On Multi-Class Cost-Sensitive Learning

Zhi-Hua Zhou and **Xu-Ying Liu**
National Laboratory for Novel Software Technology
Nanjing University, Nanjing 210093, China
{zhouzh, liuxy}@lamda.nju.edu.cn

Abstract

A popular approach to cost-sensitive learning is to rescale the classes according to their misclassification costs. Although this approach is effective in dealing with binary-class problems, recent studies show that it is often not so helpful when being applied to multi-class problems directly. This paper analyzes that why the traditional rescaling approach is often helpless on multi-class problems, which reveals that before applying rescaling, the *consistency* of the costs must be examined. Based on the analysis, a new approach is presented, which should be the choice if the user wants to use rescaling for multi-class cost-sensitive learning. Moreover, this paper shows that the proposed approach is helpful when unequal misclassification costs and class imbalance occur simultaneously, and can also be used to tackle pure class-imbalance learning. Thus, the proposed approach provides a unified framework for using rescaling to address multi-class cost-sensitive learning as well as multi-class class-imbalance learning.

Introduction

In classical machine learning and data mining settings, the classifiers usually try to minimize the number of errors they will make in classifying new instances. Such a setting is valid only when the costs of different errors are equal. Unfortunately, in many real-world applications the costs of different errors are often unequal. For example, in medical diagnosis, the cost of erroneously diagnosing a patient to be healthy may be much bigger than that of mistakenly diagnosing a healthy person as being sick, because the former kind of error may result in the loss of a life.

Actually, cost-sensitive learning has attracted much attention from the machine learning and data mining communities. As it has been stated in the Technological Roadmap of the MLnetII project (European Network of Excellence in Machine Learning) (Saitta 2000), the inclusion of costs into learning has been regarded as one of the most relevant topics of future machine learning research.

During the past years, many works have been devoted to cost-sensitive learning. The learning process may involve many kinds of costs, such as the test cost, teacher cost, intervention cost, etc., among which the most studied one is the misclassification cost (Turney 2000). The works on misclassification cost can be categorized into two classes further, i.e. works on *example-dependent cost* and on *class-dependent cost*. The former assumes that different examples can have different misclassification costs even though they belong to the same class and are misclassified to another same class, while the latter assumes that the cost of misclassifying any example of a certain class to another certain class will be the same. Although there are a few works on the former (Zadrozny & Elkan 2001; Zadrozny, Langford, & Abe 2002; Brefeld, Geibel, & Wysotzki 2003; Abe, Zadrozny, & Langford 2004), more investigations are on the latter (Breiman *et al.* 1984; Domingos 1999; Elkan 2001; Ting 2002; Drummond & Holte 2003; Maloof 2003). Note that the example-dependent cost-sensitive learning and class-dependent cost-sensitive learning are with quite different properties. On one hand, misclassifying a concerned example into different classes will result in the same cost in the former but usually different costs in the latter. On the other hand, in most real-world applications it is feasible to ask a domain expert to specify the cost of misclassifying a class to another class, while only in some special tasks it is easy to get the cost for every training example. It is noteworthy that the outputs of the research on class-dependent cost-sensitive learning have been deemed as good solutions to learning from imbalanced data sets (Chawla *et al.* 2002; Weiss 2004). This paper will focus on this kind of cost-sensitive learning and hereafter *class-dependent* will not be mentioned explicitly for convenience.

A popular approach to cost-sensitive learning is to *rescale* (or *rebalance* called by Elkan (2001)) the classes such that the influences of different classes on the learning process are in proportion to their costs. A typical process is to assign the training examples of different classes with different weights, where the weights are in proportion to the misclassification costs. Then, the weighted examples are given to a learning algorithm such as C4.5 decision tree to train a model which can be used in future predictions (Elkan 2001; Ting 2002). Note that besides weighting the training examples, the rescaling approach can also be realized in many other ways, such as sampling the training examples (Elkan 2001; Drummond & Holte 2003; Maloof 2003) and moving the

decision thresholds (Domingos 1999; Elkan 2001).

