An Approach to Automation Selection of Decision Tree based on Training Data Set

D.Saravana Kumar
Asst Professor (SG) in MCA
SVS Institute of Computer
Applications, Coimbatore-109
Tamil Nadu, India

N.Ananthi
Asst Professor in Computer
Science, Dr.N.G.P. Arts and
Science College
Coimbatore-48, TN, India

M.Devi
Research Scholar
Dept. Of Computer Science
Dr.N.G.P. Arts and Science
College, Coimbatore, TN, India

ABSTRACT

In Data mining applications, very large training data sets with several million records are common. Decision trees are very much powerful and excellent technique for both classification and prediction problems. Many decision tree construction algorithms have been proposed to develop and handle large or small training data. Some related algorithms are best for large data sets and some for small data sets. Each algorithm works best for its own criteria. The decision tree algorithms classify categorical and continuous attributes very well but it handles efficiently only a smaller data set. It consumes more time for large datasets. Supervised Learning In Quest (SLIQ) and Scalable Parallelizable Induction of Decision Tree (SPRINT) handles very large datasets. But SLIQ requires that the class labels should be available in main memory beforehand. SPRINT is best suited for large data sets and it removes all these memory restrictions.

The research work deals with the automatic selection of decision tree algorithm based on training dataset size. This proposed system first prepares the training dataset size using the mathematical measure. The result training set size problem will be checked with the available memory space. If memory is very sufficient then the tree construction will continue. After the classifying the data, the accuracy of the classifier data set is estimated. The main advantages of the proposed method are that the system takes less time and avoids memory problem.

Keywords

Decision Tree Algorithm, Classification, Data Mining, Data Set.

1. INTRODUCTION

Classification has been identified as an important problem in the emerging field of data mining [1], [2]. While classification is a well-studied problem, [3] only recently has there been focus on algorithms that can handle large databases. Several classification models have been proposed over the years, e.g. neural networks [4], [5] statistical models like linear/quadratic Discriminator, decision trees [3] and genetic models. Among these models, decision trees are particularly suited for data mining [6]. Decision trees can be constructed to relatively very fast compare to other Algorithms, Another advantage is also available decision tree models are simple and are easy to understand [7]. Moreover, trees can be easily converted into SQL statements that can be used to access databases efficiently [8]. Also, decision tree classifiers obtain similar and sometimes better accuracy when compared with other classification methods.

A decision tree [9], [10] is a classification method that can be used to divide up a large collection of records into successively smaller sets of records by applying algorithms. There are so many algorithms for automatically generating decision trees. The set of records are generally divided into two disjoint subsets – a training set and a test data. The former is used to deriving the classifier methods, while the latter is used to measure the accuracy of the classifier. The input called training set has a set of example records, where each record consists of several fields or attributes. Attributes are continuous, coming from an ordered the domain, or the categorical methods, coming from an unordered method or domain. One of the main attributes are called the classifying attribute, indicates which the class to change the value belongs. The objective of decision tree is to build a model of the classifying attribute are based on the other main attributes. The accuracy of the classifier is determined by the percentage of the test examples that are correctly classified.

1.1.1 Basic Algorithm for Decision Tree

Decision tree method classifiers developed classification in two phases a) Tree Building b) Tree Pruning

The various decision tree construction algorithms available are

i) ID3 (Iterative Dichotomizer 3)
ii) C4.5CHAID (Chi-squared Automatic Interaction Detection)
iii) SLIQ (Supervised Learning in Quest)
iv) SPRINT (Scalable Parallelizable Induction of Decision Tree)
v) Rainforest
vi) CLOUDS(Classification of Large or Out-of core Data Sets)
vii) BOAT(Bootstrap Optimistic Algorithm for Tree Construction)
viii) PUBLIC(Pruning and Building Integrated in Classification)

The tree built in the first phase completely classifies the training set. This implies that the branches are created in the tree even for spurious “noise” data and statistical fluctuations [11]. These branches can lead to errors when classifying training test data. Tree pruning is aimed to removing these branches from the decision tree by selecting the sub tree with the least estimated error rate.

