Pattern Anal Applic (2017) 20:769–782
DOI 10.1007/s10044-016-0533-3

THEORETICAL ADVANCES

Decision tree induction based on minority entropy for the class imbalance problem

Kesinee Boonchuay[1] · **Krung Sinapiromsaran**[1] · **Chidchanok Lursinsap**[1]

Received: 15 April 2015 / Accepted: 5 January 2016 / Published online: 22 January 2016

Abstract Most well-known classifiers can predict a balanced data set efficiently, but they misclassify an imbalanced data set. To overcome this problem, this research proposes a new impurity measure called minority entropy, which uses information from the minority class. It applies a local range of minority class instances on a selected numeric attribute with Shannon's entropy. This range defines a subset of instances concentrating on the minority class to be constructed by decision tree induction. A decision tree algorithm using minority entropy shows improvement compared with the geometric mean and *F*-measure over C4.5, the distinct class-based splitting measure, asymmetric entropy, a top–down decision tree and Hellinger distance decision tree on 24 imbalanced data sets from the UCI repository.

Keywords Decision tree · Minority entropy · Minority range · Geometric mean · *F*-measure

1 Introduction

In 2011, the decision tree was voted one of the most used data mining algorithms [1]. It was included in 2008 [2] for the C4.5 [3] algorithm as one of the top 10 algorithms for data mining. The idea of the decision tree derived from the concept learning system (CLS) [4], which applies a recursive partitioning method to construct a tree, and CLS-inspired descendant algorithms, such as ID3 [5], C4.5, classification and regression tree (CART) [6], top–down decision tree (DKM) [7–9], asymmetric entropy (AE) [10, 11], Hellinger distance decision tree (HDDT) [12] and distinct class-based splitting measure (DCSM) [13].

Despite much research based on C4.5, there are some limitations of the decision tree that can be improved further. This paper focuses on a well-known problem in classification called a class imbalanced problem, which occurs in a data set with a highly different number of instances among classes. For example, in a two-class data set, one class has only 1 % of instances, while the remaining instances are in the other class. Several classifiers predict all instances as the second class because this achieves an accuracy of 99 %, which misclassifies all instances in the first class. In an imbalanced problem, a class with a small number of instances is called the minority class, while the other class is known as the majority class. In real-world classification, users are frequently more interested in the accuracy of predicting minority class instances than the accuracy of predicting majority class instances, especially in cases in which the cost of misclassifying minority class instances is higher than that of majority class instances, such as network intrusions [14] and the detection of oil spills using satellite radar images [15].

In 2010, [16] proposed an insensitive measure for a decision tree called the class confidence proportion decision tree (CCPDT), which replaced the use of Shannon's entropy (SE) [17] as the split measure. The CCPDT computed the class confidence proportion for each attribute and selected the best split from the best confidence value. The

✉ Krung Sinapiromsaran
Krung.S@chula.ac.th

Kesinee Boonchuay
bkasinee@hotmail.com

Chidchanok Lursinsap
lchidcha@chula.ac.th

[1] Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University, Bangkok 10330, Thailand

class confidence proportion used by the CCPDT was modified from the traditional confidence in [18] to focus on instances in each class instead of instances in each partition. Additionally, it integrated Fisher's exact test [19] to prune the branches, which improved overall performance. According to the results in the paper, the proposed measure yielded statistically improved performance on imbalanced data sets compared with traditional confidence.

Other methods exist that are based on a sampling technique that aims to balance the number of instances between classes by either over-sampling, under-sampling or both. For an over-sampling technique, the number of minority instances is synthesized to balance instances between the majority and minority instances, for example, adaptive synthetic (ADASYN) [20], synthetic minority over-sampling technique (SMOTE) [21], borderline-SMOTE [22], density-based synthetic minority over-sampling technique (DBSMOTE) [23] and safe-level-SMOTE [24]. For an under-sampling technique, some majority instances are removed instead of synthesizing minority instances, for example, majority under-sampling technique (MUTE) [25]. These sampling techniques can be applied with any existing classifier, but they change the distribution within the data set. Additionally, an increase in the number of instances requires extra processing time.

In this paper, we propose a new impurity measure called minority entropy (ME) to improve the performance of decision tree induction on an imbalanced data set. This technique aims to reduce the effect of overwhelming majority class instances, while maintaining all minority class instances in the current data set. The concept of an under-sampling technique permanently eliminates majority class instances, while ME ignores majority class instances along an examining attribute. ME therefore preserves all instances in the data set. ME is designed to focus on the minority class instances surrounded by the instances from another class, which increases the ability to recognize minority class instances within an attribute range. The minority range on the selected attribute is defined as the difference between the largest and smallest values of all minority class instances. According to the results in the fifth section of this paper, ME improves performance to manage an imbalanced problem compared with the geometric mean and the harmonic mean of the detection rate and false alarm rate (F-measure).

The next section in this paper elaborates on decision tree induction. The third section presents related works and the fourth section explains the details and proofs of ME properties. The fifth section presents experimental results of ME compared with C4.5, DCSM, AE, DKM and HDDT. The final section contains the conclusion and future work.

2 Decision tree

A decision tree is a tree structure model that consists of multiple nodes connected by branches. There are three types of nodes, which are the root, internal and leaf nodes. The root node reaches every other node in the tree and an internal node represents an attribute that is used to test instances. Each internal node connects to its child nodes by branches that satisfy specified conditions. A leaf node contains instances that are classified in a specific class.

In this paper, we use a two-class data set, which consists of instances from a positive class and negative class. Decision tree induction is an algorithm to construct a decision tree from training instances. At each node, all instances are separated into partitions based on their values in the specified attribute to reduce impurity of the data set. Therefore, the impurity measure plays an important role in determining the best split. If each partition contains only instances in the same class, the impurity is zero. Conversely, if each partition contains instances of all classes equally, the impurity is set to the highest value. The widely used impurity measures are entropy [17], Gini [26] and classification error.

The structure of the data set is presented in Fig. 1 to describe the impurity formulae. Given the data set $\boldsymbol{D}$, the attribute $a \in \{A_1, A_2, ..., A_m\}$ represents the selected attribute. $\boldsymbol{D}$ consists of instances from a set of two classes $\boldsymbol{C} = \{+, -\}$. Note that $\boldsymbol{D}_+ \cap \boldsymbol{D}_- = \emptyset$ and $\boldsymbol{D} = \bigcup_{c \in \boldsymbol{C}} \boldsymbol{D}_c$.

Let z be the value of the attribute $\boldsymbol{a}$. Let $\text{proj}_a(i)$ denote the projection of the instance $\boldsymbol{i}$ of the attribute $\boldsymbol{a}$. From Eq. 1, $sl_a(D, z)$ denotes a set of instances with values in the attribute $\boldsymbol{a}$ less than or equal to z. From Eq. 2, $\boldsymbol{sr}_a(\boldsymbol{D}, z)$

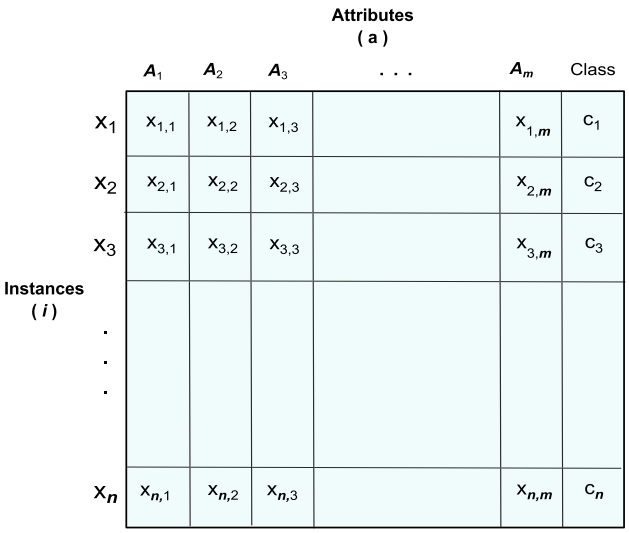

Fig. 1 A structure of a data set

denotes a set of instances with values in the attribute a greater than z.

$$sl_{\mathrm{a}}(D,z) = \{i \in D | proj_{\mathrm{a}}(i) \leq z\} \quad (1)$$

$$sr_{\mathrm{a}}(D,z) = \{i \in D | proj_{\mathrm{a}}(i) > z\} \quad (2)$$

In a standard decision tree, SE, denoted by Ent, is used as an impurity measure in Eq. 3:

$$Ent(D) = -\frac{|D_+|}{|D|}\log_2\frac{|D_+|}{|D|} - \frac{|D_-|}{|D|}\log_2\frac{|D_-|}{|D|} \quad (3)$$

$$S_{\mathrm{a}}(D,z) = \frac{|sl_{\mathrm{a}}(D,z)|}{|D|}Ent(sl_{\mathrm{a}}(D,z)) + \frac{|sr_{\mathrm{a}}(D,z)|}{|D|}Ent(sr_{\mathrm{a}}(D,z)) \quad (4)$$

$$Ent_{\mathrm{a}}(D) = \min_{z\in\{proj_{\mathrm{a}}(i)|i\in D\}} S_{\mathrm{a}}(D,z) \quad (5)$$

Equation 4 presents the formula that applies entropy as the impurity measure for the value z of the attribute a. From Eqns. 1 and 2, all instances are separated into two partitions by z. Then the first and second terms compute entropies for the first and second partitions, respectively. In these terms, entropies for each partition are weighted by the ratio of instances in the data set. Equation 5 selects the minimum entropy among the split values of the attribute a. Then, the best attribute is selected as the maximum value of $Ent(D) - Ent_{\mathrm{a}}(D)$ over the attribute a.

