# A Cost-sensitive Decision Tree under the Condition of Multiple Classes

Shaorong Feng
School of Information Science and Engineering
Xiamen University
Xiamen, China
shaorong@xmu.edu.cn

**Abstract—Cost-sensitive learning is one of the top ten problems in the field of data mining, its target is to produce the least cost in the classification process under the condition of achieve a given classification accuracy. The decision tree is a kind of cost-sensitive classification algorithm. There are many typically cost-sensitive decision tree algorithms based on greedy method to build a single model such as PM, MinCost, etc. This kind of algorithms has good comprehensibility, requires less time and space complexity compared to other cost-sensitive classification algorithm. However, , currently a lot of research works are limited to binary classification problem, very few people study the performance of classification cost and accuracy of this kind of algorithm under the condition of multiple classes. This paper puts forward a cost-sensitive decision tree based on score-evaluation under the condition of multiple classes (SECSDT_MC for short). Experiments show that SECSDT_MC compared with PM and MinCost can produce fewer classification costs or achieve higher classification accuracy in most cases under the condition of multiple classes.**

***Keywords-Cost-sensitive; Multiple Classes; Decision Tree; Classification; Accuracy***

## I. INTRODUCTION

As one of the 10 most challenging problems in the field of data mining, cost-sensitive learning attracts the attention of many researchers in recent years [1]. The traditional classification algorithms (such as decision trees, neural networks, and support vector machines [2]) are intended to make accurate sample classification. Decision tree classification algorithm, as one of the most classic algorithms, has been widely used since it has a higher classification accuracy, better intelligibility, less time and space complexity[3]. Because traditional classification algorithm does not consider the cost of misclassification for samples of different categories, and the costs to obtain the required test attribute value, so this type of algorithm does not apply to credit risk assessment [4], fraud detection [5], software security forecasting [6], and other fields.

For this reason, some researchers introduced the concept of cost-sensitive learning to the traditional classification algorithms, and produced many outstanding cost-sensitive classification algorithms, and cost-sensitive decision tree [7] is one of them. Cost-sensitive decision tree inherits the advantages of traditional decision tree algorithm, while the cost-sensitive decision tree also overcomes the traditional decision tree problem for ignoring various costs arising from the high cost of classification.

Misclassification cost and attribute detection cost are the most common factors relates to the cost in cost-sensitive detection [8, 9]. Misclassification cost refers to the cost to classify a sample of a classification erroneously as B; attribute detection means the cost to obtain the attribute value of the samples. Through establishing the decision tree model to classify the centralized and unknown sample, after paying the cost of certain attribute to detect and obtain the corresponding attribute value according to the decision path, the sample classification can be as accurate as possible. In the classification results, if the category is correct, it generally has no corresponding misclassification cost, if it is misclassification, you need to pay the appropriate misclassification costs, which depending on the detailed classification. Cost-sensitive decision tree is to make the sum of misclassification cost and attributes detection the least.

The current cost-sensitive decision tree algorithm can be divided into two categories, one is to establish a single cost-sensitive decision tree classification model based on greedy method, such as PM [9] and MinCost [10]; Another one is to take use of integrated learning method with Boosting, Bagging and other ways to build the ultimate cost-sensitive classification model compositing multiple tree models, such as MetaCost [11] and AdaBoost [12] and so on. The second algorithm can generate smaller total cost of classification or higher classification accuracy comparing to the first algorithm. However, the second method requires more time and space complexity, and when the data concentration sample number or the attribute dimension of the sample is large, the second algorithm will need to pay a very large time and space complexity. For this reason, many researchers focus on the first category algorithm and make the split attribute of inner nodes in line with the selections in building cost-sensitive decision tree model so that the model can obtain smaller total classification cost or better classification accuracy. For cost-sensitive learning problems, classification and classification accuracy, to some extent, shows the trade-off relationship, and it is difficult to obtain the results with smaller total classification cost and accuracy. It is the key to solving these problems about how to rank these two evaluation criteria in the same frame for balance. As two algorithms with better performance in first-class algorithms, PM and MinCost represent two important thoughts for building cost-sensitive decision tree, that is, to build a better cost-sensitive decision tree model, researchers need to fully consider the impact on the overall model of cost and classification accuracy factors [13].

