Applied Soft Computing 53 (2017) 27–33

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Randomly selected decision tree for test-cost sensitive learning

Chen Qiu[a], Liangxiao Jiang[a,b,*], Chaoqun Li[c]

[a] Department of Computer Science, China University of Geosciences, Wuhan 430074, China
[b] Hubei Key Laboratory of Intelligent Geo-Information Processing, China University of Geosciences, Wuhan 430074, China
[c] Department of Mathematics, China University of Geosciences, Wuhan 430074, China

ARTICLE INFO

Article history:
Received 14 January 2016
Received in revised form 6 October 2016
Accepted 21 December 2016
Available online 29 December 2016

Keywords:
Decision tree learning
Test-cost sensitive learning
Random selection
Test cost
Classification accuracy

ABSTRACT

In many real-world applications, decision trees that take account of the cost of acquiring attributes for decision making have been the research focuses. The decision-making process must learn which sequence to perform, and how to build an inexpensive and reliable inductive learning model to accomplish its task. Many previous works in the area of test-cost sensitive decision tree learning have successfully reduced the total test cost, unfortunately also degraded the classification accuracy simultaneously. This paper works on a new idea, i.e., it does not has to reduce the total test cost at the cost of the loss of classification accuracy. For that, we propose a multi-target adaptive attribute selection measure and a simple but effective method for building and testing decision trees. Instead of using a greedy attribute selection measure like many other decision tree learning algorithms, our algorithm uses a random attribute selection measure to find an appropriate attribute to test at each node in the tree. Specifically, we conduct a random search through the whole space of attributes in tree building, and we call the resulting model randomly selected decision tree (RSDT). By this way, RSDT significantly reduces the total test cost, yet at the same time maintains the higher classification accuracy compared to its competitors. The experimental results on 36 UCI datasets validate the effectiveness of our proposed RSDT.

1. Introduction

As a kind of inductive learning algorithm, decision tree algorithms have been successful to build classifiers with the aim to maximize the classification accuracy. The well-known ID3 [1], C4.5 [2], CART [3], Random Forests [4], and so on all center around inducing decision trees for the high classification accuracy.

However, one of the main difficulties of tree building in practice is that the majority of variables tests have associated cost, which may be diverse for each test [5,6]. Since data is not free, instead of only focusing on classification accuracy, a learner should perform an economic yet effective induction in practical application. That is to say, when building decision trees on a training data or performing a test on a new instance, if the tests incur the cost themselves, we should consider the total test cost and decide if it is worthwhile to pay the test cost.

Test-cost sensitive learning is more practical than simple traditional classification in many applications such as intelligent medical diagnostic systems [7]. As an example, in medical diagnosis, an expert needs to evaluate the tradeoff between the accuracy (the proportion of patients diagnosed correctly) and efficiency (the cost of measuring attribute values). Before diagnosing a patient, some tests for this patient, such as the diastolic blood pressure test or the serum insulin test, may not yet be known and generally take different cost. Like in the Pima Indians Diabetes dataset [8], a serum insulin test takes $22.78 for a patient while a diastolic blood pressure test only takes $1. These tests provide different informational values towards maximizing the classification accuracy, while performing them will incur extra cost. So, we have to pursue the balance between classifiers' reliability and low-cost testing.

To the best of our knowledge, some existing test-cost sensitive learning algorithms are about balancing the act of two types of cost, namely the misclassification cost and the test cost, to determine which test will be done [8–13]. The others focus on the balance between the classification accuracy and the minimal test cost directly [14–19]. Dealing with the high-cost test classification problems, decision trees are a kind of feasible candidate. When a test case is classified by a decision tree, some algorithms [20–24] have tried to find a tradeoff between the accuracy and the test cost. These algorithms are all the improved test-cost sensitive versions based on ID3 or C4.5 and they directly adapt existing information theoretic measures by including costs. Through the experiment and study, the results show that, compared with C4.5, all these algorithms reduce the test cost, unfortunately, yet at the same time degrade the classification accuracy.

[*] Corresponding author at: Department of Computer Science, China University of Geosciences, Wuhan 430074, China.
E-mail address: ljiang@cug.edu.cn (L. Jiang).

http://dx.doi.org/10.1016/j.asoc.2016.12.047
1568-4946/© 2016 Elsevier B.V. All rights reserved.

In this paper we focus on building decision trees which have not only the lower test cost but also the higher classification accuracy. Previous works [20–24] reduce the test cost while also degrade the classification accuracy. In the medical diagnosis and other fields, the higher classification accuracy is also one of the most important factors. This fact raises the question of whether we can build decision trees which reach the same classification accuracy as C4.5, meanwhile reduce the test cost significantly. To this end, instead of using the greedy attribute selection measures employed by previous works [20–24], the randomness is introduced to the tree building to select appropriate attributes. More specifically, we carry a random search through the whole useful candidate attributes. We call the resulting model Randomly Selected Decision tree (RSDT). When selecting the current attribute to build a tree, RSDT cannot only consider the total test cost, but also the classification accuracy. The experimental results on 36 UCI datasets [25] validate the effectiveness of our proposed RSDT.

The rest of this paper is organized as follows. Section 2 introduces some related works on attribute selection measures in decision tree learning and test-cost sensitive decision tree. Section 3 proposes our test-cost sensitive decision tree learning algorithm. Section 4 conducts a series of experiments on a large suite of benchmark datasets to validate our algorithm. Section 5 concludes the paper and outlines the main directions for future study.

2. Related work

2.1. Attribute selection measures in decision tree learning

A decision tree consists of a tree structural model and a set of decision nodes and leaves. The structural model is a directly decision-making process in which a leave specifies a class value and a decision node specifies a test over one of the attributes, called the attribute selected at the node. Attribute selection is quantified for the root node using a statistical measure given a set of examples. The examples are then filtered into subsets according to values of the selected attribute. The same process is applied recursively to each of the subsets until all nodes are leaves. Decision tree learning algorithms such as C4.5 [2] are often used for classification problems. On the base of the information gain ratio, a selection measure is utilized in C4.5, which can be defined as follows.

$$GainRatio(S,A) = \frac{Gain(S,A)}{SplitInfomation(S,A)} \tag{1}$$

where $Gain(S,A)$, called information gain, denotes the reduction of impurity from the parent node (before splitting) to the child nodes (after splitting). $Gain(S,A)$ is defined as

$$Gain(S,A) = Entropy(S) - \sum_{i=1}^{k} \frac{|S_i|}{|S|} Entropy(S_i) \tag{2}$$

where $Entropy(S)$, called entropy, describes the purity of the given instances set, k is the number of the split attribute values, S_i is the subset of instances at the ith child node of the parent node.