2. CLASSIFIER ACCURACY

Estimating classifier accuracy [12] is important that it allows one to evaluate how accurately a given classifier will label the future data, that is, data on which the classifier has not been trained. Performance metric accuracy is defined as follows, **Accuracy = Number of correct predictions / Total number of predictions** The performance of a model can be expressed in terms of its error rate is given by the following formula, **Error rate = Number of wrong predictions / Total number of predictions**

3. ALGORITHMS USED

The decision tree algorithms C4.5, SLIQ, SPRINT are taken for the research work because of its advantages comparing to other algorithms. C4.5 [9] is an incremental version of ID3; it handles both numerical and categorical attributes. It works efficiently for only small datasets [13]. The well-known C4.5 classifier grows trees depth first and repeatedly sorts the data at every node of the tree to arrive at the best splits for numeric attributes.

SLIQ [11] on the other hand, replaces this repeated sorting with one-time sort by using separate lists for each attribute and also handles very large training sets. SLIQ uses a data structure called a class list which is memory resident at all times. The size of this structure is proportional to the number of input records; this limits the amount of data that can be classified by SLIQ.

SPRINT is a decision-tree classifier for data mining. It is able to handle large disk-resident training sets, with no restrictions on training-set size, and is easily parallelizable. One list is maintained for each attribute in the dataset. The SPRINT removes all memory restrictions.

The size of the training set is checked with the available memory before choosing an algorithm. If the size of the training set size is less than 25 megabytes then the system chooses C4.5 algorithm to classify the dataset. If the training set size is above 25 megabytes then the automated system selects SLIQ algorithm. If the size of the training set size is above 50 megabytes then the system chooses SPRINT algorithm to generate the tree.

Finally, the accuracy of the classifier based on the measures like time taken for classifying the dataset, accuracy and error rate of the classifier are estimated.

4. DATABASE USED

To test the proposed work a large amount of database is necessary, i.e., larger in both depth and width. This framework uses the U.S. Bureau Census data set which is publicly available from U.C. Irvine repository. The Census-income dataset is a multivariate data from the web site [14] .This data set contains weighted census data extracted from the 1994 and 1995 current population surveys conducted by the U.S. Census Bureau. The data contains 40 demographic and employment related variable details.

Age, Class of worker, Detailed industry recode, Detailed occupation recode, Education, Wage per hour, Enroll in edu inst last wk, Marital stat, Major industry code, Major occupation code based on , Race, Member of a main labor union, The main reason for unemployment, Full or part time employment stat etc. The census database has 1, 99,523 instances (records) and 42 attributes. Attributes are of both categorical and numerical.

The splitting point for categorical attributes is different. If S (A) is the set of possible values of the categorical attribute A, then the split test is of the form AЄ S’ where S’ЄS. For an attribute with n values, there are 2n possible splits. If n is small, the split index value is found for all the possible combinations and the best split is taken. If n is large, then the split is made by some heuristics and the best split among them is found.

The construction of an attribute list is similar to that of numerical attributes. But instead of having a class histogram, a count matrix is maintained for the categorical attribute. The count matrix has n rows (for n distinct values of the attribute) and k columns (for k classes). Each entry, say (I, $_{j}$) entry, represents the number of records in the data set having i^{th} value of the attribute and in the j^{th} class.

4.1.1 MDL pruning algorithm is used for pruning the tree.

Pseudocode

```
for each continuous feature $A_j$
    do sort $AL_j$
while( з a  mixed leaf) do
    for each feature Aj do
        for each mixed leaf V do
            determine Q(v,Aj)
            if(Q(v,Aj)) better than previous best) then update Q(V)
    for each feature Aj do
        for each mixed leaf V such that Q(V) = Q(V,Aj) do
            split  v using Q(v,Aj)
    for each feature Aj do
        for each mixed leaf V such that Q(V) = Q(V,Aj) do
            split  v using Q(v,Aj)
```

