Soft-CsGDT: Soft Cost-sensitive Gaussian Decision Tree for Cost-sensitive Classification of Data Streams

Ning Guo
gn@bupt.edu.cn

Yanhua Yu
yuyanhua@bupt.edu.cn

Meina Song
mnsong@bupt.edu.cn

Junde Song
jdsong@bupt.edu.cn

Yu Fu
ddskyfuyu@gmail.com

PCN&CAD Center, Beijing University of Posts and Telecommunications
Beijing 100876, China

ABSTRACT

Nowadays in many real-world scenarios, high speed data streams are usually with non-uniform misclassification costs and thus call for cost-sensitive classification algorithms of data streams. However, only little literature focuses on this issue. On the other hand, the existing algorithms for cost-sensitive classification can achieve excellent performance in the metric of total misclassification costs, but always lead to obvious reduction of accuracy, which restrains the practical application greatly. In this paper, we present an improved folk theorem. Based on the new theorem, the existing accuracy-based classification algorithm can be converted into soft cost-sensitive one immediately, which allows us to take both accuracy and cost into account. Following the idea of this theorem, the soft-CsGDT algorithm is proposed to process the data streams with non-uniform misclassification costs, which is an expansion of GDT. With both synthetic and real-world datasets, the experimental results show that compared with the cost-sensitive algorithm, the accuracy in our soft-CsGDT is significantly improved, while the total misclassification costs are approximately the same.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*data mining*; I.2.6 [**Artificial Intelligence**]: Learning—*concept learning*; I.5.2 [**Pattern Recognition**]: Design Methodology—*classifier design and evaluation*

Keywords

Data stream mining, cost-sensitive classification, decision tree, incremental learning

BigMine'13, August 11-14 2013, Chicago, IL, USA

ACM 978-1-4503-2324-6/13/08 ...$15.00.
http://dx.doi.org/10.1145/2501221.2501223

1. INTRODUCTION

In data mining applications, such as medical diagnosis, credit fraud detection, and intrusion detection, the data volume is always very large and continuously growing. These applications can be modeled as a typical classification problem of data streams. At present, in the field of data stream classification the major concern is accuracy-based learning [2, 3, 4, 15], which aims at minimizing error rate. However, the real-world data streams always come with non-uniform misclassification costs, therefore the existing data stream learning algorithms, which don't take misclassification cost into account, do not perform well.

In the scenario of credit fraud detection, the misclassification error, which is made in judging the bad user behaviors that cause huge economic losses as normal behaviors, is always much more serious (in other words, high cost) than the opposite type of error, which judges someone as bad when they are in fact with normal behaviors. The similar scenarios also exist in intrusion detection, medical diagnosis and real-time business decision-making. In this paper, we model these applications as a cost-sensitive classification problem of data streams.

Another important issue is that, to minimize total misclassification costs, existing cost-sensitive classification algorithms always result in high error rate [5]. The low accuracy of the algorithms restricts their practical application greatly. Here we concentrate on how to build a soft cost-sensitive classifier model for data streams, which takes both accuracy and cost into account.

In this paper, firstly, based on the folk theorem [14] and GDT [8], we present the Cost-sensitive Gaussian Decision Tree (CsGDT) algorithm to deal with data streams with non-uniform misclassification costs. Then, following the idea of the folk theorem, we propose an improved theorem, which is proved and analyzed in Section 3.2. The new theorem allows us to convert the existing accuracy-based algorithms to be soft cost-sensitive immediately. Furthermore, based on the new theorem and GDT and OcVFDT [6], soft-CsGDT algorithm is proposed for cost-sensitive classification of data streams, which can make a trade-off between accuracy and cost. At last, we evaluate the classification performance of CsGDT and soft-CsGDT on both synthetic and real-world datasets.

The rest of this paper is organized as follows. Section 2 gives a brief review of related work. In Section 3, the problem is formulated and the motivating theories are an-

alyzed. Then in Sections 4 and 5, algorithms CsGDT and soft-CsGDT are proposed respectively. The experimental results are shown in Section 6. Finally, Section 7 gives a conclusion to this work.

2. RELATED WORK

Our work focuses on the soft cost-sensitive classification of data streams, which is related to two groups of research: data stream classification and cost-sensitive classification. Both of them are the challenging problems in data mining research in recent years [13].

Data stream classification has been extensively studied in data mining research [3, 9, 4, 15]. There are mainly two kinds of solutions: single-model learning [3] and ensemble learning [2, 15]. The former keeps updating the single classification model with the new instance, while the latter uses parts of the data stream to construct a series of base models. Domingos and Hulten have proposed a single-model decision tree learning system , VFDT, which is one of the art-of-the-state methods [3]. In recent years, Rutkowski *et al*. have corrected a mathematical error of VFDT and proposed the GDT (Gaussian Decision Tree) [9, 8].

Cost-sensitive classification, which focuses on minimizing total cost, is practical in various domains, thus lots of attempts have been made to study this issue in the last decades [5, 14, 10, 7]. The existing algorithms can be grouped into two categories. The first is a direct method, in which the classifier is constructed all over again without any former classifier, such as the ICET [11] and some decision trees [7]. The other category is the meta-learning method, which improves the existing accuracy-based classifiers to be cost-sensitive, representative ones such as [14, 10]. The main idea of our work to construct cost-sensitive classifier of data streams belongs to the latter category.