$SplitInfomation(S,A)$ is the split information of the selected split attribute, which is defined as follows.

$$SplitInfomation(S,A) = -\sum_{i=1}^{k} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|} \tag{3}$$

In addition to the information gain ratio measure, some other attribute selection measures in decision tree learning can be found from Jiang et al. [26] and Jiang [27].

Note that, such greedy attribute selection measures may have the potential to suffering from local optimum. Aiming at this problem, Breiman [4] provides a framework of random split selection for tree ensembles, which is well known as the "random forests". It is a classifier consisting of many decision trees. Its output class is the mode of the classes output by individual trees. The randomizing variable is the key factor in the algorithm, and it is typically used in the selection of the node and coordinates to split when a tree is built.

2.2. Test-cost sensitive decision trees

Traditional decision tree learning algorithms such as C4.5 aim to maximize the classification accuracy. However, in many real-world applications, the cost of acquiring attribute values is diverse and expensive [14,28,15–19], and thus it is more reasonable to induce decision trees that take account of test cost of attributes.

As shown in the previous subsection, top-down greedy algorithms for inducing decision trees use information theoretic measures, such as the information gain ratio measure, to select an appropriate attribute during the tree induction process. Naturally, many scholars adapt those measures by introduce the test cost of attributes. Extended algorithms that considering the test cost include EG2 [20], IDX [21], CS-ID3 [22], CSGR [23], CS-C4.5 [24] and so on. By introducing the test cost of attributes, these works mainly focus on minimizing the total test cost and adapting information theoretic measures towards attributes that cost less.

An advantage of the above adaptive decision tree learning algorithms is that it naturally extends the information theoretic measures by introducing the test cost. In Ling and Charles [9], the test-cost sensitive learning is converted as the theory of *Decision Trees with Minimal Cost* (DTMC). Instead of adapting the information gain to introduce the test cost, Ling and Charles [9] use the misclassification cost and the test cost directly as the cost reduction splitting criteria. Besides, Sheng et al. [29] present an approach where a decision tree is built for each new test case. For a given new case, depending on the expected cost calculated so far, the optimal policy suggests a best attribute to minimize the total costs. Their research adopts an optimal strategy, which may also have the potential to local optimum.

Another related work [16] is the filter attribute selection method that takes into account the test cost of features, which proposes a framework for test-cost sensitive feature selection (CS-CFS) based on CFS (Correlation-based Feature Selection). CS-CFS consists of adding a new term to the evaluation function of a filter feature selection method so that the test cost is taken into account. It is defined as

$$MC_s = \frac{k\bar{r}_{ci}}{\sqrt{k + k(k-1)\bar{r}_{ii}}} - \lambda \frac{\sum_{i=1}^{k} C_{test}(A_i)}{k} \tag{4}$$

where MC_s is the merit of the selected attribute subset S affected by the cost of the features, k is the size of attribute subset, $\bar{r_{ci}}$ is the average feature-class correlation, $\bar{r_{ii}}$ is the average feature-feature inter-correlation, $C_{test}(A_i)$ is the test cost of the feature A_i, and λ is a parameter introduced to weight the influence of the cost in the evaluation function.

Through experimentation with these algorithms, we have found that the classification accuracy of these decision tree algorithms may have not been recognized enough. To make up the disadvantage, it is the goal of this paper that building decision trees which keep high classification accuracy meanwhile reduce the total test cost significantly. Since those algorithms that adapt existing information theoretic measures by introducing the test cost [20–24] degrade the classifiers' accuracy, our algorithm is not going to adapt the existing information theoretic measures to introduce the test cost. At the same time, we also do not use the optimal strategies like Ling and Charles [9] and Sheng et al. [29] in order to avoid the potential to local optimum. In contrast, our work adopts the random selected strategy. An random factor is introduced to regulate the influence of our strategy and makes the built decision trees more biased in favor of the test cost or the classification accuracy.

3. Randomly selected decision tree

In this section, we propose a test-cost sensitive decision tree learning algorithm called randomly selected decision tree (RSDT) which aims to pursue both the higher classification accuracy and the lower test cost. For this purpose, a random strategy, instead of a greedy strategy, is used to find an appropriate attribute for each splitting. Specifically, given an attribute set *AS* containing *m* attributes and a random factor β, an adaptive attribute selection operator is performed. Within the rang of (0, β), the best attribute Att_{best}, which has the highest information gain ratio among *m* attributes, is selected. Because Att_{best} is the attribute with the highest information gain ratio, in this case, the process of tree-building pays more attention to the built decision tree's accuracy. Otherwise, RSDT selects an attribute, denoted as Att_{Proper}. The process of looking for Att_{Proper} is given later. Att_{Proper} is an attribute with the lowest test cost among all candidates, in this case, the process of tree-building pays more attention to the built decision tree's test cost while it still pursues a certain accuracy. Let Att_s be the selected split attribute at the current split node, which is defined as

$$Att_s = \begin{cases} Att_{best}, & \text{if } \ rand(0,1) < \beta \\ Att_{Proper}, & \text{otherwise} \end{cases} \tag{5}$$

Now, the only left thing is how to find Att_{Proper}. To an attribute set *AS* containing *m* attributes, we firstly perform a ranking operator, in terms of the attributes' information gain ratio, to rank all attributes in descending order. Based on the ordered attribute array *AS*, we can obtain an attribute subset $\bar{AS} = \{\bar{A}_1, \bar{A}_2, \ldots, \bar{A}_\eta\}$, which has only the top η attributes with the highest information gain ratio. Here η is defined as

$$\eta = \min\{1 + \log_2 m, g\} \tag{6}$$

where *g* is the number of the attributes whose information gain ratio is greater than 0.