C4.5 is a descendant of ID3 that uses information gain to measure the impurity of a discrete valued attribute. In Eq. 6, $InfoGain_{\mathrm{a}}(D)$ denotes the information gain of the attribute a. The information gain is computed from the difference between entropy before and after the split. The attribute a that provides the highest value of information gain is selected as the best split. However, ID3 tends to favor an attribute with many distinct values, while C4.5 avoids this situation by using split information in Eq. 7. Split information is used to estimate the distribution of instances after the split for the value z of the attribute a. If the number of instances in each partition is equal, the split information is 1. In other cases, this value is less than 1. The gain ratio uses split information as a denominator to reduce the bias of the information gain. Therefore, if all instances are located only in a single partition, the gain ratio will obtain the highest value. $SplitInfo_{\mathrm{a}}(D)$ is defined as the split information.$GainRatio_{\mathrm{a}}(D)$ denotes the gain ratio for the attribute a that provides the minimum value of the gain ratio.

$$InfoGain_{\mathrm{a}}(D) = Ent(D) - Ent_{\mathrm{a}}(D) \quad (6)$$

$$SplitInfo_{\mathrm{a}}(D,z) = -\frac{|sl_{\mathrm{a}}(D,z)|}{|D|}\log_2\frac{|sl_{\mathrm{a}}(D,z)|}{|D|} - \frac{|sr_{\mathrm{a}}(D,z)|}{|D|}\log_2\frac{|sr_{\mathrm{a}}(D,z)|}{|D|} \quad (7)$$

$$GainRatio_{\mathrm{a}}(D) = \min_{z\in\{proj_{\mathrm{a}}(i)|i\in D\}} \frac{S_{\mathrm{a}}(D,z)}{SplitInfo_{\mathrm{a}}(D,z)} \quad (8)$$

Another interesting measure proposed in recent years is DCSM, which combines two concepts. The first concept addresses the number of distinct classes in each partition after the split. The fewer distinct classes in partitions, the purer the partitions. The results in the paper [13] showed that DCSM could improve the performance of C4.5, and it produced a compact decision tree. The next section presents related works that focus on techniques based on decision tree induction targeting an imbalanced data set.

3 Related works for an imbalanced problem

As discussed in the previous section, C4.5 and DCSM are not designed to solve an imbalanced problem; hence they tend to yield unsatisfactory performance for an imbalanced data set. Many researchers have addressed this problem and provided remedies for C4.5.

In C4.5, SE is used as a split measure that is a symmetric entropy. It achieves its maximum value at 1. In a two-class data set, the maximum of SE occurs when both class ratios are 0.5 and it achieves its minimum value at 0, when the ratio of one class is 0 and the other is 1. By contrast, AE modifies this entropy. It achieves its maximum value when one class ratio equals the parameter called θ, which is the entropy skewness. This parameter can be set within the range 0 to 1 to favor instances in the minority class. θ can be determined using experiments such as in [27]. However, an unsuitable θ can be a drawback for AE. Our experiments in the fifth section of this paper use the highest performing θ from the training process.

Regarding other techniques, DKM [8, 9] and HDDT [12] are skew-insensitive split measures that are designed to handle an imbalanced data set. The skew-insensitive split measure is not affected by the ratio of the number of instances among classes. The authors of DKM introduced a new split measure for top–down decision tree induction in [7] in 1996. In that paper, the authors aimed to improve the performance of decision tree induction without focusing on an imbalanced problem. In [8, 9], DMK was adapted to run on an imbalanced data set. In 2008, the technique in [12] used a measure of distributional divergence as the splitting criteria, called Hellinger distance. The authors presented the proof that the Hellinger distance was less sensitive for class distribution than DKM, and it yielded improvement over DKM.

Improvement for an imbalanced data set can be achieved by the use of ME, which is outlined in the following section.

4 Minority entropy

4.1 Motivation

Most classifiers are unable to recognize minority class instances because of the very small number of instances compared with the majority class instances. To address this behavior, most techniques compensate for a tiny number of minority class instances or diminish the significance of a large number of majority class instances, which increases the likelihood of minority class prediction. The classifier is thus able to yield enhanced accuracy with regard to these minority class instances. ME improves decision tree induction by diminishing majority instances outside the range for all minority instances, called the minority range. In Fig. 2, the minority range is the size of the middle box, which provides sufficient information to construct a decision tree as illustrated at the top of the figure. All instances outside the range that would not affect the ability to predict minority class instances are excluded. Therefore, ME considers only instances in the minority range to construct a decision tree.

Given a sample data set, where the minority class is represented as the positive class and the majority class is represented as the negative class, Fig. 3a shows a split value at 0.55 for the first attribute and Fig. 3b shows a split value at 0.35 for the second attribute. ME provides a higher value for the second attribute than the first because the significance of minority class instances increases after the split, while the standard decision tree splits by the first attribute because of its entropy. ME achieves this by ignoring all majority class instances outside the minority range.

The effect of removing majority class instances is illustrated in Fig. 4a, b. Figure 4a shows the data set after removing all majority class instances with values in the first attribute less than 0.15 and Fig. 4b removes all

