ComboSplit: Combining Various Splitting Criteria for Building a Single Decision Tree

Md Nasim Adnan and Md Zahidul Islam
Centre Centre for Research in Complex Systems (CRiCS)
School of Computing and Mathematics, Charles Sturt University
Bathurst, NSW 2795, Australia
{madnan, zislam}@csu.edu.au

ABSTRACT

Typically existing decision tree building algorithms use a single splitting criterion such as Gain Ratio and Gini Index. In this paper three existing splitting criteria are compared within the framework of the C4.5 decision tree building algorithm. We also propose a technique called ComboSplit for combining the existing splitting criteria to build a single decision tree. We experimentally evaluate the decision trees obtained by various existing splitting criteria and ComboSplit. Ten publicly available datasets are used in the experiments. Decision Trees obtained by ComboSplit generally have higher prediction accuracy than the trees obtained by the existing splitting criteria.

KEYWORDS

Data Mining, Classification, Decision Tree, Splitting Criteria, Prediction Accuracy.

1 INTRODUCTION

Due to the advances in the technology related to the data storage and information processing a huge amount of data is being collected these days. Around 2.5 quintillion bytes of data are being added in a single day — so much that 90% of the data in the world today have been generated in the last two years alone [1]. Knowledge discovery and pattern recognition from the collected data are crucial in order to make the best use of the data. Therefore, various data mining tasks such as classification and clustering are typically applied on a dataset. In this paper, we consider a dataset D as a two-dimensional table where rows are records $R = \{R_1, R_2, ..., R_n\}$ and columns are attributes $A = \{A_1, A_2, ..., A_{m'}\}$. We also consider that a dataset can have two types of attributes; numerical (such as Age) and categorical (such as Country). Out of the categorical attributes of the dataset, one is chosen as the class attribute. All other attributes are termed as the classifier (or non-class) attributes. From this point we denote the set of non-class attributes by $\{A_1, A_2, ..., A_m\}$, the class attribute by C, and the complete set of attributes by $\{A_1, A_2, ..., A_m, C\}$. The classification task aims to map the set of classifier attributes $\{A_1, A_2, ..., A_m\}$ to a predefined class attribute C [2]. That is, it discovers the relationship between the non-class attributes and the class attribute. The function is commonly known as a classification model, which can be used for the pattern extraction (knowledge discovery) from a dataset. A classification model can also be used for the prediction of the class value of an unlabelled record. An unlabeled record is a record that does not have a class value assigned. For example, in a hospital dataset the record related to a patient who has not been diagnosed yet with a disease is an unlabeled record, where Diagnosis is the class attribute.

There are different types of classifiers including Decision Trees, Bayesian classifiers, and Artificial Neural Networks [2]. There are also many decision tree algorithms such as ID3 [2], C4.5 [4, 5], and Explore [9]. We now give a brief introduction to the C4.5 algorithm since our proposed technique uses C4.5 in this study.

A decision tree induction is a recursive process which builds a decision tree from a training dataset, i.e. a dataset where all records are labeled with class attribute values. The induction process starts by selecting a non-class attribute A_i (also known as a splitting attribute) to divide the training dataset D into a set of mutually exclusive horizontal segments/partitions [3, 4, 5, 9]. If the splitting attribute A_i is categorical with k different domain values i.e. $A_i= \{a_{i1}, a_{i2}, \ldots a_{ik}\}$ (domain values of an attribute are the set of all possible values for the attribute) then D is divided (split) into k segments $D_1, D_2, \ldots D_k$, where all records of a segment D_j have the same value a_{ij}, and the records belonging to different segments have different values [3, 4]. However, if the splitting attribute is numerical $A_i=[l,u]$ (l is the lower limit and u is the upper limit of the domain values of A_i) then the dataset is typically divided into two segments; D_1, and D_2. All records in the segment D_1 have values of A_i “lower than or equal to” a splitting point p and the records in the other segment D_2 have values higher than the splitting point p, where $l<p<u$ [5, 9].