Then, we define Att_{Proper} as the attribute with the lowest test cost among all these top η attributes. Namely,

$$Att_{Proper} = argmin_{\bar{A}_i \in \bar{AS}} C_{test}(\bar{A}_i) \tag{7}$$

where $C_{test}(\bar{A}_i)$ denotes the test cost of the attribute $\bar{A}_i$.

It can be seen that our algorithm exactly uses the random factor β and the parameter η to handle the tradeoff between the test cost and the classification accuracy. The role of the random factor β is to regulate the influence of strategy, and make trees more biased in favor of the test cost or the classification accuracy. When the parameter β varies, different weights are given to two strategies for choosing attributes. When β is small, the attribute Att_{best} is hardly selected and the decision tree's accuracy may be poor. With the increase of β, the decision tree will be more accurate. If β tends to 1, it will restrain the influence of the test cost in attribute selection. According to the size of η, we select different attributes to be Att_{Proper}. Att_{Proper} is an attribute which has the lowest test cost while its information gain ratio is not too low and even may be relatively high. Algorithm 1 outlines the training algorithm of RSDT.

Algorithm 1. RSDT-training (*TD*, *AS*, *TC*, β).

Input: *TD*-a training dataset; *AS*-an attribute set; *TC*-an array listing the test cost of each attribute; β-a random factor
Output: *DT*-the built test-cost sensitive decision tree

```
if the number of training instances is under 2 then
    Create a leaf node for the tree
else
    $m := sizeof(AS)$
    for $i = 1$ to $m$ do
        Calculate the ith attribute's GainRatio using Eq. (1)
    end for
    Sort AS in descending order of GainRatio
    if the maximum GainRatio is zero then
        Create a leaf node for the tree
    else
        if rand(0,1) < $\beta$ then
            Use the attribute $Att_{best}$ to split the tree
        else
            Calculate $\eta$ using Eq. (6)
            Obtain the attribute subset $\bar{AS}$
            Find $Att_{Proper}$ using $\bar{AS}$ and Eq. (7)
            Use the attribute $Att_{Proper}$ to split the tree
        end if
        Create a child node for each possible value of the split attribute
        For each child node, recursively call the algorithm
    end if
end if
Return a test-cost sensitive decision tree
```

Compared to the time complexity $O(nm^2)$ of the standard decision-tree learning algorithm C4.5 [2,30], Algorithm 1 needs some additional time to sort *m* attributes. The additional time complexity for sorting *m* attributes is only $O(mlog_2 m)$, where *n* is the number of training instances and *m* is the number of attributes. Therefore, we can conclude that Algorithm 1 almost maintains computational simplicity and efficiency that characterize standard decision-tree learning algorithm C4.5.

After the tree is built, the following discussion is how to deal with test instances in order to predict the class of the test instances with the minimal total test cost. In this paper, we consider the strategy to follow the tree built in the previous section. Yang [31] note that the decision trees had already specified an order in which to perform the tests. Aimed at reducing the total test cost without any loss of accuracy in the process of building a decision tree, it is reasonable to follow the test sequential. Algorithm 2 outlines the testing algorithm of RSDT.

Algorithm 2. RSDT-testing (*DT*, *TC*, *x*).

Input: *DT*-the built test-cost sensitive decision tree by Algorithm 1; *TC*-an array listing the test cost of each attribute; *x*-a test instance
Output: *c*-the predicted class; T_{test}-the total test cost for *x*

```
Sort x down the built tree DT from the root node to someone leaf node L
Estimate the class membership probabilities of x using the training instances dropping into the leaf node L and then predict its class label c
According to TC, calculate the total test cost $T_{test}$ of all split attributes in this path
Return c and $T_{test}$
```

Since RSDT is inherently unstable, we stabilize the estimated class membership probabilities by building an ensemble of RSDT using bagging and averaging the estimated class membership probabilities across the ensemble like Breiman [4], Jiang [32] and Jiang et al. [33]. The calculation of the average total test cost for ensemble trees is as follows: firstly, the average total test cost for a single tree is calculated by T_{test} of all testing instances. We will then calculate the average total test cost for ensemble decision trees. Bagging has two parameters: the number of bagging iterations and the percentage of the training data to use for learning a RSDT in each iteration. In our experiments, we use the parameter settings with 30 and 100, respectively. To our knowledge, Hall [34] stabilizes the estimated attribute weights by building multiple decision trees using bagging. Ahmad [35] creates ensembles of decision trees such that

Table 1
Classification accuracy (%) comparisons for C4.5 versus CSGR, CS-C4.5, CS-CFS, and RSDT.