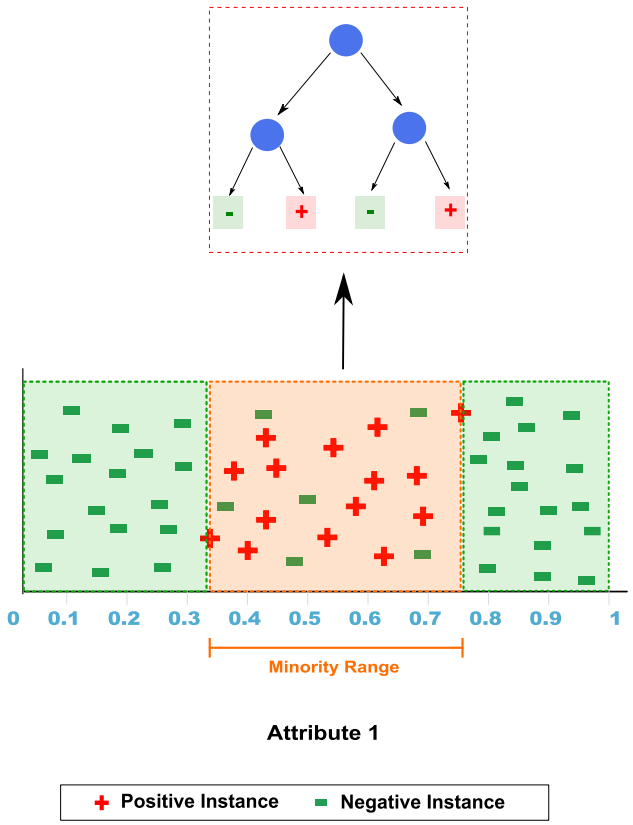

Fig. 2 An example of a decision tree

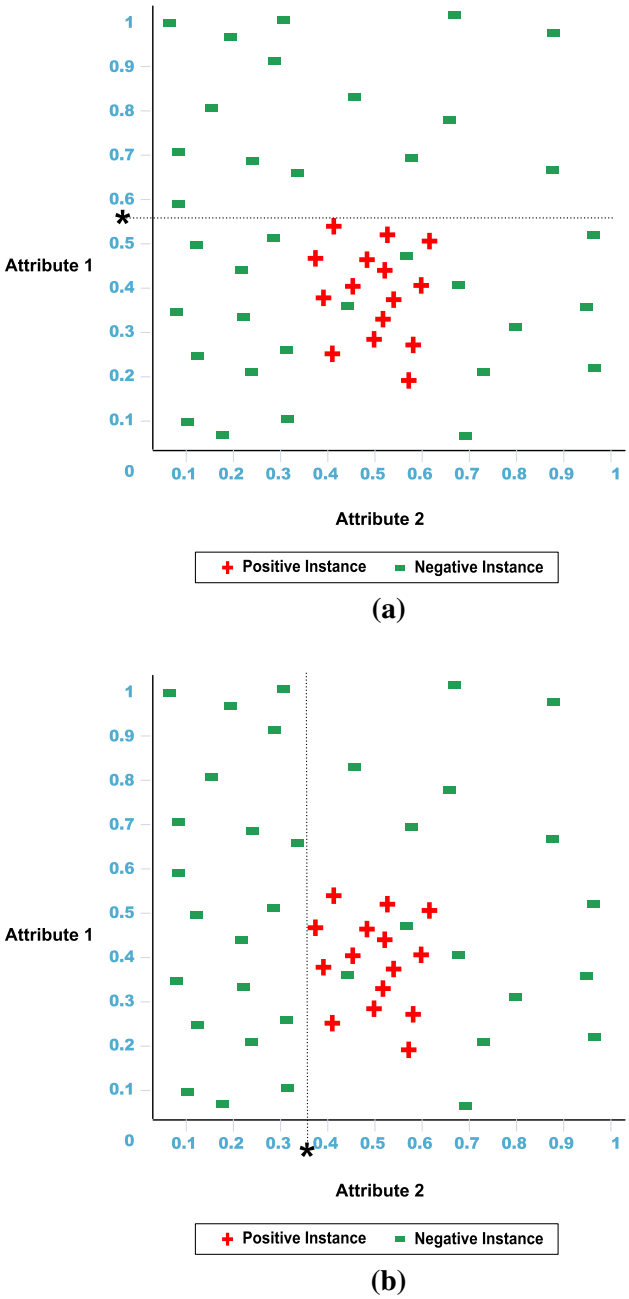

Fig. 3 **a** Split by attribute 1 and **b** split by attribute 2

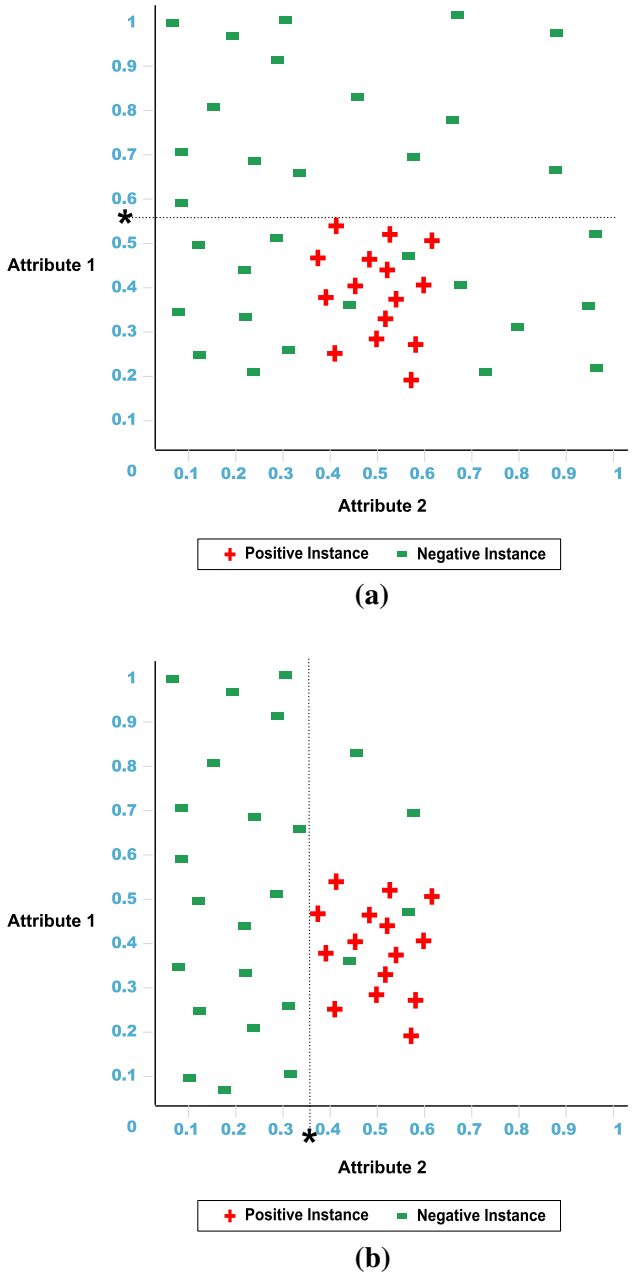

Fig. 4 **a** Split by attribute 1 and **b** split by attribute 2

majority class instances with values in the second attribute greater than 0.65. The second attribute obviously provides a better split than the first, as shown in Fig. 4b. ME considers the minority class instance distribution and ignores majority class instances outside the minority range, which is explained in the next section.

4.2 Minority entropy

Minority entropy is a new impurity measure to be applied with decision tree induction, which is computed from the minority class instances within the minority range. To compute ME, the minority range is defined by the range of values between $\min_{k\in D_+} \text{proj}_a(k)$ and $\max_{l\in D_+} \text{proj}_a(l)$ of the attribute a. Then a set of instances within the minority range is defined:

$$spr_a(D) = \{i \in D | \min_{k\in D_+} proj_a(k) \le proj_a(i) \le \max_{l\in D_+} proj_a(l)\} \quad (9)$$

Using $spr_a(D)$ instead of D in the entropy formula, the ratio of minority class instances in the minority range is greater than or equal to the ratio of minority class instances from SE, and the ratio of majority class instances in the minority range is less than or equal to the ratio of majority class instances from SE, which is proved in Theorem 1. This theorem, proposed by us, is used to demonstrate the idea of ME. The details of the theorem are provided as follows:

Theorem 1 Define D as a set of instances, $\boldsymbol{D} = \{x_1, x_2, ..., x_n\}$, where n is the number of instances. Each instance consists of m attributes, $\{A_1, A_2, ..., A_m\}$. Let $a \in \{A_1, A_2, ..., A_m\}$. D_+ is a nonempty set of positive instances and D_- a set of negative instances, such that $D_+ \cap D_- = \emptyset$ and $D_+ \cup D_- = D$. $spr_a(D)_+$ and $spr_a(D)_-$ are sets of positive and negative instances in $spr_a(D)$. The following statements are true.

1. $\dfrac{|D_+|}{|D|} \le \dfrac{|spr_a(D)_+|}{|spr_a(D)|}$
2. $\dfrac{|D_-|}{|D|} \ge \dfrac{|spr_a(D)_-|}{|spr_a(D)|}$

Proof Because $spr_a(D) \subseteq D$, $|spr_a(D)| \le |D|$, Moreover, all positive instances in D lie within the minority range. Therefore, $spr_a(D)_+ = D_+$. We can conclude that the first statement is true.

Let $Q = D - spr_a(D)$. Q_+ is a set of positive instances in Q. Q_- is a set of negative instances in Q. Because no positive instances lie in Q, $Q = Q_-$. Let c be the number of instances in Q_-.

$$\begin{aligned}\frac{|D_-|}{|D|} &= \frac{|spr_a(D)_- \cup Q_-|}{|spr_a(D) \cup Q|} \\ &= \frac{|spr_a(D)_-| + c}{|spr_a(D)| + c} \\ &\ge \frac{|spr_a(D)_-|}{|spr_a(D)|}.\end{aligned}$$

Thus, the second statement is true.

Minority entropy is defined according to Theorem 1 as follows: $ME_a(D)$ denotes the minority entropy formula of a data set (D) for an attribute (a).

$$ME_a(D) = Ent_a(spr_a(D)) \quad (10)$$

From this equation, ME provides the highest value, which is 1, when the number of instances in the positive and negative classes are the same. It also provides the lowest value, which is 0, when all instances are in the same class, as for SE. Moreover, ME provides the zero value when minority class instances are embedded between majority class instances and no majority class instances appear inside the range of minority class instances along the attribute, while decision tree induction using SE may not always provide the zero value. The proof of this case is provided in Theorem 2. □

Theorem 2 Define D as a set of instances, $D = D_+ \cup D_-$. If $spr_a(D) = D_+$, then $ME_a(D)$ provides the lowest value = 0.

