INTERNATIONAL JOURNAL OF COMPUTER APPLICATIONS IN ENGINEERING, TECHNOLOGY AND SCIENCES (IJ-CA-ETS)

A COMPARATIVE STUDY ON DECISION TREE CLASSIFICATION ALGORITHMS IN DATA MINING

[1]VENKATADRI.M, [2]LOKANATHA C. REDDY

[1] Research Scholar, Dept. Of Computer Science School Of Science & Technology, Dravidian University, Kuppam-517425, Andhra Pradesh,
[2] Professor, Dept. Of Computer Science School Of Science & Technology, Dravidian University, Kuppam-517425, Andhra Pradesh, India.

venkatadri.mr@gmail.com, lokanathar@yahoo.com

ABSTRACT :
In Data mining, Classification of objects based on their features into pre-defined categories is a widely studied problem with rigorous applications in fraud detection, artificial intelligence methods and many other fields. Among the various classification algorithms available in literature the decision tree is one of the most practical and effective methods and uses inductive learning. In this paper we reviewed various decision tree algorithms with their limitations and also we evaluated their performance with experimental analysis based on sample data.

1.INTRODUCTION

Data mining is an automated discovery process of nontrivial, previously unknown and potentially useful patterns embedded in databases[1]. Research has shown that, data doubles every three years[2]. Thus data mining has become an important tool to transform these data into information. The datasets in data mining applications are often large and so new classification techniques have been developed and are being developed to deal with millions of objects having perhaps dozens or even hundreds of attributes. Hence classifying these datasets becomes an important problem in data mining[3]. Classification is the problem of automatically assigning an object to one of several pre-defined categories based on the attributes of the object. Classification is also known as supervised learning[4]. In classification a given set of data records is divided into training and test data sets. The training data set is used to build the classification model, while the test data records are used in validating the model. The model is then used to classify and predict new set of data records different from both the training and test data sets. Some of the commonly used classification algorithms are neural networks[21], logistic regression[22] and decision trees[23] etc. Among these decision tree algorithms are most commonly used[7]. Decision tree provides a modeling technique that is easy for humans to comprehend and is simplifies the classification process[8].

This paper attempts to provide a comparative analysis of various commonly used decision tree algorithms with an experimental approach on their classification and prediction accuracy. It is organized as follows: Section 2 provides an overview on decision tree algorithms and phases of decision tree construction and its implementation patterns. Section 3 provides experimental analysis and comparisons of various decision tree algorithms. Section 4 provides a summary and conclusions.

1.1 DESCRIPTION OF TECHNICAL TERMS/NOTATIONS USED

Entropy is a measure of the number of random ways in which a system may be arranged. For a data set S containing n records the information entropy is defined as $Entropy(S) = -\sum P_I \log_2 P_I$.(Here P_I is the proportion of S belonging to class I.)

Gain or the expected information gain is the change in information entropy from a prior state to a state that takes some information. , the information gain of example set S on attribute A is defined as

$$Gain(S, A) = Entropy(S) - \sum_{v \in Value(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

where $Value(A)$ is the set of all possible values of attribute A, S_v is the subset of S for which attribute A has value v, $|S_v|$ is the number of elements in S_v and $|S|$ is the number of elements in S.

Gini index for a data set S is defined as $gini(S) = 1 - \sum P_I^2$ and for a 2-split $gini_{split}(S) = \frac{n_1}{n} gini(S_1) + \frac{n_2}{n} gini(S_2)$ and so on for a k-split.

Hunts method algorithm[12] for decision tree construction trains the data set with recursive partition using depth first greedy technique, till all the record data sets belong to the class label.

2. DECISION TREE CLASSIFICATION

Decision tree is considered to be one of the most popular data-mining techniques for knowledge discovery. It systematically analyzes the information contained in a large data source to extract valuable rules and relationships and usually it is used for the purpose of classification/prediction. Compared to other data-mining techniques, it is widely applied in various areas since it is robust to data scales or distributions [16, 17].

