An Efficient Mining Procedure For Gene Selection By Using Tree Algorithms

S.ANUSUYA*1, R.KARTHIKEYAN*2

*1 *II. m. tech student*
Department of CSE
Bharath University
Chennai, India.
anusuya.es@gmail.com

*2 *Assistant Professor*
Department of CSE
Bharath University
Chennai, India.
rkarthikeyan78@yahoo.in

ABSTRACT

Tree induction methods and linear regression are popular techniques for supervised learning tasks, both for the prediction of discrete classes and numeric quantities. The two schemes have somewhat complementary properties. The simple linear models fit by regression exhibit high bias and low variance, while tree induction fits more complex models which results in lower bias but higher variance. For predicting numeric quantities, there has been work on combining these two schemes into model trees, i.e. trees that contain linear regression functions at the leaves .This algorithm that adapts this idea for classification problems. For solving classification tasks in statistics, the analogue to linear regression is linear logistic regression, so our method builds classification trees with linear logistic regression functions at the leaves. A stagewise fitting process allows the different logististic regression functions in the tree to be fit by incremental refinement using the recently proposed LogitBoost algorithm , and we show how this approach can be used to automatically select the most relevant attributes to be included in the logistic models.

Keywords: Trees, Leaves, DNA microarray.

I .INTRODUCTION

Machine learning provides tools that automatically analyze large datasets, looking for informative patterns, or use accumulated data to improve decision making in certain domains. Recent advances in computer hardware and software have led to a large increase in the amount of data that is routinely stored electronically, and this has made techniques of machine learning both more important and more feasible. There has been an impressive array of successful applications of machine learning: programs that learn from data are used to make or support decisions in medical domains, recognize spoken words, detect fraudulent use of credit cards, or classify astronomical structures. Often, the learning task can be stated as a classification problem: given some observations about a specific object, the goal is to associate it with one of a predefined set of classes. For example, a patient could be diagnosed as having a particular disease or not depending on the outcome of a number of medical tests. Solving classification tasks is an important problem in machine learning and has received a lot of attention.

Where is gene located?

A cell is made up of chromosomes. 23 pairs of chromosomes present in a single cell. The Chromosome is made up of genes. The genes consist of DNA which is made up of four chemical letters .i.e A, C, T, G. In a cell both DNA and RNA are present. Both did a replication process.DNA was convert into RNA is called transcription process. RNA was convert into DNA is called reverse transcription process. Protein was obtained from the RNA with the help of translation process. The advantages of gene mining is, it is used in clonning method,pregnancy cases,agriculture. The disadvantage of gene mining is,high tech,costly. In this paper large number of attributes are present,so using the select attributes and then classify with various algorithm it produce different output for various algorithms.compare all output and find which one is low output, it is the final accuracy. Furthermore, using the selected genes on the cancer

classification, the robust classification accuracy has been produced by some different classification methods

II. METHODS

J48 algorithm:

A decision tree is a predictive machine-learning model that decides the target value of a new sample based on various attribute values of the available data. The internal nodes of a decision tree denote the different attributes , the branches between the nodes tell us the possible values that these attributes can have in the observed samples, while the terminal nodes tell classification of the dependent variable. The attribute that is to be predicted is known as the dependent variable, since its value depends upon, or is decided by, the values of all the other attributes. The other attributes, which help in predicting the value of the dependent variable, are known as the independent variables in the dataset.

The J48 Decision tree classifier follows the following simple algorithm. In order to classify a new item, it first needs to create a decision tree based on the attribute values of the available training data. So, whenever it encounters a set of items training set it identifies the attribute that discriminates the various instances most clearly. This feature that is able to tell us most about the data instances so that we can classify them the best is said to have the highest information gain. Now, among the possible values of this feature, if there is any value for which there is no ambiguity, that is, for which the data instances falling within its category have the same value for the target variable, then we terminate that branch and assign to it the target value that we have obtained.

For the other cases, we then look for another attribute that gives us the highest information gain. Hence we continue in this manner until we either get a clear decision of what combination of attributes gives us a particular target value, or we run out of attributes. In the event that we run out of attributes, or if we cannot get an unambiguous result from the available information, we assign this branch a target value that the majority of the items under this branch possess. Now that we have the decision tree, we follow the order of attribute selection as we have obtained for the tree. By checking all the respective attributes and their values with those seen in the decision tree model, we can assign or predict the target value of this new instance. The above description will be more clear and easier to understand with the help of an example. Hence, let us see an example of J48 decision tree classification.

