994) show that the application of dynamic reducts leads to increase in the quality of classification and decrease in the size of the decision rule sets. Further, inspired by the partition method of noise handling by Zhu *et al.* (2003), the use of dynamic RDT is expected to help to cope up with the problem of noise in the real time dataset.

Figure 1 explains the overall architecture of the dynamic RDT model. In this figure, the training data refers to the collection of examples used for learning the rules for a given domain. Training data is sampled from the given dataset. RS requires the data to be described with discrete values. Hence a continuous domain is replaced with a discrete one using a process called discretization. A number of algorithm are available in the literature for discretization (Fayyad and Irani (1992), Chmielewski and Grzymala-Busse (1994), Dougherty *et al.* (1995), Nguyen *et al.* (1998)). The next step involves computation of the reducts and selection of the most stable dynamic reduct. Definition and description of dynamic reduct computation is presented in Section 2.2.2. The dynamic reduct distinguishes between most of the examples belonging to different decision classes in the presence of noise. Like a reduct, it also assists in reducing the training data by removing the attributes not present in the reduct. The reduced training data is finally used for decision tree induction. C4.5 is an algorithm for DT induction and was proposed by Quinlan (1993). Java implementation of C4.5 algorithm called J4.8 by Witten and Frank (2000), is used for decision tree induction. The tree may be mapped to rules by following each possible path from the root towards the leaf.

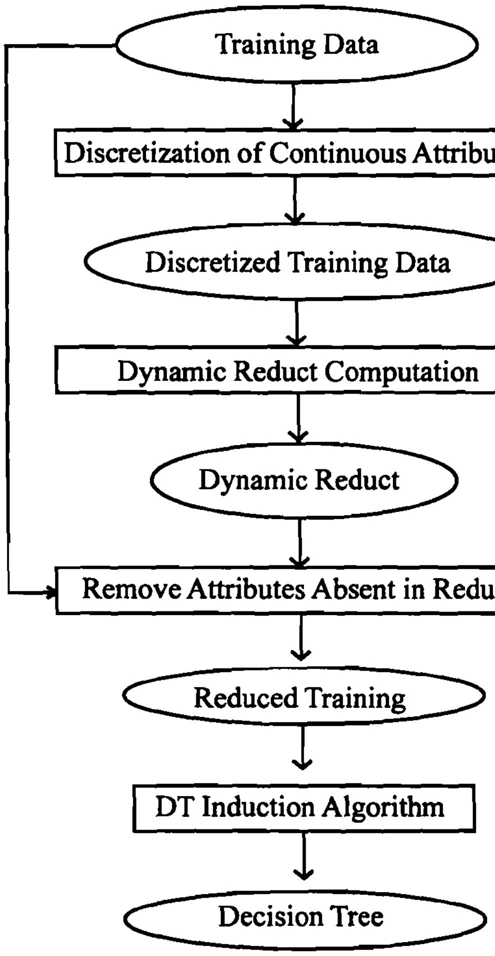
Training Data
Discretization of Continuous Attributes
Discretized Training Data
Dynamic Reduct Computation
Dynamic Reduct
Remove Attributes Absent in Reduct
Reduced Training
DT Induction Algorithm
Decision Tree

Figure 1. The architecture of dynamic RDT model

The computational method for dynamic reduct as suggested by Bazan *et al.* (1994) requires samples of varying size from the dataset. Considering samples of the same size, an alternative algorithm for the computation of dynamic reduct is proposed. The modification in the algorithm is inspired by the usage of the sampling in the n-fold cross validation.

Algorithm *DynamicReductCompute;*
Input: *n, dataset*; **Output:** *Dynamic Reduct*

1. Divide the dataset into folds.
2. For i =1 to n, repeat step 3.
3. Leaving fold[i], compute decision relative reducts for the remaining n-1 folds and denote the set of reducts by R[i].
4. The set of reducts as obtained from the steps 1 to 3 is n in number. Select the reduct with the highest frequency as the dynamic reduct.