Decision tree algorithm recursively partitions a data set of records using depth-fist greedy approach[5] or breadth-first approach, until all the data items belong to a particular class are identified. A decision tree structure is made of root, internal and leaf nodes. Most decision tree classifiers perform classification in two phases: tree-growing (or building) and tree-pruning. The tree building is done in top-down manner. During this phase the tree is recursively partitioned till all the data items belong to the same class label. In the tree pruning phase the full grown tree is cut back to prevent over fitting and improve the accuracy of the tree[10] in bottom up fashion. It is used to improve the prediction and classification accuracy of the algorithm by minimizing the over-fitting (noise or much data in training data set)[14]. Decision tree algorithm structure is given in two phases as under:

BuildTree (data set *S*)
if all records in *S* belong to the same class,
return;
for each attribute A_i
evaluate splits on attribute A_i ;
use best split found to partition *S* into S_1 and S_2 ;
BuildTree (S_1);
BuildTree (S_2);
endBuildTree;

Fig. 1 Algorithm for decision tree growth phase

PruneTree (node *t*)
if t is leaf
return C(S) +1
/* C(S) is the cost of encoding the classes for the records in set S */
$minCost_1$:= PruneTree (t_1);
$minCost_2$:= PruneTree (t_2); /* t_1, t_2 are t's children*/
$minCost_t$:= min{ C(S)+1, $C_{split}(t)+1+minCost_1+minCost_2$ };
return $minCost_t$; /* C_{split}: cost of encoding a split */
EndPruneTree;

Fig.2 Algorithm for decision tree prune phase

Decision tree algorithms are implementable in both serial and parallel form. Parallel implementation of decision tree algorithms is desirable in-order to ensure fast generation of results especially with the classification/prediction of large data sets; it is also possible to exploit the underlying computer architecture[26]. However when small-medium data sets are involved serial implementation of decision algorithms is easy to implement and desirable.

2. 1 ID3

ID3(Iterative Dichotomized) algorithm is based on the Concept Learning System (CLS) algorithm. CLS algorithm is the basic algorithm for decision tree learning. The tree growth phase of CLS is the matter of choosing attribute to test at each node is by the trainer. ID3 improves CLS by adding a heuristic for attribute selection. ID3 is based on Hunt's algorithm[12] and is implemented in serially[13]. This algorithm recursively partitions the training dataset till the record sets belong to the class label using depth first greedy technique. In growth phase of the tree construction, this algorithm uses information gain, an entropy based measure, to select the best splitting attribute, and the attribute with the highest information gain is selected as the splitting attribute. ID3 doesn't give accurate result when there is too much noise or details in the training data set, thus an intensive pre-processing of data is carried out before building a decision tree model with ID3[13]. One of the main drawbacks of ID3 is that the measure Gain used tends to favor attributes with a large number of distinct values[15]. It only accepts categorical attributes in building a tree model. This decision tree algorithm generates variable branches per node.

2. 2 C4.5

C4.5 algorithm is an improved version of ID3, this algorithm uses Gain Ratio as a splitting criteria, instead of taking gain in ID3 algorithm for splitting criteria[14] in tree growth phase. Hence C4.5 is an evolution of ID3[13]. This algorithm handles both continuous and discrete attributes- In order to handle continuous attributes, C4.5 creates a threshold and then splits the list into those whose attribute value is above the threshold and those that are less than or equal to it[31]. Like ID3 the data is sorted at every node of the tree in order to determine the best splitting attribute. The splitting ceases when the number of instances to be split is below a certain threshold. The main advantages of C4.5 is when building a decision tree, C4.5 can deal with datasets that have patterns with unknown attribute values. C4.5 can also deal with the case of attributes with continuous domains by discretization. This algorithm handles training data with attribute values by allowing attribute values to be marked as missing. Missing attribute values are simply not used in gain and entropy calculations. It has an enhanced method of tree pruning that reduces misclassification errors due to noise or too much detail in the training data set.