The purpose of this splitting is to create a purer distribution of class values in the succeeding partitions/segments than the distribution in D. Therefore, for a numerical attribute all possible split points (i.e. all values between l and u) are used to find out the split point that gives the best distribution of class values. Finally, the splitting attribute that gives the purest class distribution among all splitting attributes is selected as the test attribute.

The process of selecting the test attribute continues recursively in each subsequent data segment D_i until either every partition gets the purest class distribution or a stopping criterion is satisfied. By “purest class distribution” we mean the presence of a single class value $c_i \in C$ for all records.

A decision tree consists of nodes and leaves as shown in Fig. 1, where the nodes are denoted by rectangles and leaves are denoted by ovals. The shortest path from the root node to a leaf node makes up a logic rule that identifies a relationship between the non-class attributes and the class attribute. For example, the logic rule for Leaf 1 is “*if Degree = Masters AND Income < 85K* → *Profession = Lecturer*”. Every record of a training dataset satisfies one and only one logic rule (i.e. leaf) of a decision tree.

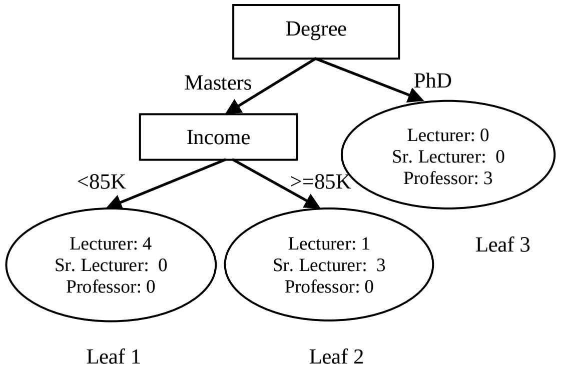

Figure 1: Decision tree

While a decision tree is built on a training dataset D it can be applied on a testing dataset D' (where the records are unlabeled) in order to predict the class values of the records of D'. Based on the non-class attribute values of an unlabeled record; the leaf where the record falls in, is first identified. The majority class value (i.e. the class value of the majority records) of the leaf is then predicted as the class value of the unlabeled record. It is desirable that a tree achieves high prediction accuracy on a testing dataset.

For the quest of high prediction accuracy, it is important that we find the best splitting attribute and the splitting point/s for each node. In the literature, there are many splitting criteria that aim to find the best splitting attribute. Some of the most renowned splitting criteria are Gain Ratio [4, 5], Gini Index [6] and DCSM [7]. While there are some studies [10, 11, 12] that discuss various splitting criteria they do not offer a clear empirical analysis to demonstrate which one is better than the others.

Therefore, in this study we first use different splitting criteria one by one within the C4.5 algorithm and compare the prediction accuracies of the trees obtained by them. We use 10 publicly available datasets [13], where five of them have low prediction accuracy (less than 75%) and the remaining five have high accuracy (greater than 90%). We find that the prediction accuracies are very close to each other and there is no clear winner among the criteria.

Hence, in this study we also propose a technique called ComboSplit for combining all splitting criteria to find the best splitting attribute and the best splitting point/s. We build decision trees on the 10 datasets using ComboSplit within the C4.5 algorithm. Our experimental results indicate a clear superiority (in terms of prediction accuracy) of ComboSplit over the other splitting criteria, especially on the low accuracy datasets. It is also indicated that while combining the splitting criteria ComboSplit does not overly rely on a single criterion. This also indicates the absence of a clear winner among the criteria and further emphasizes the importance of ComboSplit.

ComboSplit may not be suitable for time-critical applications where a decision needs to be made on the fly. However, there are many other scenarios where a higher execution time is tolerated in order to achieve higher prediction accuracy. An example of such scenarios can be medical research. Moreover, it is also seen from the literature that often it is not possible to achieve a huge increase in prediction accuracy [3, 14]. Therefore, we thankfully accept any possible improvement in accuracy. Although ComboSplit is a simple idea it was never suggested in the literature (to the best of our knowledge), and it generally achieves higher prediction accuracy than other splitting criteria.

The structure of the paper is organized as follows. Section 2 presents a Background Study. In Section 3 we describe our novel technique. Experimental results are presented in Section 4 and Section 5 draws the concluding remarks.