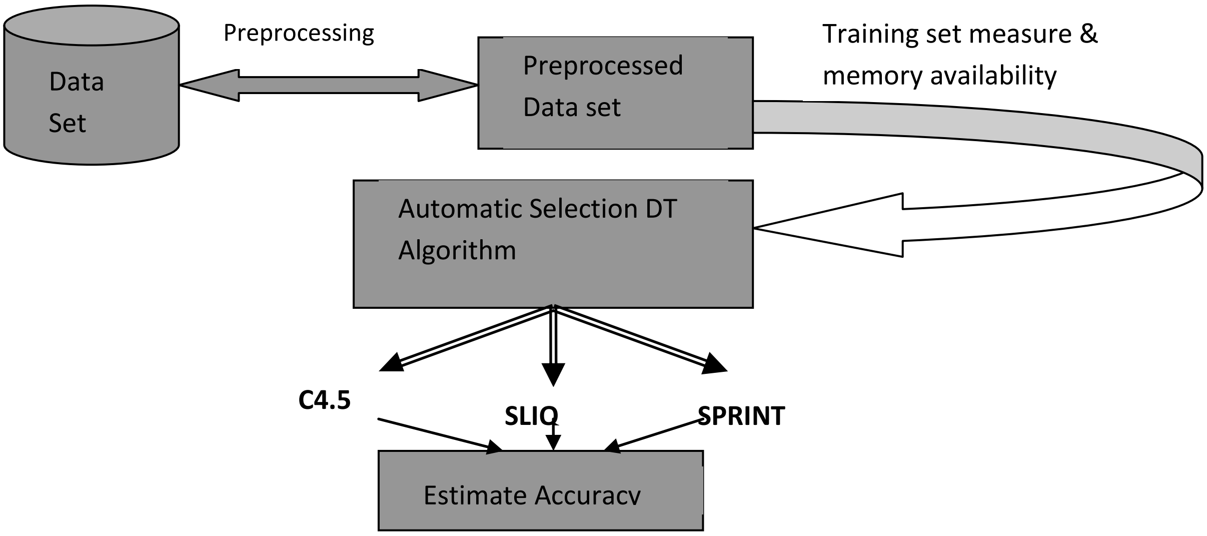

Fig 1: Proposed System Architecture

5. SPRINT CLASSIFIER

Pseudocode

```
for each continuous feature Aj do sort ALj,root
while ( ∃ a  mixed leaf) do
for each mixed leaf v do
for each feature Aj do

determine Q(v,Aj)
if (Q(v,Aj) better than previous best) then update Q(v)
split v into v1 and v2 using Q(v)
for each feature Aj do split ALj,v into ALj,v1 and ALj,v2
```

SPRINT, a decision-tree-based classification algorithm, removes all of the memory restrictions, and is fast and scalable. It is similar to SLIQ except its data structure.
Attribute lists

SPRINT initially creates an attribute list for each attribute in the data entries in these lists, which we call attribute records are an attribute value, label, and the index of the record (rid) from which these value were obtained. Initial lists for continuous attributes are sorted by attribute value once when first created. If the entire data does not fit in memory, attribute lists are maintained on disk. The initial lists created from the training set are associated with the root of the classification tree. As the tree is grown and nodes are split to create new children, the attribute lists belonging to each node are partitioned and associated with the children. When a list is partitioned, the order of therecords in the list is preserved; thus, partitioned lists never require resorting.

6. EXPERIMENTAL RESULT

The proposed system is successfully developed and tested with the census dataset. Three algorithms C4.5, SLIQ and SPRINT are chosen for this research work. Each algorithm works best for its own criteria. The end user is not aware of the dependent criteria for each algorithm and also the memory requirements for each. The user does not know which algorithm takes less time and gives more accuracy. So the user repeatedly runs the three algorithms. From the results, the user knows that the SPRINT algorithm takes less time for large (consists of 1,99,523 records and 42 attributes) dataset and C4.5 is best for small (consists of 12 records and 42 attributes) dataset. The manual results are shown in the tabular column as follows

Algorithm	Time Taken (Seconds)	Accuracy	Error Rate
1. C4.5	19.52	92.90	7.10
2. SLIQ	19.27	93.44	6.56
3. SPRINT	18.92	93.44	6.56

Fig 2: Results

The above result shows that C4.5 takes more time to classify large number of records comparing to SLIQ. From the screen results, it can be studied that the accuracy decreases when C4.5 classifies the large number of records.