Several recent works also address the cost-sensitive classification of data streams and soft cost-sensitive classification issues. Wang *et al*. [12] have proposed CSOGD-I and CSOGD-II algorithms to deal with the data streams with non-uniform misclassification costs, which use the online gradient descent approach. Our work here is different, because our algorithm is a decision tree with a strong interpretability. In [5], Jan *et al*. have introduced a simple method to improve cost-sensitive classification algorithms to be soft cost-sensitive immediately. Our work is distinct in the point that we focus on converting accuracy-based classification to be soft cost-sensitive.

3. PROBLEM AND THEORY

3.1 Problem Formulation

In this paper, for simplicity, we only consider steady binary data streams with discrete attribute values[1]. The cost-sensitive classification problem of data streams is generally formulated as follows. $S = \{s_1, s_2, \dots\}$ represents a data stream, the i-th training instance is denoted as $s_i = (\mathbf{x}, y, c)$ where $\mathbf{x}$ is the vector of attributes, y is class label, and c represents the misclassification cost of the instance.

$$\begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{bmatrix} \tag{1}$$

[1]Concept drift is not considered in this paper.

For a binary data stream, the cost matrix is listed as (1), where the row represents the actual class and the column represents the predicted class. Thus, c_{ij} means the cost of predicting an instance of i-th class as belonging to j-th class. The 0-th class and 1-th class misclassification additional cost are denoted as c_0 and c_1 respectively, the relation between c_i and c_{ij} is as (2). In the rest of this paper, for simplicity, we assume $c_{ii} = 0$ and denote $\mathbf{c}$ as cost array $[c_0, c_1]$.

$$\begin{cases} c_0 = c_{01} - c_{00} \\ c_1 = c_{10} - c_{11} \end{cases} \tag{2}$$

Assume that the instances are from a stationary distribution D. We denote D as $\mathcal{X} \times \mathcal{Y} \times \mathcal{C}$, where the $\mathcal{X}$ represents the domain of the input vector of the attributes, $\mathcal{Y}$ is the domain of the class label, which is $\{0, 1\}$, and $\mathcal{C}$ is the domain of the misclassification cost, therefore $\mathcal{C} \subset [0, +\infty)$. The accuracy-based classification algorithms aim at constructing a model $h : \mathcal{X} \to \mathcal{Y}$, which can minimize the expected Error Rate (ER):

$$ER_D(h) = \frac{1}{N} \sum_{(\mathbf{x},y,c)\sim D} I[y \neq h(\mathbf{x})]$$

The $N = \sum_{(\mathbf{x},y,c)\sim D} 1$ means the total number of the instances of distribution D. The $I[\cdot]$ is the indicator function, its value is 1 when the argument is true and 0 otherwise. The cost-sensitive classification focuses on minimizing the sum of the misclassification cost:

$$S_D(h) = \sum_{(\mathbf{x},y,c)\sim D} c \times I[y \neq h(\mathbf{x})]$$

Without loss of generality, the sum of the misclassification cost $S_D(h)$ can be replaced with the expected Cost Rate (CR):

$$CR_D(h) = \frac{1}{N_c} \sum_{(\mathbf{x},y,c)\sim D} c \times I[y \neq h(\mathbf{x})]$$

The $N_c = \sum_{(\mathbf{x},y,c)\sim D} c$ means the sum of the instances' cost.

Following the view of [5], the soft cost-sensitive algorithms aim at minimizing the weighted sum of CR and ER:

$$Sum_D(h) = \alpha \times CR_D(h) + (1-\alpha) \times ER_D(h), \quad \alpha \in [0, 1] \ .$$

Thus, only when the lowest value of $Sum_D(h)$ is acquired, can we construct a soft cost-sensitive classifier learning model, which takes both accuracy and cost into account.

3.2 Motivating Theory

In this section, firstly, we review the folk theorem [14], and furthermore a more general theorem is put forward on the basis of it.

A basic folk theorem states that if we build an accuracy-based classifier from $\widehat{D}$:

$$\widehat{D}(\mathbf{x}, y, c) \equiv c \times D(\mathbf{x}, y, c) \tag{3}$$

then the classification model is cost-sensitive for distribution D. From (3), we can know that an instance $(\mathbf{x}, y, c)$ from D is corresponding to a number of c identical instances from

$\widehat{D}$. Then the instance number $\widehat{N}$ of $\widehat{D}$ is:

$$\widehat{N} = \sum_{(\mathbf{x},y,c)\sim\widehat{D}} 1 = \sum_{(\mathbf{x},y,c)\sim D} c = N_c.$$

THEOREM 1. *(Translation Theorem) For all distributions, D, there exists a constant $N_c = \sum_{(\mathbf{x},y,c)\sim D} c$, then for any classifier h:*

$$CR_D(h) = ER_{\widehat{D}}(h)$$

PROOF.

$$\begin{aligned} CR_D(h) &= \frac{1}{N_c} \sum_{(\mathbf{x},y,c)\sim D} c \times I[y \neq h(\mathbf{x})] \\ &= \frac{1}{\widehat{N}} \sum_{(\mathbf{x},y,c)\sim D} c \times I[y \neq h(\mathbf{x})] \\ &= \frac{1}{\widehat{N}} \sum_{(\mathbf{x},y,c)\sim \widehat{D}} I[y \neq h(\mathbf{x})] \\ &= ER_{\widehat{D}}(h) \end{aligned}$$

□

From Theorem 1, we can get that, if we create a distribution $\widehat{D}$, which is constructed from the original distribution D according to (3), then the accuracy-based classifier learned from $\widehat{D}$, would be cost-sensitive for the instances from D.

