International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 5– No.4, March 2013 – www.ijais.org

Performance Study on Rule-based Classification Techniques across Multiple Database Relations

M. Thangaraj, Ph.D.
Associate Professor
Madurai Kamaraj University,Madurai
Tamil Nadu,India

C.R.Vijayalakshmi
Research Scholar
Madurai Kamaraj University,Madurai
Tamil Nadu,India

ABSTRACT
Classification is an important task in data mining and machine learning which has been studied extensively and has a wide range of applications. There are many classification problem occurs and need to be solved. There are different types of classification algorithms like tree-based, rule-based etc, are widely used. In this paper, a performance comparison of different rule-based classifiers across multiple database relations is presented. Empirical study on both real world and synthetic databases shows their efficiency and accuracy.

General Terms
Classification, Rule-based classifier, Decision Tree.

Keywords
Multi-relational classification, RIPPER, RIDOR, PART, Tuple ID propagation.

1. INTRODUCTION
Most real world applications such as marketing surveys, medical records, and inventory management contains structured data are stored in multiple relations. This leads to the evolution of multi-relational data mining (MRDM). Multi-relational data mining learns the interesting patterns directly from multiple interrelated tables with the support of primary key /foreign keys. Multi-relational classification (MRC) is one of the rapidly rising subfields of multi relational data mining which constructs a classification model that utilizes information gathered in several relations.

Numerous approaches are available for classification, such as support vector machine and neural networks. But, they are appropriate for single flat relations. In general, there are two techniques are available for MRC. i) Exploit traditional data mining methods, known as propositionalization, which alter multiple relational data into a single flat data relation by means of physical joins and aggregations. As a consequence, some crucial information taken by the links is lost and produces an enormously fat table with huge number of additional attributes and plentiful missing values [13]. ii) We can revise the existing classification algorithm to handle the data in the multiple tables [15], [6], [4], [18].

Research Road Map:
In this paper, we apply four well known rule-based classification techniques namely Decision tree, PART, RIPPER and RIDOR based on tuple-id propagation techniques [23], [24], [25]. The objective of this paper is to compare the performance of four different rule based classifiers across multiple database relations using tuple-id propagation technique based on the following criteria: number of tuples, number of relations, number of foreign-keys and classification accuracy and runtime. The research road map is shown in Figure 1. To implement these algorithms, the whole work is divided into three phases:

- **Class propagation** – Class propagation element propagate vital information from the target relation to the background relations based on the foreign key links using tuple id propagation technique. In this way, each resulting relation contains efficient and various information which then enables a propositional learner to efficiently learn the target concept.
- **Rule Generation** - Classification algorithm builds the classifier by learning from a training set made up of database tuples and their associated class labels. In this step, the learning model is represented as a set of If-then rules.
- **Classification and Result analysis** – Next, the model is used for classification and a test data are used to estimate the accuracy of the model and then results are analyzed.

The rest of the paper is organized as follows. Section 2 review the related work and section 3 presents the background details. We describe the algorithms in section 4 and the empirical results are presented in section 5. Finally, we conclude this study in section 6.

2. RELATED WORK
Evaluation of classification techniques have been the focal point of several previous studies. In [16], performance evaluation is done for rule based classifiers namely RIPPER, RIDOR and PART for the analysis of genetic association studies at three different levels such as Models level, Rules levels and Attributes level. RIPPER algorithm outperforms

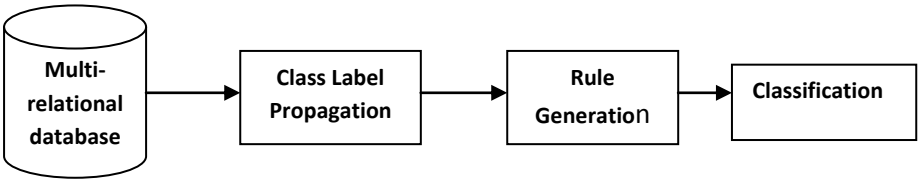

Fig. 1. Research Road Map

well than other two rule based classifiers at all levels. PART and RIDOR algorithms performed poorly at model detection and performed well at rules level. At attributes level, RIDOR gives satisfactory performance and PART gives poor performance. An evaluation of different classifiers such as Bayesian Network, Radial Basis Function (RBF) using WEKA for breast cancer based on kappa statistics, Mean absolute error and Root mean squared error is given in [12]. Among these classifiers, Bayesian Network classifier has the lowest average error than others and it is well suited for use in medical or bioinformatics field.