7. RESULT ANALYSIS

In automatic method, the algorithm selection is done by the system automatically based on the training set size. The training dataset used here has size more than 50 MB which consists of 1, 99,523 records with 42 attributes. So the system automatically chooses the SPRINT algorithm and thus the time is saved much.

8. CONCLUSION

Currently, classification using decision trees is one of the most active research areas in data mining research. The amount of data creates a need for the development of scalable and efficient data mining algorithms. Most of the methods available at present do not take into account the challenges posed by large amounts of data. This thesis addressed automated selection of decision tree algorithm based on the training set size considers the dataset size as well as the memory available.

9. FUTURE DIRECTIONS

The future work provides the following issues for preceding this research work.

1. The proposed approach checked with only numerical and categorical attributes. In future it can be extended to handle real valued attributes like floating point temperature.
2 Existing decision tree algorithms finds the root node using entropy, gini index. Some other optimal measures will be used for finding the root node in further work. Decision tree methods can also be easily extended to learning functions with more than two possible output values.

10. ACKNOWLEDGMENT

First author thank the college management for their continues support and Encouragement.

11. REFERENCES

[1] Amir Bar-Or, Daniel Keren, Assaf Schuster, and Ran Wolff, "Hierarchical Decision Tree Induction in Distributed Genomic Databases", IEEE Transactions on Knowledge and Data Engineering, vol.17, No.8, August 2005 .

[2] Arun K Pujari, "Data Mining Techniques", Universities Press, 2001

[3] Banerjee M., and Chakraborty M.K., "Rough Logics: A survey with further directions," Rough Sets Analysis, Physica Verlag, Heidelberg, 1997.

[4] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, "Classification and Regression Trees. Wadsworth, Belmont", 1984.

[5] J. Bala, J. Huang and H. Vafaie K. DeJong and H. Wechsler "Hybrid Learning Using Genetic Algorithms and Decision Trees for Pattern Classification", 2003.

[6] Carla E. Brodley Paul E. Utgoff, "Multivariate versus Univariate Decision Trees", COINS Technical Report 92-8, Jan 1992

[7] Andrew B. Nobel, "Analysis of a complexity based pruning scheme for classification trees", IEEE Transactions on Information Theory, vol. 48, pp.2362-2368, 2002.

[8] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami, "Database mining: A performance perspective. IEEE Transactions on Knowledge and Data Engineering", 5(6):914{925, December 1993.

[9] Donato Malerba, Floriana Esposito and Giovanni Semeraro , "A Further Comparision of Simplification Methods for Decision –Tree Induction" , Springer-verlag, 1996.

[10] Floriana Esposito, Donato Malerba, and Giovanni Semeraro "A Comparative Analysis of Methods for Pruning Decision Trees" , IEEE Transactions on pattern analysis and machine intelligence, vol.19,No.5, May 1997

[11] Johannes Gehrke, Raghu Ramakrishnan, Venkatesh Ganti_, "RainForest - A Framework for Fast Decision Tree Construction of Large Datasets", Proceedings of the 24th VLDB Conference New York, USA, 1998.

[12] V. Corruble D.E. Brown and C.L. Pittard, "A comparison of decision classifiers with back propagation neural networks for multimodal classification problems", *Pattern Recognition*, 26:953–961, 1993.

[13] Deborah R. Carvalho, Alex A. Freitas , "A hybrid decision tree/genetic algorithm for coping with the problem of small disjuncts in data mining" 2004.

[14] Haixun Wang, Carlo Zaniolo "CMP: A Fast Decision Tree Classifier Using Multivariate Predictions", *University of* D. Hand, H. Mannila, P. Smyth," Principles of Data Mining", MIT Press, Cambridge, MA, 2001.