Proof Suppose $spr_a(D) = \boldsymbol{D_+}$. Because $spr_a(D) = spr_a(D)_+ \cup spr_a(D)_-$, then $spr_a(D)_- = \emptyset$.

$$Ent_a(spr_a(D)) = -\frac{|spr_a(D)_+|}{|spr_a(D)|}\log_2\frac{|spr_a(D)_+|}{|spr_a(D)|} - \frac{|spr_a(D)_-|}{|spr_a(D)|}\log_2\frac{|spr_a(D)_-|}{|spr_a(D)|} = -1\log_2 1 - 0 = 0$$

Because $ME_a(D) = Ent_a(spr_a(D))$ from Eq. 10, then $ME_a(D) = 0$.

If the range along the attribute a contains only minority class instances, then ME has the lowest value for this attribute. Decision tree induction using ME, therefore, selects this attribute as the best split (see Figs. 3 and 4). Example 1 illustrates the steps for computing ME, which supports Theorem 2. □

Example 1 The sample data set (D) has 10 positive instances and 28 negative instances, as shown in Fig. 5. For the first attribute, the first partition consists of all instances with values less than or equal to 0.7 and the second partition consists of all instances with values greater than 0.7. Therefore, ME for the first attribute is 0.4019.

For the second attribute, the first partition consists of all instances with values less than or equal to 0.6. The second partition consists of all instances with values greater than 0.6. ME for the second attribute is zero. Decision tree induction using ME selects the second attribute as a split.

SE is 0.4019, which is the same value as ME for the first attribute, but it is 0.4373 for the second attribute, which is higher than the value for the first attribute. Therefore, the decision tree using SE selects the first attribute as a split.

The worst case scenario: If the minority range covers all majority instances along the attribute, then $spr_a(D)_+ = D_+$ and $spr_a(D)_- = D_-$. In this case, ME has the same value as SE, so no benefit is gained from using ME as shown in Fig. 6. The following algorithm provides details for decision tree induction using ME:

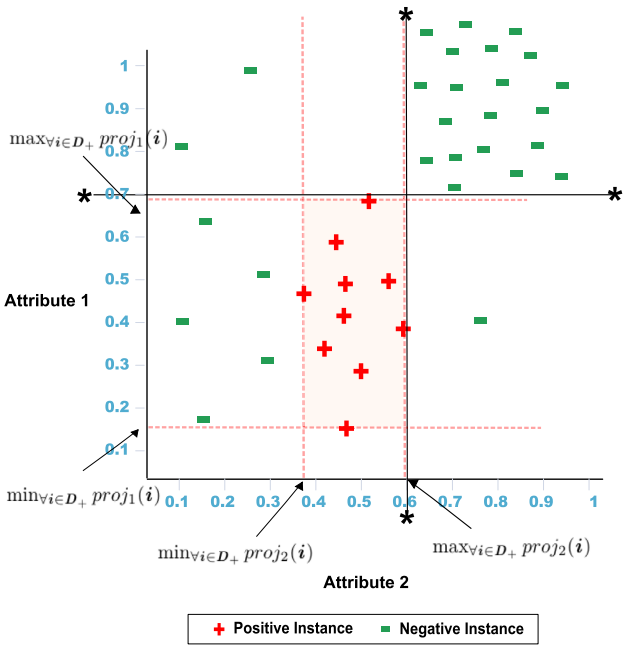

Fig. 5 An example data set

Algorithm: Decision Tree Induction using ME

Input: A data set ($\boldsymbol{D}$) including minority class instances and majority class instances
Output: A decision tree

1. Create a node of the tree
2. If all instances are in the same class then
3. Return the node labeled as that class
4. For each attribute $\boldsymbol{a}$ in $\boldsymbol{D}$
5. Compute $\boldsymbol{spr_a(D)}$ for attribute $\boldsymbol{a}$
6. For each value($\boldsymbol{z}$) of ($\boldsymbol{spr_a(D)}$) in attribute $\boldsymbol{a}$
7. Split the instances by $\boldsymbol{spr_a(D)}$ into a left partition and right partition
8. Compute ME for a value($\boldsymbol{z}$) of $\boldsymbol{spr_a(D)}$
9. End For
10. Select the best split for attribute $\boldsymbol{a}$ from the value to provide the minimum value of $\boldsymbol{ME}$
11. End For
12. Separate the instances into partitions corresponding to the selected attribute
13. Iterate for each partition

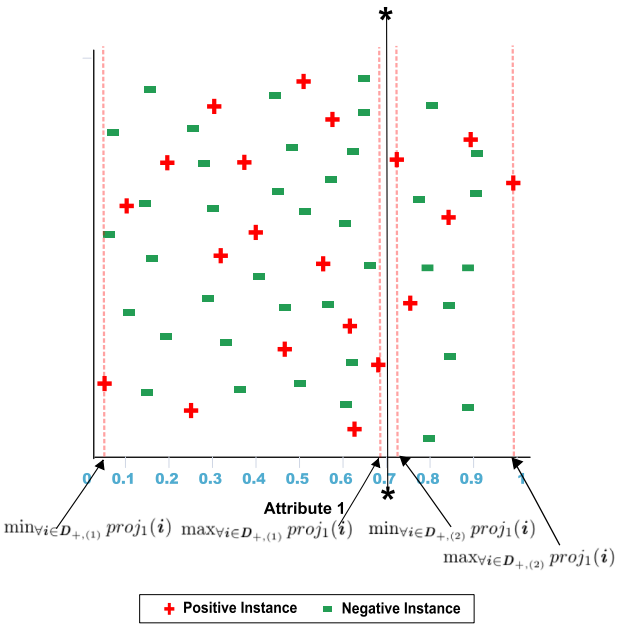

Fig. 6 An example data set where ME and SE provide the same information

---

Minority Entropy (D, a)

---

1. Find the set of instances in a *minority range* $(\boldsymbol{MR})$ for attribute $\boldsymbol{a}$
2. $\boldsymbol{Prob\_Pos} = \frac{\text{number of positive instances}}{\text{number of instances}}$
3. $\boldsymbol{Prob\_Neg} = \frac{\text{number of negative instances}}{\text{number of instances}}$
4. $\boldsymbol{Prob\_PosMR} = \frac{\text{number of positive instances in } \boldsymbol{MR}}{\text{number of instances in } \boldsymbol{MR}}$
5. $\boldsymbol{Prob\_NegMR} = \frac{\text{number of negative instances in } \boldsymbol{MR}}{\text{number of instances in } \boldsymbol{MR}}$
6. $\boldsymbol{ME}$ = Compute entropy from ($\boldsymbol{Prob\_PosMR}$ and $\boldsymbol{Prob\_NegMR}$)
7. Return $\boldsymbol{ME}$

Time complexity: To find the best split, the time complexity for computing the gain ratio in C4.5 is $O(m \cdot n)$ for each level, where n is the number of instances and m is the number of attributes. The time complexity of C4.5 was derived in [28]. For ME, the time complexity for each level is also $O(m \cdot n)$, which is the same as for C4.5. Theorem 3 shows the proof of the time complexity for ME.

Theorem 3 The time complexity of computing ME for all attributes is $\boldsymbol{O(m \cdot n)}$, where $\boldsymbol{n}$ is the number of instance and $\boldsymbol{m}$ is the number of attributes.

Proof Let $\boldsymbol{T(n)}$ be the time complexity of computing ME. $\boldsymbol{T_i(n)}$ denotes the time complexity of the $\boldsymbol{i}$th task. The details of all tasks shown in Minority Entropy() are shown as follows:

- $\boldsymbol{T_1(n)}$ denotes the time complexity of *find_inst_in_MR*. To find the minority range, the minimum and maximum values of positive instances have to be identified first. The algorithm loops through all instances to find the minimum and maximum values. Therefore, $\boldsymbol{T_1(n)} = \boldsymbol{O(n)}$. In the worst case scenario for identifying this group of instances, it takes $\boldsymbol{O(n)}$.
- $\boldsymbol{T_2(n)}$ denotes the time complexity of *count_pos_inst*. $\boldsymbol{T_2(n)} = \boldsymbol{O(n)}$ because each instance is examined once.
- $\boldsymbol{T_3(n)}$ denotes the time complexity of *count_neg_inst*. $\boldsymbol{T_3(n)} = \boldsymbol{O(n)}$ because each instance is examined once.
- $\boldsymbol{T_4(n)}$ denotes the time complexity of *count_pos_inst_in_MR*. $\boldsymbol{T_4(n)} = \boldsymbol{O(n)}$ because each instance is examined once.
- $\boldsymbol{T_5(n)}$ denotes the time complexity of *count_neg_inst_in_MR*. $\boldsymbol{T_5(n)} = \boldsymbol{O(n)}$ because each instance is examined once.
- $\boldsymbol{T_6(n)}$ denotes the time complexity of *compute_ME*. $\boldsymbol{T_6(n)} = \boldsymbol{O}(1)$.