The paper [14] gives an assessment of rule-based classifiers for Iris data set from UCI machine learning repository using an open source machine learning tool WEKA. The classification accuracy, mean absolute error and root mean squared error are calculated for each machine learning algorithm. Among the nine classifiers, Non-Nested Generalized Exemplars (NNGE) performs well in the classification problem. OneR classifier, RIDOR Classifier and JRIP classifier are coming in the next category to classify the data. In [10], comparison of classifiers such as FOIL (First Order Inductive Learner), PRM (Predictive Rule Mining), Classification based on predictive association (CPAR) with Predictive association rule mining is given. CPAR and PRM generate more rules than FOIL. CPAR is more competent than PRM because much recurring calculations are avoided and multiple literals can be chosen to generate multiple rules at the same time. The study of classification techniques namely fuzzy sets, Genetic algorithms for breast cancer diagnosis and prognosis is given in [20]. The prognostic problem is mostly examined under Artificial Neural Networks and its accuracy is higher than other techniques. A comparative study of different classification techniques such as Support Vector Machine (SVM), Artificial Neural Networks and decision tree for cardiovascular disease prediction is given in [17**]**. These techniques are evaluated based on Sensitivity, Specificity, Accuracy, Error Rate, True Positive Rate and False Positive Rate and Support Vector Machine model is turned out to be best for cardiovascular disease prediction. All the above work focuses the data stored in single table.

3. BACKGROUND

Under this section we will present the following tuple id propagation technique for providing realistic joins among the tables in multi-relational environment and propositional rule-based classification techniques.

3.1 Tuple ID propagation technique

Tuple id propagation is a technique used to connect target tables with non-target tables using primary key/ foreign key relationship. They are joined by adding two more columns such as ID of the target tuples and its class labels to the tuples of the non-target relations. This process takes only small amount of time and space compared to the physical joins used by the existing classifiers and it will boost up the effectiveness of the multi-relational classification techniques. Consider an example multi-relational database that is shown in Table 1. The target table is student and its primary key consists of set of integers which denotes the ID of each target tuples. The physical join between these tables is shown in Table 2 and the virtual join using tuple id propagation is shown in Table 3 which is more powerful than physical join.

Table 1: Example Multi Relational Database

Student			
SId	**Cno**	**Grade**	**Class**
1	11	A	Yes
2	12	B	No
3	13	C	Yes
4	12	A	Yes
5	13	C	No

Course		
Cno	**Cname**	**Credit**
11	Java	3
12	Perl	2
13	PHP	4

Table 2: The join of student and course

SId	Cno	Grade	Cname	Credit
1	11	A	Java	3
2	12	B	Perl	2
3	13	C	PHP	4
4	12	A	Perl	2
5	13	C	PHP	4

Table 3: Result of Tuple id propagation

Cno	Cname	Credit	IDs	Class
11	Java	3	1	1+,0-
12	Perl	2	2,4	1+,1-
13	PHP	4	3,5	1+,1-

Tuple ID propagation is an effective and expandable method since little bit of information is moved between target and non-target tables that needs only small volume of additional storage area.

3.2 Propositional Rule-based Classifier

Classification is one of the popular techniques of data mining, which is used to predict group membership for the instances of data. Many approaches have been proposed for classification problem. Among them Decision tree, Bayesian Networks, SVM, Rule-based classification, associative classifications are widely used approaches.

Classification involves searching for rules that separates the data into disjoint groups. Rules are an excellent way of representing information. A rule-based classifier uses set of IF-THEN rules for classification. Rule: IF (Condition) –THEN conclusion, where Condition is a conjunction of attributes and Conclusion contains class prediction. For e.g.

R1: (age=youth) ^ (student=Yes) -> buys_computer=yes).

