A Novel Prototype Decision Tree Method Using Sampling Strategy

Bhanu Prakash Battula[1], Debnath Bhattacharyya[2],
C.V.P.R. Prasad[3], and Tai-hoon Kim[4](✉)

[1] Department of CSE, Vignan College, Guntur, AP, India
battulaphd@gmail.com
[2] Department of Computer Science and Engineering,
Vignan's Institute of Information Technology, Visakhapatnam, AP, India
debnathb@gmail.com
[3] Research Scholar, Acharya Nagarjuna University, Guntur, Andhra Pradesh, India
prasadcvpr@gmail.com
[4] Department of Convergence Security, Sungshin Women's University, 249-1,
Dongseon-dong 3-ga, Seoul 136-742, Korea
taihoonn@daum.net

Abstract. Data Mining is a popular knowledge discovery technique. In data mining decision trees are of the simple and powerful decision making models. One of the limitations in decision trees is towards the data source which they tackle. If data sources which are given as input to decision tree are of imbalance nature then the efficiency of decision tree drops drastically, we propose a decision tree structure which mimics human learning by performing balance of data source to some extent. In this paper, we propose a novel method based on sampling strategy. Extensive experiments, using C4.5 decision tree as base classifier, show that the performance measures of our method is comparable to state-of-the-art methods.

Keywords: Knowledge discovery · Data mining · Classification · Decision trees · Sampling strategy

1 Introduction

In Machine Learning community, and in data mining works, classification has its own importance. Classification is an important part and the research application field in the data mining [1].

A decision tree gets its name because it is shaped like a tree and can be used to make decisions. ―Technically, a tree is a set of nodes and branches and each branch descends from a node to another node. The nodes represent the attributes considered in the decision process and the branches represent the different attribute values. To reach a decision using the tree for a given case, we take the attribute values of the case and traverse the tree from the root node down to the leaf node that contains the decision [2]. "A critical issue in artificial intelligence (AI) research is to overcome the

O. Gervasi et al. (Eds.): ICCSA 2015, Part I, LNCS 9155, pp. 90–97, 2015.
DOI: 10.1007/978-3-319-21404-7_7

so-called-knowledge-acquisition bottleneck" in the construction of knowledge-based systems. Decision tree can be used to solve this problem. Decision trees can acquire knowledge from concrete examples rather than from experts [3]. In addition, for knowledge-based systems, decision trees have the advantage of being comprehensible by human experts and of being directly convertible into production rules [4].

A decision tree not only provides the solution for a given case, but also provides the reasons behind its decision. So the real benefit of decision tree technology is that it avoids the need for human expert. Because of the above advantages, there are many successes in applying decision tree learning to solve real-world problems.

To summarize, the contributions of this paper are as follows:

1. A sampling strategy is extended in the decision learning model.
2. Empirical evaluation on a wide variety of real world datasets, and establishing the superiority of the new framework.
3. Analyzing the performance of the methods using the measures of diversity.

The paper is organized as follows. In Sect. 2 we present the recent advances in decision tree learning. This will directly motivate the main contribution of this work presented in Sect. 3, where we propose a new framework for sampling strategic learning. Evaluation criteria's for decision tree learning is presented in section 4. Experimental results are reported in Sect. 5. Finally, we conclude with Sect. 6 where we discuss major open issues and future work.

2 Recent Advances in Decision Trees

In Data mining, the problem of decision trees has also become an active area of research. In the literature survey of decision trees we may have many proposals on algorithmic, data-level and hybrid approaches. The recent advances in decision tree learning have been summarized as follows:

A parallel decision tree learning algorithm expressed in MapReduce programming model that runs on Apache Hadoop platform is proposed by [5]. A new adaptive network intrusion detection learning algorithm using naive Bayesian classifier is proposed by [6]. A new hybrid classification model which is established based on a combination of clustering, feature selection, decision trees, and genetic algorithm techniques is proposed by [7]. A novel roughest based multivariate decision trees (RSMDT) method in which, the positive region degree of condition attributes with respect to decision attributes in rough set theory is used for selecting attributes in multivariate tests is proposed by [8].