In the remaining part of this section, we will discuss how to build a soft cost-sensitive classifier, which can get the similar CR with the cost-sensitive algorithm, and achieve the lower ER. Motivated by the idea of [5], we focus on minimizing the weighted sum of ER and CR.

THEOREM 2. *For all distributions, D, for any fixed variable $\alpha \in [0,1]$, we create a new distribution $\widetilde{D}$:*

$$\widetilde{D}(\mathbf{x},y,c) \equiv \left(\alpha \times c + (1-\alpha) \times \frac{N_c}{N}\right) D(\mathbf{x},y,c) \quad (4)$$

we note that:

$$\widetilde{N}_\alpha = \sum_{(\mathbf{x},y,c)\sim D} \left(\alpha \times c + (1-\alpha) \times \frac{N_c}{N}\right)$$

then for any classifier h:

$$Sum_D(h) = ER_{\widetilde{D}}(h) \quad (5)$$

PROOF.

$$\begin{aligned} Sum_D(h) &= \alpha \times CR_D + (1-\alpha) \times ER_D \\ &= \frac{\alpha}{N_c} \sum_{(\mathbf{x},y,c)\sim D} c \times I[y \neq h(\mathbf{x})] \\ &\quad + \frac{(1-\alpha)}{N} \sum_{(\mathbf{x},y,c)\sim D} I[y \neq h(\mathbf{x})] \\ &= \frac{1}{N_c} \sum_{(\mathbf{x},y,c)\sim D} \Big(\alpha \times c \\ &\quad + (1-\alpha) \times \frac{N_c}{N}\Big) \times I[y \neq h(\mathbf{x})] \\ &= \frac{\widetilde{N}_\alpha}{N_c} \times \frac{1}{\widetilde{N}_\alpha} \sum_{(\mathbf{x},y,c)\sim \widetilde{D}} I[y \neq h(\mathbf{x})] \\ &= \frac{\widetilde{N}_\alpha}{N_c} \times ER_{\widetilde{D}}(h) \\ &= ER_{\widetilde{D}}(h) \end{aligned}$$

where $\widetilde{N}_\alpha = \sum_{(\mathbf{x},y,c)\sim D} \left(\alpha \times c + (1-\alpha) \times \frac{N_c}{N}\right)$

$$\begin{aligned} &= \alpha \sum_{(\mathbf{x},y,c)\sim D} c + (1-\alpha) \times \frac{N_c}{N} \sum_{(\mathbf{x},y,c)\sim D} 1 \\ &= \alpha \times N_c + (1-\alpha) \times \frac{N_c}{N} \times N \\ &= N_c \end{aligned}$$

□

The left side of (5) is the weighted sum of CR and ER that we want to minimize, while the other side is the expected Error Rate (ER) that many existing algorithms focus on. In other words, Theorem 2 states that by creating a distribution $\widetilde{D}$, the accuracy-based classifier can be improved to soft cost-sensitive one immediately.

For a finite dataset, the N and N_c are both constant, thus the ratio $\frac{N_c}{N}$ is known. However, since the data stream is endless, as time progresses, both N and N_c tend to infinity. While for the steady data stream, the instances are drawn from a stationary distribution D, therefore, the true value of ratio $\frac{N_c}{N}$ is also a constant. As long as the ratio $\frac{N_c}{N}$ is known, Theorem 2 can be applied to the scenario of data streams. In fact, an individual process is added to estimate the ratio $\frac{N_c}{N}$ during handling the data stream, in order to make use of Theorem 2.

4. CSGDT

Theorem 1 shows that the accuracy-based classifier can be converted into a cost-sensitive one immediately, by constructing a distribution $\widehat{D}$. In this section, on the basis of Theorem 1 and GDT [8], CsGDT (Cost-sensitive Gaussian Decision Tree) is proposed to process a data stream with non-uniform misclassification costs.

4.1 Constructing Distribution $\widehat{D}$

According to Theorem 1, an instance $(\mathbf{x},y,c)$ from D corresponds to c instances of $\widehat{D}$, then we can add a weight $w(k)$ to each instance from distribution D whose class label is k, this method is equivalent to building a separate distribution $\widehat{D}$.

$$w(k) = c_k \quad (6)$$

c_k is the k-th class's misclassification cost. Since the distribution D is stable and misclassification cost c of each class is fixed, then the weight $w(k)$ would be fixed.

There are several details of how to use the instance's weight in the decision tree algorithm. Firstly, the instance's weight is used to alter the probability $p(k|q)$ to weighted probability $p_w(k|q)$ (7), which will be used in the calculation of information gain.

$$p_w(k|q) = \frac{w(k)\, n_q^k}{\sum_i w(i)\, n_q^i} \quad (7)$$

$p_w(k|q)$ means the k-th class's weighted probability at node q and n_q^i means the number of the instances whose class label is i at node q.

Secondly, after the decision tree model has been constructed, the k-th (8) class should be used to label the leaf q, this strategy is essential to achieve the same effect of building a separate distribution $\widehat{D}$.