2. 3 CART

CART (Classification and Regression Trees)[18], is characterized by the fact that it constructs binary trees, namely each internal node has exactly two outgoing edges whereas both ID3, C4.5 algorithms generate the decision trees with variable branches per node. CART is unique from other Hunt's based algorithm as it is also use for regression analysis with the help of regression trees. The regression analysis feature is used in forecasting a dependent variable (result) given a set of predictor variables over a given period of time. The CART decision tree is a binary recursive partitioning procedure capable of processing continuous and nominal attributes both as targets and predictors[18]. In CART trees are grown, uses gini index for splitting procedure, to a maximum size without the use of a stopping rule and then pruned back (essentially split by split) to the root via cost-complexity pruning. The CART mechanism is intended to produce not one, but a sequence of nested pruned trees, all of which are candidate optimal trees. The CART mechanism includes automatic (optional) class balancing, automatic missing value handling, and allows for cost-sensitive learning, dynamic feature construction, and probability tree estimation[6].

2. 4 BEST FIRST TREE

BEST FIRST TREE[30] Standard algorithms such as ID3, C4.5 and CART for the top-down induction of decision trees expand nodes in depth-first order in each step using the divide-and-conquer strategy. The selection attribute in C4.5 is based on entropy gain in tree grown phase. The selection attribute is based on gini index in CART algorithm, and then split training instances into subsets; one for each branch extending from the root node, the number of subsets is the same as the number of branches. A fixed order is used to expand nodes (normally, left to right) these decision trees. In Best-first decision trees the selection of best split is based on boosting algorithms[29] which is used to expand nodes in best-first order instead of a fixed order. This algorithm uses the both the gain and gini index in calculating the best node in tree grown phase of the decision tree. This method adds the "best" split node to the tree in each step. The best node is the node that maximally reduces impurity among all nodes available for splitting (i. e. not labeled as terminal nodes). Although this results in the same fully-grown tree as standard depth-first expansion, it enables us to investigate new tree pruning methods that use cross-validation to select the number of expansions. Both pre-pruning and post-pruning can be performed in this way.

2. 5 SLIQ

SLIQ[25] (Supervised Learning In Quest) was one of the first scalable algorithms for decision tree induction. This can be implemented in serial and parallel pattern. It is not based on Hunt's algorithm for decision tree classification[13]. It partitions a training data set recursively using breadth-first greedy strategy that is integrated with pre-sorting technique during the tree building phase. SLIQ uses a vertical data format, meaning all values of an attribute were stored as a list, which was sorted at the start of the algorithm. This meant that the attributes need not be sorted repeatedly at each node as was the case in existing algorithms like CART and C4.5. With the pre-sorting technique sorting at decision tree nodes is eliminated and replaced with one-time sort, with the use of list data structure for each attribute to determine the best split point. The calculation of gini index for each possible split point can be done efficiently by storing class distributions in histograms, one per class per node. However SLIQ uses a memory-resident data structure called *class list* which stores the class labels of each record. This data structure limits the size of the datasets SLIQ can handle[19]. In building a decision tree model SLIQ handles both numeric and categorical attributes. One of the disadvantages of SLIQ is that it uses a class list data structure that is memory resident thereby imposing memory restrictions on the data. It uses Minimum Description length Principle(MDL)[11] in pruning the tree after constructing it MDL is an inexpensive technique in tree pruning that uses the least amount of coding in producing tree that are small in size using bottom –up technique[13, 24]. The SLIQ decision tree algorithm produces accurate decision trees that are significantly smaller than the trees produced using C4.5 and CART. At the same time, SLIQ executes nearly an order of magnitude faster than CART[32].

