Songklanakarin J. Sci. Technol.
39 (5), 665-673, Sep - Oct. 2017

http://www.sjst.psu.ac.th

Original Article

Boundary expansion algorithm of a decision tree induction for an imbalanced dataset

Kesinee Boonchuay*, Krung Sinapiromsaran, and Chidchanok Lursinsap

Department of Mathematics and Computer Science, Faculty of Science,
Chulalongkorn University, Pathum Wan, Bangkok 10330 Thailand

Received: 5 January 2016; Revised: 23 July 2016; Accepted: 23 August 2016

Abstract

A decision tree is one of the famous classifiers based on a recursive partitioning algorithm. This paper introduces the Boundary Expansion Algorithm (BEA) to improve a decision tree induction that deals with an imbalanced dataset. BEA utilizes all attributes to define non-splittable ranges. The computed means of all attributes for minority instances are used to find the nearest minority instance, which will be expanded along all attributes to cover a minority region. As a result, BEA can successfully cope with an imbalanced dataset comparing with C4.5, Gini, asymmetric entropy, top-down tree, and Hellinger distance decision tree on 25 imbalanced datasets from the UCI Repository.

Keywords: C4.5, decision tree, classification, boundary expansion algorithm

1. Introduction

A decision tree is one of the most widely used classifiers (KDnuggets, 2011; Wu *et al*., 2008) mentioned by Barros *et al.* (2012) and Safavian *et al.* (1991) for four reasons. First, a decision tree is robust with regard to noise in a dataset compared with other classifiers. Second, it requires low computational cost for building a tree and classifying new instances. Third, a decision tree can handle a dataset with a multicollinearity problem. Fourth, the model of a decision tree combines local decisions to represent a complex global decision, which makes it easy to generate rules.

Many research studies rely on a decision tree to solve problems in several areas. For examples, Yeon *et al.* (2010) applied a decision tree to analyze landslide susceptibility in Injae, Korea. Yu *et al.* (2010) used a decision tree to build an energy demand modeling. In the work of Garcia *et al.* (2013), they applied a decision tree to predict the prognosis of severe traumatic brain injury in Brazilian patients from Florianopolis City. Lee *et al.* (2013) used personality traits to construct a decision tree that was used to analyze the ability of students to win a prize.

However, a decision tree still does not cope well with one particular type of problem: the class imbalanced problem. This occurs in a dataset having a huge different number of instances among classes. A classifier tends to misclassify instances from the class that has a small number of instances. Numerous research studies are taking different approaches to cope with this problem. More details are discussed in section 3.2.

Since a traditional decision tree induction was designed based on a balanced dataset, it tends to have this limitation. In our work, we focus on adapting a decision tree induction to handle an imbalanced dataset on an algorithmic level. A technique is proposed to remedy this situation, called the *Boundary Expansion Algorithm* (BEA). This technique aims to improve the performance of a decision tree induction to handle the class imbalanced problem.

In a dataset, a group of instances located together tends to be from the same class sharing the group characteristics. In an imbalanced dataset, the member of minority class instances is small, so they need to be handled carefully. By

* Corresponding author.
Email address: bkasinee@hotmail.com

considering one dimension at a time, a group of minority instances may be split, which would in turn tear apart their main characteristics. In order to identify this group, all dimensions have to be examined at the same time. Therefore, the idea of BEA is to maintain groups of minority class instances in a multi-dimensional space, while a traditional decision tree considers only a single dimension at a time. Instances in these groups should not be separated by any split from a decision tree, so that their common characteristics are not disparate. To apply this idea, BEA selects a group and creates a boundary by expanding from a centroid. The boundary will be used to define non-splittable intervals for all attributes during the partitioning step. This guarantees that the dense minority region will not be partitioned, while a traditional decision tree induction may split it.

This paper consists of five sections. The first section is the introduction. The second section contains related works to the class imbalanced problem. The third section introduces our technique. The fourth section contains all experimental results. The last section is the conclusion and future work.

2. Related Works

2.1 Decision tree

Since our proposed algorithm is based on a decision tree induction, a detailed description of a decision tree induction is included in this paper. A decision tree induction is an algorithm for splitting a dataset among all attributes to guide the decision at the leaf nodes using the current information from all instances in a current dataset or the current internal node. A tree consists of multiple connected nodes. Each node represents a condition based on a selected attribute for splitting instances into partitions. If a partition consists of instances from the same class or reaches a stopping criterion, it is a leaf, and it identifies all instances within that class. After each iteration the best split will be selected using a splitting measure. The traditional decision tree uses an impurity measure called Shannon's entropy. The formula for Shannon's entropy appears below. Let D denote a set of instances. c denotes the number of classes in a whole dataset. D_w denotes a set of instances beginning in the class w.

$$Entropy(D) = -\sum_{w=1}^{c} \frac{|D_w|}{|D|} \, log_2 \, \frac{|D_w|}{|D|}$$

All plausible values along an attribute are examined to locate the minimum splitting entropy for the current dataset. After that, the algorithm will select the best split among all attributes. However, this method cannot cope with a dataset having the checkerboard pattern as shown in Figure 1.

In 2010, a new node splitting measure was proposed called the distinct class based splitting measure (DCSM) (Chandra *et al.*, 2010). This splitting measure applies the idea of the number of distinct class of instances, which improves the performance of a decision tree. The result of comparing this technique with our algorithm will be shown in the fourth section. There also are several researches proposed to improve decision tree inductions, such as Zighed *et al.* (2010), Lenca *et al.* (2010), Chandra *et al.* (2011), Sinapiromsaran *et al.* (2012), and Sirisomboonrat *et al.* (2012). The main difference between our method and their techniques is that our method blocks a range of values from examining as the best split. More details are provided in section 3.