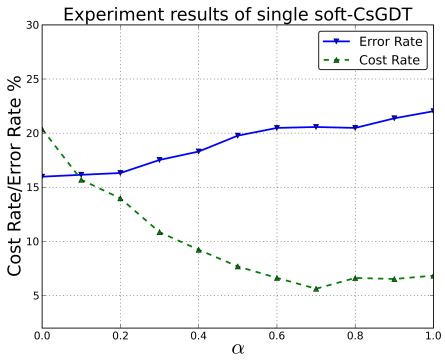

Figure 1: Experiment with α of single soft-CsGDT

$$k = arg\max_j\{w\,(j)\,n_q^j\}, \qquad j \in \{0,1\} \tag{8}$$

4.2 Building CsGDT

The Gaussian Decision Tree (GDT) is designed to classify data streams with high accuracy, which uses the Gaussian approximation bound to select the best attribute during the growth of the decision tree. In this section, based on the framework of GDT [8], CsGDT (Cost-sensitive Gaussian Decision Tree) is proposed by doing as follows: firstly, use weighted probability $p_w(k|q)$ (7) to calculate the information gain; secondly, label each leaf with k-th class (8).

5. SOFT-CSGDT

Based on Theorem 2, GDT and OcVFDT [6], we propose soft Cost-sensitive Gaussian Decision Tree (soft-CsGDT) algorithm to take both cost and accuracy into account.

5.1 Constructing Distribution $\widetilde{D}$

Based on the analysis in Section 4.1, to construct distribution $\widetilde{D}$, we just need to modify $w(k)$ as (9).

$$w\,(k) = \alpha \times c_k + (1-\alpha) \times \frac{N_c}{N} \tag{9}$$

c_k is the k-th class's misclassification cost.

Fig.1 demonstrates a property of soft cost-sensitive classification. The experiment is conducted on synthetic data with training data size $N = 2000$k, please refer to Section 6.1 for more details. Fig.1 shows that, with various values of α, the Cost Rate always changes, and the lowest Cost Rate sometimes does not occur at either $\alpha = 0$ nor $\alpha = 1$, while some intermediate value of α can result in better performance. That's because the cost-sensitive algorithm lead to overfitting when only costs are concerned. This property also has been observed in [5]. On the basis of this property, we build a forest of decision trees with various values of α, and select the most appropriate one as the final classifier.

5.2 Building Soft-CsGDT

As in the scenario of data stream, the ratio $\frac{N_c}{N}$ in (9) can't be known in advance, then a process is added before the core algorithm to estimate the ratio. The measured value of $\frac{N_c}{N}$ keeps being updated so as to approaching the real value. Finally, after the model has been finished, the leaf will be labeled with class k according to (8). As the best setting of α in (9) is unknown, we select ten possible values of α, from 0.1 to 1. Then, we obtain a forest, T, with ten different soft-CsGDTs.

The most appropriate tree in T is then chosen by evaluating the performance of each separate soft-CsGDT with a block of validating instances. The instances in the validating block are randomly selected from S with probability $p_{validate}$, and the size of the validating block is $n_{validate}$. As soon as the validating block is full, the instances inside it will be used to assess each tree in T, and then the validating block is cleared. Finally, the assessment data is used to select the most appropriate tree as the final classifier. The algorithm for constructing soft cost-sensitive gaussian decision tree is listed in Algorithm 1.

Algorithm 1 : soft-CsGDT algorithm

Input:
a stream of instances, S;
a set of discrete attributes, $\mathcal{A}$;
an array of misclassification cost $[c_0, c_1]$, $\mathbf{c}$;
the relative importance of CR, β;
the threshold of the proportion of minority class, th;
the number of instances used to estimate $\frac{N_c}{N}$, n_{ratio};
one minus the desired probability of choosing the correct attribute at any given node, δ;
user-specified tie threshold, τ;
the number of instances between check the leaf, n_{min};

Output: soft cost-sensitive gaussian decision tree.

1: $ValidatingBlock = \phi, \quad T = \phi$;
2: **for** each $i \in [1, 10]$ **do**
3: Initialize a tree T_i with only a leaf L_0 (the root);
4: $T_i.L_0.A = \mathcal{A}, \quad T = T \cup T_i$;
5: **end for**
6: Estimate $\frac{N_c}{N}$ using the first n_{ratio} instances of S;
7: **for** each latter instance $s \in S$ **do**
8: Update $\frac{N_c}{N}$;
9: **for** each $i \in [1, 10]$ **do**
10: Compute $\mathbf{w} = [w(0), w(1)]$ with $\alpha = \frac{i}{10}$, following Formula (9);
11: $Grow(T_i, \mathbf{w}, s, \delta, \tau, n_{min})$
12: **end for**
13: **if** $Random() \leq p_{validate}$ **then**
14: $ValidateBlock = ValidateBlock \cup s$;
15: **if** $|ValidateBlock| == n_{validate}$ **then**
16: $Assess(T, ValidateBlock)$;
17: $ValidateBlock = \phi$;
18: **end if**
19: **end if**
20: **end for**
21: **if** $ValidateBlock \neq \phi$ **then**
22: $Assess(T, ValidateBlock)$;
23: **end if**
24: **return** $GetBestTree(T)$;

In Algorithm 1, Step 1 to 6 are an initialization process; a forest of 10 trees can be obtained from Step 7 to 12; then in Step 13 to 23, the trees are assessed with the validating block. The procedure $Grow(T_i, \mathbf{w}, s, \delta, \tau, n_{min})$ can construct a single soft cost-sensitive decision tree, where $\mathbf{w} = [w(0), w(1)]$ is the array of class weights, more details will be introduced in Section 5.3. The function $Random()$ generates a random number in $[0, 1]$. The $Assess(T, ValidateBlock)$

is used to assess the classification performance of trees in T. Finally, the procedure $GetBestTree(T)$ returns the most appropriate tree chosen from forest T, more details are illustrated in Section 5.4.