2 BACKGROUND STUDY

In this section we describe some of the most popular node splitting criteria namely Gain Ratio [4, 5], Gini Index [6] and DCSM [7]. For convenience, we use the notations shown in Table 1.

Table 1: Training Dataset

Notations	Description
$D=\{R_1, R_2, \dots R_n\}$	A Dataset having n records $R_1, R_2, \dots R_n$.
$\|D\|$	The number of the records of the dataset D.
$A=\{A_1, A_2, \dots A_m\}$	The set of attributes in D.
$A_i=\{a_{i1}, a_{i2}, \dots a_{ip}\}$	An attribute having p domain values.
$\|A_i\|$	The domain size (p) of A_i.
$C = \{c_1, c_2, \dots c_q\}$	The class attribute having q class values.
$\|C\|$	The domain size of the class attribute.
D_i	The i-th partition (i.e. segment) of D.
$\|D_i\|$	The total number of the records in D_i.
$c(D_i)$	The domain size of the class attribute in D_i.
$\|D_i^j\|$	The number of the records having the j-th class value within the partition D_i.
$\|D_i^M\|$	The number of the records having the majority class value within the partition D_i.
$P(D, j)$	The proportion of records in D belonging to the j-th class value. That is, $\|D_i^j\|$ over $\|D_i\|$.

Gain Ratio: The splitting criterion called Gain Ratio is an advanced version of “Information Gain”, which is used in ID3 [8]. Again the Information Gain of an attribute is calculated using the Entropy, which can be computed for the whole dataset D as follows [4, 5].

$$Entropy\,(D) = -\sum_{j=1}^{|C|} p(D,j) log_2\left(p(D,j)\right) \qquad (1)$$

A dataset D can be divided into p partitions (D_1, D_2, ... D_p) based on an attribute A_i (if the domain size of A_i is p). Now, the overall Entropy of the class distribution following the partitioning based on the values of A_i is calculated as follows.

$$Entropy\ (A_i, D) = \sum_{i=1}^{p} \frac{|D_i|}{|D|} \cdot Entropy(D_i) \qquad (2)$$

The Information Gain of the attribute A_i (when it divides D) is calculated as follows:

$$Info\ Gain\ (\mathrm{A_i}, D) = Entropy\ (D) - Entropy(A_i, D) \qquad (3)$$

Finally, the Gain Ratio of the attribute A_i is calculated as follows:

$$Gain\ Ratio\ (\mathrm{A_i}, D) = \frac{Info\ Gain\ (\mathrm{A_i}, D)}{Split\ Info\ (\mathrm{A_i}, D)} \qquad (4)$$

The Split Info of the attribute A_i (for D) is calculated as follows.

$$Split\ Info\ (\mathrm{A_i}, D) = \sum_{i=1}^{p} \frac{|D_i|}{|D|} \cdot log_2 \frac{|D_i|}{|D|} \qquad (5)$$

Gini Index: Gini Index [6] calculates the *Gini Gain* of an attribute A_i (similar to the Information Gain presented in Eq. 3) as follows:

$$Gini\ Gain\ (\mathrm{A_i}, D) = Gini\ (D) - Gini\ (\mathrm{A_i}, D) \qquad (6)$$

$$Gini\ (D) = 1 - \sum_{j=1}^{|C|} (p(D, j))^2 \qquad (7)$$

DCSM: A recent splitting criterion called DCSM [7] calculates the *DCSM Gain* by *DCSM* (D) – *DCSM* $(\mathrm{A_i}, D)$, where *DCSM* is calculated by Equation 8. It is worth mentioning that Gini Index and DCSM do not take the Split Info of an attribute into consideration and thereby only compute Gain instead of Gain Ratio.

$$DCSM\ (D) = \sum_{i=1}^{p} [\frac{|D_i|}{|D|} \cdot c(D_i) \exp(c(D_i)) \sum_{j=1}^{|C|} p(D_i, \mathrm{j}) \cdot \exp\ (\frac{c(D_i)}{c} \cdot (1 - (p(D_i, j))^2))] \qquad (8)$$

Both *Gini* $(\mathrm{A_i}, D)$ and *DCSM* $(\mathrm{A_i}, D)$ are calculated in a similar way how *Entropy* $(\mathrm{A_i}, D)$ is calculated in Eq. 2.

