A Survey of Logic Based Classifiers

Aftab Ali Haider and Sohail Asghar

Abstract—**Classification is most challenging and innovative problem in data mining. Classification techniques had been focus of research since years. Logic, perception, instance and statistical concepts based classifiers are available to resolve the classification problem. This work is about the logic based classifiers known as decision tree classifiers because these use logic based algorithms to classify data on the basis of feature values. A splitting criterion on attributes is used to generate the tree. A classifier can be implemented serially or in parallel depending upon the size of data set. Some of the classifiers such as SLIQ, SPRINT, CLOUDS, BOAT and Rainforest have the capability of parallel implementation. IDE 3, CART, C4.5 and C5.0 are serial classifiers. Building phase has more importance in some classifiers to improve the scalability along with quality of the classifier. This study will provide an overview of different logic based classifiers and will compare these against our pre-defined criteria. We conclude that SLIQ and SPRINT are suitable for larger data sets whereas C4.5 and C5.0 are best suited for smaller data sets.**

Index Terms—**Classification, data mining, decision trees, logic based classifiers.**

I. Introduction

There is an ardent need to automate data mining techniques due to recent advances in storage and data collection methods. Tremendous increase in data received from current information systems used for weather forecasting, market sales, daily stocks trade and others has increased the need to find proactive and highly efficient data mining techniques to cope with these advances. Various techniques based on logic, perception or statistical algorithms are available. Decision trees use logic based algorithms to describe, classify and generalize data. Some of the applications of decision trees in various knowledge areas are pattern recognition, expert systems, signal processing, decision theory, machine learning, and artificial neural networks and statistics [1], [2].

This study is concerned with the analytical review and criticism of logic based classifiers. A comparison is given in Table I.

II. Decision Tree Algorithms

Concept learning system (CLS) presented by Hunt in 1966 was the beginning of decision tree classifiers. The objective of CLS was to reduce the cost of classifying objects which can be either misclassification cost or finding the value of object or both. IDE 3.0 (Iterative Dichotomizer 3 is successor of CLS but it focused on the induction task. Induction is basically to derive such classification rules from attributes of objects for which classifiers are equally successful for both training as well as test data set. IDE 3.0 suggested to build all possible decision trees and finally select the simplest one was the optimal result. However, limitation of IDE 3.0 can be more explicitly found for larger databases, where construction of all possible chains of decision trees is not an easy task. Second drawback is evident from the fact that this algorithm selects the simplest decision tree as the result. The optimal or accurate decision tree may or may not be the simplest tree in the chain of possible decision trees. Only ordered values of attributes can be handled in IDE 3.0 and there is no method to deal with continuous values. Selection best attribute at root node is found through information gain method. Due to inability of IDE 3.0 to deal with noisy data, lot of pre-processing is required for accurate results [4], [5].

C 4.5 is successor of IDE 3.0 and is based on Hunt's tree construction method. It can be rightly said that C 4.5 is an improved version of IDE 3.0. In C 4.5, the inherent drawbacks of IDE 3.0 are removed to make the algorithm more accurate for noisy data and with rich information. C 4.5 can handle both ordered and unordered vales, however, is less accurate for continuous attributes [6]. This problem was later addressed [10]. In order to avoid over-fitting problem, pruning method has been enhanced. The selection of best attribute is done through gain ratio impurity method. Instability problem of decision trees has also been considered in C 4.5. C 5.0 has combined boosting algorithm Adaboost and C 4.5 as a software package [9].