2.2 Class imbalanced problem

For a class imbalanced problem, a minority class is the class having the smallest number of instances. A majority class is the class having the largest number of instances. For a two-class dataset case in this paper, a majority class and a minority class are also represented as a negative class and a positive class, respectively. There are two main approaches to cope with a class imbalance. First, this problem can be handled on a data level using a sampling method. It aims to balance the number of instances between classes. There are two techniques of sampling: over-sampling and under-sampling. Examples of sampling methods are SMOTE (Chawla *et al.*, 2002), Borderline-SMOTE (Han *et al.*, 2005), ADASYN (He *et al.*, 2008), Safe-Level-SMOTE (Bunkhumpornpat *et al.*, 2009), and DBSMOTE (Bunkhumpornpat *et al.*, 2012). The second approach handles class imbalance on an algorithmic level. A method of this approach is developed to increase the performance in an imbalanced dataset without changing the dataset. Since a decision tree is a widely used classifier, there are many algorithms based on a decision tree that are applied to an imbalanced dataset. Zighed *et al.* (2010) proposed that asymmetric entropy (AE) replace symmetric entropy as the new impurity measure. Another example is off-centered entropy (OCE) (Lenca *et al.*, 2010), which uses the probability of instances for each class. Then the new probability is applied to Shannon's entropy. As a result, both AE and OCE provide the skew entropy, which favors minority

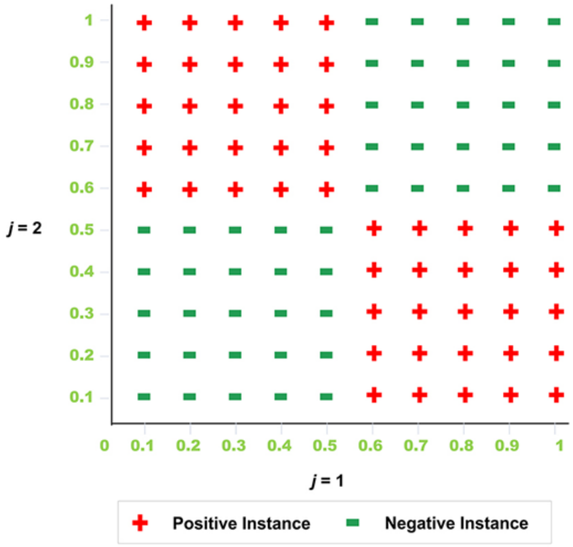

Figure 1. Sample dataset having the checkerboard pattern.

instances. In Dietterich *et al.* (1996), a top down decision tree induction called DKM was proposed. It applies an insensitive measure to handle the class imbalanced problem. In Cieslak *et al.* (2008), they use a measure of distributional divergence called Hellinger Distance as the splitting criteria. In our work, we focus on adapting the decision tree induction algorithm on an algorithmic level. The details of this will be described in the next section.

3. Boundary Expansion Algorithm

3.1 Motivation

The traditional decision tree induction is a recursive partitioning algorithm that splits a dataset based on a single attribute. There are some patterns that cannot be handled by this concept, such as the checkerboard pattern. By using Shannon's entropy as the splitting measure, all plausible splits yield the same entropy value. Hence, the decision tree induction must blindly guess the best split for the dataset. In order to handle this pattern, multiple attributes should be considered simultaneously. Our method (BEA) was motivated by a desire to solve the checkerboard pattern. It will make the split at the lower or upper boundary of any cell in the checkerboard.

Since a checkerboard pattern rarely appears in a real-world dataset, BEA is designed to work with an imbalanced dataset, which frequently appears. First, it utilizes all attributes simultaneously, and then it specifies a non-splittable range along all attributes. A value in this range is prohibited from being examined as a candidate for the best split. Each range presents the contiguous values for a region of a unique minority class. To allow the split in this range, therefore, a group of minority class instances will be separated, which will cause disparate relationships among this minority group.

BEA extends a region for positive instances that are located together without a negative instance. These positive instances will share the dominant characteristic of this region. By using a traditional decision tree induction, there is a chance to split this positive group if it generates the largest entropy gain. BEA will be described in more detail in the next section.

3.2 Boundary expansion algorithm

BEA identifies the boundary from a centroid of positive instances. A positive centroid (*PC*) is defined as a positive instance that is the nearest neighbor of a pseudo-positive centroid (*PSC*). *PSC* combines the mean values of positive instances from all attributes. Let *n* denote the number of instances in a dataset. Then for each attribute *j*,

$$PSC_j = \frac{1}{n}\sum_{i=1}^{n} p_{i,j}$$

At each iteration, BEA expands the boundary, called *positive boundary*, from *PC* for each attribute j^{th}. It decreases a single δ step from PC_j toward the lower bound and at the same time, it increases a single δ step from PC_j toward the upper bound. δ is set as the shortest distance between any two instances. Let δ be computed by the following, in which $x_{l,j}$ and $x_{k,j}$ denote instance l^{th} and instance k^{th} for an attribute j^{th} $(l \neq k)$.

$$\delta = \min_{j} \min_{l,k} | x_{l,j} - x_{k,j} |$$

The global upper bound and the global lower bound of the whole dataset are defined by the maximum and minimum values of all positive instances from all attributes. These global boundaries are represented by pseudo instances called P_{pmax} and P_{pmin}. These two pseudo instances combine the boundaries of all attributes from all positive instances. For an attribute j^{th}, $P_{pmax,j}$ denotes the upper bound, and $P_{pmin,j}$ denotes the lower bound. They are defined as the following, in which $p_{i,j}$ denotes a positive instance for an instance i^{th} and attribute j^{th}.