A novel splitting criteria which chooses the split with maximum similarity and the decision tree is called mstree is proposed by [9]. An improved ID3 algorithm and a novel class attribute selection method based on Maclaurin-Priority Value First method is proposed by [10]. A modified decision tree algorithm for mobile user classification, which introduced genetic algorithm to optimize the results of the decision tree algorithm, is proposed by [11]. A new parallelized decision tree algorithm on a CUDA (compute unified device architecture), which is a GPGPU solution provided by NVIDIA is proposed by [12]. A Stochastic Gradient Boosted Decision Trees based method is proposed by [13]. A modified Fuzzy Decision Tree for the fuzzy rules extraction is proposed by [14].

Obviously, there are many other algorithms which are not included in this literature. A profound comparison of the above algorithms and many others can be gathered from the references list.

3 The Proposed Method

In this section, the proposed approach is presented.

The proposed approach follows a sampling strategic approach for continuous improvement. The decision tree performs classification in two stages. In the first stage it builds model from the training instances available and in the second stage it validates the testing instances using the build model. The efficiency of the decision tree is evaluated on the testing instances. If a normal or balance data source is provided as input to the decision tree then the model build is efficient enough to classify the testing instances with considerable efficiency.

If the data source provided to the decision tree is of imbalance nature i.e; Let us consider the dataset is of binary class. One class has predominantly more number of instances than the other class; then we may say that type of dataset as an imbalance dataset. The instances in one class can be 95% and in other class it can be 5%. If the decision tree uses the dataset for both training and testing and it follows training-testing strategy of 66-33% or 10 Fold cross validation (CV) there is a great chance that the training set will contain instances of only one class(class of 95% instances). The model build by decision tree using training instances of only one class may not be an efficient model. In the validation phase when the above build model is used for testing instances then definitely the model will encounter some of the instances which it has not seen, then the question comes, "IF IT HAS SEEN NO INSTANCES, HOW CAN IT KNOW?". We proposed A Novel Prototype Decision Tree Method using Sampling Strategy as our problem for investigation.

We designed a sampling strategy which can solve the above limitation of decision trees. One of the solutions is to allow decision trees to build an efficient model by using the instances of all the classes in the dataset. If a binary imbalance datasets encountered in the decision tree learning process the selective sampling can be performed to the class which has very less percentage of instances.

The above said strategy is implemented in the proposed system. In the initial stage the decision tree learning process will initiate with the identification of data source as normal or imbalance dataset. A threshold (Imbalance ratio) value is provided for classification of the data source as a normal or imbalance dataset.

In the next stage, if the data source is identified as an imbalance dataset then the class with less percentage of instances is identified and the proposed sampling strategy is implemented. The resampling is done by replication and hybridized instances. The percentage of synthetic instances generated will range from 0 – 100 % depending upon the percentage of difference of majority and minority classes in the original dataset. The synthetic minority instances generated can have a percentage of instances which can be a replica of the pure instances and reaming percentage of instances are of the hybrid type of synthetic instances generated by combing two or more instances from the pure minority subset. In the next and final phase a base algorithm is used to evaluate the improved dataset.

4 Experimental Design and Evaluation Criteria's

We used the open source tool Weka [16] and implemented our proposed model. In order to test the robustness of our method it is compared to existing methods C4.5 [17], Classification and Regression Trees (CART) [18], Functional Trees [FT], Reduced Error Pruning Tree (REP), and SMOTE[19] in our experiments.

In order to compare the classifiers, we use 10-fold cross validation. In 10-fold cross validation, each dataset is broken into 10 disjoint sets such that each set has (roughly) the same distribution. The classifier is learned 10 times such that in each iteration a different set is withheld from the training phase, and used instead to test the classifier. We then compute the measures as the average of each of these runs.

To assess the classification results we count the number of true positive (TP), true negative (TN), false positive (FP) (actually negative, but classified as positive) and false negative (FN) (actually positive, but classified as negative) examples. It is now well known that error rate is not an appropriate evaluation criterion when there is class imbalance or unequal costs. In this paper, we use AUC, Precision, F-measure, TP Rate and TN Rate as performance evaluation measures.