Because task 1 through task 6 are performed for each attribute, their time complexities have to be multiplied by the number of attributes $(\boldsymbol{m})$. Task 2 through task 5 can be computed in $\boldsymbol{O(n)}$. Therefore,

$$\begin{aligned} T(n) &= m \cdot (T_1(n) + T_2(n) + T3(n) + T_4(n) \\ &\quad + T_5(n) + T_6(1)) \\ &= m \cdot (O(n) + O(n) + O(n) + O(n) \\ &\quad + O(n) + O(1)) \\ &= m \cdot (O(n) + 1) \\ &= O(m \cdot n) \end{aligned}$$

□

5 Experimental results

5.1 Data sets

The experiments were performed on 24 data sets from the UCI repository [29] with minority class instances of less than 30 %. SatImage, OpticDigits, Adult and PenDigits data sets had training and testing available in the UCI repository, while the other data sets were validated using tenfold cross-validation. Table 1 presents the data sets in the order of their percentage of minority class instances. The first and the second columns contain the data set

Table 1 Detail of imbalanced two-class data sets

No.	Data sets	Minor. class / major. class	#att.	#inst.	% Minority (%)
1	Page blocks	1 / The remainder	10	5473	0.51
2	Thyroid	1 / The remainder	21	720	2.36
3	Letter	A / The remainder	16	20,000	3.95
4	Abalone	18 / 9	8	731	5.75
5	Glass	5 / The remainder	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 remainder	13	988	9.11
9	PenDigit*	5 / The remainder	16	10,992	9.60
10	OpticDigits*	4 / The remainder	36	6435	9.76
11	Ecoli(imU)	O / The remainder	64	5620	9.86
12	SatImage(4)*	imU / The remainder	7	336	10.42
13	Fertility	O / The remainder	10	100	12.00
14	Breast tissue	con / The remainder	10	106	13.21
15	Segmentation	1 / The remainder	19	2310	14.29
16	Ecoli(pp)	pp / The remainder	7	336	15.48
17	Vertebral column	DH / The remainder	6	310	19.35
18	Shuttle*	The remainder / 1	9	58,000	21.40
19	SatImage(1)*	1 / The remainder	4	748	23.80
20	Transfusion	1 / The remainder	36	6435	23.82
21	Adult*	0 / The remainder	10	195	24.62
22	Parkinsons	1 / The remainder	14	45,099	24.76
23	Haberman	2 / The remainder	3	306	26.47
24	Wine	3 / The remainder	13	178	26.97

"*" indicates the data sets that use train/test partitions

number and name, and the third column contains the selected class as a minority class and the remaining classes as majority classes. For example, in the Letter data set, class A is selected as the minority class, while the remaining classes are defined as majority classes. SatImage(1) and SatImage(4) are SatImage data sets for which class 1 and class 4 are selected as minority classes, respectively. Ecoli(imU) and Ecoli(pp) are Ecoli data sets for which class imU and class pp are selected as minority classes, respectively. The fourth column contains the number of attributes in the data sets and the fifth column contains the number of instances in the data sets. The last column contains the percentage of minority class instances. These data sets were used in all experiments, which aimed to show the results for handling imbalanced data sets using C4.5, DCSM, DKM, HDDT, AE and ME.

5.2 Performance measures and evaluation

C4.5, DCSM, DKM, HDDT, ME and AE were implemented using MATLAB. Only AE required the parameter θ, which was obtained from the best performance in the training data set. The experimental results show a comparison of ME with C4.5, DCSM, DKM, HDDT ME and AE using the F-measure [30]. According to [31], the F-measure is one of the performance measures that is suitable for a class imbalanced problem, which harmonizes the recall in Eq. 11 and precision in Eq. 12. β is used as the weight of importance between the recall and precision, so it is set to 1, which means the recall and precision are equally important. The formulae for the F-measure and geometric mean are provided in Eqs. 13 and 14, respectively. The recall and precision performance measures are extracted from the contingency table shown in Table 2. In this table, true positive (TP) denotes the number of positive instances that are correctly predicted as positive instances, true negative (TN) denotes the number of negative instances that are correctly predicted as negative instances, false positive (FP) denotes the number of negative instances that

Table 2 Contingency table

	Actual positive	Actual negative
Predicted positive	True positive (TP)	False positive (FP)
Predicted negative	False negative (FN)	True negative (TN)

Table 3 Simulate testing result

	***F*-measure**	Geometric mean
SE (5 % of minority class instances)	0.6667	0.7705
ME (5 % of minority class instances)	1	1
SE (10 % of minority class instances)	0.9091	0.9888
ME (10 % of minority class instances)	1	1
SE (15 % of minority class instances)	0.9677	0.9941
ME (15 % of minority class instances)	1	1
SE (20 % of minority class instances)	1	1
ME (20 % of minority class instances)	1	1

Table 4 The comparison result by the geometric mean for imbalanced data sets

No.	Data sets	Geometric mean					
		C4.5	DCSM	AE	DKM	HDDT	ME
1	Page blocks	0.8230 (3)	**0.9058 (1)**	0.7555 (4)	0.4225 (6)	0.7062 (5)	0.8446(2)
2	Thyroid	**0.8732 (1)**	**0.8732 (1)**	**0.8732 (1)**	0.5404 (5)	0.4174 (6)	0.8726 (4)
3	Letter	0.9700 (3)	0.9581 (5)	0.9739(2)	0.8953 (6)	0.9621 (4)	**0.9760(1)**
4	Abalone	0.4291 (3)	0.4805 (2)	0.3987(4)	0.3710 (5)	0.3395 (6)	**0.5259(1)**
5	Glass	0.8237 (4)	**0.9584 (1)**	**0.9584 (1)**	0.7647 (5)	0.6709 (6)	**0.9584 (1)**
6	Cleveland	0.6104 (2)	0.3873 (5)	0.5353(4)	0.0000 (6)	**0.6665 (1)**	0.5460(3)
7	LED display domain	**0.8904 (1)**	0.8754 (5)	**0.8904 (1)**	0.8732 (6)	**0.8904 (1)**	**0.8904 (1)**
8	Vowel	0.9629 (2)	0.9402 (4)	0.9423 (3)	0.5823 (6)	0.7769 (5)	**0.9743 (1)**
9	PenDigit	0.8848 (4)	**0.9221 (1)**	0.9056 (3)	0.7660 (6)	0.7920 (5)	0.9209 (2)
10	OpticDigits	0.9627 (5)	0.9765 (2)	0.9726 (3)	0.8057 (6)	0.9705 (4)	**0.9806 (1)**
11	Ecoli(imU)	0.6841 (3)	0.6602 (4)	**0.7742 (1)**	0.4908 (6)	0.6186 (5)	0.7420 (2)
12	SatImage(4)	0.7397 (3)	0.7338 (4)	**0.7479 (1)**	0.6612 (6)	0.7104 (5)	0.7419 (2)
13	Fertility	0.3769 (4)	0.2735 (5)	0.3844 (3)	0.2647 (6)	**0.5573 (1)**	0.5401 (2)
14	Breast tissue	0.9003 (4)	**0.9208 (1)**	**0.9208 (1)**	0.7137 (6)	0.8767 (5)	**0.9208 (1)**
15	Segmentation	0.9843 (4)	**0.9944 (1)**	0.9922 (2)	0.9031 (6)	0.9031 (5)	0.9894 (3)
16	Ecoli(pp)	0.8506 (2)	0.8320 (4)	0.8397 (3)	0.7337 (5)	0.6987 (6)	**0.8522 (1)**
17	Vertebral column	0.6693 (4)	0.6708 (3)	0.7189 (2)	0.6481 (5)	0.6432 (6)	**0.7298 (1)**
18	Shuttle	0.9998 (3)	0.9989 (4)	0.9998 (2)	0.9934 (6)	0.9988 (5)	**1.0000 (1)**
19	SatImage(1)	0.9617 (3)	0.9655 (2)	0.9588 (4)	0.9238 (6)	0.9360 (5)	**0.9677 (1)**
20	Transfusion	0.5239 (3)	**0.5563 (1)**	0.5327 (2)	0.4986 (5)	0.4745 (6)	0.5239 (3)
21	Adult	0.7460 (2)	0.7366 (4)	0.7361 (5)	0.6686 (6)	**0.7474 (1)**	0.7447 (3)
22	Parkinsons	0.7470 (5)	0.8081 (2)	0.8063 (3)	0.6776 (6)	0.7673 (4)	**0.8274 (1)**
23	Haberman	**0.5346 (1)**	0.5111 (5)	0.5185 (3)	0.4120 (6)	0.5121 (4)	0.5277 (2)
24	Wine	**0.9638 (1)**	0.9494 (3)	0.9608 (2)	0.8540 (5)	0.7145 (6)	0.9282 (4)
	Average rank	2.92	2.92	2.50	5.71	4.46	**1.83**
	Friedman test	**0.010515**	**0.033006**	**0.049535**	**0.000001**	**0.000393**	Base