5.3 Growth of Single Soft-CsGDT

Based on GDT [8], the construction procedure of single soft-CsGDT is illustrated in Algorithm 2.

Algorithm 2 can be divided into two parts. In the first part, from Step 1 to 2, the instance s passes throw soft-CsGDT tree from the root to a leaf (denoted as L_q) and then the class's label of node L_q is updated (Step 2). In the second part, Step 3 to 20 are the growth process of the tree. For each available attribute at L_q, information gain is calculated with weighted probability $p_w(k|q)$. Once the condition of Step 13 is satisfied, leaf node L_q will be replaced with an internal node by splitting on the best attribute A_a.

Algorithm 2 : $Grow(T_i, \mathbf{w}, s, \delta, \tau, n_{min})$

Input: Refer to Algorithm 1 for the details of input parameters;

1: $L_q = T_i.sort(s)$;
2: Label node L_q with class k, following Formula (8);
3: $N_{class} = numOfClassesAtNode(L_q)$;
4: $N_{instances} = numOfInstancesAtNode(L_q)$;
5: $P_{minClass} = proportionOfMinorityClassAtNode(L_q)$;
6: **if** $N_{class} > 1$ and $N_{instances}\%n_{min} == 0$ and $P_{minClass} > th$ **then**
7: **for** each $A_i \in L_q.A$ **do**
8: Compute $G(A_i)$ using weighted probability following Formula (7);
9: **end for**
10: Choose attribute A_a and A_b with the highest and second-highest G;
11: $\triangle G = G(A_a) - G(A_b)$;
12: Compute ϵ following the Formula in [8];
13: **if** $\triangle G > \epsilon$ or $\triangle G \leq \epsilon < \tau$ **then**
14: Split on attribute A_a;
15: **for** each branch of the split **do**
16: $addChildLeaf()$;
17: Set the attribute set of the ChildLeaf as $L_q.A \backslash \{A_a\}$;
18: **end for**
19: **end if**
20: **end if**

5.4 Tree Selection

Ten single trees are trained in Algorithm 1, then aiming at the goal to balance cost and accuracy, we will select the most appropriate tree as the final classifier model.

By using the validating block to assess each tree $T_i \in T$, the following statistics are collected:

- n_k, the count of the instances whose actual class label is k in the validating block.
- m_k^i, the count of the instances misclassified by tree T_i, whose actual class label is k in the validating block.

As soon as the validating block is full, the procedure $Assess(T, ValidateBlock)$ will be called to collect the statistics listed above. The statistical data of each run of the same tree T_i should be accumulated together.

The following formula is used to evaluate the performance of each tree $T_i \in T$:

$$\begin{aligned} wSum(T_i) &= \beta \times CR_i + (1-\beta) \times ER_i \\ where \quad CR_i &= \frac{\sum_k c_k m_k^i}{\sum_k c_k n_k}, \quad ER_i = \frac{\sum_k m_k^i}{\sum_k n_k} \end{aligned}$$

β is a user-specified parameter, which represents the relative importance of CR.

The most appropriate tree is selected by:

$$T_j = arg\min_i(wSum(T_i)), \quad 1 \leq i \leq 10$$

Let $|\mathcal{A}|$ be the number of attributes, v be the maximum number of values for an attribute, $|C|$ be the number of classes, d be the depth of the decision tree, N be the number of instances of data stream S, then the time complexity of Algorithm 1 is $O(N|\mathcal{A}||C|dv) + O(N|\mathcal{A}||C|dvp_{validate}) = O(N)$.

6. EXPERIMENTS

In this section, we evaluate the classification performance of CsGDT and soft-CsGDT, and the experiments are conducted on both synthetic data and real-world data. Furthermore, as CsGDT is a cost-sensitive classification of data streams, thus the comparison of CsGDT and soft-CsGDT can be used to show the strong ability of soft-CsGDT to balance cost and accuracy.

All of the experiments are implemented on the platform of MOA [1], and the environment is a PC with Windows 7 OS, Core 2 Duo CPU and 4G memory.

6.1 Synthetic Data

In this part, each algorithm runs twelve times on synthetic dataset, and the average statistics are used as the results. After each run, a size of 50k instances are used for testing. Please refer to Section 6.1.1 for more details about the synthetic dataset.

The default parameter values of each algorithm are set as Table 1. Those settings remain unchanged unless there is another statement.