3 OUR PROPOSED TECHNIQUE

We propose ComboSplit to determine the best splitting attribute (and corresponding splitting point/s) by simultaneously using multiple splitting criteria at every node, while building a single decision tree. ComboSplit can use as many splitting criteria as desired by a user. In this paper we use three existing splitting criteria namely Gain Ratio [4, 5], Gini Index [6] and DCSM [7]. Once the best splitting attribute and corresponding splitting point/s are determined for a node ComboSplit uses any existing decision tree building algorithm. In this study we use C4.5 [8]. We next present the ComboSplit algorithm (see Algorithm 1) and then explain the two main steps (Step 1 and Step 2) in details. The other steps of ComboSplit (as shown in Algorithm 1) are identical to any existing decision tree building algorithm such as C4.5 [8].

```
BuildTree (D):
    S ← FindCandidateSplits (D)     /* Explained in Step 1*/
    B = FindBestSplit (S)           /* Explained in Step 2*/
    if Gain Ratio (B, D) < $\theta$       /* $\theta$ is a user defined
        MakeLeaf (D)                threshold as used in C4.5 */
    end if
    else
        D′ ← Split (B, D)           /* D′ is a set of data segments */
        for each $D_i \in D'$
            BuildTree (Di)
        end for
    end else
end BuildTree (D)
```

Algorithm 1: ComboSplit

Step 1: Find the Candidate Splits

In this step a set S of splitting attributes and corresponding splitting point/s are selected using a number of splitting criteria such as Gain Ratio and Gini Index. Each splitting criteria selects the best splitting attribute and point/s according to the criteria. If a splitting attribute is numerical then the best splitting point is chosen, however if the splitting attribute is categorical then all values of the attribute are used as the splitting points. Each $S_i \in$

S contains information on a splitting attribute and corresponding splitting point/s.

This step can be useful if different splitting criteria choose different splitting attributes and points. From the equations discussed in Section 2 it is clear that different splitting criteria are not unlikely to choose different splitting attributes and points. To further investigate the possibility of different splitting criteria choosing different splitting attributes and points, we also carry out an empirical analysis on the *Si* (meaning *Silicon*) attribute of the *Glass Identification* dataset [13]. The *Si* attribute is numerical with the domain size of 133. Form our experimental results the best splitting point of *Si* according to Gain Ratio is the 114th domain value, whereas the best splitting point according to Gini Index is the 112th domain value of the attribute (see Table 2). DCSM also selects the 112th domain value as the splitting point.

Moreover, in our experiments (as presented in Section 4) we find that the trees built by ComboSplit choose splitting attributes and points from all three splitting criteria used in this study instead of choosing all splitting attributes from a single criterion (see Table 6). For example, the ComboSplit tree built from the Glass Identification dataset has 61.60% splitting attributes according to Gain Ratio and 36.29% attributes according to Gini Index. Therefore, Table 6 shows that different splitting criteria are likely to choose different splitting attributes and point/s.

Step 2: Select the Best Splitting Attribute.

In this step the splitting quality of each splitting attribute and corresponding splitting point/s are evaluated using any existing rule evaluation criterion. In this study we use m-estimate (see Eq. 9) [2]. If a splitting attribute $A_i = \{a_{i1}, a_{i2}, \ldots a_{ip}\}$ divides a dataset D into p mutually exclusive horizontal segments $D = \{D_1, D_2, \ldots D_p\}$ then the m-estimate M_j of each segment D_j is calculated using Eq. 9. Finally, the overall m-estimate for the splitting attribute is calculated using Eq. 10.

$$M_j = \frac{\left|D_j^M\right| + \left(|\mathrm{C}| \cdot \frac{\left|D_j^M\right|}{\left|D_j\right|}\right)}{\left|D_j\right| + |C|} \tag{9}$$

$$M = \sum_{j=1}^{p} \frac{|D_j|}{|D|} \cdot \mathrm{M_j} \tag{10}$$

Table 2: The Best Splitting Points for the *Si Attribute* for Different Splitting Criteria

Splitting Criteria	The Best Splitting Point
Gain Ratio	114th domain value
Gini Index	112th domain value
DCSM	112th domain value
m-estimate	33rd and 34th domain value

Out of all candidates, the splitting attribute that produces the maximum m-estimate is chosen as the ultimate splitting attribute for the data segment as shown in Algorithm 1. Once the splitting attributes and corresponding splitting points are selected a decision tree is built using an approach similar to the C4.5 algorithm, as shown in Algorithm 1 for ComboSplit.