The related misclassification costs of two types of cost-sensitive classification issues mainly include the cost of positive case misclassification being counter-examples (ie FP) and the cost of counterexample misclassification being positive case (ie FN), and usually FN>FP. Many classic cost-sensitive decision tree algorithms (such as PM and MinCost) show very good performance for binary classification, that is, meeting certain classification accuracy while generating smaller total cost of classification. However, few researchers have studied how these algorithms needle perform under multi-classification cost-sensitive problems. This article draws on the ways of PM and MinCost splitting attribute selection against multi-classification cost-sensitive problem and presents a score-policy-based sensitive decision tree (denoted as SECSDT_MC), and compare with the PM and MinCost algorithms. Through the experiment, researchers can see that SECSDT_MC can produce smaller total cost of classification or higher classification accuracy in most cases of experimental data sets.

## II. RELATED WORK

Lomax [7] had a profound summary for cost-sensitive decision tree in the paper and gave some classical cost-sensitive decision tree algorithm based on greedy method, such as EG2, CS-ID3, IDX, CS-C4.5, MinCost and PM, in which MinCost and PM are the most representative algorithms. The used formula in splitting attributes selection is as follows:

$$\text{MinCost: } ICF=DMC_A-C_A \tag{1}$$

$$\text{PM: } ICF=\frac{(2^{InfoGain_A}-1)*DMC_A*\omega}{C_A+1} \tag{2}$$

Wherein, *ICF* is short for Information Cost Function; $DMC_A$ represents the difference of the expected misclassification cost of current node and the expected misclassification cost sum of each sub-node after the based attribute $A$ of samples of the internal nodes of decision tree model being allocated to each sub-node (expected misclassification cost calculation will be described in the next section). $InfoGain_A$ represents the information gain of attribute $A$; $C_A$ represents the sum of all samples relating to the detection of attribute $A$ on the current nodes; $\omega$ is the given metric about the importance of attribute $A$ according to the experts' experience.

PM algorithms exist two disadvantages: (1) Due to the cost factors don't have the same value scale with information gain, so the equation (2) is likely to cause the calculation results depending on the quotients of $DMC_A$ and *ICF* ($C_A$ + 1); (2) the calculation on the classification accuracy in the formula uses information gain. The higher gain, the better classification accuracy; in order to obtain higher information gain, algorithm tends to choose the attribute with smaller value [14], however, the attribute with smaller value does not necessarily bring the misclassification cost smaller.

MinCost only conducts related calculation for cost factors in the heuristic function of selecting classification attributes and doesn't introduce information gain, the Gini coefficient, fuzzy rules, membership functions and other information theory methods to calculate the accuracy of classification. Te-Kang Jan [15] et al pointed out to build the cost-sensitive decision tree algorithm with better performance, that is, to obtain the least total classification cost under the conditions with certain accuracy, it will need to consider classification accuracy and various costs related to classification problems.

The research about uncertainties data is a hot topic in recent years in data mining field. Since the data sampling process has a number of objective factors, the data indicators of the obtained samples may have some deviation with the true value. Liu [16] et al had made some modification for cost-sensitive decision tree algorithm PM and made it suitable for solving the cost-sensitive classification problems under conditions of uncertainty data.

Wang [17] et al combined with the cost-sensitive classification gradient descent algorithm and online gradient technology, proposed two online cost-sensitive classification algorithms, and proved the effectiveness of the proposed method in various applications experimentally.

The research on cost-sensitive decision tree algorithm based on greedy methods mainly focus on binary classification, cost-sensitive classification of multi-classes (multiple classes) now is one hotspot of cost-sensitive classification field. Therefore, the paper compares the performance of two aspects of SECSDT_MC, PM, MinCost, which build single decision tree model algorithm based on greedy method under the conditions of multi-class classification, on the accuracy and classification total cost.