Dataset	C4.5	CSGR	CS-C4.5	CSCFS	RSDT
anneal	98.54 ± 0.90	98.74 ± 1.03	98.56 ± 1.04	93.64 ± 2.17 ●	98.42 ± 1.23
anneal.ORIG	91.17 ± 2.51	82.30 ± 2.54 ●	90.70 ± 2.59	84.91 ± 3.11 ●	90.01 ± 2.56
audiology	78.55 ± 7.66	73.80 ± 7.45 ●	75.96 ± 7.97	76.91 ± 6.87	72.73 ± 7.04
autos	81.25 ± 8.56	79.06 ± 8.22	68.39 ± 11.18 ●	59.17 ± 9.30 ●	75.12 ± 8.10
balance-scale	64.14 ± 4.16	62.62 ± 4.69	67.93 ± 5.51	64.14 ± 4.16	74.04 ± 5.70 ○
breast-cancer	75.26 ± 5.04	67.82 ± 4.73 ●	69.91 ± 5.70 ●	67.16 ± 6.04 ●	70.48 ± 6.22 ●
breast-w	93.85 ± 3.49	93.28 ± 3.23	92.83 ± 3.27	93.89 ± 3.41	95.28 ± 2.65
colic	84.26 ± 5.99	84.77 ± 5.32	84.53 ± 5.71	83.82 ± 6.62	85.16 ± 5.12
colic.ORIG	81.30 ± 5.57	83.91 ± 5.40	82.31 ± 5.29	65.54 ± 2.81 ●	78.12 ± 5.48
credit-a	84.45 ± 3.99	85.51 ± 3.96	84.62 ± 4.08	85.29 ± 3.98	85.80 ± 3.95
credit-g	72.22 ± 3.47	70.00 ± 0.00	70.99 ± 3.45	71.39 ± 3.52	72.48 ± 3.25
diabetes	73.89 ± 4.71	65.26 ± 1.10 ●	74.44 ± 4.62	72.94 ± 4.64	73.41 ± 4.34
glass	58.20 ± 8.29	57.86 ± 8.18	59.93 ± 7.98	48.99 ± 8.13 ●	57.82 ± 9.11
heart-c	79.14 ± 6.33	76.50 ± 6.77	78.97 ± 6.78	78.55 ± 6.55	78.98 ± 7.00
heart-h	79.75 ± 7.24	80.91 ± 7.47	78.80 ± 7.18	80.70 ± 7.34	80.79 ± 7.01
heart-statlog	78.96 ± 7.23	75.30 ± 7.72	73.22 ± 7.44 ●	79.81 ± 7.76	77.52 ± 7.69
hepatitis	81.06 ± 8.48	83.05 ± 8.31	81.76 ± 7.42	84.16 ± 8.12	82.34 ± 7.59
hypothyroid	93.24 ± 0.44	93.31 ± 0.45	93.23 ± 0.44	93.28 ± 0.44	93.22 ± 0.44
ionosphere	87.98 ± 5.04	87.07 ± 4.52	87.18 ± 4.45	87.29 ± 4.64	91.00 ± 4.38 ○
iris	96.00 ± 4.64	86.13 ± 6.68 ●	97.20 ± 4.04	96.00 ± 4.64	96.60 ± 4.29
kr-vs-kp	99.44 ± 0.37	94.32 ± 1.33 ●	99.14 ± 0.58	94.14 ± 1.33 ●	98.72 ± 0.62 ●
labor	84.97 ± 14.24	81.60 ± 13.84	81.60 ± 13.84	84.97 ± 14.24	82.70 ± 13.75
letter	81.36 ± 0.77	54.59 ± 0.95 ●	75.96 ± 0.87 ●	81.75 ± 0.81 ○	80.79 ± 0.91
lymph	77.81 ± 9.33	75.50 ± 10.03	75.73 ± 10.09	76.61 ± 9.59	79.52 ± 9.41
mushroom	100.00 ± 0.00	99.56 ± 0.22 ●	100.00 ± 0.00	99.01 ± 0.34 ●	100.00 ± 0.00
primary-tumor	42.19 ± 6.93	38.35 ± 5.50	37.63 ± 6.75	39.36 ± 6.57	41.87 ± 6.35
segment	93.44 ± 1.69	83.98 ± 2.21 ●	93.78 ± 1.67	91.94 ± 1.61 ●	94.08 ± 1.47
sick	98.17 ± 0.67	97.67 ± 0.76 ●	97.74 ± 0.75 ●	96.98 ± 0.88 ●	97.97 ± 0.62
sonar	71.09 ± 8.30	62.81 ± 8.84 ●	64.41 ± 11.76	72.35 ± 9.40	73.92 ± 9.61
soybean	92.55 ± 2.75	88.97 ± 3.70 ●	82.46 ± 3.90 ●	92.17 ± 3.40	91.29 ± 2.97
splice	94.17 ± 1.28	91.14 ± 1.52 ●	91.78 ± 1.84 ●	94.37 ± 1.36	92.80 ± 1.65 ●
vehicle	68.28 ± 3.58	63.38 ± 3.62 ●	67.49 ± 3.99	61.15 ± 4.22 ●	70.26 ± 3.60
vote	96.27 ± 2.79	95.63 ± 2.76	95.54 ± 2.71	95.31 ± 2.77	95.72 ± 2.81
vowel	74.29 ± 4.02	58.28 ± 4.74 ●	64.04 ± 4.77 ●	70.38 ± 4.87 ●	78.01 ± 4.12 ○
waveform-5000	72.63 ± 1.81	58.52 ± 2.53 ●	67.89 ± 2.12 ●	73.36 ± 2.04	77.65 ± 2.14 ○
zoo	92.61 ± 7.33	89.82 ± 6.94	90.60 ± 7.66	91.45 ± 7.52	93.55 ± 7.09
Average	82.57	78.37	80.48	80.08	82.73
W/T/L	–	0/20/16	0/27/9	1/23/12	4/29/3

each member of an ensemble can use the expressive power of the kernel functions.

4. Experiments and results

The purpose of these experiments is to validate the effectiveness of the proposed RSDT in terms of the classification accuracy and the average total test cost. We run our experiments on 36 UCI datasets [36,37] published on the main web site of WEKA platform. In our experiments, we conducted the same preprocessing steps on these datasets as Jiang et al. [38]: missing values are replaced with the modes and means from the available data. Numeric attribute values are discretized using the unsupervised ten-bin discretization implemented in WEKA platform. Besides, we manually delete three useless attributes: the attribute "Hospital Number" in the data set "colic.ORIG", the attribute "instance name" in the data set "splice", and the attribute "animal" in the data set "zoo".

At first, we design a group of experiments to find the best value of the random factor β. We range the random factor β from 0 to 0.9 and investigate our RSDT algorithm using 10-fold cross-validation. Based on 36 UCI datasets, we measure the classification accuracy and the average total test cost, respectively. The detailed results are shown in Fig. 1. From Fig. 1, we can see that the performance of RSDT with different β values in terms of the classification accuracy and the average total test cost. With the increase of the β value, the classification accuracy of RSDT rises very relaxedly while the average total test cost rises sharply. When the β value reach to 0.2, the classification accuracy of RSDT is already higher than that of C4.5. In order to balance the classification accuracy and the average total test cost, we set β to 0.2 in our later comparison experiments.