LMT algorithm:

A logistic model tree basically consists of a standard decision tree structure with logistic regression functions at the leaves, much like a model tree is a regression tree with regression functions at the leaves. As in ordinary decision trees, a test on one of the attributes is associated with every inner node. For a nominal attribute with k values, the node has k child nodes, and instances are sorted down one of the k branches depending on their value of the attribute. For numeric attributes, the node has two child nodes and the test consists of comparing the attribute value to a threshold. An instance is sorted down the left branch if its value for that attribute is smaller than the threshold and sorted down the right branch otherwise.

RANDOM TREE ALGORITHM

The Random Forest algorithm was developed by Leo Breiman, a statistician at the University of California: Berkeley. Random Forests, a meta-learner comprised of many individual trees, was designed to operate quickly over large datasets and more importantly to be diverse by using random samples to build each tree in the forest. Figure 1 shows how the training data is sampled to create an in-bag portion to construct the tree, and a smaller out-of-bag portion to test the completed tree to asses its performance. This performance measure is known as the out-of-bag error estimate.

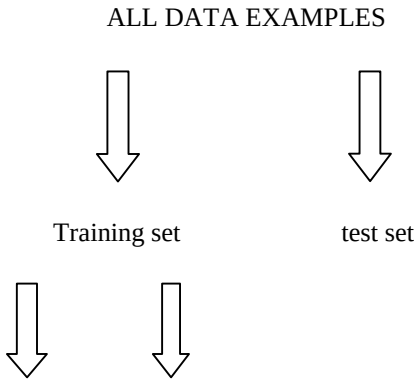


Figure 1: Breakdown of data used to build trees

Construction of a Tree:

1. Randomly sample with replacement (bootstrap) the training set and select 2/3 of data to be used for tree construction (inBag).
2. Choose a random number of attributes from the inBag data and select the one with the most information gain to comprise each node.
3. Continue to work down the tree until no more nodes can be created due to information loss.

4. Compute out-of-bag error estimates by running (ooB) dataset through tree and measuring its correctness.

Diversity is obtained by randomly choosing attributes at each node of the tree and then using the attribute that provides the highest level of learning. The importance of this cannot be overstated as the performance of the Random Forests algorithm is linked to the level of correlation between any two trees in the forest. The more the correlation increases, the lower the overall performance of the entire forest of trees. The way to vary the level of correlation between trees is by adjusting the number of random attributes to be selected when creating a split in each tree. Increasing this variable (m) will both increase the correlation of each tree and the strength of each tree. At some point the tree correlation and tree strength will complement each other providing the highest performance. In addition, increasing the number of trees will provide a more intelligent learner just as having a large diverse group will make intelligent decisions.

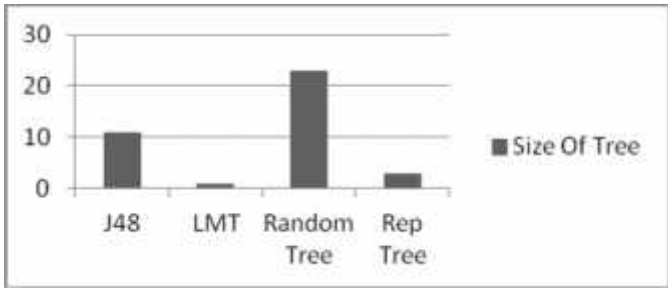
III. Numerical Experiments:

In order to exam our algorithm, we carry out some experiments on datasets, namely Breast cancer datasets. Breast cancer dataset have an attribute is 11 and instance is 700. Use classification and find number of leaves and size of the tree. Decision stump algorithm, j48 algorithm, lmt algorithm, random forest algorithm, random tree algorithm, rep tree algorithm are placed. But here use only j48 algorithm, lmt algorithm, random tree algorithm, rep tree algorithm. At first the instance are divided into 7 equal parts. Each part have a 100 instance but the same attributes. Then apply tool to all part of data and find the tree and leaves.