CART (Classification and Regression Trees) builds both regression and classification trees. In CART, data is taken in raw form and it has the capability to handle both continuous and discrete attributes. CART uses binary recursive partitioning procedure. A sequence of nested pruned trees is obtained in CART. Pruning in CART is done on a portion of training set. Unlike C 4.5, predictive performance of optimal decision tree is obtained on independent data set rather than internal performance measure used in C 4.5 for tree selection. This is the basic deviation from other decision tree algorithms that use training data set to identify the best tree. Another distinct feature of CART is that it has automatic tree balancing and missing values handling mechanism. Missing variables are not dropped and "surrogate" variables with similar information contained in the primary splitter are substituted. Automatic learning makes it simpler than other multivariate modeling methods. Gini index is used to select the best attribute at root. However, it has the capability to use other single variable such as symgini or multi-variable splitting criteria e.g. linear combinations to determine the best split point. Each and every node is tested for best split based on these criterions. Regression tree building is another distinct feature which is not found in C 4.5 and IDE 3.0 [11], [12].

Manuscript received September 19, 2012; revised October 14, 2012.

Aftab Ali Haider is with Center for Software Dependability, Muhammad Ali Jinnah University (MAJU), Islamabad, Pakistan (email: aftab775@yahoo.com).

Sohail Asghar is with Centre for Research in Data Engineering (CORD), Muhammad Ali Jinnah University (MAJU) Islamabad, Pakistan (email: sohail.asghar@jinnah.edu.pk).

DOI: 10.7763/IJFCC.2013.V2.135

TABLE I: A COMPARISON OF LOGIC BASED CLASSIFIERS

Parameters	IDE 3.0	C 4.5 & C 5.0	CART	SLIQ	SPRINT	CLOUDS	BOAT	Rainforest	PUBLIC
Pruning Technique	Simple Pruning	Error Based Pruning	Cost Complexity Pruning	MDL principle	MDL principle	MDL principle	MDL principle	MDL principle	MDL principle
Accuracy of Classification	Poor	Moderate	Moderate, however poor than IDE 3 in certain cases	Very High for small data set even better than SLIQ	Very High for large data set	Better than SPRINT	Better than SPRINT	Better than SPRINT	Better than SPRINT
Scalability	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Basic methodology of tree construction	Hunt's	Hunt's	Hunt's	Shafer's	Shafer's	Shafer's	BOAT framework	RainForest Framework	Shafer's
Method to select root node	Information Gain	Gain /purity ratio	Single variable and multivariable	Gini Index	Gini Index	Gini Index	Wide range of split selection methods based on impurity	Wide range of split selection methods based on impurity	entropy
Type of data to be handled	Continuous	Numeric and discrete	Numeric and Discrete	Numeric and Discrete	Numeric and Discrete	Numeric and Discrete	Numeric and Discrete	Numeric and Discrete	Numeric and Discrete
Tree coverage approach	Depth First, divide and Conquer	Depth First, divide and Conquer	Depth First, divide and Conquer	Breadth First, divide and Conquer	Breadth First, divide and Conquer	Breadth First, divide and Conquer	Top down, divide and conquer	Top down, divide and conquer	Breadth First, divide and Conquer
Ancestor algorithm	CLS	IDE 3.0 , C 4.0	IDE 3.0	C 4.5	SLIQ	SPRINT	SPRINT	SPRINT	SPRINT
Special features	No	Quite successful in experimental physics	Does not use statistics	Disk Resident	Disk resident and successful for parallel machines	Uses SS and SSE for small subset	Allows dynamic insertion and deletion	AVC groups instead of Attribute list is used It provides framework for all known algorithms	Integrates pruning and tree building

SLIQ and SPRINT are based on Shafer's tree construction method that uses breadth first approach. Another distinct feature of these algorithms is that these are not memory resident and are highly suitable for large data sets. SLIQ (Supervised Learning In Quest) can handle both numeric and categorical attributes. To reduce the cost of evaluating numeric attributes, pre sorting techniques are used in tree growth phase. Pre sorting replaces the sorting at nodes with one time sort and uses list data to find the best split point. List data structure is memory resident and is a constraint on SLIQ to handle larger data sets. Pruning is based on Minimum Description Length (MDL) principle which is an inexpensive and accurate method. These characteristics make SLIQ capable to handle large data sets with ease and lesser time complexity [7]. SLIQ uses gini index to find the best attribute to reside at root node. However, drawbacks in suing gini are addressed in [13].