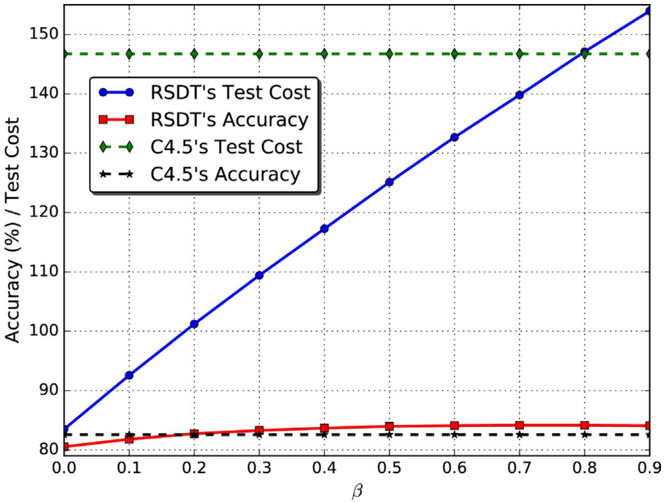

Fig. 1. The performance of RSDT with different β values in terms of the classification accuracy and the average total test cost on 36 UCI datasets.

Then, we compare our proposed RSDT with its state-of-the-art competitors including C4.5 [2], CSGR [23], CS-C4.5 [24], and CS-CFS [16]. Now, we introduce established algorithms and their abbreviations used in our experiments.

Table 2
Average total test cost comparisons for C4.5 versus CSGR, CS-C4.5, CS-CFS, and RSDT.

Dataset	C4.5	CSGR	CS-C4.5	CS-CFS	RSDT
anneal	354.40 ± 59.53	363.64 ± 90.53	433.16 ± 45.71 ○	217.99 ± 58.96 •	171.54 ± 14.34 •
anneal.ORIG	275.28 ± 16.51	141.04 ± 34.51 •	275.00 ± 19.70	108.00 ± 44.91 •	238.30 ± 11.00 •
audiology	430.79 ± 55.12	369.60 ± 45.39 •	287.98 ± 31.16 •	306.79 ± 67.75 •	292.55 ± 16.61 •
autos	152.70 ± 16.76	112.89 ± 6.49 •	24.75 ± 6.84 •	104.08 ± 40.37 •	62.18 ± 6.70 •
balance-scale	51.41 ± 9.72	25.66 ± 4.38 •	37.00 ± 3.54 •	96.26 ± 33.19 ○	47.76 ± 2.12
breast-cancer	54.76 ± 9.59	1.77 ± 6.21 •	11.25 ± 5.55 •	48.37 ± 34.35	19.54 ± 4.85 •
breast-w	53.67 ± 15.55	42.88 ± 0.87 •	16.93 ± 4.03 •	60.87 ± 34.42	41.34 ± 3.53 •
colic	53.26 ± 7.46	46.80 ± 2.44 •	47.10 ± 2.80 •	72.53 ± 32.73	52.47 ± 2.95
colic.ORIG	67.02 ± 9.97	56.46 ± 7.23 •	58.53 ± 9.82 •	4.81 ± 16.73 •	37.90 ± 4.13 •
credit-a	84.58 ± 21.85	21.90 ± 0.00 •	71.55 ± 24.96	65.72 ± 34.65	62.73 ± 4.98 •
credit-g	180.82 ± 19.60	0.00 ± 0.00 •	154.93 ± 16.19 •	85.11 ± 32.59 •	175.66 ± 10.02
diabetes	83.53 ± 8.30	4.96 ± 18.18 •	73.85 ± 2.83 •	56.54 ± 30.89 •	78.98 ± 4.54
glass	115.35 ± 38.57	141.61 ± 9.04	74.07 ± 10.56 •	91.57 ± 40.54	122.51 ± 11.48
heart-c	135.33 ± 20.07	36.76 ± 3.54 •	119.04 ± 13.95 •	99.23 ± 40.24 •	114.02 ± 6.67 •
heart-h	106.27 ± 33.34	58.86 ± 8.96 •	41.78 ± 18.41 •	78.66 ± 32.38	67.27 ± 8.42 •
heart-statlog	163.27 ± 16.30	61.32 ± 0.00 •	106.82 ± 34.98 •	112.66 ± 32.70 •	146.86 ± 7.54 •
hepatitis	53.83 ± 31.04	39.64 ± 8.46	48.23 ± 16.56	46.09 ± 28.32	42.04 ± 6.17
hypothyroid	12.13 ± 0.36	11.62 ± 0.00 •	12.09 ± 0.35	46.19 ± 27.96 ○	47.27 ± 8.55 ○
ionosphere	150.14 ± 13.61	105.36 ± 0.80 •	71.25 ± 19.02 •	90.12 ± 37.64 •	60.89 ± 7.74 •
iris	85.15 ± 5.33	43.53 ± 0.00 •	60.19 ± 8.58 •	53.59 ± 28.16 •	65.19 ± 6.04 •
kr-vs-kp	234.75 ± 8.41	168.08 ± 3.55 •	229.58 ± 6.52 •	154.65 ± 45.24 •	181.91 ± 5.89 •
labor	85.21 ± 17.26	69.96 ± 26.16	62.44 ± 21.63 •	90.95 ± 39.30	40.90 ± 5.81 •
letter	242.62 ± 1.90	80.34 ± 0.48 •	106.85 ± 1.63 •	255.91 ± 49.94	122.88 ± 1.74 •
lymph	103.76 ± 29.58	88.49 ± 16.07	40.18 ± 16.25 •	129.39 ± 50.67	47.16 ± 6.63 •
mushroom	87.68 ± 4.04	81.47 ± 0.75 •	78.12 ± 1.70 •	89.98 ± 35.69	72.06 ± 3.30 •
primary-tumor	464.49 ± 28.87	287.41 ± 28.09 •	348.99 ± 30.88 •	295.24 ± 66.46 •	343.10 ± 20.02 •
segment	93.38 ± 4.78	48.15 ± 0.45 •	44.24 ± 2.76 •	133.41 ± 36.79 ○	51.17 ± 2.44 •
sick	33.99 ± 1.59	27.22 ± 0.12 •	31.46 ± 1.30 •	49.70 ± 28.15	41.77 ± 3.11 ○
sonar	94.70 ± 10.77	23.37 ± 6.43 •	16.27 ± 5.54 •	48.46 ± 30.82 •	50.19 ± 4.56 •
soybean	377.63 ± 12.95	297.14 ± 11.99 •	167.15 ± 11.27 •	275.78 ± 62.83 •	260.82 ± 8.65 •
splice	99.84 ± 3.93	58.93 ± 10.08 •	38.97 ± 3.90 •	195.29 ± 48.58 ○	49.22 ± 2.83 •
vehicle	158.08 ± 20.62	53.73 ± 2.43 •	86.08 ± 6.77 •	107.77 ± 37.38 •	83.58 ± 4.58 •
vote	88.78 ± 8.22	48.66 ± 0.00 •	80.13 ± 15.98	61.82 ± 30.48 •	77.39 ± 5.10 •
vowel	190.55 ± 5.80	74.83 ± 14.62 •	57.56 ± 2.86 •	145.91 ± 42.41 •	96.46 ± 5.46 •
waveform-5000	152.18 ± 4.03	28.21 ± 6.19 •	40.50 ± 5.05 •	118.84 ± 36.50 •	81.09 ± 2.52 •
zoo	110.95 ± 8.01	98.73 ± 9.39 •	101.67 ± 13.03 •	124.13 ± 34.57	96.81 ± 7.99 •
Average	146.73	89.47	98.77	114.51	101.21
W/T/L	–	31/5/0	30/5/1	19/13/4	28/6/2