Table 1: Experiment Parameters

PARAMETER	VALUE
c	$[0.1, 0.9]$
β	0.65
th	0.05
n_{ratio}	2000
δ	10^{-7}
τ	0.05
n_{min}	200
$n_{validate}$	200
$p_{validate}$	0.1

6.1.1 Generating Synthetic Data Stream

The synthetic data is similar with [8], which is generated by a synthetic random tree. At the first d_{min} levels of the tree, all the nodes are split to spread. At each following level, p percent of nodes will be replaced by leaves, while the others will go on splitting with a random assigned attribute, which hasn't ever been used in the path from the root to

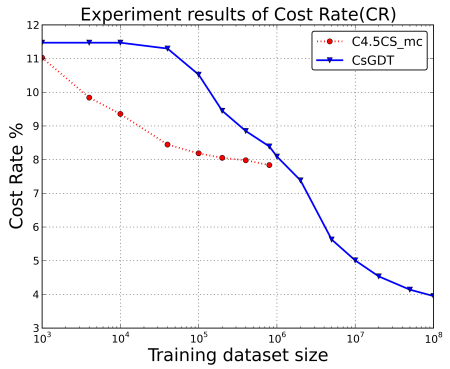

Figure 2: Comparison of CsGDT and C4.5CS_mc

this node. There is also a parameter d_{max}, which limits the depth of the synthetic tree. After the tree is built up, each leaf is labeled with a class randomly. In the experiments, we create twelve different synthetic random trees, all of which have the same basic setting with 100 binary attributes, two-class label, $p = 0.15$, $d_{min} = 3$ and $d_{max} = 18$.

6.1.2 *CsGDT vs. C4.5CS_mc*

In order to evaluate the classification performance of CsGDT, we carry out the first experiment to compare CsGDT with C4.5CS_mc [10]. The parameters of C4.5CS_mc are set as [10]. In Fig.2, the horizontal axis represents the training data size N, the vertical axis represents Cost Rate (CR). For the limitation of random access memory, the maximal data size for C4.5CS_mc is $N = 800$k. From Fig.2, we get to know that, when $N \leq 800$k, with the same size of dataset, CR of C4.5CS_mc is lower than CsGDT. Most importantly, when $N > 800$k, as the growth of data size, CsGDT can achieve a significant improvement in CR. Thus, CsGDT is an efficient algorithm to build the cost-sensitive classifier model of data streams.

6.1.3 *Analysis of Parameters in soft-CsGDT*

Another experiment is conducted, to study the influence of β. For soft-CsGDT, we set $N = 4000$k, $\mathbf{c} = [0.1, 0.9]$, both GDT and CsGDT are selected as benchmark algorithms. The results are shown in Fig.3.

From Fig.3, we can know that as the growth of β, ER increases gradually, while CR decreases rapidly when $\beta \leq 0.6$. In fact, the parameter β represents the relative importance of Cost Rate (CR), user can set it according to the specific environment. In the following experiments, we focus on reducing ER, while guaranteeing the lowest CR, therefore we set $\beta = 0.65$.

6.1.4 *CsGDT vs. soft-CsGDT*

In this group of experiments, we compare the performance of CsGDT and soft-CsGDT, GDT is used as the benchmark algorithm. For soft-CsGDT, we set $n_{ratio} = 2000$ to estimate $\frac{N_c}{N}$.

First, we conduct the experiment with different training data size. The results are shown in Fig.4. In Fig.4(A) and Fig.4(B), the horizontal axis represents dataset size N, the vertical axis represents ER and CR, respectively. From Fig.4 all of the following can be observed. First, with different size of training dataset, GDT can get lower error rate (ER) than CsGDT, but leads to unacceptable high cost rate (CR). That's because GDT is an accuracy-based algorithm which focuses on minimizing the ER. Second, CsGDT can achieve obvious low CR, but results in high ER, which is due to that CsGDT is just a cost-sensitive algorithm which aims at minimizing the CR. Last, soft-CsGDT can achieve the similar CR with CsGDT, while a significant reduction in ER compared with CsGDT.

Next, we examine the performance of both CsGDT and soft-CsGDT with a variety of misclassification costs, and the GDT is used as benchmark algorithm. A new parameter $c_{ratio} = \frac{c_0}{c_1}$ is used to represent the cost's difference, where $c_k, k \in \{0, 1\}$ is the misclassification cost of the k-th class. Since the distribution of the synthetic data streams are even, then we only conduct the experiments with $c_{ratio} \geq 1$. The experiments are carried out with training data size $N = 4000$k. Fig.5 shows that, as c_{ratio} increases from 1 to 35, CR of CsGDT greatly decreases, with a significant growth of ER. While ER of soft-CsGDT changes smoothly, which is significantly lower than CsGDT, and CR is about the same. That is, with a wide range of c_{ratio}, soft-CsGDT can achieve better performance than CsGDT. The explanation for this result is that soft-CsGDT can take both CR and ER into consideration, while CsGDT only focuses on CR.

6.1.5 *Running Time and Size of the Tree*

In this part, we give a view of the final decision trees of the three algorithms. The experiment is conducted with various values of training data size N, and the other input parameters are set as Table 1. The results are illustrated in Table 2, the first column represents training data size N, the remaining columns are the running time, the total number of nodes and the number of leaves of the final decision tree. Note that, the running time here only contains the time of building the classifier model. The time of both generating synthetic data streams and I/O are not included.

6.2 Real-world Data

In the following experiment, we choose the real-world KDD CUP'99 dataset to investigate the performance of both CsGDT and soft-CsGDT.

6.2.1 *Dataset and Preprocessing*

The KDD CUP'99 dataset contains 4898431 instances. A typical instance is composed of 7 nominal attributes and 34 numerical attributes. Since the algorithm GDT can only deal with the nominal attribute, then we should add a preprocessing step to discrete the numerical attributes. we adopt the similar processing with [8], each numerical attribute's range is divided into 15 chunks, the value of these attributes are replaced with the index number of the chunk. As GDT can only deal with two-class data streams, we also need to aggregate the four types of networks attacks ('dos','u2l','r2l','probe') as one 'bad' class, the 'normal' class remains the same. After preprocessing, the proportion of '*bad*' class is 80.14%.