Bold values represent the best performance measure comparing among all classifiers

are inaccurately predicted as positive instances and false negative (FN) denotes the number of positive instances that are inaccurately predicted as negative instances.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{11}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{12}$$

$$F\text{-measure} = \frac{(1+\beta)^2 \cdot (\text{Recall} \cdot \text{Precision})}{\beta^2 \cdot (\text{Recall} + \text{Precision})} \tag{13}$$

$$\text{Geometric mean} = \sqrt{\frac{\text{TP}}{\text{TP} + \text{FN}} \cdot \frac{\text{TN}}{\text{TN} + \text{FP}}} \tag{14}$$

In the next section, the comparison of the results using C4.5, DCSM, AE, DKM, HDDT and ME are presented by the *F*-measure, geometric mean, precision and recall. In this paper,

Table 5 The comparison result by the *F*-measure for imbalanced data sets

No.	Data sets	*F*-measure					
		C4.5	DCSM	AE	DKM	HDDT	ME
1	Page blocks	0.6667 (3)	**0.8070 (1)**	0.6400 (4)	0.2857 (6)	0.5000 (5)	0.7273 (2)
2	Thyroid	**0.8125 (1)**	**0.8125 (1)**	**0.8125 (1)**	0.3704 (5)	0.2069 (6)	0.7879 (4)
3	Letter	0.9442 (3)	0.9314 (5)	0.9494 (2)	0.8303 (6)	0.9349 (4)	**0.9544 (1)**
4	Abalone	0.2192 (3)	0.2740 (2)	0.1728 (4)	0.1644 (5)	0.1449 (6)	**0.3158 (1)**
5	Glass	0.6923 (4)	**0.9231 (1)**	**0.9231 (1)**	0.5161 (5)	0.5000 (6)	**0.9231 (1)**
6	Cleveland	0.4348 (2)	0.2105 (5)	0.2857 (4)	0.0000 (6)	**0.4800 (1)**	0.3636 (3)
7	LED display domain	**0.7895 (1)**	0.7733 (5)	**0.7895 (1)**	0.7532 (6)	**0.7895 (1)**	**0.7895 (1)**
8	Vowel	0.9333 (2)	0.9143 (3)	0.8852 (4)	0.4593 (6)	0.7051 (5)	**0.9451 (1)**
9	PenDigit	0.8373 (4)	**0.8885 (1)**	0.8683 (3)	0.6119 (6)	0.6782 (5)	0.8777 (2)
10	OpticDigits	0.9379 (5)	0.9474 (3)	0.9575 (2)	0.6919 (6)	0.9389 (4)	**0.9609 (1)**
11	Ecoli(imU)	0.5397 (3)	0.4923 (4)	**0.6197 (1)**	0.2857 (6)	0.4516 (5)	0.6061 (2)
12	SatImage(4)	0.5541 (4)	0.5634 (3)	**0.5891 (1)**	0.4734 (6)	0.5450 (5)	0.5823 (2)
13	Fertility	0.1481 (4)	0.0909 (5)	0.1667 (3)	0.0741 (6)	**0.3636 (1)**	0.2963 (2)
14	Breast tissue	0.7742 (5)	**0.8889 (1)**	**0.8889 (1)**	0.5000 (6)	0.8148 (4)	**0.8889 (1)**
15	Segmentation	0.9742 (4)	**0.9894 (1)**	0.9834 (2)	0.8286 (6)	0.8427 (5)	0.9818 (3)
16	Ecoli(pp)	0.7723 (2)	0.7238 (4)	0.7600 (3)	0.5941 (5)	0.5625 (6)	**0.7800 (1)**
17	Vertebral column	0.5172 (4)	0.5217 (3)	0.5862 (2)	0.4615 (6)	0.4640 (5)	**0.6239 (1)**
18	Shuttle	0.9997 (3)	0.9985 (4)	0.9998 (2)	0.9899 (6)	0.9985 (5)	**1.0000 (1)**
19	SatImage(1)	0.9463 (4)	**0.9538 (1)**	0.9469 (3)	0.8682 (6)	0.8936 (5)	0.9511 (2)
20	Transfusion	0.3522 (3)	**0.3939 (1)**	0.3620 (2)	0.3302 (5)	0.3000 (6)	0.3522 (3)
21	Adult	0.6301 (2)	0.6213 (4)	0.6199 (5)	0.5295 (6)	**0.6343 (1)**	0.6300 (3)
22	Parkinsons	0.6263 (5)	0.6990 (3)	0.7416 (2)	0.5294 (6)	0.6739 (4)	**0.7609 (1)**
23	Haberman	**0.3669 (1)**	0.3415 (5)	0.3522 (3)	0.2411 (6)	0.3462 (4)	0.3625 (2)
24	Wine	0.9388 (2)	0.9184 (3)	**0.9474 (1)**	0.8043 (5)	0.6207 (6)	0.9130 (4)
	Average rank	3.08	2.88	2.38	5.75	4.38	**1.88**
	Friedman test	**0.010515**	0.088082	**0.049535**	**0.000001**	**0.000393**	Base

Bold values represent the best performance measure comparing among all classifiers

we focus on the *F*-measure and geometric mean rather than the precision and recall. If a classifier blindly predicts a minority class for all instances, it yields the highest recall while the precision is low. If there is another classifier that focuses on providing the highest precision, the number of misclassifications in minority class instances tends to increase. Therefore, a consideration based on only the precision or recall seems to be biased. To eliminate these drawbacks, the measure must combine both the precision and recall, such as the *F*-measure and geometric mean.

The Friedman test is used to compare ranking across imbalanced data sets. It is a non-parametric statistical test that is suitable for the comparison of classifiers [32]. All experiments use the significance level of $\alpha = 0.05$.

5.3 Results

To show the effectiveness of ME over SE, the simulated data sets were generated and evaluated by the *F*-measure and geometric mean. These data sets consisted of ten attributes of 100 instances. For the first attribute, a specific range was selected and fixed, such as the range between 0.1 and 0.2, so that uniform sampling of minority instances was performed within this range, while the other class instances were located outside this range. The values for the other attributes were selected at random between 0 and 1. Experiments were performed on four groups of data sets with 5, 10, 15 and 20 % of minority class instances. Each group contained 10 simulated data sets. The average results for ME compared with SE by the *F*-measure and geometric mean are presented in Table 3.

For SE, the values for both the *F*-measure and geometric mean increase when the percentage of the minority class instances increases, which means that it can handle a balanced data set better than an imbalanced data set. Note that ME provides the average *F*-measure and geometric mean of 1, which is shown for all data sets. This result is evidence that ME outperforms SE when a data set contains pure minority instances in the minority range.

Despite the effectiveness of ME, real-world data sets rarely exhibit pure minority instances within the minority