## III. SECSDT_MC ALGORITHM UNDER MULTI-CLASSES CONDITIONS

### *A. Problem description*

Assume data set S has n samples; m detection attributes and 1 classification attribute of each sample; classification attribute has $t$ kinds of attribute value totally (namely sample has $t$ kinds of classifications). Wherein, detection attribute is marked as $A_1$、$A_2$、...、$A_m$; classification attribute is marked as $A_C$; $t$ kinds of classifications are marked as $Class_1, Class_2, \cdots, Class_t$. respectively.

The definition of misclassification cost matrix is shown as Fig. 1.

$$\begin{bmatrix} C(1,1) & C(1,2) & ... & C(1,t) \\ C(2,1) & C(2,2) & ... & C(2,t) \\ ... & ... & ... & ... \\ C(t,1) & C(t,2) & ... & C(t,t) \end{bmatrix}_{t\times t}$$

Figure 1. Cost matrix

In Fig. 1, $C(i, j)$ refers to the true classification of the sample is Classj, which will need the paid misclassification cost($1 \le i,\ j \le t$). $C(i,\ i)$ is 0, namely there will be no misclassification cost under the conditions of correct classification.

The cost function of cost-sensitive decision tree can be represented with formula (3).

$$F(x,i)=\sum_{j=1}^{t}(p(j\mid x)\times C(i,j))+totalTestCost \tag{3}$$

Wherein, $F(x, i)$ means the total cost produced through cost-sensitive decision tree model to classify sample $x$ as $Class_i$ (the sum of misclassification and attribute detection cost); $p(j|x)$ refers to the probability of true classification $Class_j$; *totalTestCost* means the sum of detection cost for related attributes.

The target of cost-sensitive decision tree model is to classify the samples in set S, so the cost of each sample is the minimum for being classified as $Class_i$(ie, $F(x, i)$), and for all classification sample, it has better classification accuracy.

*B. Selection method of splicing attribute*

Our goal is to construct to meet the conditions of better classification accuracy producing a cost-sensitive decision tree model with the least total cost. Although these two indicators are not contradictory, but it does not mean better classification accuracy decision tree model can produce less cost of classification; vice versa. Therefore, the core of building cost-sensitive decision tree model is the selection of internal node splitting attribute. In the paper, Te-Kang Jan [15] pointed out that the cost-sensitive decision tree model with less total cost of classification and higher classification accuracy is a typical combinatorial optimization problem. Although nowadays there are many heuristic algorithms, such as genetic algorithms, simulated annealing algorithm, swarm intelligence algorithm, and these algorithms can solve these problems, but they require much time and space complexity, especially when the sample attribute dimension is large, these algorithms are not suitable. The author also noted that weighted sum is an effective way to solve these problems, and have good scalability. Specifically, if the cost-sensitive decision tree issues are related to other constraints, these factors can be added directly to algorithms, and these factors are assigned with the appropriate weights, researchers able to get satisfactory results at last.

The general idea of cost-sensitive decision tree under the conditions of multi-class based on score strategy method is to use information theory (such as information gain, the Gini coefficient, the membership function, etc.) as the heuristic function to assess classification accuracy factors in selecting splitting attributes on the internal nodes of the model, using misclassification costs and attributes detection costs as a heuristic function to assess the cost factors, and perform weighted sum for the results of these two heuristic functions. The highest final result of each candidate attribute will be the splitting attribute of this node, here researchers call this process as score for each candidate attribute vividly. The specific formula is as follows:

$$score(A_i) = \alpha \times (AvgInfoGain(A_i))_{normalized} + (1-\alpha) \times (CostRed(A_i))_{normalized} \quad (4)$$

Wherein, $score(A_i)$ indicates the scoring results of candidate attribute $A_i$; $AvgInfoGain(A_i)$ represents the index by using the average information gain for assessing the accuracy of classification, specifically defined as Equation (5) below; $CostRed(A_i)$ represents the reduction of classification cost, as specified in equation (6), here using $CostRed(A_i)$ as the evaluation indicator of cost factors. Since $AvgInfoGain(A_i)$ is numerically far smaller than $CostRed(A_i)$ of the value scale, in order to prevent as PM algorithm that a whole formula calculations are mainly affected by the price factor, here researchers need to normalize $AvgInfoGain(A_i)$ and $CostRed(A_i)$, as shown in (11) (12) equation in details. After getting the normalized $AvgInfoGain(A_i)$ and $CostRed(A_i)$, then make weighted sum of these two indicators, the final candidate attribute with the highest score will be taken as splitting attribute on internal nodes of cost-sensitive decision tree model.