$$P_{pmax,j} = \max_{j}\left(p_{i,j}\right), \text{ and,}$$

$$P_{pmin,j} = \min_{j}\left(p_{i,j}\right)$$

The expansion will stop when it meets one of following two criteria: (1) Negative instances are included within the boundary and (2) The boundary reaches the global upper bound or global lower bound. For example, in Figure 2, P_{pmax} is (0.7, 0.7). P_{pmin} is (0.4, 0.4). Therefore, the boundary will cover only positive instances. Our idea is that the group of positive instances within the ranges of this boundary should not be separated by any split. Therefore, all values lying within this range will not be selected for a split; they are called non-splittable values. A decision tree will be constructed without including these non-splittable values into the tree. For example, in Figure 3, positive instances are located within a circular area. The region of positive instances could not be captured by a single rectangle as shown in Figure 2. In this case, BEA constructs the minority boundary by using the rectangle inside the circle. Since the rectangle inside the circle consists of only positive instances, it should not be separated. BEA then uses a split measure to decide the best split. It is possible that the best split might be around the border of the circle, which is handled by a decision tree induction.

The details of BEA are shown below.

Boundary Expansion Algorithm

Input: A dataset (*D*) including positive class instances and negative class instances

Output: Positive Upper Bound (*UPB*) and Positive Lower Bound (*LPB*)

```
1. Compute Positive Centroid (PC)
2. Compute $\delta$
3. Set initial value for UPB and LPB equal to PC
4. Set the number of iteration ($r$) = 0
5. While $LPB \geq P_{pmin}$ and $UPB \leq P_{pmax}$ and $nTotal$=0
```

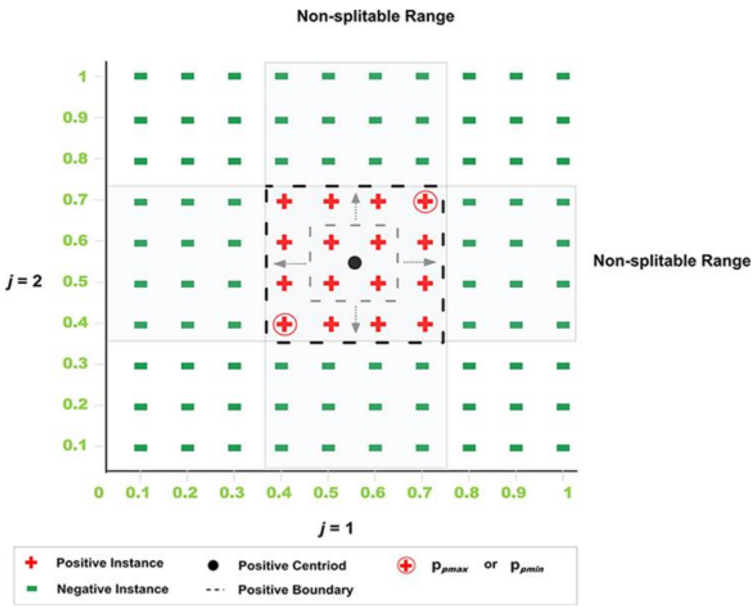

Figure 2. Demonstration of Boundary Expansion Algorithm.

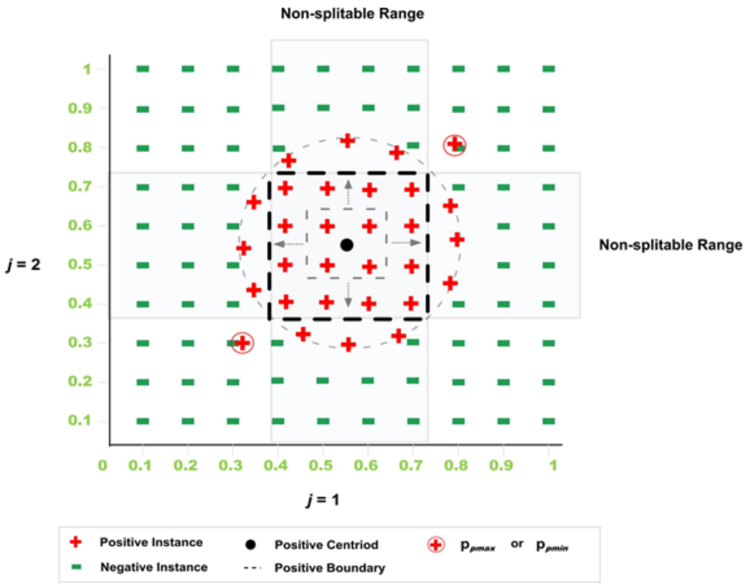

Figure 3. BEA on a dataset with positive instances located in a circle shape.

```
For attribute (j) in UPB and LPB
    $UPB^{r}_{j} = UPB^{r-1}_{j} + \delta$
    $LPB^{r}_{j} = LPB^{r-1}_{j} + \delta$
End For
$nTotal$ = number of negative instances in positive boundary
$r = r + 1$
If $nTotal = 0$
    $UPB = UPB^{r}$
    $LPB = LPB^{r}$
End If
Loop
Return UPB and LPB
```

BEA yields the positive upper bound (*UPB*) and positive lower bound (*LPB*) for all attributes. These bounds will be applied to the decision tree induction. The splits between *UPB* and *LPB* are prohibited. Hence, the algorithm for a decision tree applying BEA is shown below.

```
Decision Tree Induction applying BEA
Input: A dataset ($D$) including positive class instances and negative class instances
Output: A decision tree
1. Create a root node
2. If all instances are in the same class then
3.     Return node labelled as that class
4. Find $UPB$ and $LPB$ for all attributes by using $BEA$
5. For each attribute $j$ in $D$
6.     $S$ = the set of values in an attribute $j$
7.     $S\_BEA$ = the set of values in $S$ which excludes
       all values between $UPB$ and $LPB$
8.     Select the best split for attribute $j$ from $S\_BEA$
9. End For
10. Separate instances into partitions corresponding
    to the selected attribute
11. Recursive for each partition
```

The result of a decision tree applying BEA will be presented in the next section.