A rule *R1* covers an instance *x* if the attributes of the instance satisfy the condition of the rule. There are various propositional rule based algorithms are available namely OneR, PART, Conjunctive Rule, RIPPER etc. which assumes that data reside in a single table.

The classification rules are constructed in two ways - a) Direct method b) Indirect method. Direct or sequential methods are those that extract rules directly from data as RIPPER. The Fig.2 shows the pseudo code for sequential covering method. The Learn_one_Rule() function discovers the "finest" rule for the existing class, given the current set of training tuples. Indirect methods are those that extract rules from other classification model like decision trees e.g. C4.5rules. Advantages of rule-based classifications are,

- easy to understand
- easy to construct
- highly expressive as decision trees
- can classify new instances quickly

In this paper, propositional rule based classifiers such as PART, RIPPER, RIDOR, Decision tree are extended based on tuple id propagation techniques to compare the effectiveness of those classifiers in a multi relational classification environment.

```
Algorithm : Sequential Covering (D, Att-vals)
Input     : A relational database D with class-labeled
            Tuples, the set of all attributes and their
            possible values Att-vals.
Output    : A set of IF-Then rules R.
Procedure:
      R = empty set; // Initial set of rules learned is
                        empty
      For each class c do
        Repeat
          Rule = Learn_one-Rule (D, Att_vals, c)
          D = D- Rule  // remove the tuples covered
                          by Rule from D
        Until records in D = NIL;
        R = R + Rule; // add new rule to rule set
      End For
     Return R;
     End
```

Fig. 2. Pseudo code for sequential covering

4. MULTIRELATIONAL RULE-BASED CLASSIFIERS

A multi-relational database D consists of a collection of relations *R={X1, X2, ... Xn}*, and a set of relationship between pairs of relations. One of the relations will be central for analysis known as target relation attached with the class labels and other relations are treated as non-target relations. Each relation has one primary key and some foreign keys and each foreign key is linked to the primary key of other tables. As in tuple id propagation, this work also considers two types of relationships between relations: Link between a primary key *k* and some foreign key pointing to *k*, Link between two foreign keys *K1* and *K2*, which point to the identical primary key *k* and ignore the other types of links because they does not signify strong relationship. The following parameters are used in this performance study.

MIN_INFO_GAIN = 0.05 MIN_SUP = 10

X= If target > 10% X= NULL else X=ruleset

4.1 PART Classifier

PART (Partial Decision Tree) [5] is an indirect technique for constructing classification rules. It employ partial decision tree to generate the individual rules and the tree is induced with C4.5 classifier. After tree generation, rules are derived directly from the partial tree starting with the deepest leaf node in combination with every node along the path towards the root. Then, the partial decision tree is removed. The algorithm for generating partial decision tree and PARTRule algorithm for classification across multiple database **r**elations are shown in Fig.3 and Fig.4.

```
Algorithm : PARTree (D, Rt)
Input     : A relational database D with a target
            Relation Rt
Output    : A Partial decision tree N for predicting
            class labels of target tuples
Procedure :
      N = empty; \\ Tree node initially empty
      A = empty;   \\ Attribute for storing
                      max.information gain
    If |Rt| < MIN_SUP then return
    Evaluate all attributes in any active relation or
    Relations joinable with active relation using
                                  Information gain
    A = attribute with max information gain
    If (info_gain (A) < MIN_INFO_GAIN) then return
       N = A;  set relation of A to active
  Do
      Divide Rt into subsets R' according to A
      and add those nodes as children of node N
  While (there are subsets that have not been expanded
     and all the subsets expanded are so far are leaves)
  If all the subsets expanded are leaves and estimated
     Error for sub tree >= 0.5
       Undo expansions into subsets and make node a
       leaf
  For each relation R' that is set active by this function
       Set R' inactive
  Return N
  End
```

Fig 3. Multi relational PARTree algorithm

```
Algorithm : PARTRule(D, Rt)
Input     : A relational database D with a target
            Relation Rt.
Output    : A decision rules R for predicting class
            labels of target tuples
Procedure:
          R = empty set;  // Initial set of rules learned is
                             empty
          Do
            Treenode T = PARTree (D, Rt)
            Create a rule r from T
            R = R + r;
            X=X – r  // Remove all positive target tuples
                        satisfying r from X;
          While (X)
          Return R;
End
```