- C4.5: The standard decision-tree learning algorithm developed by Quinlan [2].
- CSGR: CSGR with C4.5 as the base classifier. The adjustable parameter γ is set to 0.1. Davis et al. [23] recommend the range of γ from 10^{-6} to 10^{+6}.
- CS-C4.5: CS-C4.5 with C4.5 as the base classifier. The factor of risk ϕ and the cost scale factor ω are set to 1 and 0.5, respectively. Freitas et al. [24] recommend the range of ω from 0.5 to 1.
- CS-CFS: CS-CFS with C4.5 as the base classifier. The parameter λ is set to 1. Bolón-Canedo et al. [16] recommend the range of λ from 0 to 10.
- RSDT: RSDT with C4.5 as the base classifier. The random factor β is set to 0.2.

The classification accuracy and average total test cost of each algorithm on each dataset are obtained via 10 runs of 10-fold cross-validation. Runs with the various algorithms are carried out on the same training sets and evaluated on the same test sets. In particular, the cross-validation folds are the same for all the experiments on each dataset. Tables 1 and 2 show the detailed comparison results in terms of the classification accuracy and the average total test cost. In Table 2, the test cost of each attribute is randomly created from an uniformly distribution between 0 and 100.

The symbols ○ and • in the tables respectively denote statistically significant upgradation or degradation over C4.5 with a corrected paired two-tailed t-tests at 95% significance level [39]. Besides, the **Averages** and the **W/T/L** values are summarized at the bottom of the tables. Each entry's **W/T/L** in the table means that, compared to C4.5, CSGR, CS-C4.5, CS-CFS and RSDT win on W datasets, tie on T datasets, and lose on L datasets. It is worth noting that the two-tailed t-tests in Table 2 are opposite to those in Table 1. With regard to the average total test cost, a large number is worse than a small number. Thus, for the average total test cost, ○ marks significantly worse than C4.5 and • marks better.

From these results, we can see that the existing algorithms reduce the total test cost at the cost of the loss of classification accuracy. RSDT significantly reduces the total test cost, yet at the same time maintains the higher classification accuracy that characterizes C4.5. Now, we summarize some highlights briefly as follows:

1 In terms of the classification accuracy, C4.5 is notably better than CSGR (zero win and 16 losses), CS-C4.5 (zero win and 9 losses) and CS-CFS (one win and 12 losses). Further, the average classification accuracy (82.57) of C4.5 is significantly higher than those of CSGR (78.37), CS-C4.5 (80.48) and CS-CFS (80.08).
2 In terms of the classification accuracy, RSDT almost ties C4.5 with 4 wins and 3 losses. Additionally, the average classification accuracy (82.73) of RSDT is even a littler higher than that of C4.5.
3 In terms of the average total test cost, C4.5 is markedly worse than CSGR (31 wins and zero loss), CS-C4.5 (30 wins and one loss) and CS-CFS (19 wins and 4 losses). In addition, the average total test cost (146.73) of C4.5 is substantially higher than those of CSGR (89.47), CS-C4.5 (98.77) and CS-CFS (114.51).
4 In terms of the average total test cost, RSDT significantly outperforms C4.5 with 28 wins and 2 losses. Further, the average total test cost (101.21) of RSDT is significantly lower than that of C4.5.

Table 3
Average rankings of the algorithms obtained by applying the Friedman test in terms of the classification accuracy.

Algorithm	Ranking
C4.5	2.2639
CSGR	3.8194
CS-C4.5	3.3472
CSCFS	3.3194
RSDT	2.25

Table 4
p-values for $\alpha = 0.05$ in terms of the classification accuracy.

i	Algorithms	$z = (R_0 - R_i)/SE$	p
10	CSGR vs. RSDT	4.211261	0.000025
9	C4.5 vs. CSGR	4.173994	0.00003
8	CS-C4.5 vs. RSDT	2.944156	0.003238
7	C4.5 vs. CS-C4.5	2.906888	0.00365
6	CSCFS vs. RSDT	2.869621	0.00411
5	C4.5 vs. CSCFS	2.832353	0.004621
4	CSGR vs. CSCFS	1.341641	0.179712
3	CSGR vs. CS-C4.5	1.267105	0.205118
2	CS-C4.5 vs. CSCFS	0.074536	0.940584
1	C4.5 vs. RSDT	0.037268	0.970271

Nemenyi's procedure rejects those hypotheses that have an unadjusted p-value ≤ 0.005.
Bergmann's procedure rejects these hypotheses:

- C4.5 vs. CSGR.
- C4.5 vs. CS-C4.5.
- C4.5 vs. CSCFS.
- CSGR vs. RSDT.
- CS-C4.5 vs. RSDT.
- CSCFS vs. RSDT.