For the calculation of average information gain(i.e. $AvgInfoGain(A_i)$), firstly, researchers need to calculate the information entropy about data set S on current nodes and the information entropy on sub nodes; and then calculate the information gain $Gain(A_i)$ about attribute Ai; then in order to prevent the algorithm intending to choose a candidate attribute values as attribute splitting attribute, where the average gain information $AvgInfoGain(A_i)$ instead of $Gain(A_i)$, the specific formula is as follows:

$$AvgInfoGain(A_i) = \frac{Gain(A_i)}{|A_i|} \quad (5)$$

In equation (5), if the is discrete attribute $A_i$, $|A_i|$ represents the number of $A_i$ attribute value; and if the attribute $A_i$ is the numerical continuous attribute, firstly, use the discrete method based on information entropy to make it discrete, at this time, $|A_i|$ denotes the number of discrete intervals of the discretization attribute.

For the calculation of classification cost reducing amount of $CostRed(A_i)$, as specified in equation (6) below:

$$Cost\,Red(A_i) = ExpMisCost\,Red(A_i) - TestCost(S) \quad (6)$$

Wherein, $TestCost(S)$ represents the sum of attribute detection cost of attribute $A_i$ for all the samples in the detection set $S$ of current nodes; $ExpMisCostRed(A_i)$ represents the expected misclassification cost reductions of current nodes, particularly as equation (7).

$$ExpMisCost\,Red(A_i) = ExpMisCost(S) - \sum_{j=1}^{|A_i|} ExpMisCost(S_j) \quad (7)$$

Wherein, $ExpMisCost(S)$ represents the expected misclassification cost of all samples on the current nodes. To calculate $ExpMisCost(S)$, you first need to calculate the misclassification cost (namely $CostOfClass(Class_i)$) that generates by classifying the sample as certain class ($Class_i$), as specified in equation (8); Secondly, researchers also need to calculate the probability (ie $ProOfClass(Class_i)$) classifying the sample of current nodes as certain class ($Class_i$), as specified in the formula(9).

$$CostOfClass(class_i) = \sum_{j=1}^{t} C_{ij} \times n_j \quad (8)$$

Wherein, $t$ refers to the number of sample types, $1 \le i, j \le t$, $n_j$ refers to the sample $Class_j$ quantity on current nodes.

If the produced cost of misclassification for marking all the samples on the nodes as $Class_i$ is larger, then the probability that the samples on the node is determined as $Class_i$ will be smaller, so the calculation of the probability to determine the sample on the node as *Classi*(*ProOfClass* ($Class_i$)) is shown in equation (9):

$$ProOfClass(Class_i) = 1 - \frac{CostOfClass(Class_i)}{totalClassCost} \quad (9)$$

Wherein, *totalClassCost* is the sum of each classification $CostOfClass(Class_i)$.

After calculating the $ProOfClass(Class_i)$ and $CostOfClass(Class_i)$ of $A_i$ on each classification, researchers can calculate the expected misclassification cost on current nodes $ExpMisCost(S)$, for details, see the formula (10):

$$ExpMisCost(S) = \sum_{i=1}^{t} ProOfClass(class_i) \times CostOfClass(class_i) \quad (10)$$

Calculate the $AvgInfoGain(A_i)$ and $CostRed(A_i)$ of each candidate attribute with the formula aforesaid, then conduct normalization processing. The detailed method is to get the maximum value of each attribute on $AvgInfoGain(A_i)$ and $CostRed(A_i)$, denoted as *MaxAvgInfoGain* and *MaxCostRed*. The results after normalization should be denoted as $AvgInfoGain(A_i)_{normal}$ and $CostRed(A_i)_{normal}$, see formula (11) and (12):

$$AvgInfoGain(A_i)_{normal} = {AvgInfoGain(A_i)} \Big/ {MaxAvgInfoGain} \quad (11)$$

$$Cost\,Red\,(A_i)_{normal} = {Cost\,Red\,(A_i)} \Big/ {MaxCost\,Red} \quad (12)$$