Fig 4. Multi relational PARTRule algorithm

4.2 RIPPER Classifier

The Repeated Incremental Pruning to Produce Error Reduction (Ripper**)** [21], [22] is straight approach for generating classification rules. It is considered to be more efficient than decision trees on large and noisy datasets. It undergoes four phases: i) Growth ii) Pruning iii) Optimization iv) Selection. In the growth phase, it produces a sequence of individual rules by adding predicates until the rule satisfies stopping criteria. The rules that reduce the performance of the algorithm are pruned in the second phase. In the optimization step, each rule is optimized by adding up attributes to the original rule or generates a new rule using phase 1 and phase 2. In the last stage, the best rules are retained and others are ignored from the model. It employs Description_Length() function to calculate the description length of the rule. The algorithm for multi-relational Ripper is shown in Fig.5.

Algorithm: **Ripper (D, P, N**)
Input : A relational database **D w**ith P positive and N negative target tuples.
Output : A set of rules R for predicting class labels of Target tuples
Procedure:
R = empty; // Intial set of rule learned is empty
p1 = empty; // For storing best predicate
r= empty; // initially rule learned is empty
DL = Description_Length(R, P, N)
Set Rt to active
While P ≠ {} // Grow and Prune a new rule
Evaluate Predicates in all active relation or relations joinable with active relation
p1 = predicate with highest foil gain
r = r + p1
set relation of p1 to active
Prune the rule using reduced error pruning
R=R + r;
L = Description_Length(R, P, N) // compute new description length L
If (L > DL+64) **then**
Remove all positive target tuples satisfying r
For each rule r in the Rule set R
If Description_Length(R -{r}, P, N) < DL then
Delete r from R
DL = Description length(R, P, N)
End if
End for
End if
Delete from P and N all examples covered by r
End while
Set all active relations into inactive
Return R
End

Fig 5. Multi relational RIPPER algorithm

4.3 RIDOR Classifier

Ripple Down Rule learner (RIDOR) [9], [8], [1], [2], [3**]** is also a direct classification method. First and foremost it constructs the default rule and then produces the exceptions for the default rule with lowest error rate. For every exception, the most excellent exceptions are created thereby produces the tree-like expansion of exceptions. The exceptions represent a set of rules that foretell classes other than the default. Incremental Reduced Error Pruning IREP [7] is used to create the exceptions. The RIDOR classification algorithm for multiple database relations is shown in Fig.6.

Algorithm : **Ridor (D, Rt)**
Input : A relational database D with target Relation Rt that contains P positive and N negative tuples
Output : A set of rules for predicting class labels of target tuples
Procedure:
Rule set R = empty set;
if |Rt| < MIN_SUP **then** return
Rule r =empty rule
Set R_t active
Repeat
Find a rule r in active relation or relations joinable with active relation
Learn except branch and if not branch
Set relation of r to active
R= R + r
X= X- r // Remove all positive target tuples satisfying r from X;
Until (X=NULL)
Set all active relations into inactive
Return R;
End

Fig 6. Multi relational Ridor algorithm

4.4 Decision Tree

A decision tree [19], [11] has flow-chart like tree structure and is generated by a recursive divide-conquer algorithm. In this tree, every internal node indicates a test on an attributes and each branch specifies result of the test and each leaf node contains related target class. The multi-relational decision tree starts at the root node, which denotes the tuples in the target relations and set that relation to active. At every step, it will look for most optimal attribute recursively in active or unactive relations to split the target tuples into several portions with the help of Find_Gainmax () function and propagate the IDs. The function takes relational database **D** along with target table **Rt** as input. Then it will prevent the process on one branch if there is not adequate tuples or no attribute with satisfactory information gain can be found. The pseudo code for multi relational decision tree algorithm is shown in Fig.7. The algorithm for finding the optimal attribute is shown in Fig. 8.