4. EXPERIMENTAL DETAILS

4.1 Dataset

Nutrition data (Table 1) is extracted from the primary survey data of 180 rural households for a study (Adhiguru and Ramasamy (2003)) on nutritional security at National Centre for Agricultural Economics and Policy Research (NCAP). The objective is to identify the poor households who are not food-secure. Average calories consumed per household is estimated by considering the expenditure on various food items into account. Based on the knowledge of the agricultural experts, the households consuming less than 1500 Calories per capita are labelled as 0 (vulnerable to food insecurity) for the attribute *CLASS* while others as 1 (not vulnerable to food security). Hence, the households consuming average energy less than 1500 Calories per person are classified certainly as very poor and deserve external help to prevent starvation. The resulting classifier should be able to predict the vulnerability of the unseen household (data not used for training the classifier) towards starvation on the basis of the morphological attributes.

4.2 Learning Algorithms

Learning algorithms used for the experiments are summarized in Table 2. Suitable references are also provided for some of the approaches or part of an approach. The column *Algorithm* in Table 2 lists the name of the algorithm and the column *Description* gives the important components of the learning algorithm. Algorithm RS is a pure rough set based approach while CJU and CJP are the Java implementations of the pruned and unpruned versions of C4.5 algorithm respectively. CJU and CJP are also pure algorithms for DT induction. Other rows in Table 2 except the one of LDA present the hybrid approaches and are self explanatory.

4.3 Performance Evaluation

Standard method of predicting the evaluation parameters of a learning technique given a fixed sample of data is 10 × 10 Cross-Validation (CV) (Witten and Frank (2000)). In this paper, classification accuracy, complexity, number of rules, and number of variables are used as evaluation parameters. Classification accuracy is estimated by applying the algorithm to the examples not used for rule induction and is measured by the percentage of the examples for which decision class is correctly predicted by the model (Witten and Frank (2000)). A set of rules induced for classification is called rule-set. The condition of the form attribute = value is called a selector. Total number of selectors in a rule-set is used as a measure of complexity of the rule-set (Minz and Jain (2003a)). To determine number of rules, induced DT is mapped to rules by traversing all paths from root to each of the leaves.

Table 1. Characteristics of nutrition dataset

Attribute	Description	Type	Values
LAND	Whether owner of land	Nominal	0,1
HEDU	Education code of the household head	Continuous	1-26
HAGE	Age of the household head in years	Continuous	20-90
CHLD	Presence of children below certain age	Nominal	0,1
FLSIZE	Number of family members in the household	Continuous	2-20
PRWM	Percentage of women in the household	Continuous	0-100
PEAR	Percentage of earning members in the household	Continuous	0-100
HSTD	Whether homestead garden is available	Nominal	0,1
CLASS	Decision attribute to describe whether food secure	Nominal	0,1

Table 2. Learning approaches used for nutrition dataset

Id	Algorithm	Description
1	RS	Rough Set approach using full discernibility based reduct (Pawlak (1991), Ohrn (1999))
2	CJU	Continuous data, J4.8 algorithm (Witten and Frank (2000)), Unpruned Decision Tree (Quinlan (1993))
3	CJP	Continuous data, J4.8 algorithm, Pruned Decision Tree (Quinlan (1993))
4	DID	Discretization (RS based) (Nguyen *et al.* (1998)), no reduct, ID3 algorithm (Quinlan (1993))
5	RDT	Discretization (RS based), full discernibility based reduct (Ohrn (1999)), ID3 (Minz and Jain (2003a))
6	DJU	Discretization (RS based), J4.8, Unpruned Decision Tree
7	DJP	Discretization (RS based), J4.8, Pruned Decision Tree
8	RJU	RS based discretization and full discernibility based reduct, J4.8, Unpruned DT
9	RJP	RS based discretization and full discernibility based reduct, J4.8, Pruned DT
10	DRJU	RS based discretization and Dynamic reduct, J4.8, Unpruned DT
11	DRJP	RS based discretization and Dynamic reduct, J4.8, Pruned DT
12	LDA	Linear Discriminant Analysis (Johnson *et al.* (2002))