4. Experimental Results

The experiments were performed on 25 datasets from the UCI repository (Blake *et al.*, 1998). All datasets are validated by ten-fold cross-validation. In order to apply BEA, multiple-class datasets are transformed to binary-class datasets as one-against-all, having the target class in the first part of the third column in Table 1. Then rest of the classes represents a negative class as shown in the second part of the same column. In Table 1, the first column presents the number of datasets. The second column contains the dataset names. The third column consists of the first part showing the names of the original classes that selected as a positive class and the second part presenting the names of original classes that selected as a negative class. The fourth column, the fifth column, and the sixth column are the number of attributes, the number of instances, and the percent of positive class instances, respectively. All codes (C4.5, Gini, DCSM, AE, DKM, HDDT, and BEA) were implemented in MATLAB.

4.1 Performance measure and evaluation

In the experiment, the F-measure (Buckland & Gey, 1994) and geometric mean are used as the performance measures. The formula of F-measure is as follows: TP denotes the number of true positive instances; FP denotes the number of false positive instances; FN denotes the number of false negative instances. In the experimental result, β is 1.

Table 1. Dataset characteristics.

No.	Datasets	Positive Class / Negative Class	#Attributes	#Instances	%Positive
1	Page Blocks	1 / The rest	10	5473	0.51 %
2	Thyroid	3 / The rest	21	720	2.36 %
3	Letter	A / The rest	16	20000	3.95 %
4	Abalone	18 / 9	8	731	5.75 %
5	Glass(5)	5 / The rest	9	214	6.07 %
6	Cleveland	0 / 4	13	173	7.51 %
7	LED Display Domain	0, 2, 4, 5, 6, 7, 8, 9 / 1	7	443	8.35 %
8	Vowel	0 / The rest	13	988	9.11 %
9	Ecoli(imU)	imU / The rest	7	336	10.42 %
10	Fertility	O / The rest	10	100	12.00 %
11	Breast Tissue	con / The rest	10	106	13.21 %
12	Segmentation	1 / The rest	19	2310	14.29 %
13	Ecoli(pp)	pp / The rest	7	336	15.48 %
14	Vertebral Column	DH / The rest	6	310	19.35 %
15	Transfusion	1 / The rest	4	748	23.80 %
16	Parkinsons	0 / The rest	10	195	24.62 %
17	Haberman	2 / The rest	3	306	26.47 %
18	Wine	3 / The rest	13	178	26.97 %
19	Yeast	CYT / The rest	8	1484	31.20 %
20	Glass(1)	1 / The rest	9	214	32.71 %
21	Seeds	2 / The rest	10	210	33.33 %
22	Waveform	2 / The rest	21	5000	33.92 %
23	Pima	1 / The rest	8	768	34.90 %
24	Inonosphere	b / The rest	34	351	35.90 %
25	Wisconsin BC	M / The rest	30	569	37.26 %

$$F - measure = \frac{(1+\beta^2)\times TP}{(1+\beta^2)\ TP + \beta^2 FN + FP}$$

For the geometric mean, the formula is as follows:

$$Geometric\ \ mean = \sqrt{\frac{TP}{TP+FN}\frac{TN}{TN+FP}}$$

For evaluation, the Friedman test is a non-parametric statistical test that is suitable for the comparison of classifiers (Dem¡sar, 2006). Therefore, our experimental results are compared using the Friedman test with a significance level of $\alpha = 0.05$.

4.2 Experimental results

The experiment shown in Table 2 is the comparison of C4.5, Gini, DCSM, AE, DKM, HDDT, and BEA by F-measure. For each dataset, all techniques are ranked by F-measure values, which are shown in the blanket after them. BEA provides the best average ranking over all datasets at 2.00. The second and third best average rankings are AE and DCSM at 2.88 and 3.08, respectively. For the Friedman test, BEA yields a significant improvement over C4.5, Gini, AE, DKM, and HDDT for imbalanced datasets compared with F-measure at a 0.05 significance level.

In Table 3, all techniques are ranked by geometric mean values, which are shown in the blanket after the name. BEA provides the best average ranking over all datasets at 2.12. The second and third best average rankings are AE and Gini at 2.88 and 3.08, respectively. For the Friedman test, BEA yields a significant improvement over C4.5, Gini, AE, DKM, and HDDT for imbalanced datasets compared with the geometric mean at a 0.05 significance level.

From our experimental results, BEA provides better performance than other techniques using F-measure or geometric mean, especially in the abalone, vowel, ecoli, and vertebral column datasets. However, it yields unsatisfactory performance in the wine and the transfusion datasets. BEA attempts to locate a group of contiguous minority instances and utilizes this information to determine the best split. In the wine and transfusion datasets, minority instances are not formed into significant groups without including some majority instances. This causes BEA to generate scattered unsplittable ranges. Therefore, BEA would not gain the benefit from these datasets. In contrast, if minority instances are formed in large contiguous groups, BEA will utilize this information to help determine the best split. Accordingly, BEA tends to yield improvement in these datasets.