5. EMPIRICAL RESULTS AND OBSERVATIONS

The classifiers we have chosen are –RIPPER, PART, Decision tree and RIDOR. To compare the performance of these rules based classifiers using tuple-id propagation techniques, the following criteria are used:

- Classification accuracy
- Number of relations
- Number of attributes
- Number of tuples
- Number of foreign-keys

- Run time.

All experiments were performed on HP Pavilion dv6 laptop with Intel® Core™ i3 CPU and 2 GB RAM running on Windows XP Professional. In each database, a 10-fold cross-validation is used.

```
Algorithm: DTree (D, Rt)
Input    : A relational database D with target
           relation Rt
Output   : A decision tree N for predicting class
           labels of target tuples
Procedure:
    Bestattribute A = empty;
    If |Rt| < MIN_SUP then return
    Treenode N = empty;
    A = Optimal_refinement (D,Rt);
     N=A
    Set relation of A to active
    Partition Rt into Rt'according to R
    For each partition Rt'
   Treenode Ni = DTtree (D, Rt')
   Add Ni as a child of N
   Set all active relations into inactive
  Return N
End
```

Fig 7. Multi relational Tree algorithm

```
Algorithm: Optimal_ refinement(D, Rt)
Input    : A relational database D with a target
           Relation Rt, and the attributes.
Output   : The optimal attribute Amax with highest
           Information gain.
Procedure:
    Set Rt to active;
    For each active relation P,
    Amax: = Find_Gainmax (D, Rt);
    End For
   For each unactive relation R
      For each key/foreign_key k of R
        if R can be joined to some active relation P
        with P.k then
        Propagate IDs from P to R;
        Amax1: = Find_Gainmax (D, Rt);
        End if
      End For
     For each foreign key k'≠k of R
       propagate IDs from R to relation P that is
        pointed to by R.k
       Amax2: = Find_Gainmax (D,Rt);
     End For
  End For
  If (Amax> Amax1) and ( Amax >Amax2) then
   break;
    Else If Amax1> Amax2 then  Amax= Amax1
        Else Amax= Amax2
        End If
 End If
 Return Amax
End
```

Fig 8. Optimal refinement Algorithm

5.1. Synthetic Multi-relational Database

To carry out an experimental analysis, a group of synthetic relational databases is generated, which imitate the real relational databases. Our data generator gets the parameters in Table 4 as almost similar in CrossMine. Keeping other parameters stable, we change three parameters when generating each database, which are the number of relations, the expected number of tuples in each relation and the expected number of foreign keys in each relation. We utilize *Rx.Ty.Fz* to denote a database with *x* relations, *y* expected tuples in each relation and z expected foreign keys in each relation. In each of the subsequent experiments, the runtime and accuracy of these algorithms are compared.

Table 4: The parameters of synthetic multi-relational database generator

Name	Descriptions	Value
$\|R\|$	# relations	*X*
Tmin	Min # tuples in each relation	50
T	Expected # tuples in each relation	*Y*
Amin	Min # attributes in each relation	2
A	Expected # attributes in each relation	5
Vmin	Min # values in each relation	2
V	Expected # values in each relation	10
Fmin	Min # foreign-keys in each relation	2
F	Expected # foreign-keys in each relation	*Z*
$\|r\|$	# rules	10
Lmin	Min # complex predicates in each rule	4
max	Max # complex predicates in each rule	6

To assess the scalability with respect to number of relations, we create four databases with 10, 15, 20 and 25 relations correspondingly. In each relation, the expected number of tuples is 1000 and the expected number of foreign keys is 2. The runtime and accuracy of these four methods are shown in Fig.9 and 10.