After normalization process, select the appropriate weights, calculate the score of each attribute $score(A_i)$ according to the formula (4), and then select the attribute with the highest score, if the attribute meet certain convergence condition (which will be set forth in Section 3.3) then denote the current node as a leaf node, otherwise marked as internal node, and select the attribute as the splitting attribute on the current node. On the choice of $\alpha$, according to different data sets, through many experiments, select the values that meet the required accuracy of classification and classification total cost as the best weight, but this will increase the time complexity of the calculation, which is not in line with the original intention to create sensitive cost model of single model decision tree, so researchers try to explore the relationship between the value of $\alpha$ and misclassification cost matrix, as well as the number of each sample on the current node. Assume the $CostOfClass(Class_i)$ of *maxMisCost* and *minMisCost* current nodes having the maximum and minimum value, and set $\alpha$ as the quotient of *minMisCost* and *maxMisCost*. Researchers used several tests and found that the value set in this way, in most cases, would make algorithm SECSDT_MC obtain the least total classification cost or the best classification accuracy.

### C. Convergence conditions of algorithm

SECSDT_MC is essentially a recursive plus depth-first approach, therefore, the algorithm must be specified appropriate convergence conditions, when a node on the decision tree model to meet a particular convergence condition, the node is marked as the leaf node, calculating the probability of each class of the nodes. Corresponding convergence conditions are as follows:

(1)The ratio of sample number of certain class on the nodes taking up the sample sum is larger or equal to certain threshold value (such as 90%), and the samples quantity is also larger or equal to the given threshold value. Under such conditions, according to the statistics knowledge, it’s more possible that samples reaching to this node belong to this classification. To ensure the accuracy of classification, the node is marked as leaf node, and the classification is also classified as the leaf node classification.

(2)The total number of samples on the node is less than a specified threshold. Then if researchers continue to build sub-nodes recursively, classification accuracy may be reduced, and increasing the cost of attributes detection. Therefore, it should be determined to be leaf nodes, and calculate the probability that .the node being determined as leaf node

(3)The path from the root node to the parent node has exhausted all test attributes, this time, the node should be labeled as leaf node, and calculate the probability of the node being marked as each classification according to the formula (9) in section *B*.

(4)If the selected splitting attribute cannot reduce the total cost of classification according to the way described in section *B*, then the node is marked as the leaf node, and calculates the probability of the node being marked as each classification according to the formula (9) in section *B*.

The aforesaid convergence condition is not absolute, and it can be customized according to the convergence condition in more details for the actual problem. For example, if the actual problem defines attribute detection cost, while the sum of detection cost of related attributes on certain node that starting from the root node is higher than the given limited cost, then the node should be marked as leaf node; If the total number of nodes on the sample is too small, then according to the aforesaid convergence

### D. Pseudo code of SECSDT_MC algorithm

SECSDT_MC is the same with PM, MinCost, and they all build decision tree model recursively. Firstly, build root node, and then select the appropriate splitting attribute on the current node based on the equation (4). If the current node does not meet the convergence conditions described in section 3.3, then allocate the samples on current nodes to the corresponding sub-nodes based on the break point of splitting attribute, and recursive call SECSDT_MC algorithm for each sub-node, then the current node is set to the leaf node, calculating the probability of being marked as each class. The current node recursion reaches to this end and return.

The algorithm pseudo-code of cost-sensitive decision tree SECSDT_MC under multi-class problem based on score strategy under description of multi-class issues, as follows:

Algorithm: SECSDT_MC(*SampleSet*, *CandidateAttrSet*, *AttrCost*, *CostMatrix*)

Input: (1) training samples set *SampleSet*; (2) candidate attribute set *CandidateAttrSet*; (3) the detection cost of each attribute *AttrCost*; (4) misclassification cost matrix *CostMatrix*;

Output: cost-sensitive decision tree with *root* as root node;