It has also been observed that sometimes a learning scheme may relatively perform better in terms of one or some of the performance parameters but not so in terms of the remaining performance measures. Jain and Minz (2003b) used Cumulative Score (CS) as a criterion for the comparison of learning schemes to deal with such situations. Depending on the importance and preferences for various performance parameters, the user may determine the weights (w_i) for the corresponding performance parameters (X_i). The sum of all the weights under consideration must be 1. For the experiments in this paper, all the evaluation parameters are assigned equal weights. CS is computed using the following expression

$$CS = \sum_{i=1}^{k} w_i X_i \text{ where } 0 \le CS \le 1 \text{ and } \sum_{i=1}^{k} w_i = 1 \quad (2)$$

5. RESULTS AND DISCUSSION

Average of the performance parameters as obtained by using different learning schemes and 10 × 10 CV are shown in Table 3. The column CS shows the cumulative score as obtained using Equation (2) and equal weights for all the performance parameters. For algorithm LDA, only the performance parameter accuracy is shown as other performance parameters are not applicable due to the nature of the approach. The results for LDA are obtained using SPSS and using leave one out validation scheme. The table clearly shows that accuracy, complexity and the number of rules for the algorithms DJP, RJP, DRJP and CJP are almost same but for the algorithm DRJP number of attributes requirement is much less in the resulting classifier.

Classification models obtained from each of the algorithm are compared graphically in Figure 2. It is observed that RS produces largest number of rules that are specific and therefore does not perform well for unseen data. Further DJP, RJP and DRJP show good classification accuracy and are comparable to accuracy obtained from Quinlan's C4.5 algorithm (CJP) but the performance of each one is better than RS, CJU and their respective unpruned counterparts (DJU, RJU and DRJU). The paired sample t-test confirms that difference in classification accuracy of DJP, RJP, DRJP and CJP are not significant at 99% confidence interval of the difference. Similar pattern is observed on comparing the complexity and number of rules. Number of attributes in the learned classifier is reduced by more than 40% for RDT based classifiers as compared to CJP. This reduction in attribute requirement is a boon for users who wish to use the classifier for identification of vulnerable households by considering as few attributes as possible. This is also useful for conducting future surveys of similar nature as only the significant attributes may be considered for data collection. DRJP induces a classifier having 9 simple rules based on 4 variables only with an estimated classification accuracy of 73%. The classifier appears to be interesting and useful to users as a viable alternative to a detailed survey to identify vulnerable group. In general, the estimated classification accuracy of 73% may not look impressive. However, it is worth mentioning here that the dataset is a real dataset, which has been collected for a different purpose; hence it may be lacking some more relevant morphological attributes, which may impact the value of the attribute *CLASS* directly or indirectly. Also, the size of the available dataset is too small to capture the entire real time behaviour. Use of Linear Discriminant Analysis (LDA), a standard statistical tool, for learning classifier model from Nutrition dataset using leave one out scheme has resulted in the average cross validated accuracy of 71.1%. LDA method does not provide attribute selection, rules and the complexity of the rules; hence the computation of these parameters is not done for LDA. This suggests that dynamic RDT not only gives comprehensible rules but a more accurate classifier than LDA.