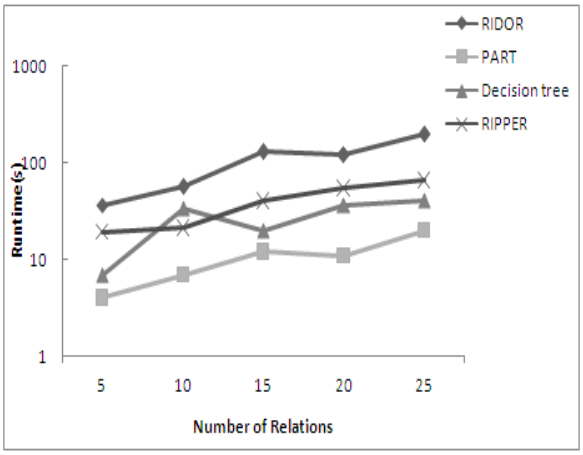

Fig 9: Runtime on R*.T1000.F2

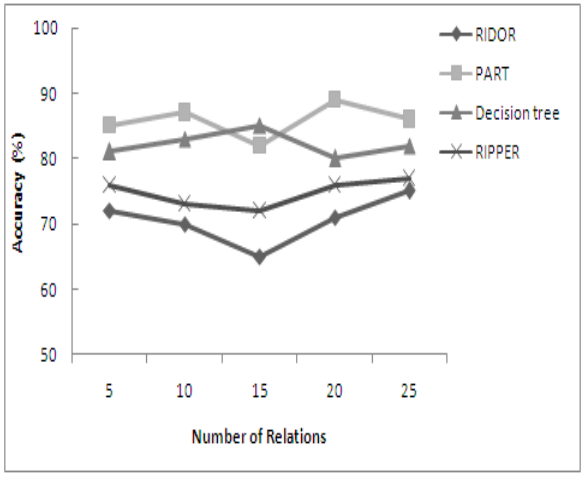

Fig 10: Accuracy on R*.T1000.F2

To estimate the scalability with regard to number of tuples, we generate five databases with 1000, 3000, 5000, 7000 10000 expected tuples per relation respectively. For all of these databases, the number of relation is 10 and the expected number of foreign keys is 2. The runtime and accuracy are shown in Fig. 11 and 12.

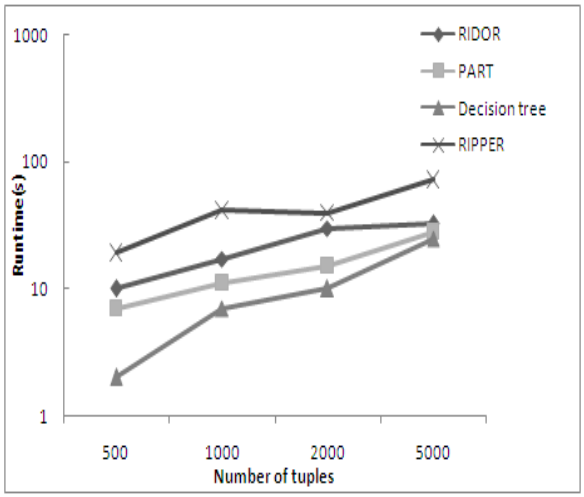

Fig 11: Runtime on R10.T*.F2

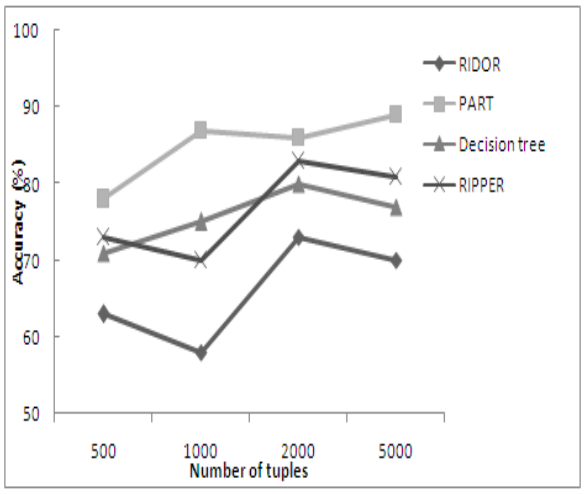

Fig 12: Accuracy on R10.T*.F2

To test the scalability with respect to the number of foreign keys per relation, three databases are created with 1, 2, and 3 expected foreign keys per relation. In each of these databases, the expected number of tuples is 1000 and number of relation is 10. The Fig. 13 and 14 shows their runtime and accuracy.