Table 5
Average rankings of the algorithms obtained by applying the Friedman test in terms of the average total test cost.

Algorithm	Ranking
C4.5	4.5
CSGR	2.0556
CS-C4.5	2.3611
CSCFS	3.3889
RSDT	2.6944

In addition, we use the KEEL Data-Mining Software Tool [40] to conduct a Friedman test with the corresponding post-hoc tests [41], such as the Nemenyi and Bergmann tests, for the comparison of more algorithms over multiple data sets. Tables 3 and 5 show the average rankings of the algorithms obtained by applying the Friedman test [41], respectively. For the 5 algorithms and 36 data sets, F_F is distributed according to the F distribution with $5 - 1 = 4$ and $(5 - 1) \times (36 - 1) = 140$ degrees of freedom. F_F calculated from the mean ranks is 28.777778 and 54.644444, respectively, all of which values are greater than the critical values of $F(4, 140)$ for $a = 0.5$. Therefore, we reject the null hypothesis and proceed with the Nemenyi and Bergmann tests to determine accurately the significant difference between each pair of algorithms. Tables 4 and 6 respectively show the z-values and the p-values obtained, where the detailed results using the Nemenyi and Bergmann tests indicate which pairs of algorithms are significantly different. These non-parametric statistical test results show that:

1 In terms of the classification accuracy, the average rankings of them are C4.5 (2.2639), CSGR (3.8194), CS-C4.5 (3.3472), CSCFS (3.3194) and RSDT (2.25), respectively. RSDT is notably better than all the other exiting competitors: CSGR, CS-C4.5 and CSCFS. Further, there is no significant difference between C4.5 and RSDT.

Table 6
p-values for $\alpha = 0.05$ in terms of the average total test cost.

i	Algorithms	$z = (R_0 - R_i)/SE$	p
10	C4.5 vs. CSGR	6.559133	0
9	C4.5 vs. CS-C4.5	5.739241	0
8	C4.5 vs. RSDT	4.844814	0.000001
7	CSGR vs. CSCFS	3.577709	0.000347
6	C4.5 vs. CSCFS	2.981424	0.002869
5	CS-C4.5 vs. CSCFS	2.757817	0.005819
4	CSCFS vs. RSDT	1.86339	0.062407
3	CSGR vs. RSDT	1.714319	0.08647
2	CS-C4.5 vs. RSDT	0.894427	0.371093
1	CSGR vs. CS-C4.5	0.819892	0.412278

Nemenyi's procedure rejects those hypotheses that have an unadjusted p-value ≤ 0.005.
Bergmann's procedure rejects these hypotheses:

- C4.5 vs. CSGR.
- C4.5 vs. CS-C4.5.
- C4.5 vs. CSCFS.
- C4.5 vs. RSDT.
- CSGR vs. CSCFS.
- CS-C4.5 vs. CSCFS.

2 In terms of the average total test cost, the average rankings of them are C4.5 (4.5), CSGR (2.0556), CS-C4.5 (2.3611), CSCFS (3.3889) and RSDT (2.6944), respectively. RSDT significantly outperforms C4.5. In addition, there are no considerable differences between RSDT and all the other exiting competitors: CSGR, CS-C4.5 and CSCFS.

5. Conclusions and further study

A new test-cost sensitive decision tree learning algorithm RSDT is proposed in this paper, which aims to keep the high classification accuracy meanwhile reduce the total test cost. Compared to C4.5, existing test-cost sensitive decision tree learning algorithms by adapting information theoretic measures to introduce the test cost degrade the classification accuracy when they reduce the total test cost, while RSDT maintains the same classification accuracy as C4.5 and at the same time significantly reduces the total test cost. This paper provides a new idea for research, i.e., it does not has to reduce the test costs at the cost of the loss of classification accuracy. We can reduce the total test cost and maintain the same classification accuracy as C4.5 simultaneously. For this purpose, a random attribute selection measure is presented. Instead of the greedy strategy, the proposed random attribute selection measure employs a random strategy to guide the selection of the optimal attribute for splitting. A random factor is introduced to tree building to make trees more biased in favor of the test cost or the classification accuracy.

As already pointed out, there are two objectives in the task of test-cost sensitive classification; one is decreasing the test cost, the other is improving the classification accuracy. Our current version transforms the test-cost sensitive classification problem into a constrained single-objective optimization problem. We believe that the use of more sophisticated multi-objective optimization methods could further improve the performance of the current RSDT and make its advantage stronger. This is a main direction for our future study. Besides, for simplicity, we assume that all features have discrete values only in this paper and thus all continuous features are discretized using a preprocessing step. However, in many real-world applications, continuous features are widespread and, therefore, extending it to directly handle applications with continuous features is another direction for our future study.

Acknowledgements

The work was partially supported by the National Natural Science Foundation of China (61203287), the Program for New Century Excellent Talents in University (NCET-12-0953), and the Chenguang Program of Science and Technology of Wuhan (2015070404010202).