Let us define a few well known and widely used measures:

The Accuracy (ACC) measure is computed by equation (1) ,

$$AUC = \frac{1 + TP_{RATE} - FP_{RATE}}{2} \tag{1}$$

The Area under Curve (AUC) measure is computed by equation (2),

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FN} + \mathrm{FP} + \mathrm{FN}} \tag{2}$$

The Precision measure is computed by equation (3),

$$\Pr ecision = \frac{TP}{(TP) + (FP)} \tag{3}$$

The F-measure Value is computed by equation (4),

$$F - measure = \frac{2 \times \Pr ecision \times \mathrm{Re}\, call}{\Pr ecision + \mathrm{Re}\, call} \tag{4}$$

The True Positive Rate measure is computed by equation (5),

$$TruePositi\ veRate = \frac{TP}{(TP)+(FN)} \tag{5}$$

The True Negative Rate measure is computed by equation (6),

$$TrueNegati\ veRate = \frac{TN}{(TN)+(FP)} \tag{6}$$

DATASETS USED IN DECISION TREE LEARNING

Table 1 summarizes the datasets used in the proposed study from UCI [15].

Table 1. Summary of imbalanced datasets

S.no Datasets	# Ex.	# Atts.	Class (_,+)	IR
1. Abalone19	4174	9	(32; 1412)	1:130
2. Abalone19-18	731	9	(42; 689)	1:17
3. Shuttle-c0-vs-c4	1829	10	(123:1706)	1: 14
4. Vowel0	988	14	(90:898)	1: 10
5. Yeast-0-5	528	9	(51:477)	1: 9.4

The details of the datasets are given in table 1. For each data set, S.no., name of the dataset, number of instances, Classes, imbalance ratio (IR) are descried in the table for all the datasets.

5 Results

In this section, we carry out the empirical comparison of our proposed algorithm with the benchmarks. Our aim is to answer several questions about the proposed learning algorithms in the scenario of two-class imbalanced problems.

1. In first place, we want to analyze which one of the approaches is able to better handle a large amount of imbalanced data-sets with different IR, i.e., to show which one is the most robust method.
2. We also want to investigate their improvement with respect to classic decision tree methods and to look into the appropriateness of their use instead of applying a unique preprocessing step and training a single method. That is, whether the trade-off between complexity increment and performance enhancement is justified or not. Given the amount of methods in the comparison, we cannot

afford it directly. On this account, we compared the proposed algorithm with each and every algorithm independently. This methodology allows us to obtain a better insight on the results by identifying the strengths and limitations of our proposed method on every compared algorithm.

Table 2 shows the detailed experimental results of the mean classification accuracy, AUC, Precision, Recall, F-measure of C4.5, CART, FT, REP, SMOTE and Proposed Algor. on all the data sets. From Table 2 we can see that the performance of accuracy of our proposed model achieved substantial improvement over C4.5, CART, FT, REP and SMOTE on most data set which suggests that the proposed model is potentially a good technique for decision trees.

Table 2. Summary of tenfold cross validation performance for proposed algorithm on all the datasets