5. Conclusions and Future Work

There are limitations of using a decision tree, such as the handling of an imbalance dataset. Our research developed new methodology for building a decision tree. It protected the minority range from all attributes to avoid the splitting of the minority group. Our experiments showed that BEA yields better performance over C4.5, Gini, AE, DKM, and HDDT for imbalanced datasets compared with the F-measure.

BEA requires additional computational time for a boundary expansion process, which must be computed at every step due to the centroid update. However, like all methods on an algorithmic level, BEA does not change the distribution of datasets. As a result, it does not have to process new instances or build more complex classifiers.

For F-measure comparison on an imbalanced algorithm, BEA provides better performance than asymmetric entropy. One main disadvantage of using asymmetric entropy is found in selecting the appropriate parameter for a dataset. If a given parameter is not suitable, using asymmetric entropy can affect the performance of the decision tree significantly. Because it is a parameter-free method, BEA does not have this weakness.

BEA can cope with an imbalanced dataset as shown in the experimental results. However, it may not be able to handle the spreading of minority instances that do not form a group. Therefore, BEA will capture each instance separately and may not improve the performance of a decision tree induction in this case.

For future work, the computational time of BEA can be improved by utilizing a previously computed centroid. If the expansion starts at the appropriate centroid, the computational time should be dropped. Moreover, it can be applied directly with other impurity measures, such as Gini, DCSM, and asymmetric entropy to improve the performance.

Acknowledgements

We thank the Strategic Scholarships Fellowships Frontier Research Networks (specific for Southern region) for the Ph.D. Program Thai doctoral degree from the Commission on Higher Education, Thailand for its financial support.

References

Barros, R. C., Basgalupp, M. P., De Carvalho, A. C., & Freitas, A. A. (2012). A survey of evolutionary algorithms for decision-tree induction. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(3), 291-312.

Blake, C. L., & Merz, C. J. (1998). UCI Repository of machine learning databases. Irvine, CA: University of California. Retrieved from http://archive.ics.uci.edu/ml/index.php

Buckland, M. K., & Gey, F. C. (1994). The relationship between recall and precision. *Journal of the American Society for Information Science*, *45*(1), 12-19.

Bunkhumpornpat, C., Sinapiromsaran, K., & Lursinsap, C. (2009). Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. *Advances in Knowledge Discovery and Data Mining*, 475-482.

Table 2. Experimental results comparing by F-measure.