An interesting characteristic of our algorithm is that while m-estimate is used to evaluate the splitting attributes produced by existing splitting criteria m-estimate is not independently used to identify the best splitting attribute and point/s, like Gain Ratio. That is, we do not replace Gain Ratio by m-estimate in C4.5. There are two main reasons for this as follows.

First, if m-estimate is used to identify the splitting attribute and point/s independently (like Gain Ratio) then it may choose a totally different splitting attribute and point/s. Table 2 shows that for the *Si* attribute it chooses the 33rd or 34th domain value as the splitting point which is very different from the splitting points chosen by other splitting criteria.

Second, it appears to us that existing splitting criteria are more sophisticated than m-estimate and thereby should identify better splitting attributes than m-estimate. It is evident from the equations (from Eq. 1 to Eq. 10) that while Gain Ratio, Gini Index and DCSM consider the distribution of the class values in succeeding partitions, m-estimate only emphasizes on the majority class values in the succeeding partitions. That is, while Gain Ratio cares for the proportion of records having the majority class value (i.e. $|D_i^M|/|D_i|$) it also considers the distribution of other class values. M-estimate does not care for the distribution of other class values. We explain this with an example as follows.

Let, D be a training dataset with the attribute set $\{A_1, A_2, C\}$, where A_1 and A_2 are numerical attributes and the domain values of C are c_1, c_2 and c_3. D has 40 records in total, where 30 records have the c_1 class value, 5 records have c_2 and 5 records have c_3. The attribute A_1 divides the dataset D into two segments, $D_{1,1}$ and $D_{1,2}$ whereas A_2 divides D into two other segments $D_{2,1}$ and $D_{2,2}$. In Table3, we present the distributions of the class values in the succeeding partitions for A_1 and A_2. We find that the m-estimate values for A_1 and A_2 are exactly the same indicating that m-estimate is unresponsive to the distribution of the class values. However, all three other criteria achieve higher values for the distribution in A_2 than for the distribution in A_1.

Table 3: Example Splitting of a Dataset

Splitting Attribute	Succeeding Partitions	Class Distribution			Gain Ratio	Gini Index	DCSM	m-estimate
		c_1	c_2	c_3				
A_1	$D_{1,1}$	20	3	2	0.024	0.008	50.053	0.125
	$D_{1,2}$	10	2	3				
A_2	$D_{2,1}$	20	5	0	0.278	0.040	95.531	0.125
	$D_{2,2}$	10	0	5				

Now, although m-estimate is not used independently we consider it as a useful metric to evaluate the splitting quality of the existing criteria. This way instead of giving m-estimate the option to choose a splitting attribute from the set of all attributes and points it is given the option to choose the best splitting attribute from the set of candidate splits, where each element of the set is already chosen by a sophisticated splitting criterion. This approach should improve the prediction accuracy since for a future unlabeled record the class value is predicted based on the majority class value only, and m-estimate chooses the candidate split that has the largest number of records with the majority class value. Our experimental results in Section 4 also indicate the effectiveness of the approach.

4 EXPERIMENTAL RESULTS

We evaluate ComboSplit by comparing it with three existing splitting criteria on 5 low-accuracy datasets (where the prediction accuracy is less than 75%) and 5 high accuracy datasets (where the prediction accuracy is higher than 90%) that are publicly available from the UCI Machine Learning Repository [13]. We remove the records with missing values from the datasets. We also remove the identifier attributes such as Staff_ID. We use 10-fold-cross-validation (10-CV) [3] for every dataset. The minimum Gain Ratio value (i.e. θ in Algorithm 1) in this study is 0.01. The minimum number of records in each leaf is 2.