We select a number of 50k instances for testing randomly, and the other parts used to train the classifier model. The parameters of the algorithms are set as follows: $th = 0.005, \delta = 10^{-5}, \beta = 0.65$. In particular, to evaluate the general classification performance of CsGDT and soft-CsGDT, we conduct a group of experiments with various values of $\mathbf{c}$, the other parameters are set as Table 1.

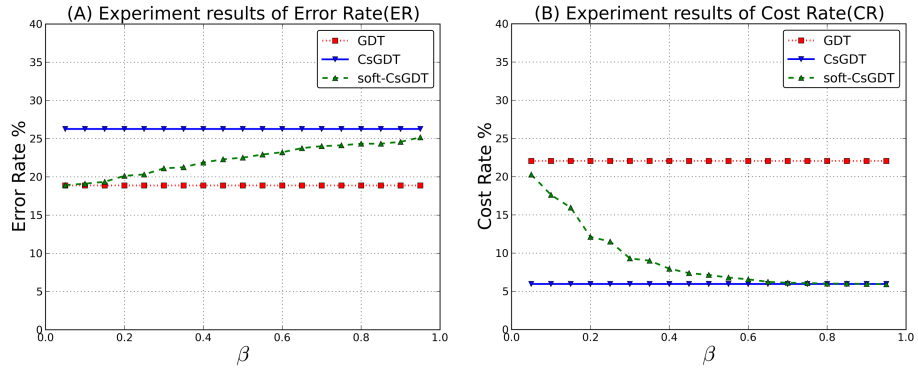

Figure 3: Experiment with β of soft-CsGDT

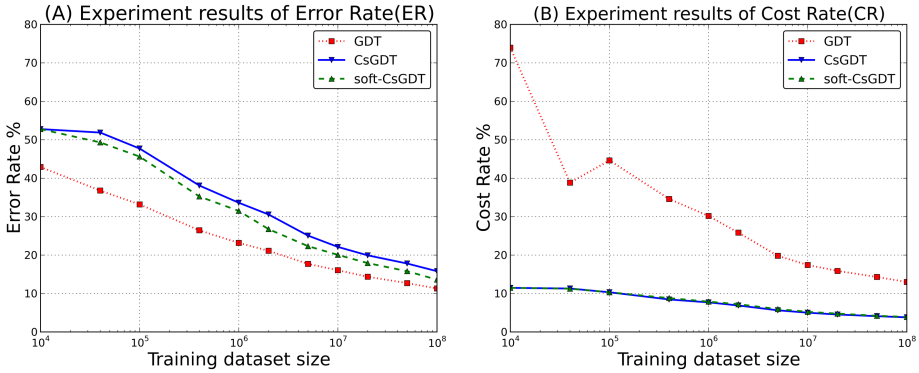

Figure 4: Comparison of CsGDT and soft-CsGDT

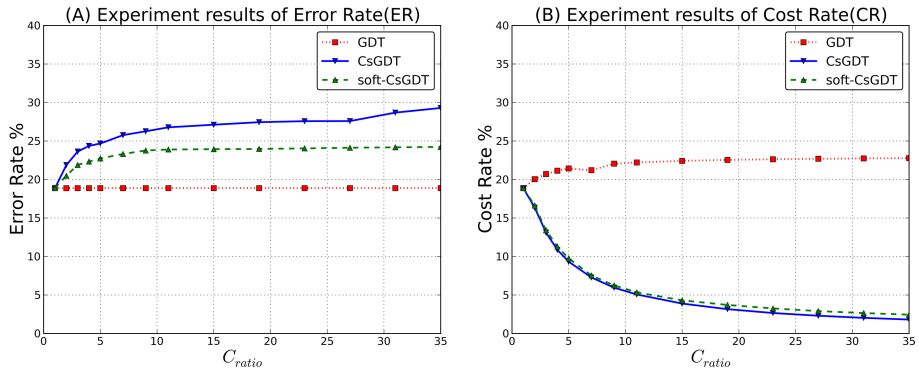

Figure 5: Experiment with C_{ratio} of CsGDT and soft-CsGDT

6.2.2 *Experiment Result*

With each value of cost array **c**, we run each algorithm twelve times with different order of sequence S, and the averaged experiment results are displayed in Table 3.

In Table 3, the first two columns give the values of cost array **c**, $'norm'$ represents the $'normal'$ class and $'bad'$ represents the other four classes. The other columns summarize the experiment results: the middle three columns are ER, and the last three are CR.

It is obvious from Table 3, despite of the uneven distribution of the data, CsGDT always achieves a low CR, while soft-CsGDT acquires a similar CR (always a bit lower CR), and a more lower ER than CsGDT. In addition, the performance of soft-CsGDT remains stable with various values of cost array **c**.