Other advantages of SLIQ are its ability to handle disk resident data sets that cannot be handled by previously discussed algorithms. However, this algorithm uses training data set for classification as compared to CART [7].

Unlike SLIQ, SPRINT (Scalable Parallelizable Induction of decision Tree algorithm) can also be used both as parallel as well as serial decision tree algorithm. It is definitely an extension of SLIQ algorithm. SPRINT is also based on breadth first technique presented by Shafer for tree construction. SPRINT is also fast and highly scalable like SLIQ. Unlike SLIQ, where data list structure is memory resident, in SPRINT attribute list is not memory resident. Hence, there is no storage constraint on larger data sets in SPRINT. Memory resident attribute list is useful for smaller data sets because there is no need to rewrite the list for each split. For larger data sets, disk resident attribute list of SPRINT is better than SLIQ [3].

PUBLIC classifier is distinct from previously discussed algorithms as it integrates pruning and building phases together. Growing a decision tree without pruning may need extra effort, which can be reduced if pruning is done in parallel. PUBLIC uses entropy to select the best split at root node. Results deduced in indicate that PUBLIC performs better than SPRINT [8].

CLOUDS (Classification for Large or OUt of core Data Sets) is an enhancement to SPRINT. It has lower complexity and lesser input / output requirements as compared to SPRINT for real data sets. CLOUDS is based on shafer's breadth first methodology. Sampling the splitting points (SS) and sampling the splitting point with estimation (SSE) are used to build the tree from randomly small subset of training data [14].

RainForest framework provides a tree induction schema that can be applied to all known algorithms. This framework separates the quality and scalability concerns. A group of attributes at each nodes is evaluated which are later used to find the best split criteria. The algorithms based on this framework have lower computational complexity with better performance as compared to SPRINT [15].

Bootstrapped Optimistic Algorithm for **T**ree construction (BOAT) further improves the computational efficiency by building decision trees at faster speeds than other tree building algorithms and performs fewer database scans. An added feature of BOAT is its ability to dynamically update the tree for any modification including insertion and deletion without the need to re-construct it. BOAT exploits bootstrapping technique to select the splitting criteria and a small subset of the training dataset for which tree is constructed. This tree is built on a small subset of training dataset but it reflects the properties of entire training data set. In case there is some error or difference, the tree can be reconstructed for the affected portion with nominal cost [16]. Brief comparison is given in Table 1.

III. Evaluation and Analysis of Classifiers

Major Logic based classifiers are evaluated as under;

A. IDE 3.0

IDE 3.0 was an initial work by Quinlan which was based on Hunt's tree construction. Some of the shortcomings as compared to other algorithms are;

- Simplest tree selected out of the chain of possible trees may or may not be the optimal tree.
- This algorithm was unable to handle numeric attribute.
- This algorithm has low accuracy of classification for large data sets and was not scalable as well.
- There was need of pre processing to improve the accuracy of IDE 3.0 in information rich and noisy data set.

B. C 4.5 and C 5.0

C 4.5 was the successor of IDE 3.0 and C 4.0. This algorithm was able to handle both numeric and discrete data sets. But later work found that C 4.5 has low accuracy in dealing with numeric attributes. A new version with modifications to cater for the numeric attribute in better way was released. Even then the accuracy of classification of C 4.5 was not comparable to other statistics based classifiers. Later boosting and bagging on C 4.5 helped to improve the performance of C 4.5. In some cases, C 4.5 is far better than ANN.

Some of the shortcomings are listed below

- The pruning method of C 4.5 is biased to under-pruning
- Tree selection in C 4.5 is based on training data set, whereas same task in CART is done on test data set.
- C 4.5 is based on classical statistics and requires skill for better understanding.
- C 4.5 lacks the means to partial automatic learning as are found in CART.
- Missing values are dropped in C 4.5 whereas there is a mechanism to deal with missing values in CART.
- C 4.5 is memory dependant and is not successful for very large data sets.
- C 4.5 is not speedy and fast as SLIQ and SPRINT
- C 4.5 cannot use multivariate methods for attribute selection at root node.