No.	Datasets	C4.5	Gini	DCSM	AE	DKM	HDDT	BEA
1	Page Blocks	0.6290 ± 0.3094 (3)	0.5538 ± 0.3574 (5)	**0.7414 ± 0.3206 (1)**	0.5550 ± 0.4246 (4)	0.2682 ± 0.3157 (7)	0.4471 ± 0.3327 (6)	0.6489 ± 0.3965 (2)
2	Thyroid	**0.8103 ± 0.3189 (1)**	**0.8103 ± 0.3189 (1)**	**0.8103 ± 0.3189 (1)**	**0.8103 ± 0.3189 (1)**	0.3222 ± 0.3865 (6)	0.2333 ± 0.3865 (7)	**0.8103 ± 0.3189 (1)**
3	Letter	0.9440 ± 0.0176 (4)	**0.9541 ± 0.0128 (1)**	0.9315 ± 0.0187 (6)	0.9493 ± 0.0228 (3)	0.8307 ± 0.035 (7)	0.9349 ± 0.0167 (5)	0.9522 ± 0.0185 (2)
4	Abalone	0.2035 ± 0.1562 (4)	0.2702 ± 0.2188 (3)	0.2857 ± 0.1498 (2)	0.1422 ± 0.1657 (5)	0.1378 ± 0.1882 (6)	0.1369 ± 0.1879 (7)	**0.3095 ± 0.2145 (1)**
5	Glass(5)	0.7067 ± 0.3239 (5)	0.8067 ± 0.3492 (4)	**0.8667 ± 0.3220 (1)**	**0.8667 ± 0.3220 (1)**	0.4905 ± 0.3815 (6)	0.4667 ± 0.4500 (7)	**0.8667 ± 0.3220 (1)**
6	Cleveland	**0.4333 ± 0.4727 (1)**	0.2667 ± 0.4389 (4)	0.1400 ± 0.3273 (6)	0.2667 ± 0.3702 (4)	0.0000 ± 0.0000 (7)	0.3524 ± 0.4640 (2)	0.3167 ± 0.4335 (3)
7	LED Display Domain	0.7799 ± 0.1051 (2)	0.7799 ± 0.1051 (2)	**0.7827 ± 0.1113 (1)**	0.7799 ± 0.1051 (2)	0.7418 ± 0.1304 (7)	0.7799 ± 0.1051 (2)	0.7799 ± 0.1051 (2)
8	Vowel	0.9307 ± 0.0706 (2)	0.9099 ± 0.0687 (4)	0.9138 ± 0.0479 (3)	0.8854 ± 0.0788 (5)	0.4552 ± 0.1667 (7)	0.7000 ± 0.1237 (6)	**0.9521 ± 0.0570 (1)**
9	Ecoli(imU)	0.4362 ± 0.3346 (5)	0.4903 ± 0.2929 (3)	0.4617 ± 0.2674 (4)	**0.5496 ± 0.2442 (1)**	0.2652 ± 0.1941 (7)	0.3569 ± 0.3115 (6)	0.4990 ± 0.3057 (2)
10	Fertility	0.1667 ± 0.3600 (3)	0.1667 ± 0.2833 (3)	0.1000 ± 0.3162 (6)	0.1667 ± 0.3600 (3)	0.1000 ± 0.3162 (6)	**0.3667 ± 0.4830 (1)**	0.2667 ± 0.3784 (2)
11	Breast Tissue	0.7800 ± 0.3425 (5)	**0.9417 ± 0.1245 (1)**	**0.9417 ± 0.1245 (1)**	**0.9417 ± 0.1245 (1)**	0.4633 ± 0.3818 (7)	0.7467 ± 0.4095 (6)	**0.9417 ± 0.1245 (1)**
12	Segmentation	0.9741 ± 0.0212 (5)	0.9756 ± 0.0225 (4)	**0.9894 ± 0.0073 (1)**	0.9835 ± 0.0147 (2)	0.8288 ± 0.0783 (7)	0.8435 ± 0.0436 (6)	0.9819 ± 0.0119 (3)
13	Ecoli(pp)	0.7523 ± 0.1798 (3)	0.7414 ± 0.2063 (4)	0.7146 ± 0.1328 (5)	0.7692 ± 0.1043 (2)	0.5476 ± 0.2502 (6)	0.5433 ± 0.2757 (7)	**0.8018 ± 0.1650 (1)**
14	Vertebral Column	0.5053 ± 0.1582 (5)	0.5722 ± 0.1690 (3)	0.5058 ± 0.2303 (4)	0.5907 ± 0.2164 (2)	0.4483 ± 0.1979 (7)	0.4701 ± 0.105 (6)	**0.6182 ± 0.2835 (1)**
15	Transfusion	0.3500 ± 0.1044 (5)	0.3645 ± 0.0958 (2)	**0.3877 ± 0.1212 (1)**	0.3573 ± 0.0901 (3)	0.3303 ± 0.0996 (6)	0.2981 ± 0.1098 (7)	0.3571 ± 0.0794 (4)
16	Parkinsons	0.6008 ± 0.1712 (6)	0.6791 ± 0.1408 (5)	0.7014 ± 0.1209 (3)	**0.7365 ± 0.1344 (1)**	0.514 ± 0.1643 (7)	0.6861 ± 0.1381 (4)	0.7182 ± 0.1435 (2)
17	Haberman	**0.3579 ± 0.0883 (1)**	0.3093 ± 0.0991 (6)	0.3377 ± 0.1105 (3)	0.3281 ± 0.1799 (5)	0.2262 ± 0.1798 (7)	0.3371 ± 0.1624 (4)	0.3402 ± 0.1534 (2)
18	Wine	0.9369 ± 0.0595 (3)	0.9385 ± 0.0880 (2)	0.9237 ± 0.0926 (4)	**0.9544 ± 0.0602 (1)**	0.8323 ± 0.1322 (6)	0.5987 ± 0.1911 (7)	0.8952 ± 0.0956 (5)
19	Yeast	0.5137 ± 0.0538 (5)	0.5008 ± 0.0470 (7)	0.5263 ± 0.0578 (3)	**0.5359 ± 0.0650 (1)**	0.5019 ± 0.0481 (6)	0.5233 ± 0.0582 (4)	0.5287 ± 0.0843 (2)
20	Glass(1)	0.7498 ± 0.1368 (2)	0.7388 ± 0.1097 (3)	**0.7499 ± 0.1132 (1)**	0.7299 ± 0.1146 (5)	0.5672 ± 0.1604 (7)	0.6870 ± 0.1800 (6)	0.7320 ± 0.1116 (4)
21	Seeds	**0.9713 ± 0.0372 (1)**	**0.9713 ± 0.0372 (1)**	0.9647 ± 0.0503 (4)	0.9498 ± 0.0606 (5)	0.8475 ± 0.098 (7)	0.9031 ± 0.0968 (6)	**0.9713 ± 0.0372 (1)**
22	Waveform	0.7647 ± 0.0182 (4)	**0.7838 ± 0.0199 (1)**	0.7594 ± 0.0257 (5)	0.7806 ± 0.0328 (3)	0.6996 ± 0.0305 (7)	0.7156 ± 0.0425 (6)	0.7816 ± 0.0221 (2)
23	Pima	0.5912 ± 0.0668 (3)	0.5956 ± 0.0896 (2)	0.5609 ± 0.0637 (5)	0.5617 ± 0.0633 (4)	0.5215 ± 0.0742 (7)	0.5384 ± 0.0973 (6)	**0.5976 ± 0.0664 (1)**
24	Inonosphere	**0.9020 ± 0.0850 (1)**	0.8524 ± 0.0737 (4)	0.8986 ± 0.0719 (2)	0.8007 ± 0.0689 (6)	0.6994 ± 0.0806 (7)	0.8393 ± 0.0539 (5)	0.8550 ± 0.0856 (3)
25	Wisconsin BC	0.8957 ± 0.0819 (5)	0.9129 ± 0.0445 (3)	0.9034 ± 0.0388 (4)	0.9238 ± 0.0462 (2)	0.8691 ± 0.0600 (7)	0.8852 ± 0.0502 (6)	**0.9359 ± 0.0494 (1)**
	Average Rank	3.36	3.12	3.08	2.88	6.68	5.40	2.00
	Friedman Test	**0.010515**	**0.016377**	**0.088082**	**0.049535**	**0.000001**	**0.000045**	**Base**

Table 3. Experimental results comparing by geometric mean.