2. 6 SPRINT

SPRINT (Scalable Parallelizable Induction of decision Tree algorithm) present a more memory-efficient version of SLIQ[26]. This algorithm associates the class label along with the record identifier for each value in the attribute lists. It is a fast, scalable decision tree classifier. It is not based on Hunt's algorithm in constructing the decision tree, rather it partitions the training data set recursively using breadth-first greedy technique until each partition belong to the same leaf node or class[13, 26]. It is an enhancement of SLIQ as it can be implemented in both serial and parallel pattern for good data placement and load balancing[26]. In this paper we will focus on the serial implementation of SPRINT, Like SLIQ it uses one time sort of the data items and it has no restriction on the input data size. Unlike SLIQ it uses two data structures; attribute list and histogram which is not memory resident making SPRINT suitable for large data set, thus it removes all the data memory restrictions on data[26]. It handles both continuous and categorical attributes.

2. 7 RANDOMFOREST

RandomForest[20] are ensemble of un pruned binary decision trees, unlike other decision tree classifiers Random Forest grows multiple trees which creates a forest like classification. Each tree is grown on bootstrap[27] sample of the training set (this help in over fitting). Ensemble learning method of RandomForest is very promising technique in terms of accuracy and also providing a distributed aspect can adapted to distributed data mining[28]. In the tree grown phase of standard trees (ID3, C4.5, CART, SLIQ, SPRINT, BFTree) each node is split using the best split among all variables. In a random forest, each node is split using the best among a subset of predictors randomly chosen at that node. This somewhat counterintuitive strategy turns out to perform very well compared to many other classifiers, including discriminant analysis, support vector machines and neural networks, and is robust against over fitting[20]. The Random Forest produces better performance as compared to that of single tree classifier[9]. This method is computationally effective, does not over fit, is robust to noise and can also be applied when the number of variables is much larger than the number of samples. It includes a good method for estimating missing data and maintains accuracy when a large proportion of the data are missing.

3. EXPERIMENTAL ANALYSIS

We carried out an experiment to compare the above said decision tree algorithms based on their Accuracy, Learning Time and Tree Size. This experiment has carried out on the financial (credit-g), transportation (vehicle), science (image segmentation), handwriting recognition (letter), land usage images (sat-image), and mushroom data sets taken from the UCI Machine Learning Repository and we have chosen weka 3.6 data mining tool to carry out our experiment.

Datasets	C4.5	CART	BF Tree	SLIQ	SPRINT	Random Forest
1. Vehicle	0. 17	0. 39	0. 39	0. 13	0. 11	0. 03
2.Letter	1. 66	3. 52	6. 25	0. 75	0. 14	0. 12
3. credit-g	0. 03	1. 05	1. 06	0. 09	0. 05	0. 02
4. Segment	0. 16	1. 06	0. 99	0. 33	0. 06	0. 08
5. Mushroom	1. 26	6. 17	6. 55	6. 44	5. 13	1. 02
6. Sat-image	0. 93	0. 95	0. 97	0. 54	0. 55	0. 77

Table 1: details of the data sets

Datasets	#Attributes	#Classes	#Instances
1. Vehicle	19	4	846
2. Letter	17	26	4080
3. Credit- g	21	2	1000
4. Segment	20	7	2310
5. Mushroom	23	2	8124
6. Sat-image	37	6	1286

Table 2 : Representing Accuracy percentages of the different algorithms.

Based on the observations on our experimental results the comparison is as follows:

3.1 ACCURACY

This is the reliability of the decision tree and one of the most important parameters which is used for comparing different approaches. This parameter is relevant to the percentage of test samples that are correctly classified. Considering the experimental results shown in the Table 2 the accuracy of the ensemble decision trees (Random Forest) is better than the accuracy of the single tree decision tree algorithms like C4.5, CART, BFTree, SLIQ,

SPRINT. The accuracy of the presorting algorithms is better than the accuracy of the entropy based induction decision trees (C4.5, CART, and BFTree).