Table 3: Comparing Algorithms using Cumulative Score (CS) for Nutrition Dataset

Algo	Accuracy	Complexity	No. of Rules	No. of Attrib.	CS
RS	51.17	1003	149	6.7	0.1669
CJU	69.33	173	26	8.0	0.2156
CJP	73.16	40	10	7.0	0.2499
DID	59.72	262	79	7.3	0.1878
RDT	59.44	269	82	6.8	0.1896
DJU	67.16	188	56	7.1	0.2088
DJP	73.22	43	16	4.2	0.2647
RJU	68.17	177	55	6.4	0.2155
RJP	72.83	43	17	4.0	0.2661
DRJU	67.33	186	56	6.6	0.2120
DRJP	73.00	43	9	4.0	0.2785
LDA	71.10	-	-	-	-

6. CONCLUSIONS

The use of classical RS model gives low accuracy and weak rules with high complexity. The use of RDT algorithm on Nutrition dataset shows that the classification accuracy, complexity and number of rules

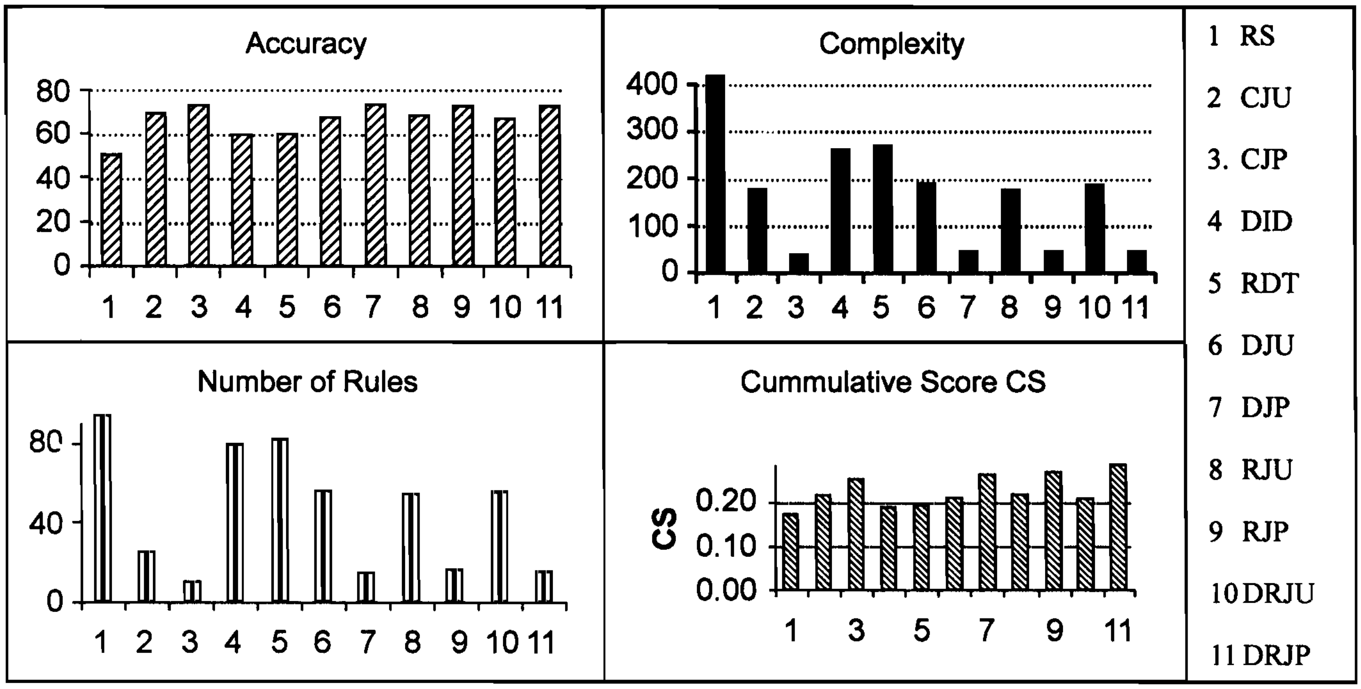

Figure 2. Comparison of learning schemes for Nutrition dataset using 10 × 10 cross validation

are comparable to widely used C4.5 algorithm. RDT schemes increase the accuracy as well as decrease the complexity, number of rules and number of attribute requirements for the learned classifier. The use of the dynamic reduct is substantiated by the good performance of DRJP with respect to all performance parameters. It emphasizes the use of dynamic reducts for RDT algorithm. In future, more experiments may be carried out to compare the performance of dynamic RDT on real time datasets of varying sizes with other learning schemes.