Part 1.
Table 1:Size of trees

Algorithm	Size of tree
J48	11
LMT	1
Random Tree	23
Rep Tree	3

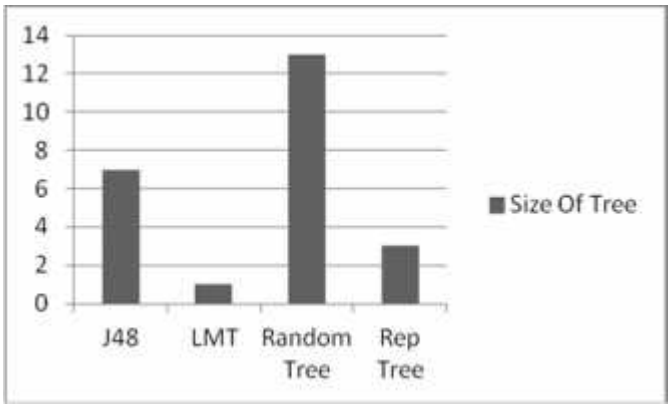
Graph 1:Chart for size of tree



Part 2
Table 2:Size of trees

Algorithm	Size Of Tree
J48	7
LMT	1
Random Tree	13
Rep Tree	3

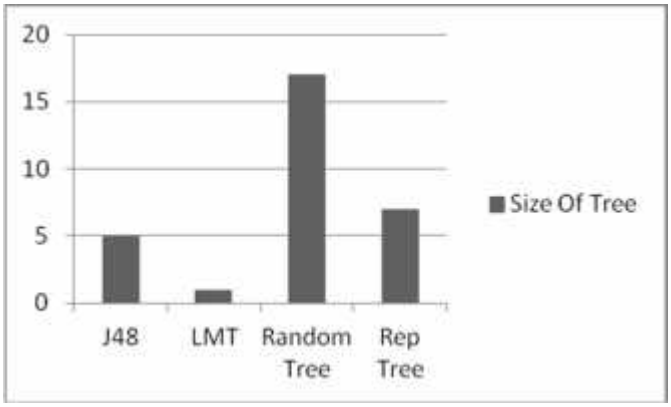
Graph 2: Chart for Size of Trees



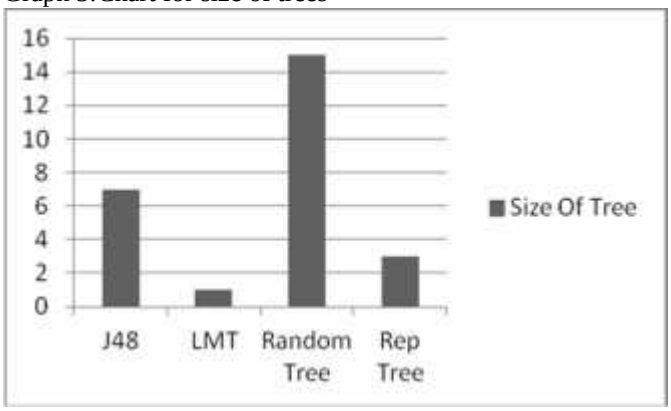
Part 3
Table 3:Size of Trees

Algorithm	Size Of Tree
J48	5
LMT	1
Random Tree	17
Rep Tree	7

Graph 3:Chart for Size of Trees



Graph 5:Chart for size of trees

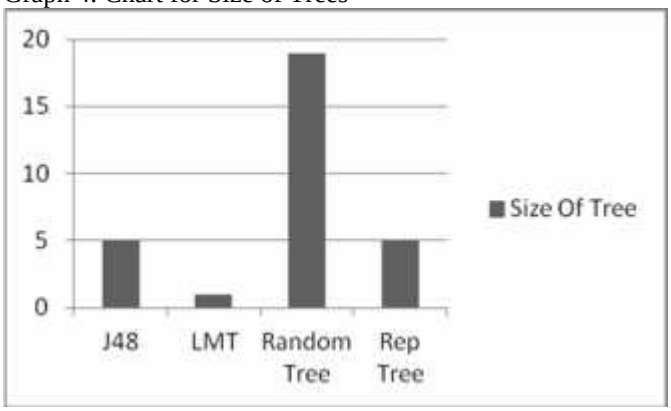


Part 4

Table 4:Size of trees

Algorithm	Size Of Tree
J48	5
LMT	1
Random Tree	19
Rep Tree	5

Graph 4: Chart for Size of Trees



Part 5

Table 5:Size of trees

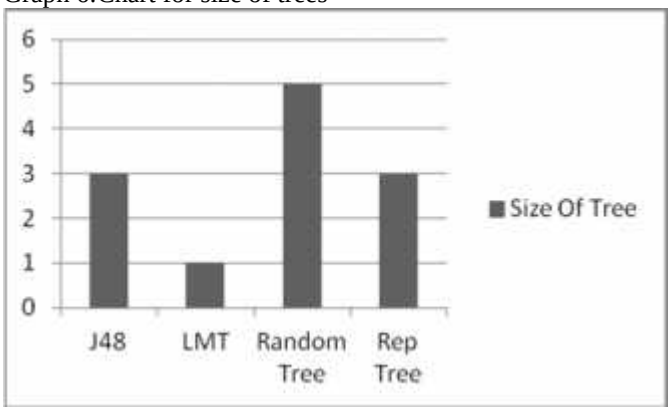
Algorithm	Size Of Tree
J48	7
LMT	1
Random Tree	15
Rep Tree	3

Part 6

Table 6:Size of tress

Algorithm	Size Of Tree
J48	3
LMT	1
Random Tree	5
Rep Tree	3

Graph 6:Chart for size of trees

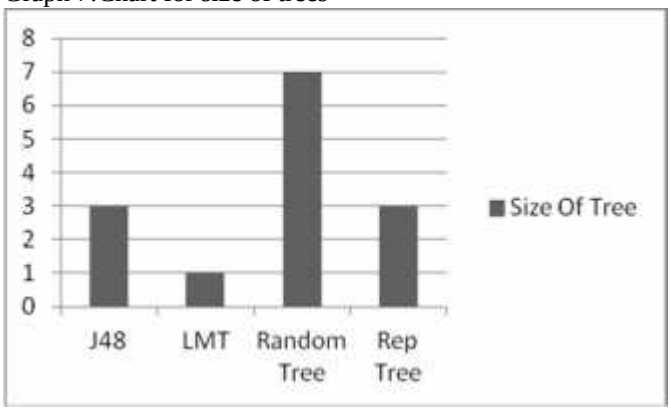


Part 7

Table 7:Size of trees

Algorithm	Size Of Tree
J48	3
LMT	1
Random Tree	7
Rep Tree	3

Graph 7:Chart for size of trees



IV. conclusion

Here used four algorithms like J48 algorithm, LMT algorithm, Random tree algorithm, Rep tree algorithm and find the size of tree for all datas. Compare the size of these algorithm and draw the chart for these various sizes. Random tree algorithm have a large number of size of tree compare with other algorithm.

V. ACKNOWLEDGMENT