A few important properties of the low accuracy datasets are presented in Table 4. For example, the Glass Identification dataset has nine numerical attributes, zero categorical attributes, 214 records and 7 class-attribute values.

Table 4: Description of the low-accuracy Datasets

Dataset Name	Number of non-class Attributes		Number of Records	Domain Size of the Class Attribute
	Num.	Cat.		
Glass Identification	9	0	214	7
Hayes-Roth	0	4	132	4
Pima Indians Diabetes	0	8	768	2
Statlog Vehicle	18	0	846	4
Liver Disorders	6	0	345	2

Prediction accuracy is one of the most commonly used performance indicators of a decision tree. Therefore, in Table 5, we present the prediction accuracies of the C4.5 trees using Gain Ratio, Gini Index, and DCSM as the splitting criteria. We also present the prediction accuracies of the ComboSplit decision trees.

The best result for each dataset is presented in a bold font**.** From Table 5, it is evident that the ComboSplit performs the best for all low-accuracy datasets except Liver Disorder (LD) where it performs the second best. In LD it performs slightly worse than Gain Ratio, 66.92% as opposed to 67.29%. In terms of the average prediction accuracy over all five datasets in Table 5 ComboSplit performs the best. The numbers in the parentheses show the relative positions of the techniques.

ComboSplit achieves the best prediction accuracy perhaps due to its ability to switch between the splitting criteria in order to pick the best splitting attribute for each node. That is, ComboSplit may pick the splitting attribute based on Gain Ratio for a node and Gini Index for another node, but the splitting attribute chosen always defeats all other candidates in terms of m-estimate, which is very similar to prediction accuracy.

Table 5: Prediction accuracy for the low-accuracy Datasets

Dataset Name	**Gain Ratio**	**Gini Index**	**DCSM**	**Combo Split**
Glass Identification	65.85	65.62	66.57	**66.80**
Hayes-Roth	46.97	46.21	44.67	**49.28**
Pima Indians Diabetes	72.57	72.84	72.48	**73.85**
Statlog Vehicle	65.96	70.85	69.99	**71.20**
Liver Disorders	**67.29**	62.55	61.00	66.92
Grand Average	63.73 [2]	63.61 [3]	62.94 [4]	**65.61 [1]**

We examine the ability of ComboSplit to switch between the splitting criteria by analyzing the number of test attributes it picks that are originally selected by Gain Ratio, Gini Index and DCSM. Table 6 presents the percentage of nodes in ComboSplit trees that use the splitting attributes and point/s that are originally selected by Gain Ratio, Gini Index and DCSM. For example, 61.60% of nodes of the ComboSplit tree obtained from Glass Identification use the splitting attributes and point/s originally selected by Gain Ratio. Table 6 shows that ComboSplit uses all splitting criteria and does not just rely on a single criterion indicating the usefulness of our approach. It also shows that none of the three splitting criterion (Gain Ratio, Gini Index and DCSM) is always better than the others.

Table 6: Percentage of Nodes, Having the Test Attributes Chosen by Gain Ratio, Gini Index and DCSM, in ComboSplit

Dataset Name	**Gain Ratio**	**Gini Index**	**DCSM**	**Total**
Glass Identification	61.60	36.29	2.11	100.00
Hayes-Roth	91.43	8.57	0.00	100.00
Pima Indians Diabetes	33.41	64.52	2.07	100.00
Statlog (Vehicle Silhouettes)	59.53	37.60	2.87	100.00
Liver Disorders	53.64	42.38	3.97	100.00
Average	59.92	37.87	2.21	100.00

We next use five high accuracy datasets to compare the accuracy of ComboSplit trees with C4.5 trees using different splitting criteria as shown in Table 7. The ComboSplit trees achieve the highest prediction accuracy in three datasets, whereas trees using DCSM and Gain Ratio achieve the highest accuracy in one dataset each. In both datasets where ComboSplit does not achieve the highest accuracy it actually reaches the second best accuracy.