REFERENCES

Adhiguru, P. and Ramasamy, C. (2003). Agricultural-based interventions for sustainable nutritional security. Policy Paper, **17**, NCAP, New Delhi.

Bazan, J., Skowron, A. and Synak, P. (1994). Dynamic reducts as a tool for extracting laws from decision tables. *LNCS,* **869**, 346-355.

Chmielewski, M.R. and Grzymala-Busse, J.W. (1994). Global discretization of continuous attributes as preprocessing for machine learning. *International Journal of Approximate Reasoning,* **11**.

Clark, P. and Niblett, T. (1986). *Induction in Noisy Domains, Expert System.* UK.

Dougherty, J., Kohavi, R. and Sahami, M. (1995). Supervised and unsupervised discretization of continuous features, machine learning. *Proceedings of Twelfth International Conference*, Morgan Kaufmann, Los Altos, CA.

Fayyad, U.M. and Irani, K.B. (1992). On the handling of continuous-valued attributes in decision tree generation. *Machine Learning,* **8**, 77-102.

Han, J., Kamber, M. (2001). *Data Mining Concepts and Techniques.* Morgan Kaufmann Publisher.

Jain, R. and Minz, S. (2003a). Classifying mushrooms in the hybridized rough set frame work. *Proceedings of 1st Indian International Conference on Artificial Intelligence -03*, India.

Jain, R. and Minz, S. (2003b). Should decision tree be learned using rough sets. *IICAI-03*, India.

Johnson, R.A. and Wichern, D.W. (2002). *Applied Multivariate Statistical Analysis.* Pearson Education Asia.

Michalski, R., Mozetic, I., Hong J. and Lavrac, N. (1986). The AQ15 inductive learning system: An overview and experiments. *Proceedings of IMAL*, Orsay.

Minz, S. and Jain, R. (2003a). Rough set based decision tree model for classification. *LNCS,* **2737**.

Minz, S. and Jain, R. (2003b). Hybridized rough set framework for classification: An experimental view. In : *Design and Application of Hybrid Intelligent Systems* A. Abraham *et al.* (Eds.), IOS Press.

Nguyen, H.S. and Nguyen, S.H. (1998). Discretization methods for data mining. In *: L. Polkowski, A. Skowron (Eds.), Rough Sets in Knowledge Discovery*, Physica-Verlag, Heidelberg, 451-482.

Ohrn, A. (1999). *Discernibility and rough sets in medicine: Tools and applications.* Ph.D. thesis, Norwegian University of Science and Technology, Department of Computer and Information Science.

Pawlak, Z. (1991). *Rough Sets - Theoretical Aspects of Reasoning about Data.* Kluwer Academic Publishers.

Parsons, S. (1998). Current approaches to handling imperfect information in data and knowledge bases. *IEEE TKDE* **10(5)**, 862.

Quinlan, J.R. (1986). Learning from noisy data. In : *Machine Learning,* **2**, Michalski, R., Carbonell J., Mitchell, T. (Eds.), Palo Alto, CA, Tioga.

Quinlan, J.R. (1993). *C4.5: Programs for Machine Learning.* Morgan Kauffmann Publishers.

UCI Machine Learning repository available at http://www.ics.uci.edu/~mlearn/MLSummary.html

Witten, I.H. and Frank, E. (2000). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.* Morgan Kaufmann Publishers.

Zhu, X., Wu, X. and Chen, Q. (2003). Eliminating class noise in large datasets. *Proceedings of the Twentieth International Conference on Machine Learning (ICML-03)*, Washington.