The authors would like to thank the management of Bharath university for their support and department of computer science and engineering encouragement for related works

References

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

[2]. L. Breiman, "Random Forests. Machine Learning," vol.45(1), pp. 5-32, 2001.

[3]. F. Esposito, D. Malerba, and G. Semeraro, "A comparative Analysis of Methods for Pruning Decision Trees", IEEE transactions on pattern analysis and machine intelligence, Vol.19(5), pp. 476-491, 1997.

[4]. J. Han and M. Kamber, "Data Mining: Concept and Techniques", Morgan Kaufmann Publishers, 2004.

[5]. WEKA:http//www.cs.waikato.ac.nz/ml/weka.

[6]. T.K.Ho, " The Random Subspace Method for constructing Decision Forest",IEEE Transcation on Pattern Analysis and Machine Intelligence,Vol.20(8),pp.(832-944),1998

[7]. G. Biau, L. Devroye, G. Lugosi, "Consisting of Random Forests and other Averaging Classifiers," Journal of Machine Learning Research, 2008.

[8]. J.R. Quinlan, "Induction of Decession Trees : Machine Learning",vol.1,pp.81-106,1986.

[9]. F. Livingston, "Implementation of Breiman's Random Forest Machine Learning algorithm," Machine learning Journal, 2008.

[10]. J.R. Quinlan, "Simplifying decision trees", Internal Journal of Human Computer Studies,Vol.51, pp. 497-491, 1999.

[11]. N. Landwehr, M. Hall, and E. Frank, " Logistic model trees". for Machine Learning.,Vol. 59(1-2),pp.161-205, 2005.

[12]. N. Laves son and P. Davidson, "Multi-dimensional measures function for classifier performance", 2nd. IEEE International conference on intelligent system, pp.508-513, 2004.