7. CONCLUSION

In this paper, we focus on the cost-sensitive classification of data streams. Following the idea of the folk theorem, we upgrade it and propose a new theorem, which allows us to improve accuracy-based classifier to be soft cost-sensitive immediately by constructing a new distribution from the o-

Table 2: Running Time and Size of the Tree

N	GDT			CsGDT			soft-CsGDT		
	Time(ms)	#Nodes	#Leaves	Time(ms)	#Nodes	#Leaves	Time(ms)	#Nodes	#Leaves
100k	776.40	9.7	5.3	792.12	8.0	4.5	8183.60	10.2	5.6
1000k	7844.40	78.8	39.9	7941.43	66.8	33.9	86496.59	72.3	36.7
10000k	93034.60	644.3	322.7	93125.91	551.5	276.2	960368.52	610.5	305.8
100000k	1010805.44	4427.2	2214.1	1102372.31	3675.5	1838.2	12615947.93	4253.2	2127.1

Table 3: Experiment on KDDCUP99

c		Error Rate(%)			Cost Rate(%)		
norm	bad	GDT	CsGDT	soft-CsGDT	GDT	CsGDT	soft-CsGDT
0.1	0.9	$\mathbf{0.40 \pm 0.08}^*$	0.70 ± 0.04	$\mathbf{0.43 \pm 0.20}$	0.35 ± 0.11	$\mathbf{0.18 \pm 0.01}^*$	$\mathbf{0.20 \pm 0.03}$
0.3	0.7	$\mathbf{0.40 \pm 0.08}$	0.63 ± 0.15	$\mathbf{0.33 \pm 0.09}^*$	$\mathbf{0.37 \pm 0.09}$	$\mathbf{0.39 \pm 0.05}$	$\mathbf{0.31 \pm 0.11}^*$
0.7	0.3	$\mathbf{0.40 \pm 0.08}$	$\mathbf{0.34 \pm 0.09}$	$\mathbf{0.31 \pm 0.08}^*$	0.45 ± 0.10	$\mathbf{0.32 \pm 0.10}$	$\mathbf{0.31 \pm 0.09}^*$
0.9	0.1	$\mathbf{0.40 \pm 0.08}^*$	0.56 ± 0.10	$\mathbf{0.42 \pm 0.18}$	0.55 ± 0.19	$\mathbf{0.23 \pm 0.03}$	$\mathbf{0.19 \pm 0.06}^*$

(those with the lowest mean are marked with *; those within one standard error of the lowest one are in bold)

riginal distribution. Based on the new theorem, GDT and OcVFDT, we propose soft-CsGDT algorithm to deal with data streams with non-uniform misclassification costs, which can make a trade-off between accuracy and cost. The experimental results on both synthetic and real-world datasets, show that soft-CsGDT can guarantee similar Cost Rate with cost-sensitive classification, at the same time, it achieves a significant improvement in Error Rate.

8. ACKNOWLEDGMENTS

This work is supported by the National Key project of Scientific and Technical Supporting Programs of China (Grant No. 2012BAH01F02,2013BAH10F01,2013BAH07F02); the National Natural Science Foundation of China (Grant No.61 072060); the National High Technology Research and Development Program of China Grant No. 2011AA100706); the Research Fund for the Doctoral Program of Higher Education (Grant No. 20110005120007); the Fundamental Research Funds for the Central Universities; Engineering Research Center of Information Networks, Ministry of Education, China.

9. REFERENCES

[1] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer. Moa: Massive online analysis. *The Journal of Machine Learning Research*, 99:1601–1604, 2010.

[2] A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavaldà. New ensemble methods for evolving data streams. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 139–148, 2009.

[3] P. Domingos and G. Hulten. Mining high-speed data streams. In *Proceedings of the 6th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 71–80, 2000.

[4] J. Gama. *Knowledge discovery from data streams*. Citeseer, 2010.

[5] T.-K. Jan, D.-W. Wang, C.-H. Lin, and H.-T. Lin. A simple methodology for soft cost-sensitive classification. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 141–149, 2012.

[6] C. Li, Y. Zhang, and X. Li. Ocvfdt: one-class very fast decision tree for one-class classification of data streams. In *Proceedings of the Third International Workshop on Knowledge Discovery from Sensor Data*, pages 79–86, 2009.

[7] C. X. Ling, Q. Yang, J. Wang, and S. Zhang. Decision trees with minimal costs. In *Proceedings of the 21th international conference on Machine learning*, page 69, 2004.

[8] L. Rutkowski, M. Jaworski, L. Pietruczuk, and P. Duda. Decision trees for mining data streams based on the gaussian approximation. *IEEE Transactions on Knowledge and Data Engineering*, PrePrints, 2013.

[9] L. Rutkowski, L. Pietruczuk, P. Duda, and M. Jaworski. Decision trees for mining data streams based on the mcdiarmid's bound. *IEEE Transactions on Knowledge and Data Engineering*, PrePrints, 2012.

[10] K. M. Ting. An instance-weighting method to induce cost-sensitive trees. *IEEE Transactions on Knowledge and Data Engineering*, 14(3):659–665, 2002.

[11] P. Turney. Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. *Journal of Artificial Intelligence Research (JAIR)*, 2, 1995.

[12] J. Wang, P. Zhao, and S. C. Hoi. Cost-sensitive online classification. In *2012 IEEE 12th International Conference on Data Mining (ICDM)*, pages 1140–1145, 2012.

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

[14] B. Zadrozny, J. Langford, and N. Abe. Cost-sensitive learning by cost-proportionate example weighting. In *Third IEEE International Conference on Data Mining(ICDM)*, pages 435–442, 2003.

[15] P. Zhang, J. Li, P. Wang, B. J. Gao, X. Zhu, and L. Guo. Enabling fast prediction for ensemble models on data streams. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 177–185, 2011.