Table 6 The comparison result by precision for imbalanced data sets

No.	Data sets	Precision					
		C4.5	DCSM	AE	DKM	HDDT	ME
1	Page blocks	0.6552 (5)	**0.7931 (1)**	0.7273 (3)	0.7143 (4)	0.5000 (6)	0.7407 (2)
2	Thyroid	**0.8667 (1)**	**0.8667 (1)**	0.8667 (1)	0.5000 (5)	0.2500 (6)	0.8125 (4)
3	Letter	0.9454 (3)	0.9429 (4)	0.9482 (2)	0.8560 (6)	0.9421 (5)	**0.9544 (1)**
4	Abalone	0.2581 (3)	0.3226 (2)	0.1795 (6)	0.1935 (4)	0.1852 (5)	**0.3529 (1)**
5	Glass	0.6923 (4)	**0.9231 (1)**	**0.9231 (1)**	0.4444 (6)	0.5455 (5)	**0.9231 (1)**
6	Cleveland	**0.5000 (1)**	0.3333 (4)	0.2667 (5)	0.0000 (6)	0.5000 (1)	0.4444 (3)
7	LED display domain	**0.7692 (1)**	0.7632 (5)	**0.7692 (1)**	0.7250 (6)	**0.7692 (1)**	**0.7692 (1)**
8	Vowel	0.9333 (3)	**0.9412 (1)**	0.8710 (4)	**0.6889 (1)**	0.8333 (6)	0.9348 (2)
9	PenDigit	0.8893 (4)	**0.9228 (1)**	0.9142 (2)	0.6119 (6)	0.7152 (5)	0.8997 (3)
10	OpticDigits	0.9432 (3)	0.9344 (4)	**0.9657 (1)**	0.7169 (6)	0.9286 (5)	0.9556 (2)
11	Ecoli(imU)	0.6071 (3)	0.5333 (4)	0.6111 (2)	0.3214 (6)	0.5185 (5)	**0.6452 (1)**
12	SatImage(4)	0.5279 (5)	0.5581 (4)	**0.5905 (1)**	0.4828 (6)	0.5600 (3)	0.5865 (2)
13	Fertility	0.1333 (4)	0.1000 (5)	0.1667 (3)	0.0667 (6)	**0.4000 (1)**	0.2667 (2)
14	Breast tissue	0.7059 (5)	**0.9231 (1)**	**0.9231 (1)**	0.4444 (6)	0.8462 (4)	**0.9231 (1)**
15	Segmentation	0.9757 (4)	**0.9879 (1)**	0.9790 (3)	0.8152 (6)	0.8492 (5)	0.9818 (2)
16	Ecoli(pp)	0.7959 (2)	0.7170 (4)	0.7917 (3)	0.6122 (6)	0.6136 (5)	**0.8125 (1)**
17	Vertebral column	0.5357 (4)	0.5455 (3)	0.6071 (2)	0.4286 (6)	0.4462 (5)	**0.6939 (1)**
18	Shuttle	0.9997 (3)	0.9990 (5)	**1.0000 (1)**	0.9904 (6)	0.9993 (4)	**1.0000 (1)**
19	SatImage(1)	0.9558 (3)	**0.9666 (1)**	0.9661 (2)	0.8384 (6)	0.8768 (5)	0.9522 (4)
20	Transfusion	0.3758 (4)	**0.4276 (1)**	0.3836 (2)	0.3796 (3)	0.3380 (6)	0.3758 (4)
21	Adult	0.6251 (4)	0.6265 (3)	0.6229 (5)	0.5333 (6)	**0.6349 (1)**	0.6291 (2)
22	Parkinsons	0.6078 (5)	0.6545 (4)	**0.8049 (1)**	0.5000 (6)	0.7045 (3)	0.7955 (2)
23	Haberman	0.3523 (4)	0.3373 (5)	0.3590 (3)	0.2833 (6)	0.3600 (2)	**0.3671 (1)**
24	Wine	0.9200 (3)	0.9000 (4)	**0.9574 (1)**	0.8409 (5)	0.6923 (6)	0.9545 (2)
	Average rank	3.38	2.88	2.33	5.63	4.13	**1.92**
	Friedman test	**0.00065**	0.0881	0.3711	**0.000007**	**0.00039**	Base

Bold values represent the best performance measure comparing among all classifiers

range for any attribute. However, this situation appears after a finite number of splits occur. To demonstrate the effectiveness of ME for general data sets, data sets from the UCI repository were used and the results of these data sets were compared with the other techniques.

For the first experiment, the results in Table 4 show that ME yielded the lowest average ranking over C4.5, DCSM, DKM, HDDT and AE at 1.83. The Friedman test showed evidence that ME provided a statistical improvement over these five techniques for imbalanced data sets compared with the geometric mean at a 0.05 significance level.

For the second experiment, the results in Table 5 show that ME yielded the lowest average ranking over C4.5, DCSM, DKM, HDDT and AE at 1.88. The Friedman test also confirmed that ME provided a significant improvement over C4.5, DKM, HDDT and AE but not DCSM for imbalanced data sets compared with the *F*-measure at a 0.05 significance level.

For the third experiment, the results in Table 6 show that ME yielded the lowest average ranking over C4.5, DCSM, DKM, HDDT and AE at 1.92. The Friedman test showed a significant improvement over C4.5, DKM and HDDT for imbalanced data sets compared with precision at a 0.05 significance level. For the comparison between DCSM and AE, ME yielded better performance, which was demonstrated by the lower average ranking. However, it could not achieve a significant improvement by the precision at a 0.05 significance level.

For the fourth experiment, the results in Table 7 show that ME yielded the lowest average ranking over C4.5, DCSM, DKM, HDDT and AE at 1.71. The Friedman test showed that ME attained a significant improvement over C4.5, DKM, HDDT and DCSM but not AE for imbalanced data sets compared with the recall at a 0.05 significance level.

5.4 Discussion

Our experimental results confirm that C4.5 is not suitable for an imbalanced data set; however, the results can be improved using ME on data sets, especially glass, OpticDigits, breast tissue, vertebral column and Parkinsons. Although ME

Table 7 The comparison result by recall for imbalanced data sets

No.	Data sets	Recall					
		C4.5	DCSM	AE	DKM	HDDT	ME
1	Page blocks	0.6786 (3)	**0.8214 (1)**	0.5714 (4)	0.1786 (6)	0.5000 (5)	0.7143 (2)
2	Thyroid	**0.7647 (1)**	**0.7647 (1)**	**0.7647 (1)**	0.2941 (5)	0.1765 (6)	**0.7647 (1)**
3	Letter	0.9430 (3)	0.9202 (5)	0.9506 (2)	0.8061 (6)	0.9278 (4)	**0.9544 (1)**
4	Abalone	0.1905 (3)	0.2381 (2)	0.1667 (4)	0.1429 (5)	0.1190 (6)	**0.2857 (1)**
5	Glass	0.6923 (4)	0.9231 (1)	0.9231 (1)	0.6154 (5)	0.4615 (6)	**0.9231 (1)**
6	Cleveland	0.3846 (2)	0.1538 (5)	0.3077 (3)	0.0000 (6)	**0.4615 (1)**	0.3077 (3)
7	LED display domain)	**0.8108 (1)**	0.7838 (5)	**0.8108 (1)**	0.7838 (5)	**0.8108 (1)**	**0.8108 (1)**
8	Vowel	0.9333 (2)	0.8889 (4)	0.9000 (3)	0.3444 (6)	0.6111 (5)	**0.9556 (1)**
9	PenDigit	0.7910 (4)	**0.8567 (1)**	0.8269 (3)	0.6119 (6)	0.6448 (5)	**0.8567 (1)**
10	OpticDigits	0.9326 (5)	0.9607 (2)	0.9494 (3)	0.6685 (6)	0.9494 (3)	**0.9663 (1)**
11	Ecoli(imU)	0.4857 (3)	0.4571 (3)	**0.6286 (1)**	0.2571 (6)	0.4000 (5)	0.5714 (2)
12	SatImage(4)	0.5829 (2)	0.5687 (4)	**0.5877 (1)**	0.4645 (6)	0.5308 (5)	0.5782 (3)
13	Fertility	0.1667 (3)	0.0833 (3)	0.1667 (5)	0.0833 (6)	0.3333 (1)	**0.3333 (1)**
14	Breast tissue	**0.8571 (1)**	**0.8571 (1)**	**0.8571 (1)**	0.5714 (6)	0.7857 (5)	**0.8571 (1)**
15	Segmentation	0.9727 (4)	**0.9909 (1)**	0.9879 (2)	0.8424 (5)	0.8364 (6)	0.9818 (3)
16	Ecoli(pp)	0.7500 (2)	0.7308 (3)	0.7308 (3)	0.5769 (5)	0.5192 (6)	**0.7500 (1)**
17	Vertebral column	0.5000 (3)	0.5000 (3)	**0.5667 (1)**	0.5000 (3)	0.4833 (6)	**0.5667 (1)**
18	Shuttle	0.9997 (2)	0.9980 (5)	0.9997 (2)	0.9894 (6)	0.9977 (2)	**1.0000 (1)**
19	SatImage(1)	0.9371 (3)	0.9414 (2)	0.9284 (4)	0.9002 (6)	0.9111 (5)	**0.9501 (1)**
20	Transfusion	0.3315 (3)	**0.3652 (1)**	0.3427 (2)	0.2921 (6)	0.2697 (5)	0.3315 (3)
21	Adult	**0.6353 (1)**	0.6163 (4)	0.6168 (4)	0.5256 (6)	0.6336 (2)	0.6309 (3)
22	Parkinsons	0.6458 (3)	**0.7500 (1)**	0.6875 (5)	0.5625 (6)	0.6458 (3)	0.7292 (2)
23	Haberman	**0.3827 (1)**	0.3457 (3)	0.3457 (3)	0.2099 (5)	0.3333 (4)	0.3580 (2)
24	Wine	**0.9583 (1)**	0.9375 (2)	0.9375 (2)	0.7708 (5)	0.5625 (6)	0.8750 (4)
	Average rank	2.50	2.75	2.38	5.50	4.54	**1.71**
	Friedman test	**0.0389**	**0.0253**	0.0593	**0.000001**	**0.00012**	Base

Bold values represent the best performance measure comparing among all classifiers

outperforms C4.5 for most data sets in our experiments, it is difficult to conclude that ME provides advantages over C4.5 for data sets with a low number of minority class instances compared with data sets with a high number of minority class instances, as demonstrated in the experiment on simulated data sets. In our opinion, real-world data sets can have several groups of minority class instances. To identify the best split, C4.5 considers the impurity of instances after the split, while ME focuses on the impurity of a single group of minority class instances. That group of minority class instances is usually identified after multiple splits. Therefore, ME cannot outperform C4.5 for some data sets from the UCI repository because of impurity of minority instances within the minority range.