No.	Datasets	C4.5	Gini	DCSM	AE	DKM	HDDT	BEA
1	Page Blocks	0.7659 ± 0.3025 (2)	0.6457 ± 0.3689 (4)	**0.8359 ± 0.3172 (1)**	0.6187 ± 0.4433 (5)	0.3144 ± 0.3513 (7)	0.6038 ± 0.4396 (6)	0.7305 ± 0.4024 (3)
2	Thyroid	**0.8776 ± 0.3152 (1)**	**0.8776 ± 0.3152 (1)**	**0.8776 ± 0.3152 (1)**	**0.8776 ± 0.3152 (1)**	0.4332 ± 0.495 (6)	0.2986 ± 0.4807 (7)	**0.8776 ± 0.3152 (1)**
3	Letter	0.9700 ± 0.0151 (4)	0.9729 ± 0.0100 (3)	0.9583 ± 0.0146 (6)	0.9738 ± 0.0156 (2)	0.8951 ± 0.0268 (7)	0.9622 ± 0.0129 (5)	**0.9808 ± 0.0139 (1)**
4	Abalone	0.3510 ± 0.2437 (4)	0.4243 ± 0.3062 (3)	0.4550 ± 0.1767 (2)	0.2703 ± 0.2912 (5)	0.2355 ± 0.3113 (6)	0.2055 ± 0.2664 (7)	**0.4552 ± 0.2585 (1)**
5	Glass(5)	0.8399 ± 0.3327 (5)	0.8475 ± 0.3363 (4)	**0.8975 ± 0.3154 (1)**	**0.8975 ± 0.3154 (1)**	0.6537 ± 0.4599 (6)	0.5437 ± 0.4923 (7)	**0.8975 ± 0.3154 (1)**
6	Cleveland	**0.4936 ± 0.5205 (1)**	0.2968 ± 0.4780 (5)	0.1500 ± 0.3375 (6)	0.3839 ± 0.4959 (3)	0.0000 ± 00000 (7)	0.3834 ± 0.4964 (4)	0.3904 ± 0.5043 (2)
7	LED Display Domain	0.8873 ± 0.0868 (2)	0.8873 ± 0.0868 (2)	**0.8938 ± 0.0997 (1)**	0.8873 ± 0.0868 (2)	0.8671 ± 0.0842 (7)	0.8873 ± 0.0868 (2)	0.8873 ± 0.0868 (2)
8	Vowel	0.9608 ± 0.0641 (2)	0.9483 ± 0.0605 (3)	0.9389 ± 0.0461 (5)	0.9406 ± 0.0591 (4)	0.5700 ± 0.1280 (7)	0.7716 ± 0.095 (6)	**0.9791 ± 0.0384 (1)**
9	Ecoli(imU)	0.5365 ± 0.3901 (5)	0.6384 ± 0.3659 (2)	0.5851 ± 0.2415 (4)	**0.6942 ± 0.2887 (1)**	0.4051 ± 0.2901 (7)	0.4781 ± 0.3578 (6)	0.6168 ± 0.3481 (3)
10	Fertility	0.1935 ± 0.4083 (4)	0.2411 ± 0.4028 (3)	0.1000 ± 0.3162 (6)	0.1935 ± 0.4083 (4)	0.1000 ± 0.3162 (6)	**0.3935 ± 0.5084 (1)**	0.3411 ± 0.4568 (2)
11	Breast Tissue	0.8617 ± 0.3088 (5)	**0.9717 ± 0.0716 (1)**	**0.9717 ± 0.0716 (1)**	**0.9717 ± 0.0716 (1)**	0.6122 ± 0.4381 (7)	0.7796 ± 0.4135 (6)	**0.9717 ± 0.0716 (1)**
12	Segmentation	0.9841 ± 0.0183 (5)	0.9856 ± 0.0190 (4)	**0.9944 ± 0.0068 (1)**	0.9921 ± 0.0099 (2)	0.9021 ± 0.0518 (7)	0.9025 ± 0.0302 (6)	0.9893 ± 0.0099 (3)
13	Ecoli(pp)	0.8367 ± 0.1608 (3)	0.8230 ± 0.1746 (4)	0.8228 ± 0.1414 (5)	0.8386 ± 0.0913 (2)	0.6927 ± 0.2190 (6)	0.6479 ± 0.2793 (7)	**0.8681 ± 0.1277 (1)**
14	Vertebral Column	0.6516 ± 0.1366 (4)	0.7061 ± 0.1223 (3)	0.6273 ± 0.2491 (6)	0.7067 ± 0.1607 (2)	0.6025 ± 0.238 (7)	0.6361 ± 0.0774 (5)	**0.7333 ± 0.2317 (1)**
15	Transfusion	0.5194 ± 0.0920 (5)	0.5335 ± 0.0871 (2)	**0.5470 ± 0.1039 (1)**	0.5272 ± 0.0852 (3)	0.4955 ± 0.0887 (6)	0.4673 ± 0.0965 (7)	0.5271 ± 0.0748 (4)
16	Parkinsons	0.7295 ± 0.1361 (6)	0.7693 ± 0.1048 (5)	**0.8110 ± 0.0985 (1)**	0.8081 ± 0.1214 (2)	0.6679 ± 0.1510 (7)	0.7800 ± 0.1297 (4)	0.8002 ± 0.1249 (3)
17	Haberman	**0.5212 ± 0.0856 (1)**	0.4753 ± 0.0921 (5)	0.5021 ± 0.0994 (2)	0.4714 ± 0.2085 (6)	0.3537 ± 0.2229 (7)	0.4916 ± 0.1441 (4)	0.4983 ± 0.1466 (3)
18	Wine	0.9667 ± 0.0405 (2)	0.9620 ± 0.0622 (3)	0.9586 ± 0.0560 (4)	**0.9700 ± 0.0467 (1)**	0.8785 ± 0.1004 (6)	0.6965 ± 0.1776 (7)	0.9278 ± 0.0706 (5)
19	Yeast	0.6328 ± 0.0441 (5)	0.6216 ± 0.0406 (7)	0.6414 ± 0.0458 (3)	**0.6496 ± 0.0539 (1)**	0.6222 ± 0.0386 (6)	0.6403 ± 0.0480 (4)	0.6446 ± 0.0700 (2)
20	Glass(1)	**0.8113 ± 0.1059 (1)**	0.8046 ± 0.0881 (3)	0.8109 ± 0.0889 (2)	0.7933 ± 0.0943 (4)	0.6568 ± 0.1300 (7)	0.7613 ± 0.1406 (6)	0.7924 ± 0.0783 (5)
21	Seeds	**0.9779 ± 0.0312 (1)**	**0.9779 ± 0.0312 (1)**	0.9745 ± 0.0384 (4)	0.9624 ± 0.0511 (5)	0.8822 ± 0.0770 (7)	0.9280 ± 0.0831 (6)	**0.9779 ± 0.0312 (1)**
22	Waveform	0.8191 ± 0.0141 (4)	**0.8352 ± 0.0159 (1)**	0.8153 ± 0.0201 (5)	0.8313 ± 0.0260 (3)	0.771 ± 0.0237 (7)	0.7847 ± 0.0340 (6)	0.8337 ± 0.0156 (2)
23	Pima	0.6767 ± 0.0560 (3)	0.6822 ± 0.0721 (2)	0.6521 ± 0.0495 (5)	0.6535 ± 0.0489 (4)	0.6223 ± 0.0579 (7)	0.6349 ± 0.0797 (6)	**0.6834 ± 0.0535 (1)**
24	Inonosphere	**0.9208 ± 0.0677 (1)**	0.8787 ± 0.0582 (4)	0.9177 ± 0.0670 (2)	0.8388 ± 0.0512 (6)	0.7545 ± 0.0683 (7)	0.8614 ± 0.0494 (5)	0.8797 ± 0.0703 (3)
25	Wisconsin BC	0.9161 ± 0.0674 (5)	0.9287 ± 0.0359 (3)	0.9229 ± 0.0328 (4)	0.9406 ± 0.0375 (2)	0.8919 ± 0.0491 (7)	0.9020 ± 0.0402 (6)	**0.9470 ± 0.0410 (1)**
	Average Rank	3.24	3.12	3.16	2.88	6.68	5.44	2.12
	Friedman Test	**0.033006**	**0.016377**	**0.393769**	**0.126630**	**0.000001**	**0.000007**	**Base**