Table 7:Prediction Accuracy for the High Accuracy Datasets

Dataset Name	**Gain Ratio**	**Gini Index**	**DCSM**	**Combo Split**
Breast Cancer	94.892	95.033	94.892	**95.039**
Car Evaluation	94.095	94.095	93.629	**94.153**
Wine	93.531	88.59	89.178	**94.707**
Nursery	96.999	97.213	**97.57**	97.286
Iris	**95.334**	93.333	93.333	94.001
Grand Average	94.970 [2]	93.652 [4]	93.720 [3]	**95.037 [1]**

We realize that it is difficult for ComboSplit to further increase accuracy in high accuracy datasets where a single splitting criterion already has a very high accuracy. On the other hand ComboSplit has more room for further accuracy improvement in low accuracy datasets, where it can switch between the splitting criteria and finally pick the one that lead to a higher accuracy. However, even for the high accuracy datasets ComboSplit achieves the highest overall accuracy (see the last row of Table 7). It is more likely to get a better accuracy by ComboSplit than any other single splitting criterion.

5 CONCLUSION

Typically existing decision tree building algorithms use a single node splitting criterion. However, it is quite understandable that different splitting criteria may have different advantages and disadvantages, and therefore they can beat each other in different scenarios. Based on this understanding in this paper we propose Combo Split: our proposed technique for building a single decision tree using different existing splitting criteria simultaneously. In the experimentation, we use prediction accuracy as a performance indicator to assess the quality of the generated trees. The experimental results indicate the superiority of the proposed ComboSplit in terms of higher prediction accuracy. Although it is generally difficult to improve prediction accuracy significantly [11], we find in this study that our technique improves the overall prediction accuracy for low-accuracy datasets. Our future research plan includes implementation of ComboSplit with more splitting criteria, and experimentation on more datasets.

7 REFERENCES

[1] IBM Co., “Bringing big data to the Enterprise,” http://www-01.ibm.com/software/au/data/bigdata/ (Last Accessed: 25 Jul 2013)

[2] P. Tan, M. Steinbach, and V. Kumar, “Introduction to Data Mining. Pearson Education,” Boston, 2006.

[3] M. Z. Islam, and H. Giggins, “Knowledge Discovery through SysFor – a Systematically Developed Forest of Multiple Decision Trees,” in Proceedings of the 9-th Australian Data Mining Conference, pp. 195-204, Ballarat, 2011.

[4] J. R. Quinlan, “C4.5: Programs for Machine Learning,” Morgan Kaufmann Publishers, San Mateo, 1993.

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

[6] L. Breiman, J. H. Friedman, R. A. Olsen, and C. J. Stone, “Classification and Regression Trees,” Wadsworth International, New York,1984.

[7] B. Chandra, R. Kothari, and P. Paul, “A new node splitting measure for decision tree construction,” in Pattern Recognition, vol. 43, pp. 2725-2731, 2010.

[8] J. R. Quinlan, “Induction of Decision Trees,” in Machine Learning,” vol. 1(1), pp. 81-106, 1986.

[9] M. Z. Islam, “EXPLORE: A Novel Decision Tree Classification Algorithm,” in Data Security and Security Data, Lecture Notes in Computer Science, vol. 6121, pp. 55-71, 2012.

[10] S. R. Safavian, and D. Landgrebe, “A Survey of Decision Tree Classifier Methodology”, in IEEE Transaction on Systems, Man and Cybernetics, vol. 21, pp. 660-674, 2002.

[11] F. Berzal, J. C. Cubero, F. Cuenca, and M. J. Martin-Bautista, “On the quest for easy-to-understand splitting rules,” in Data and Knowledge Engineering, vol.44, pp. 31-48, 2003.

[12] S. B. Kotsiantis, “Decision trees: a recent overview,” in Artificial Intelligence Review, vol. 39, pp. 261-283, 2013.

[13] UCI Machine Learning Repository, http://archive.ics.uci.edu/ml /datasets.html (Last Accessed: 15 Jul 2013).

[14] Hong Hu, Jiuyong Li, Hua Wang, Grant Dag-gard, and Li-Zhen Wang. Robustness analysis of diversi_ed ensemble decision tree algorithms for microarray data classi_cation. In Proceedings of the seventh International conference on Machine Learning and Cybernetics, pages 115-120, 12-15 July 2008.