DCSM can improve performance and provides a compact tree compared with C4.5. It does not intentionally focus on handling an imbalanced data set. According to the experimental results, the comparison between ME and DCSM is not statistically significant compared with the F-measure and precision. However, ME provides lower average ranking than DCSM for the experiments overall and is statistically significant compared with the geometric mean and recall.

One weakness of AE is the setting of θ, which must be set to a suitable value of imbalance in a data set that is derived from the training process. Accordingly, AE can handle imbalanced data sets better than C4.5, DCSM, DKM and HDDT. Overall, ME yields better performance compared with AE for all measures. It can outperform AE statistically significantly compared with the geometric mean, F-measure and recall.

In conclusion, ME provides the most improved performance of the geometric mean, F-measure, precision and recall among the six techniques: C4.5, DCSM, AE, DKM, HDDT and ME.

6 Conclusion and future work

Evidence shows that the decision tree is one of the most popular classifiers [1, 2]. Its weakness is demonstrated when applying it to an imbalanced problem because

decision tree induction was designed based on a balanced data set. Consequently, our research develops a new measure called ME, which uses SE on instances from the minority range along a single attribute. Overall, ME yields better performance over C4.5, DCSM, DKM, HDDT and AE compared with the geometric mean and F-measure.

Although ME requires additional time to locate instances in the minority range, the overall time complexity of the algorithm is the same as C4.5. As with all methods on an algorithmic level, ME does not change the distribution of the data set. As a result, it does not have to process additional instances or build multiple classifiers, which consumes less time compared with over-sampling and ensemble techniques.

In comparison with an imbalanced algorithm, ME provides better F-measure performance than AE. Moreover, AE requires a parameter (θ). If a given parameter is not suitable, using AE can affect the performance of the decision tree significantly. As a parameter-free method, ME does not have this weakness.

For future work regarding the decision tree, the minority range can be applied directly with other impurity measures, such as Gini. ME can be extended to handle nominal and ordinal data types.

Acknowledgments We thank 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

1. KDnuggets (2011) Poll results: top algorithms for analytics/data mining (Nov 2011) which methods/algorithms did you use for data analysis in 2011? http://www.kdnuggets.com/2011/11/algorithms-for-analytics-data-mining.html. Accessed 1 Feb 2013
2. Wu X, Kumar V, Ross Quinlan J, Ghosh J, Yang Q, Motoda H, McLachlan G, Ng A, Liu B, Yu P, Zhou ZH, Steinbach M, Hand D, Steinberg D (2008) Top 10 algorithms in data mining. Knowl Inf Syst 14(1):1–37. doi:10.1007/s10115-007-0114-2
3. Quinlan JR (1993) C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco
4. Hunt EB, Marin J, Stone PJ (1966) Experiments in induction. Academic, New York
5. Quinlan J (1986) Induction of decision trees. Mach Learn 1:81–106
6. Breiman L, Friedman J, Olshen R, Stone C (1984) Classification and regression trees. Wadsworth and Brooks, Monterey
7. Dietterich T, Kearns M, Mansour Y (1996) Applying the weak learning framework to understand and improve c4.5. In: ICML, Citeseer, pp 96–104
8. Drummond C, Holte RC (2000) Exploiting the cost (in) sensitivity of decision tree splitting criteria. In: ICML, pp 239–246
9. Flach PA (2003) The geometry of roc space: understanding machine learning metrics through roc isometrics. In: ICML, pp 194–201
10. Marcellin S, Zighed DA, Ritschard G (2006) An asymmetric entropy measure for decision trees, pp 1292–1299. In: 11th conference on information processing and management of uncertainty in knowledge-based systems, IPMU 2006. http://archive-ouverte.unige.ch/unige:4531, iD: unige:4531
11. Zighed D, Ritschard G, Marcellin S (2010) Asymmetric and sample size sensitive entropy measures for supervised learning. In: Ras Z, Tsay LS (eds) Advances in intelligent information systems, studies in computational intelligence, vol 265. Springer, Berlin, pp 27–42
12. Cieslak D, Chawla N (2008) Learning decision trees for unbalanced data. In: Daelemans W, Goethals B, Morik K (eds) Machine learning and knowledge discovery in databases, vol 5211, Lecture notes in computer science. Springer, Berlin, pp 241–256
13. Chandra B, Kothari R, Paul P (2010) A new node splitting measure for decision tree construction. Pattern Recognit 43(8):2725–2731
14. Fan W, Miller M, Stolfo S, Lee W, Chan P (2004) Using artificial anomalies to detect unknown and known network intrusions. Knowl Inf Syst 6(5):507–527
15. Kubat M, Holte R, Matwin S (1998) Machine learning for the detection of oil spills in satellite radar images. Mach Learn 30(2–3):195–215
16. Liu W, Chawla S, Cieslak DA, Chawla NV (2010) A robust decision tree algorithm for imbalanced data sets. SDM, SIAM 10:766–777
17. Shannon C (1948) A mathematical theory of communication. Bell Syst Tech J 27:379
18. Ma BLWHY (1998) Integrating classification and association rule mining. In: Proceedings of the fourth international conference on knowledge discovery and data mining
19. Upton GJ (1992) Fisher's exact test. J R Stat Soc Ser A Stat Soc 155(3):395–402. dio:10.2307/2982890
20. He H, Bai Y, Garcia EA, Li S (2008) ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In: Neural networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference, pp 1322–1328
21. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) Smote: synthetic minority over-sampling technique. J Artif Int Res 16(1):321–357
22. Han H, Wang WY, Mao BH (2005) Borderline-smote: A new over-sampling method in imbalanced data sets learning. In: Huang DS, Zhang XP, Huang GB (eds) Advances in intelligent computing, vol 3644, Lecture notes in computer science. Springer, Berlin, pp 878–887
23. Bunkhumpornpat C, Sinapiromsaran K, Lursinsap C (2012) Dbsmote: density-based synthetic minority over-sampling technique. Appl Intell 36(3):664–684
24. Bunkhumpornpat C, Sinapiromsaran K, Lursinsap C (2009) Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In: Theeramunkong T, Kijsirikul B, Cercone N, Ho TB (eds) Advances in knowledge discovery and data mining, vol 5476, Lecture notes in computer science. Springer, Berlin, pp 475–482
25. Bunkhumpornpat C, Sinapiromsaran K, Lursinsap C (2011) Mute: majority under-sampling technique. In: 2011 8th International Conference on information, communications and signal processing (ICICS), pp 1–4. doi:10.1109/ICICS.2011.6173603
26. Gini CW (1971) Variability and mutability, contribution to the study of statistical distributions and relations, Studi Economico-Giuridici della R. Universita de Cagliari (1912). Reviewed in: Light, RJ Margolin BH: An analysis of variance for categorical data. J Amer Stat Assoc 66
27. Lindberg DV, Lee HK (2015) Optimization under constraints by applying an asymmetric entropy measure. J Comput Gr Stat 24(2):379–393. doi:10.1080/10618600.2014.901225

28. Su J, Zhang H (2006) A fast decision tree learning algorithm. In: Proceedings of the national conference on artificial intelligence. MIT Press, Cambridge, 1999, vol 21, p 500
29. Blake C, Merz CJ (1998) UCI repository of machine learning databases
30. Buckland MK, Gey FC (1994) The relationship between recall and precision. J Am Soc Info Sci 45(1):12–19
31. He H, Garcia E (2009) Learning from imbalanced data. Knowl Data Eng IEEE Trans 21(9):1263–1284
32. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30. http://dl.acm.org/citation.cfm?id=1248547.1248548