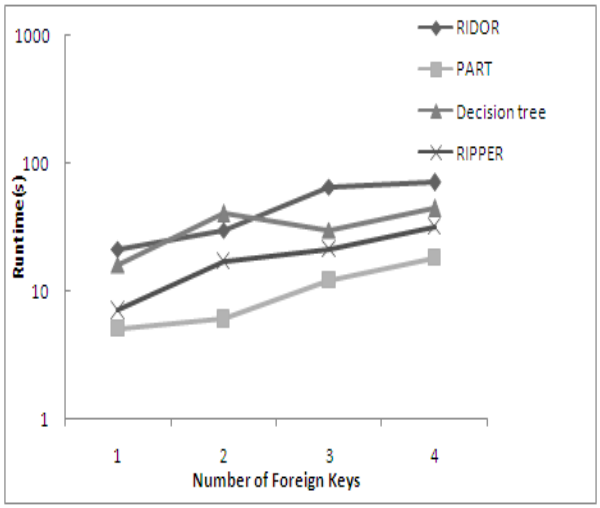

Fig 13: Runtime on R10.T1000.F*

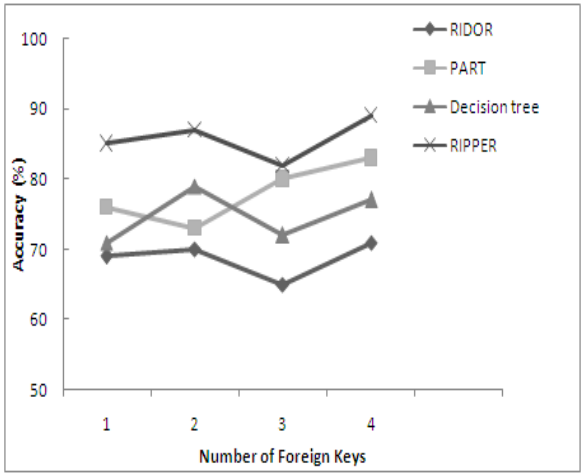

Fig 14: Accuracy on R10.T1000.F*

6. CONCLUSIONS

In this paper, four well-known rule based classification techniques namely Decision tree, RIPPER, PART, and RIDOR are applied on three real multi-relational databases. From the experimental results of this paper, we make a comparative study of these algorithms and their applicability on these databases.

- If the dataset has number of relations more than 20 then PART and Decision tree algorithms performs better than the others.
- If the dataset has number of tuples more than 25,000 then PART and RIPPER make well.
- If the dataset has number of attributes more than 25 then PART, RIPPER performs well.

- If the dataset has number of foreign keys more than 2 then Decision Tree, PART seems to be healthy.

The overall position is done based on the number of relations, number of tuples, number of attributes, number of foreign keys, classification accuracy and runtime. Based on the experimental results, PART Classifier appears to be superior to Decision tree, RIPPER and RIDOR.

7. REFERENCES

[1] Catlett, 1992. Ripple-Down-Rules as a mediating representation in interactive induction. In Proceedings of the Second Japanese Knowledge Acquisition for Knowledge-Based Systems Workshop, Kobe, Japan.

[2] Compton, P., Edwards, G., et al., Ripple down rules: turning knowledge acquisition into knowledge maintenance, Artificial Intelligence in Medicine **4**: 47-59.

[3] Compton, P., Edwards, G., Kang, B., Lazarus, L., Malor, R. 1991. Ripple down rules: Possibilities and limitations.

[4] Emde ,W., Wettschereck D. Relational Instance –Based Learning. In L.Saitta, editor, Proceedings 13th International Conference on Machine Learning, pages 122-130. Morgan Kaufmann, 1996.

[5] Frank, E., Witten, I., J.1998. Generating Accurate Rule Sets without Global Optimization, Machine Learning: Proceedings of the Fifteenth International Conference, pp.144-151, Madison, Wisconsin, Morgan Kaufmann, San Francisco.

[6] Friedman N., Getoor L., Koller D., and Pfeffer A. 1999. Learning Probabilistic Relational Models. In Sixteenth International Joint Conference on Artificial Intelligence (IJCAI).

[7] Furnkranz, J., Widmer, G. 1994. Incremental Reduced Error Pruning. In Machine Learning: Proceedings of the 11th Annual Conference, New Brunswick, New Jersey, Morgan Kaufmann.