Bunkhumpornpat, C., Sinapiromsaran, K., & Lursinsap, C. (2012). DBSMOTE: density-based synthetic minority over-sampling technique. *Applied Intelligence*, *36* (3), 664-684.

Chandra, B., & Kuppili, V. B. (2011). Heterogeneous node split measure for decision tree construction. *IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2011*, 872-877.

Chandra, B., Kothari, R., & Paul, P. (2010). A new node splitting measure for decision tree construction. *Pattern Recognition*, *43*(8), 2725-2731.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321-357.

Cieslak, D. A., & Chawla, N. V. (2008). Learning decision trees for unbalanced data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 241-256). Berlin, Heidelberg, Germany: Springer.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, *7*(Jan), 1-30.

Dietterich, T. G., Kearns, M. J., & Mansour, Y. (1996). Applying the weak learning framework to understand and improve C4. 5. In *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning* (pp. 96-104). Burlington, MA: Morgan Kaufmann.

Garcia, M. C. M., Martins, E. T., & Azevedo, F. M. (2013). Decision tree induction to prediction of prognosis in severe traumatic brain injury of Brazilian patients from Florianopolis city. *IEEE 13th International Conference on Bioinformatics and Bioengineering, 2013,* 1-4.

Han, H., Wang, W. Y., & Mao, B. H. (2005). Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. *Advances in Intelligent Computing*, 878-887.

He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. *International Joint Conference on Neural Networks, 2008*, 1322-1328.

KDnuggets. (2011). *Poll results: Top algorithms for analytics/ data mining*. Retrieved from http://www.kdnuggets.com/2011/11/algorithms-for-analytics-data-mining.html

Lee, D., Deng, L., Lin, K., Jheng, Y., Chen, Y., Chao, C., & Huang, J. (2013). Using decision tree analysis for personality to decisions of the national skills competition participants. *Information Technology Convergence*, *253*, 683-691.

Lenca, P., Lallich, S., & Vaillant, B. (2010). Construction of an off-centered entropy for the supervised learning of imbalanced classes: Some first results. *Communications in Statistics - Theory and Methods*, *39*(3), 493-507.

Safavian, S. R., & Landgrebe, D. (1991). A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, *21*(3), 660-674.

Sinapiromsaran, K., & Techaval, N. (2012). Network intrusion detection using multi-attributed frame decision tree. In *2nd International Conference on Digital Information and Communication Technology and its Applications, DICTAP 2012* (pp. 203-207). Bangkok, Thailand

Sirisomboonrat, C., & Sinapiromsaran, K. (2012). Breast cancer diagnosis using multi-attributed lens recursive partitioning algorithm. In *2012 10th International Conference on ICT and Knowledge Engineering* (pp. 40-45). Bangkok, Thailand

Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., . . . Zhou, Z. H. (2008). Top 10 algorithms in data mining. *Knowledge and Information Systems*, *14*(1), 1-37.

Yeon, Y., Han, J., & Ryu, K. (2010). Landslide susceptibility mapping in Injae, Korea, using a decision tree. *Engineering Geology*, *116*(3-4), 274-283.

Yu, Z., Haghighat, F., Fung, B., & Yoshino, H. (2010). A decision tree method for building energy demand modeling. *Energy and Buildings*, *42*(10), 1637-1646.

Zighed, D., Ritschard, G., & Marcellin, S. (2010). Asymmetric and sample size sensitive entropy measures for supervised learning. *Advances in Intelligent Information Systems*, *265*, 27-42.