(1) create a *root*;
(2) if (*root* satisfies convergence condition 1)
(3) set *root* as a leaf node and mark its class;
(4) return *root*;
(5) endif
(6) if (*root* satisfies convergence condition 2 or 3)
(7) set *root* as a leaf node and calculate the probabilities of each class;
(8) return *root*;
(9) endif
(10) select *A* as the splitting attribute with maximum score according to formula (4);
(11) if (*root* satisfies convergence condition 4)
(12) set *root* as a leaf node and calculate the probabilities of each class;
(13) return *root*;
(14) endif
(15) create *k* branches with sub-sample-set $S_i$ ($i$=1, 2, ⋯, $k$) below *root*;
(16) for (each sample in *SampleSet*)

(17) send the sample to corresponding $S_i$ ($i$=1, 2, ⋯, $k$);
(18) endfor
(19) for ($S_i$ ($i$=1, 2, …, $k$) below *root*)
(20) call SECSDT_MC($S_i$, *CandidateAttrSet-A*, *AttrCost*, *CostMatrix*) to build a sub-tree below *root*;
(21) endfor
(22) return *root*;
(23) end

## IV. EXPERIMENT DESIGN AND RESULTS ANALYSIS

### *A. Experiment design*

The computer in the experiment has Inter(R) Core(TM) i3-2100 @3.10GHz processor, 4.0GB memory and Windows 7 OS. The programming language to realize algorithm is java 1.7, programming environment Eclipse Luna.

To verify the performance of proposed algorithm in the aspects of classification total cost and accuracy, the data set used in the experiment are from the UCI [18], as shown in Table 1 below:

TABLE I. BASIC INFORMATION OF DATA SET

| data set | number of experimental attributes | Sample number | Classification number | attribute type |
|---|---|---|---|---|
| Abalone | 7 | 4177 | 29 | Categorical, Real |
| Ecoli | 7 | 336 | 8 | Real |
| Glass Identification | 10 | 214 | 7 | Real |
| Letter Recognition | 16 | 20000 | 26 | Integer |
| Image Segmentation | 19 | 2310 | 7 | Real |
| Vehicle | 18 | 946 | 4 | Integer |
| Waveform | 21 | 5000 | 3 | Real |
| Yeast | 8 | 1484 | 10 | Real |
| Pendigits | 16 | 10922 | 10 | Integer |

For conditions that sample attribute value is lost in data set, then discard the sample, because the actual background of algorithm application is only to pay the cost of the corresponding attribute detection to get the corresponding attribute value. If part of the attribute is continuous attribute (ie Integer or Real type), firstly, use the proposed method of Usama et al [19] to discrete and record related splitting points.

Experiment comparing algorithms are SECSDT_MC, PM and MinCost, which are mainly used to compare its performance in the classification total cost and classification accuracy. Because the cost-sensitive learning under multi-class issues refers to the cost of each attribute detection and misclassification cost matrix, and therefore the cost of different attribute detection and misclassification cost combination could lead to the same algorithm sometimes produces less classification total cost, sometimes produce more classification total cost, or sometimes get a better classification accuracy, sometimes they get worse classification accuracy compared to other algorithms. Because the cost of multi-class cost-sensitive classification is more complex, in order to ensure the reliability of experimental results, each experiment will use half cross-validation [2] to get results, and then get the times in 1000 experiment that the three algorithms generating minimum classification total cost and the highest classification accuracy. In experiments, the detection cost of each attribute is selected randomly from [1,5], and the misclassification cost between each classification is selected from 50 numbers of [51,100].

### *B. Experimental results and analysis*

The paper achieves the times in 1000 experiments that SECSDT_MC, PM, MinCost (each experiment relating different attribute detection cost and misclassification cost matrix)generating the minimum misclassification cost and getting the highest classification accuracy. The detailed results are shown in Fig. 2 and Fig. 3.