References

[1] J.R. Quinlan, J. Ross, Induction of decision trees, Mach. Learn. 1 (1) (1986) 81–106.
[2] J.R. Quinlan, C4.5: Programs for Machine Learning, San Mateo, Morgan Kaufmann, 1993.
[3] L. Breiman, J. Friedman, J. Stone, Classification and Regression Trees, CRC Press, 1984.
[4] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[5] C. Elkan, The foundations of cost-sensitive learning, in: International Joint Conference on Artificial Intelligence, Lawrence Erlbaum Associates, 2001, pp. 973–978.
[6] J.R. Quinlan, P.J. Ross, Inductive knowledge acquisition: a case study, in: Proceedings of the Second Australian Conference on Applications of Expert Systems, Addison-Wesley Longman Publishing Co., 1987, pp. 137–156.
[7] V. López, D. Rianõ, J.A. Bohada, Improving medical decision trees by combining relevant health-care criteria, Expert Syst. Appl. 39 (14) (2012) 11782–11791.
[8] P.D. Turney, D. Peter, Cost-sensitive classification: empirical evaluation of a hybrid genetic decision tree induction algorithm, J. Artif. Intell. Res. (1995) 369–409.
[9] C.X. Ling, X. Charles, Decision trees with minimal costs, in: Proceedings of the 21st International Conference on Machine learning, ACM, 2004.
[10] Z. Qin, S. Zhang, C. Zhang, Cost-sensitive decision trees with multiple cost scales., in: AI 2004: Advances in Artificial Intelligence, Springer Berlin Heidelberg, 2005, pp. 380–390.
[11] T. Wang, Handling over-fitting in test cost-sensitive decision tree learning by feature selection, smoothing and pruning, J. Syst. Softw. 83 (7) (2010) 1137–1147.
[12] F. Min, W. Zhu, A competition strategy to cost-sensitive decision trees., in: Rough Sets and Knowledge Technology, Springer Berlin Heidelberg, 2012, pp. 359–368.
[13] Y. Weiss, Y. Elovici, L. Rokach, The cash algorithm-cost-sensitive attribute selection using histograms, Inf. Sci. 222 (2013) 247–268.
[14] F. Min, H. He, Y. Qian, W. Zhu, Test-cost-sensitive attribute reduction, Inf. Sci. 181 (22) (2011) 4928–4942.
[15] F. Min, Q. Hu, W. Zhu, Feature selection with test cost constraint, Int. J. Approx. Reason. 55 (1) (2014) 167–179.
[16] V. Bolón-Canedo, B. Remeseiro, N. Sánchez-Maroño, A. Alonso-Betanzos, A framework for cost-based feature selection, Pattern Recognit. 47 (7) (2014) 2481–2489.
[17] V. Bolón-Canedo, B. Remeseiro, N. Sánchez-Maroño, A. Alonso-Betanzos, mC-ReliefF: an extension of ReliefF for cost-sensitive feature selection, in: Proceedings of International Conference of Agents and Artificial Intelligence, Angers, France, 2014, pp. 42–51.
[18] W. Qian, W. Shu, J. Yang, Y. Wang, Cost-sensitive feature selection on heterogeneous data, Adv. Knowl. Discov. Data Min. 9078 (2015) 397–408.
[19] G. Kong, L. Jiang, C. Li, Beyond accuracy: learning selective Bayesian classifiers with minimal test cost, Pattern Recognit. Lett. 80 (2016) 165–171.
[20] M. Núñez, Economic induction: a case study, in: Proceedings of the Third European Working Session on Learning, Pitman Publishing, Glasgow, 1988, pp. 139–145.
[21] S.W. Norton, Generating better decision trees, in: International Joint Conference on Artificial Intelligence, vol. 89, 1989, pp. 800–805.
[22] M. Tan, J.C. Schlimmer, Two Case Studies in Cost-Sensitive Concept Acquisition, 1990, pp. 854–860.
[23] J.V. Davis, J. Ha, C.J. Rossbach, H.E. Ramadan, E. Witchel, Cost-sensitive decision tree learning for forensic classification, in: Proceedings of the 17th European Conference on Machine Learning, Springer Berlin Heidelberg, 2006, pp. 622–629.
[24] A. Freitas, A. Costa-Pereira, P. Brazdil, Cost-sensitive decision trees applied to medical data, in: Data Warehousing and Knowledge Discovery, Springer Berlin Heidelberg, 2007, pp. 303–312.
[25] A. Frank, A. Asuncion, UCI Machine Learning Repository, University of California, School of Information and Computer Science, Irvine, CA, 2010.
[26] L. Jiang, C. Li, Z. Cai, Learning decision tree for ranking, Knowl. Inf. Syst. 20 (2009) 123–135.
[27] L. Jiang, Learning random forests for ranking, Front. Comput. Sci. China 5 (1) (2011) 79–86.
[28] X. Yang, Y. Qi, X. Song, J. Song, Test cost sensitive multigranulation rough set: model and minimal cost selection, Inf. Sci. 250 (2013) 184–199.
[29] S. Sheng, C.X. Ling, Q. Yang, Simple test strategies for cost-sensitive decision trees., in: Machine Learning: ECML 2005, Springer Berlin Heidelberg, 2005, pp. 365–376.
[30] J. Su, H. Zhang, A fast decision tree learning algorithm, in: Proceedings of the Twenty-First National Conference on Artificial Intelligence, AAAI Press, 2006, pp. 500–505.
[31] Q. Yang, Test-cost sensitive classification on data with missing values, Knowl. Data Eng. 18 (5) (2006) 626–638.
[32] L. Jiang, Random one-dependence estimators, Pattern Recognit. Lett. 32 (3) (2011) 532–539.
[33] L. Jiang, Z. Cai, H. Zhang, Not so greedy: randomly selected naive Bayes, Expert Syst. Appl. 39 (12) (2012) 11022–11028.
[34] M. Hall, A decision tree-based attribute weighting filter for naive Bayes, Knowl. Based Syst. 20 (2) (2007) 120–126.
[35] A. Ahmad, Decision tree ensembles based on kernel features, Appl. Intell. 41 (3) (2014) 855–869.
[36] I.H. Witten, E. Frank, M.A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, third ed., Morgan Kaufmann, 2011.
[37] L. Jiang, H. Zhang, Z. Cai, A novel Bayes model: hidden naive Bayes, IEEE Trans. Knowl. Data Eng. 21 (2009) 1361–1371.
[38] L. Jiang, Z. Cai, D. Wang, Improving tree augmented naive Bayes for class probability estimation, Knowl. Based Syst. 26 (2012) 239–245.
[39] C. Nadeau, Y. Bengio, Inference for the generalizaiton error, Mach. Learn. 52 (3) (2003) 239–281.
[40] J. Alcalá-Fdez, F. Sánchez, S. García, KEEL: a software tool to assess evolutionary algorithms for data mining problems, Soft Comput. 13 (3) (2009) 307–318.
[41] J. Demsar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.