Datasets	**C4.5**	**CART**	**FT**	**REP**	**SMOTE**	**Proposed**
			Accuracy			
Abalone19	99.23±0.096	99.23 ±0.096	99.23±0.096	99.21±0.114	91.21±2.639●	99.55±0.203
Abalone19-18	93.982±2.053●	94.51±1.338●	95.40±1.559●	94.31±1.581●	98.46±0.173○	97.07±1.679
Shuttle-c0-vs-c4	99.94±0.16	100.0±0.000	99.94±0.165	100.00 ±0.000	100.0±0.000	99.94±0.181
Vowel0	98.92±1.064●	98.23±1.394●	98.28±1.224●	98.29±1.453●	99.12±0.882	99.25±0.991
Yeast-0-5vs4	90.21±3.22●	91.06±2.927●	92.46±2.810●	91.08±3.060●	87.89±3.762●	95.05±1.725
			AUC			
Abalone19	0.500±0.000●	0.500±0.000●	0.500±0.000●	0.510±0.053●	0.745±0.098○	0.685±0.144
Abalone19-18	0.623±0.143●	0.605±0.123●	0.818±0.118●	0.631±0.134●	0.511±0.047●	0.805±0.149
Shuttle-c0-vs-c4	1.000±0.001	1.000±0.000	1.000±0.000	1.00±0.000	1.000±0.000	1.000±0.001
Vowel0	0.966±0.050●	0.949±0.065●	0.960±0.061●	0.957±0.052●	0.984±0.019○	0.968±0.054
Yeast-0-5_vs_4	0.720±0.172○	0.749±0.150○	0.769±0.110○	0.744±0.159○	0.851±0.075○	0.698±0.143
			Precision			
Abalone19	0.000±0.000●	0.000±0.000●	0.000±0.00●	0.00±0.000●	0.705±0.222○	0.297±0.236
Abalone19-18	0.384±0.034●	0.343±0.418●	0.669±0.353○	0.288±0.405●	0.010±0.100●	0.624 ±0.323
Shuttle-c0-vs-c4	0.993±0.023○	1.000±0.00○	1.000±0.000○	1.000±0.00○	1.00±0.00○	0.989±0.032
Vowel0	0.952±0.068○	0.915±0.090●	0.924±0.077●	0.923±0.102●	0.977±0.036○	0.946±0.099
Yeast-0-vs_4	0.510±0.241○	0.529±0.032○	0.683±0.244○	0.510±0.332○	0.672±0.125○	0.255±0.118
			Recall			
Abalone19	0.000±0.000●	0.000±0.000●	0.000 ±0.000●	0.000±0.000●	0.412±0.163○	0.221±0.183
Abalone19-18	0.194±0.214●	0.155±0.198●	0.360±0.225●	0.138±0.204●	0.002±0.017●	0.550±0.304
Shuttle-c0-vs-c4	1.000±0.000	1.000±0.000	0.992±0.025	1.000±0.000	1.00±0.000	1.000±0.000
Vowel0	0.933±0.082○	0.898±0.112●	0.892±0.111●	0.902±0.111●	0.972±0.036○	0.923±0.117
Yeast-0-vs_4	0.413±0.226○	0.352±0.225○	0.475±0.204○	0.351±0.239○	0.657±0.155○	0.275±0.131
			F-measure			
Abalone19	0.000±0.000●	0.000±0.000●	0.000±0.000●	0.000±0.000●	0.494±0.159○	0.240±0.182
Abalone19-18	0.242±0.250●	0.201±0.224●	0.441±0.238●	0.175±0.245●	0.003±0.029●	0.559±0.277
Shuttle-c0-vs-c4	0.996±0.012	1.000±0.000○	0.996±0.013	1.000±0.000○	1.00±0.000○	0.994±0.018
Vowel0	0.940±0.061○	0.901±0.081●	0.902±0.072●	0.906±0.081●	0.974±0.026○	0.932±0.092
Yeast-0-vs_4	0.431±0.198○	0.402±0.236○	0.534±0.190○	0.396±0.248○	0.652±0.112○	0.254±0.106

● Bold dot indicates the win of proposed method; ○ Empty dot indicates the loss of proposed method.

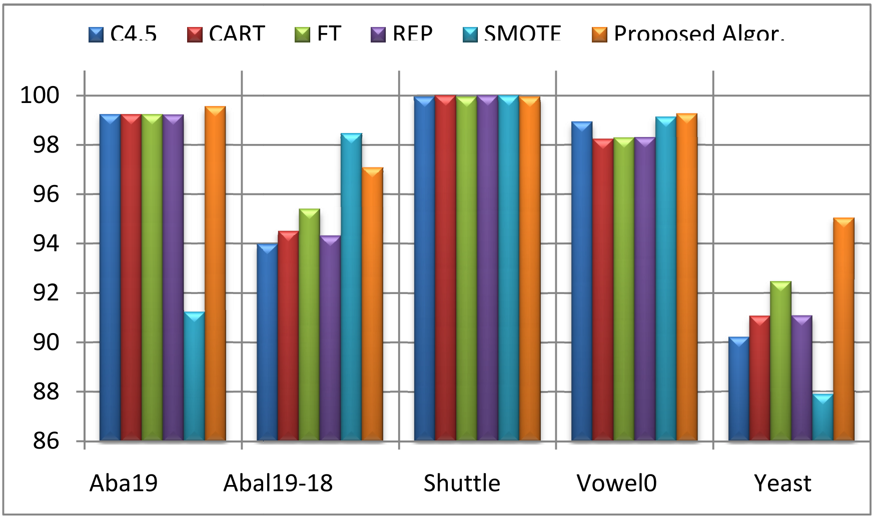

Fig. 1. Test results on accuracy on C4.5, CART, FT, REP, SMOTE and Proposed Algor. for all datasets.

The proposed method had also gained significantly improvement in terms of AUC over C4.5, CART, FT, REP and SMOTE and is comparable to two state-of-the-art technique for decision trees. The performance of the proposed method is improved on almost all the datasets for the measures of precision, recall and f-measure.