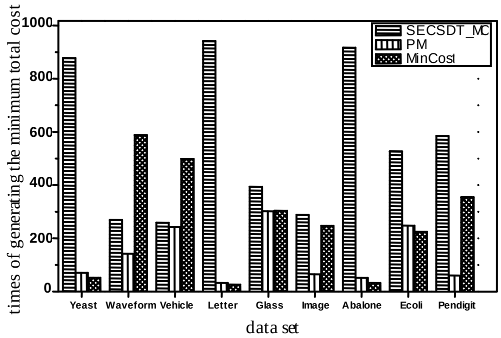


Figure 2. Times for generating minimum classification total cost

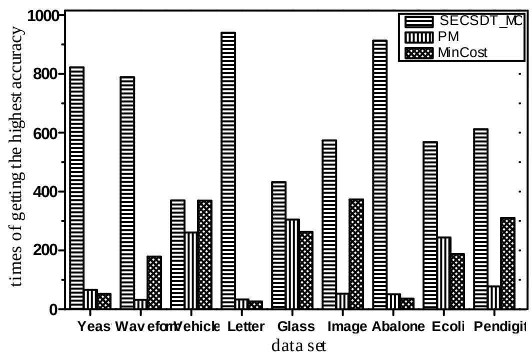


Figure 3. Times for getting the highest classification accuracy

From Fig. 2, researchers can easily find that SECSDT_MC produces less times of minimum classification total amount in data set Waveform and Vehicle than MinCost, but more than PM. For the other 7 data sets, SECSDT_MC produce more times of minimum classification total cost than PM and MinCost. For data set Yeast, Letter Recognition and Abalone, 85% can produce the least classification total cost in 1000 experiments.

From Fig. 3, researchers can easily find that only in data set Vehicle, the times that SECSDT_MC getting the maximum classification accuracy is nearly equal to MinCost, but more than PM algorithm. For the rest 8 data sets, SECSDT_MC get more times of maximum classification accuracy. For data set Yeast, Letter Recognition and Abalone, 85% can produce the highest classification accuracy in 1000 experiments.

Next, let's analyze 3 algorithms SECSDT_MC, PM, MinCost in 9 data sets, repeating 1000 experiments, the times to get the minimum classification total cost and the maximum accuracy. The results are shown as Fig. 4.

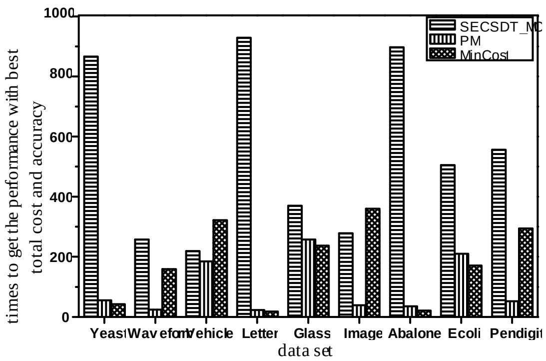


Figure 4. The times to get the performance with best total cost and accuracy

From Fig. 4, researchers can easily find that, apart from data set Vehicle,Image Segmentation, SECSDT_MC is a little less than PM in getting the minimum classification cost and the highest classification accuracy. For the other data sets, SECSDT_MC has better performance. Moreover, for the five data sets Yeast, Letter Recognition, Abalone, Ecoli, Pendigit, SECSDT_MC has more advantages.

In conclusion, compared with PM and MinCost, SECSDT_MC algorithm has better advantages in the aspects of classification total costs and classification accuracy. In the experiment, researchers also get the statistics of the times with the maximum classification accuracy and the minimum classification total cost. In 1000 experiments, researchers found that when SECSDT_MC produce the minimum classification total cost, it will almost get the maximum classification accuracy at the same time.

## V. CONCLUSIONS

This paper studies the characteristics of PM and MinCost cost-sensitive decision tree algorithm based on the greedy method. This kind of algorithm builds cost-sensitive decision tree model that has better intelligibility, higher classification accuracy and resulting in fewer classification total cost and take less time and space complexity. Multi-class cost-sensitive classification problem is the focus of cost-sensitive research study. Through analyzing the shortcomings of PM and MinCost, the paper presents a cost-sensitive decision tree algorithm SECSDT_MC based on score strategy aiming at multi-class cost-sensitive classification problem, and the experimental results show that the algorithm can produce minimum classification total cost or get the highest classification accuracy in most cases compared to the PM and MinCost.

SECSDT_MC belongs to cost-sensitive decision tree algorithm building a single model. In future research, researchers will try to apply SECSDT_MC in the complex algorithms that are build by Boosting, Bagging and other more complex methods (such as AdaBoost, MetaCost), and analyze whether the integrated approach will help to further reduce the total cost of classification or improve classification accuracy.