Despite these shortcomings, C 4.5 is preferred in data mining due to familiarity and ease to use.

C. CART

CART is also based on Hunt's tree construction methodology. This algorithm is competitor of C 4.5. Most of the drawbacks of C 4.5 are resolved in CART. However, some of the shortcomings of CART are as follow;

- CART is memory resident and is not suitable for large data sets.
- CART performs sorting at each and every node, which is not found in SLIQ and SPRINT
- CART is simpler and does not require prior knowledge of statistics. CART is new concept of tree building and is not based on classical statistics.
- Statisticians are not much familiar with CART and this is not accepted by them as other algorithms like C 4.5 and SLIQ or SPRINT are accepted.
- Due to deviation from basics of classical statistics there are only few who are familiar with it. This factor reduces its applicability because there is none to help you out in case of any difficultly to implement it.

D. SLIQ

SLIQ is fast and speedy algorithm and is highly suitable for large data sets. This algorithm uses attribute list for sorting the best attribute for splits. However, there are some of the shortcomings of the SLIQ as discussed below;

- Attribute list of SLIQ is memory resident, which puts a memory constraint on the classifier.
- SLIQ is successful for serial implementations only and cannot be applied on parallel machines.

Although there are some drawbacks of the SLIQ, but still it is best algorithm for those data sets for which memory is not an issue. However, it is not preferable to use for very large data sets.

E. SPRINT

SPRINT is an extension of SLIQ. The purpose to devise this classifier was to resolve the shortcomings of SLIQ. This algorithm has the ability to be implemented for serial as well as parallel applications. Secondly, attribute list and histogram in SPRINT are disk resident and there is no memory constraint. This algorithm makes the size of data set independent of main memory and can be rightly said a scalable algorithm with lesser time complexity [15].

Main drawbacks of this classifier are that attribute list needs to be re written and re sorted for each split, which is not a preferred thing. Some of these drawbacks are later addressed in RainForest framework.

F. PUBLIC

PUBLIC improves the accuracy of classification by integrating pruning and tree building in a single phase. The main difference between the PUBLIC and other classifiers is that this algorithm concentrates on pruning instead of tree building phase to improve the performance. Possibility of integrating pruning with tree building phase has been exploited in this algorithm.

G. CLOUDS

In pre processing phase of SPRINT data set is partitioned into attribute list which requires one read and one write operation whereas external sorting requires two read and two write operations. However, in this algorithm selection of a random sample reduces the computational cost and input output requirements. There are certain shortcomings of the CLOUDS which include;

- A small subset of the training data set is selected to build the classifier through SS and SSE. Any loss of data may affect the accuracy of classification
- Secondly, it has been assumed that entire subset fits in the main memory that may not be true always.

H. RainForest

In SPRINT, there is a need to re-write and sort the attribute list at each node. Re-writing the list undesirably increases the size of database and sorting increase computational cost. Both of these factors are highly undesirable and have room for improvements.

Rainforest has introduced **A**ttribute-**V**alue, **C**lass label (AVC) group for each node instead of generating attribute list. The size of the AVC group is much smaller than attribute lists because attribute list is proportional to the number of records in the data partition. Distinct values in columns of data partition determine the size of AVC group. RainForest framework is applicable to all known decision tree algorithms and performs faster and has better performance than SPRINT [15].

I. BOAT

RainForest requires a portion of main memory for AVC group at each node which has been eliminated in BOAT. BOAT is faster and allows for dynamic insertion and deletion of records. Some of the shortcomings of BOAT are;

- It strongly depends upon the small subset of dataset to train the classifier. This factor may lead to reduce accuracy of performance in certain cases.
- It allows dynamic insertion and deletion which requires thorough and rigorous studies to ensure that built tree is similar to the re-constructed tree.