[8] Gaines, B.R., Compton, P.J. 1992. Induction of Ripple Down Rules, AI Proceedings of the 5th Australian Joint Conference on Artificial Intelligence, Hobart, Australia, World Scientific, Singapore.

[9] Gaines, B.R., Paul Compton, J. 1995. Induction of Ripple-Down Rules Applied to Modeling Large Databases, Intell. Inf. Syst. 5(3):211-228.

[10] Gupta, P., Toshniwal, D., J. 2011. Performance Comparison of Rule Based Classification Algorithms, Computer Science & Informatics, Volume-I, Issue-II.

[11] Han, J., Kamber, M. 2001. Data Mining: Concepts and Techniques", Morgan Kaufmann.

[12] Ibrahim, F., Abu Osman, N.A., and Usman, Kadri, N.A. 2007. Comparison of Different Classification Techniques Using WEKA for Breast Cancer, (Eds.): Biomed 06, IFMBE Proceedings 15, pp. 520-523, Springer-Verlag Berlin Heidelberg.

[13] Kramer, S., N. Lavrac and P. Flach. 2001. Propositionalization approaches to relational data mining. In S.Dzeroski and N.Lavrac, eds. Relational Data Mining. Pp 262-291, Springer-Verlag.

[14] Lakshmi Devasena, J. 2011. Effectiveness Evaluation of Rule Based Classifiers for the Classification of Iris Data Set, Bonfring International Journal of Man Machine Interface, Vol. 1, Special Issue, December.

[15] Lavrac N. and Dzeroski S. 1994. Inductive Programming: Techniques and Applications. Ellis Hoodwood.

[16] Lehr, T., Yuan, J., and Zuemer, D., Jeyadev, S. J. 2011. Rule based classifier for the analysis of gene-gene and gene-environment interactions in genetic association studies, BioData Mining 4:4.

[17] Milan Kumari, Sunila Godara, J. 2011. Comparative Study of Data Mining Classification Methods in Cardiovascular Disease Prediction, Computer Science and Technology, Vol. 2, Issue 2, June.

[18] Neville J., Jensen D., Friedland L., and Hay M., 2003. Learning Relational Probability Trees. Technical Report 02-25, Dept. of Computer Science, University of Massachusetts Amherst, 2002, Revised version February.

[19] Quinlan, R., 1993. C4.5: Programs for Machine Learning San Mateo, CA: Morgan Kaufmann Publishers.

[20] Shelly Gupta, Kumar, D., J. 2011. Data Mining Classification techniques applied for breast cancer diagnosis and prognosis, Indian Journal of Computer Science and Engineering, Vol. 2 No. 2 Apr-May.

[21] William, Cohen, 1995. Fast Effective Rule Induction, In: Twelfth International Conference on Machine Learning, 115-123.

[22] Yang, J, . Tiyyagura, et al, 2001. Feature Subset selection for Rule Induction Using RIPPER.

[23] Yin, X. Han, Yu PS, J. 2006. Efficient Classification across Multiple Database Relations: A CrossMine Approach, *I*EEE Transactions on Knowledge and Data Engineering, Vol 16, No.6.

[24] Yin, X., Han, J., Yang, J. 2003. Efficient Multi-relational Classification by Tuple ID Propagation, In Proceedings of KDD workshop on MRDM.

[25] Yin, X., Han, J., Yu PS, 2004. CrossMine: Efficient Classification across Multiple Database Relations. In Proceedings of 20th Int. Conf. on Data Engineering (ICDE'04).

Muthuraman Thangaraj received his post-graduate degree in computer science from Alagappa University, Karaikudi, M.Tech degree in Computer Science from Pondicherry University and Ph.D degree in Computer Science from Madurai Kamaraj University Madurai, TN, South India in 2006. He is now the Associate Professor of Computer Science Department at M.K. University. He is an active researcher in Web mining, Semantic Web and Information Retrieval and has published more than 35 papers in Journals and Conference Proceedings.

C.R.Vijayalakshmi received MCA degree from Cauvery College Trichy, TN and M.Phil in Computer Science from Madurai Kamaraj University, Madurai in 2006. She is currently a part time Research scholar in Computer Science, Madurai Kamaraj University in the area of data mining.