Before building cost-sensitive decision tree models, SECSDT_MC, PM, MinCost have gained the detection cost of each attribute and misclassification cost matrix. And if before or during model building, these costs factors are also factors in an unknown state, or in the building process, or only in the classification phase can get these cost factors. Obviously SECSDT_MC, PM, MinCost are not suitable for solving such problems. Therefore, in future work, researchers will try to raise the cost-sensitive classification algorithms for these issues.

## ACKNOWLEDGEMENTS

In this paper, researchers get the funding support from National Natural Science Foundation of China (No.61303004); National Social Science Foundation of major projects (No.13 & ZD148); Fujian Province Natural Science Foundation (No.2013J05099).

## REFERENCES

[1] Yang Q, Wu X. 10 challenging problems in data mining research[J]. International Journal of Information Technology & Decision Making, 2006, 5(04): 597-604.

[2] Jiawei Han, Micheline Kamber, Ed., Fan Ming, Meng Xiaofeng Translation. Data Ming Concepts and Techniques(Second Edition)[M]. Beijing: China Machine Press, 2008,1.

[3] Kotsiantis S B. Decision trees: a recent overview[J]. Artificial Intelligence Review, 2013, 39(4): 261-283.

[4] Chen N, Ribeiro B, Chen A. Comparative study of classifier ensembles for cost-sensitive credit risk assessment[J]. Intelligent Data Analysis, 2015, 19(1): 127-144.

[5] Yusuf Sahin, Serol Bulkan, Ekrem Duman. A cost-sensitive decision tree approach for fraud detection[J]. Expert Systems with Applications, 2013, 40(15): 5916–5923.

[6] Siers M J, Islam M Z. Cost Sensitive Decision Forest and Voting for Software Defect Prediction[M]//PRICAI 2014: Trends in Artificial Intelligence. Springer International Publishing, 2014: 929-936.

[7] Lomax S, Vadera S. A survey of cost-sensitive decision tree induction algorithms[J]. ACM Computing Surveys (CSUR), 2013, 45(2): 16.

[8] Turney P. Types of cost in inductive concept learning[J]. 2000.

[9] Ailing Ni, Xiaofeng Zhu, Chengqi Zhang. Any-cost discovery: learning optimal classification rules[C]. In: 18th Australian Joint Conference on Artificial Intelligence, Sydney, Australia: Springer, 2005. 123-132.

[10] Charles X. Ling, Qiang Yang, Jianning Wang, et al. Decision trees with minimal costs. In: Proceedings of the twenty-first international conference on Machine learning. New York, USA: ACM, 2004. 69.

[11] Jungeun Kim, Keunho Choi, Gunwoo Kim, Yongmoo Suh. Classification cost: An empirical comparison among traditional classifier, Cost-Sensitive Classifier, and MetaCost. Expert Syst. Appl. 39(4): 4013-4019 (2012)

[12] Paolo Favaro, Andrea Vedaldi. AdaBoost [M]. Computer Vision, {A} Reference Guide, 2014:16-19.

[13] Kim J, Choi K, Kim G, et al. Classification cost: An empirical comparison among traditional classifier, Cost-Sensitive Classifier, and MetaCost[J]. Expert Systems with Applications, 2012, 39(4): 4013-4019.

[14] Mitchell, T.M. Machine Learning[M]. McGraw Hill, 1997.

[15] Te-Kang Jan, Da-Wei Wang, Chi-Hung Lin, et al. A simple methodology for soft cost-sensitive classification[C]. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. New York, USA: ACM, 2012. 141-149.

[16] Liu M, Zhang Y, Zhang X, et al. Cost-sensitive decision tree for uncertain data[M]. Springer Berlin Heidelberg, 2011: 243-255.

[17] Wang J, Zhao P, Hoi S C H. Cost-sensitive online classification[J]. Knowledge and Data Engineering, IEEE Transactions on, 2014, 26(10): 2425-2438.

[18] http://archive.ics.uci.edu/ml/

[19] Usama M. Fayyad, Keki B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning[C]. In: Proceedings of the 13th International Joint Conference on Artificial Intelligence. 1993.1022–1027