However, BOAT has provided certain improvements which can be used to achieve better performance and accuracy of classification [16].

IV. Conclusion

Selection of a classifier for certain data set is a difficult task. However, if basic features of these classifiers are known it is quite easier to select most relevant classifier that can provide better results. This perception can be strengthened by the fact that SLIQ is quite useful for smaller data sets and provides better results than SPRINT for such a dataset but when SLIQ is implemented for larger data set, the SPRINT outperforms SLIQ. Similarly IDE 3.0 has better accuracy than CART in certain cases. A careful understanding of a classifier helps to more accurately classify the training data set. There are two different implementations of classifiers i.e. serial and parallel. A parallel implementation improves the computation complexity and is mandatory for larger data sets. In such a case SPRINT, CLOUDS, BOAT or Rainforest are preferable. Whenever there is a smaller data set, the main contender to be best classifiers can be IDE 3.0, C 4.5, CART or SLIQ. A more quantified comparison of these classifiers can be done by implementing these classifiers in weka for a considerably large data set.

References

[1] S. K. Murthy, “Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey,” *Data Mining and Knowledge Discovery*, vol. 2, 1998, pp. 345–389.

[2] S. B. Kotsiantis, “Supervised Machine Learning: A Review of Classification Techniques,” *Informatica*, vol. 31, 2007, pp. 249-268.

[3] J. Shafer, R. Agrawal, and M. Mehta, “A scalable parallel classifier for data mining,” in *proceedings of the 22nd international conference on very large data base. Mumbai (Bombay)*, 1996.

[4] J. R. Quinlan, “Induction of decision trees,” *Machine Leaning*, vol. 1, 1986, pp. 81-106.

[5] J. R. Quinlan, “Simplifying decision trees,” *International Journal of Mach ine Studies*, vol. 27, 1987, pp. 221-234.

[6] J. R. Quinlan, “C45: Programs for Machine Learning,” *Morgan Kaufmann*, San Mateo, CA, 1993.

[7] M. Mehta and R. Agrawal, and J. Rissanen, “SLIQ: A fast scalable classifier for data mining,” In *EDBT 96*, Avignon, France

[8] R. Rajeev and S. Kyuseok, “PUBLIC: A decision tree classifiers that integrates building and pruning,” in *Proceedings of the 24th VLDB conference*, New York, USA, 1998.

[9] Y. Freund and M. Llew, “The alternating decision tree algorithms,” in *Proceedings of the 16th International Conference on Machine Learning*, pp. 124-133, 1999.

[10] J. R. Quinlan, “Improved use of continuous attributes in C 4.5,” *Journal of Artificial Intelligence Research*, vol. 4, 1996, pp. 77-90

[11] D. Steinberg, “The top ten algorithms in data mining,” Ch 10, *Taylor and Francis Group*, LLC, 2009.

[12] J. L. Roger, “An Introduction to Classification and Regression Tree (CART) Analysis,” in *Proc. of Annual Meeting of the Society for Academic Emergency Medicine in San Francisco*, California 2000.

[13] B. Chandra, “Elegant Decision Tree Algorithm for Classification in Data Mining,” in *Proc. of Third International Conference on Web Information Systems Engineering (Workshops)*, 2002.

[14] K. Alsabti, S. Ranka, and V. Singh, “CLOUDS: A decision tree classifier for large datasets,” in *Proc. of 4th Intl. Conf. on Knowledge Discovery and Data Mining*, Aug 1998.

[15] J. Gehrke, V. Ganti, and R. Ramakrishnan, “RainForest - A Framework for Fast Decision Tree Construction of Large Datasets,” in *Proc. of the 24th VLDB Conference New York*, USA, 1998.

[16] J. Gehrke, V. Ganti, R. Ramakrishnan, and W. Loh, “BOAT–optimistic decision tree construction,” in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, June 1999.