3.2 LEARNING TIME

This parameter is the time which is taken for learning and constructing decision trees. Different approaches try to shorten the time. Table 3 shows the learning times of different classification

algorithms in seconds. In our observation the number of instances increases the time taken to construct the decision tree increases.

Datasets	C4.5	CART	BFTree	SLIQ	SPRINT	Random Forest
1.Vehicle	71.01	70.83	72.56	79.05	79.62	83.61
2.Letter	77.05	75.29	74.00	78.5	78.52	83.85
3.credit-g	74.64	77.05	75.00	76.64	76.94	86.17
4.Segment	96.17	95.41	95.66	97.05	97.54	97.45
5.Mushroom	100	99.93	100	100	99.96	100
6.Sat-image	82.23	83.54	83.06	84.23	8442	86.95

Table 3: Representing the learning time of different algorithms

3.3 TREE SIZE

Table 4 shows the tree sizes of different decision tree classifiers. This is clear that the more the size of the decision tree grows, the more the number of operations, which has to be done for classification, i ncreases. Therefore the simplicity of this approach leads to reduction of the classification time. The Random Forest ensemble of 10 trees with 5 random features out of with different bag errors[20].

Datasets	C4.5	CART	BFTree	SLIQ	SPRINT	Random Forest
1. Vehicle	195	159	117	423	49	----
2. Letter	839	629	721	1483	353	----
3. credit- g	140	13	77	150	96	----
4. Segment	77	81	91	281	43	----
5. Mushroom	30	13	13	158	38	----
6. Sat-image	147	53	95	137	138	----

Table 4 represents the tree Size of different algorithms

4. CONCLUSION

Decision tree is one of the most popular classification techniques in data mining. In this paper we compared popular decision tree algorithms based on their accuracy, learning time and tree size. We observed that, there is a direct relationship between execution time in building the tree model and the volume of data records and also there is an indirect relationship between execution time in building the model and attribute size of the data sets. Through our experiment we conclude that SPRINT and Random Forest algorithms have good classification accuracy over above compared algorithms.

5.REFERENCES

[1] W. J Brawley, G. Piatetsky-Shapiro, and C. J Matheus ,1991, " Knowledge Discovery in data bases An overview ", MIT Press.

[2] Lyman, Peter; Hal R. Varian ,2003, "How Much Information". Web ref :URL : http://www. sims. berkeley. edu/how-much-info-2003.

[3] Gang Wang, Chenghong Zhang, Lihua Huang , 2008, "A Study of Classification Algorithm for Data Mining Based on Hybrid Intelligent Systems ", Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, pp. 371-375.

[4] S.Kotsiantis, 2007, "Supervised Machine Learning: A Review of Classification Techniques " , Informatica Journal 31. pages 249-268 .

[5] Paul E. Black ,2005, "Greedy Algorithm" ,in Dictionary of Algorithms and Data Structures[online], U. S. National Institute of Standards and Technology, February 2005, webpage: NIST-greedy algorithm.

[6] .Xin dong Wu,Vipin Kumar ,J.Ross Quinlan, Joydeep Ghosh,Qiang Yang, Hiroshi Motoda, 2007 , "Top 10 algorithms in data mining " , Spinger- Verlag London Limited ,pages 20 -28.

[7] Twa MD, Parthasarathy S, Roberts C,Mahoud AM, Raasch Tw ,Bullimore M.A, 2005, "Automated Decision Tree Classification of Corneal Shape", The journal of American academy Optometry - Volume 82 - Issue 12.

[8] Brodley, C. E. , & Utgoff, P. E, 1992, "Multivariate versus univariate decision trees".

[9] Ng Ee Ling and Yahya Abu Hasan ,2008, "Evaluation method in Random Forest as applied in Microarray data" ,Malaysian Journal of Mathematical Sciences 2(2): 73-81.

[10] Uwe. K, Dunemann,S..O, 2001, "SQL Database Primitives for Decision tree Classifiers", CIKM '01 atlanta, ACM, GA USA.