Figure 1 shows the detailed pictorial representation of the accuracy results for all the compared algorithms C4.5, CART, FT, REP and SMOTE on all the data sets. From Table 2 and Figure 1 we can see that our proposed approach had given a proper solution for the investigated question.

Finally, we can say that the proposed model is one of the best alternatives to handle class imbalance problems effectively in decision trees. This experimental study supports the conclusion that the a proper sampling strategy can improve the performance of decision when dealing with imbalanced data-sets, as it has helped the proposed method to be the best performing algorithm when compared with five classical and well-known algorithms.

6 Conclusion

In this paper, we proposed a sampling strategy for decision trees. The proposed algorithm mimics human learning approach. We posited that without building proper model the decision trees cannot perform better. Applying human learning in machine spaces will lead to an improved performance due to dynamic plaining. To test this hypothesis we ran experiments on 5 widely available datasets from UCI. We then compared this method with traditional benchmark algorithms. From these results it is apparent that our proposed approach is a competitive one amongst the benchmarks.

References

1. Juanli, H., Deng, J., Sui, M.: A new approach for decision tree based on principal component analysis. In: Proceedings of Conference on Computational Intelligence and Software Engineering, pp. 1–4 (2009)
2. Bergsma, S.: Large-scale semi-supervised learning for natural language processing. PhD Thesis, University of Alberta (2010)
3. Durkin, J.: Expert systems: design and development. Prentice Hall, Englewood Clis (1994)
4. Quinlan, J.: Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA (1993)
5. Purdila, V., Pentiuc, S.-G.: MR-Tree - A Scalable MapReduce Algorithm for Building Decision Trees. Journal of Applied Computer Science & Mathematics, 16(8) (2014). Suceava
6. Farid, D.M., Harbi, N., Mohammad Zahidur, R.: Combining naive bayes and decision tree for adaptive intrusion detect. International Journal of Network Security & Its Applications (IJNSA), 2(2) (April 2010)
7. Mohammad, K., Mahmood, A.: The Use of Genetic Algorithm, Clustering and Feature Selection Techniques in Constrcution of Decision Tree Models for Credit Scoring. International Journal of Managing Information Technology (IJMIT) 5(4) (November 2013). doi:10.5121/ijmit.2013.5402
8. Dianhong, W., Xingwen, L., Liangxiao, J., Xiaoting, Z., Yongguang, Z.: Rough Set Approach to Multivariate Decision Trees Inducing? Journal of Computers, 7(4) (April 2012)
9. Xinmeng, Z., Shengyi, J.: A Splitting Criteria Based on Similarity in Decision Tree Learning. Journal of Software, 7(8) (August 2012)
10. Ying, W., Xinguang, P., Jing, B.: Computer Crime Forensics Based on Improved Decision Tree Algorithm. Journal of Networks, 9(4) (April 2014)
11. Dong-sheng, L., Shujiang, F.: A Modified Decision Tree Algorithm Based on Genetic Algorithm for Mobile User Classification Problem. Scientific World Journal, Article ID 468324, 11 (2014). Hindawi Publishing Corporation. http://dx.doi.org/10.1155/2014/468324
12. Win-Tsung, L., Yue-Shan, C., Ruey-Kai, S., Chun-Chieh, C., Shyan-Ming, Y.: CUDT: A CUDA Based Decision Tree Algorithm. Scientific World Journal, Article ID 745640, 12 (2014). Hindawi Publishing Corporation. http://dx.doi.org/10.1155/2014/745640
13. Tarun, C., Jayashri, V.: Fault Diagnosis in Benchmark Process Control System Using Stochastic Gradient Boosted Decision Trees. International Journal of Soft Computing and Engineering (IJSCE), 1(3) (July 2011). ISSN: 2231-2307
14. Ganga Devi, S.V.S.: Fuzzy Rule Extraction for Fruit Data Classification. Compusoft, An international journal of advanced computer technology, 2(12) (December 2013)
15. Hamilton, A., Asuncion, D., Newman.: UCI Repository of Machine Learning Database (School of Information and Computer Science). Univ. of California, Irvine (2007). http://www.ics.uci.edu/~mlearn/MLRepository.html
16. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
17. Quinlan, J.: Induction of decision trees. Machine Learning 1, 81–106 (1986)
18. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Wadsworth, Belmont (1984)
19. Chawla, N.V., et al.: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research. 16, 321–357 (2002)