[11] J. Han, M. Kamber, 2006, "Data Mining: Concepts and Techniques Second Edition".

[12] A. Srivastava, E. Han, V. Kumar, V. Singh, 1998, "Parallel Formulations of Decision-Tree Classification Algorithms". International Conference on Parallel Processing (ICPP'98). pp 237.

[13] Matthew N. Anyanwu & Sajjan G. Shiva, 2009, " Comparative Analysis of Serial Decision Tree Classification Algorithms", International Journal of Computer Science and Security, (IJCSS) Volume (3): Issue (3).

[14] Andrew Colin, 1996, "Building Decision Trees with the ID3 Algorithm", Dr. Dobbs Journal, June 1996.

[15] Li_Tianrui, 2007, "Construction of Decision Trees based Entropy and Rough Sets under Tolerance Relation", ISKE-2007 Proceedings part of series: Advances in Intelligent Systems Research. ISBN: 978-90-78677-04 8.

[16] M. J. Berry and G. S. Linoff, 2000, "Mastering data mining", New York: John Wiley & Sons.

[17] K-M. Osei-Bryson, 2007, "Post-pruning in decision tree induction using multiple performance measures", Computers &Operations Research, 34: 3331-3345.

[18] Breiman, Leo, Friedman, J. H. , Olshen, R. A. , & Stone, C. J, 1984, "Classification and regression trees, Monterey", CA: Wadsworth & Brooks/Cole Advanced Books & Software, ISBN 978-0412048418.

[19] Alex A. Freitas , "A survey of parallel Data Mining",
URL:http://kar.kent.ac.uk/21570/2/A_Survey_of_Parallel_Data_Mining.pdf.

[20] Breiman, L, 2001, "Random Forests", Machine Learning, Pages 45, 5-31.

[21] PL Richard, 1987, "An introduction to computing with neural nets", IEEE ASSP Magazine.

[22] Taghi M. Khoshgoftaar, 2000, "Accuracy of software quality models over multiple releases", journal of Annals of Software Engineering Volume 9, Numbers 1-4.

[23] John Ross Quinlan, 1993, "C4.5: Programs form machine learning", Morgan Kaufmann.

[24] Anyanwu M, and Shiva S, 2009, "Application of Enhanced Decision Tree algorithm to Churn analysis", International Conference on Artificial Intelligence and Pattern Recognition (AIPR-09), Orlando Florida.

[25] M. Mehta, R. Agarwal, and J. Rissanen, 1996, "SLIQ: A fast scalable classifier for data mining", In Extending Database Technology, pages 18-32.

[26] J. C. Shafer, R. Agrawal, and M. Mehta, 1996, "SPRINT : A scalable parallel classifier for data mining", In Proceedings of 22nd International Conference in Very Large databases, VLDB, pages 544-555.

[27] B. Efron and R. Tibshirani, 1986, "Bootstrap Methods for Standard Errors, Confidence Intervals, and Other Measures of Statistical Accuracy", Statistical Science, Vol. 1, No. 1 , pp. 54-75.

[28] D.Mokkeddem, H Belbachir, 2009, "A Survey of Distributed Classification Based Ensemble Data mining methods", Journal of Applied Sciences 9(20): 3739-3745.

[29] J. Friedman, T. Hastie, and R. Tibshirani, 2000, "Additive Logistic Regression: a Statistical View of Boosting" , Annals of Statistics, 28:337-374.

[30] Friedman, Getoor, D Koller, Apfeffer, 1999, "Learning probabilistic relational models " , Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99).

[31] J. R. Quinlan, 1996, "Improved use of continuous attributes in C4.5" , Journal of Artificial Intelligence Research, Vol. 4, pp. 77-90.

[32] M Mehta, R Agarwal, JRissanen, 1996, "SLIQ: A fast Scalable classifier for data mining", Springer Publications.