Although the rescaling approach has been shown effective in dealing with binary-class problems (Breiman *et al.* 1984; Domingos 1999; Elkan 2001; Ting 2002; Drummond & Holte 2003; Maloof 2003), recent studies (Zhou & Liu 2006) show that it is often not so useful when being applied to multi-class problems directly. In fact, almost all previous research on cost-sensitive learning studied binary-class problems, and only some recent works started to investigate multi-class cost-sensitive learning (Abe, Zadrozny, & Langford 2004; Zhou & Liu 2006). Although multi-class problems can be converted into a series of binary-class problems to deal with, the user usually favors a more direct solution. This is just like that although multi-class classification can be addressed by traditional support vector machines via pairwise coupling, researchers still attempt to design multi-class support vector machines.

In this paper, the reason why the traditional rescaling approach is often not effective on multi-class problems is analyzed, which reveals that before applying rescaling directly, the *consistency* of the costs must be examined. Based on the analysis, a new approach is proposed, which should be the choice if the user wants to use rescaling for multi-class cost-sensitive learning. This paper also shows that the new approach is helpful when unequal misclassification costs and class imbalance occur simultaneously, and can also be used in tackling pure class-imbalance learning. Thus, it provides a unified framework for using rescaling to address multi-class cost-sensitive learning as well as multi-class class-imbalance learning.

The rest of this paper starts with analyzing why traditional rescaling approach is often helpless on multi-class problems. Then, the new new rescaling approach is presented and experiments are reported, which is followed by the conclusion.

Analysis

Let ε_{ij} $(i, j \in \{1..c\}, \varepsilon_{ii} = 0)$ denote the cost of misclassifying an example of the i-th class to the j-th class, where c is the number of classes. It is evident that these costs can be organized into a *cost matrix* whose element at the i-th row and the j-th column is ε_{ij}. Let n_i denote the number of training examples of the i-th class, and n denote the total number of training examples. In order to simplify the discussion, assume there is no class imbalance, that is, $n_i = n/c$ $(i \in \{1..c\})$.

Rescaling is a general approach which can be used to make any cost-blind learning algorithms cost-sensitive. The principle is to enable the influences of the higher-cost classes be bigger than that of the lower-cost classes. On binary-class problems, the optimal prediction is the 1st class if and only if the expected cost of this prediction is not bigger than the expected cost of predicting the 2nd class, as shown in Eq. 1 where $p = P(class = 1|\boldsymbol{x})$.

$$p \times \varepsilon_{11} + (1-p) \times \varepsilon_{21} \leq p \times \varepsilon_{12} + (1-p) \times \varepsilon_{22} \quad (1)$$

If the inequality in Eq. 1 becomes equality, then predicting either class is optimal. Therefore, the threshold p^* for making optimal decision should satisfy Eq. 2.

$$p^* \times \varepsilon_{11} + (1-p^*) \times \varepsilon_{21} = p^* \times \varepsilon_{12} + (1-p^*) \times \varepsilon_{22} \quad (2)$$

Elkan Theorem (Elkan 2001): To make a target probability threshold p^* correspond to a given probability threshold p_0, the number of the 2nd class examples in the training set should be multiplied by $\frac{p^*}{1-p^*}\frac{1-p_0}{p_0}$.

When the classifier is not biased to any class, the threshold p_0 is 0.5. Considering Eq. 2, the Elkan Theorem tells that the 2nd class should be rescaled against the 1st class according to $p^*/(1-p^*) = \varepsilon_{21}/\varepsilon_{12}$ (reminding $\varepsilon_{11} = \varepsilon_{22} = 0$), which implies that the influence of the 1st class should be $\varepsilon_{12}/\varepsilon_{21}$ times of that of the 2nd class. Generally speaking, the optimal *rescaling ratio* of the i-th class against the j-th class can be defined as Eq. 3, which indicates that the classes should be rescaled in the way that the influence of the i-th class is $\tau_{opt}(i,j)$ times of that of the j-th class. For example, if the weight assigned to the training examples of the j-th class after rescaling (via weighting the training examples) is w_j, then that of the i-th class will be $w_i = \tau_{opt}(i,j) \times w_j$ $(w_i > 0)$.

$$\tau_{opt}(i,j) = \frac{\varepsilon_{ij}}{\varepsilon_{ji}} \quad (3)$$

In the traditional rescaling approach (Breiman *et al.* 1984; Domingos 1999; Ting 2002; Drummond & Holte 2003; Maloof 2003), a quantity ε_i is derived according to Eq. 4 at first.

$$\varepsilon_i = \sum_{j=1}^{c} \varepsilon_{ij} \quad (4)$$

Then, a weight w_i is assigned to the i-th class after rescaling (via weighting the training examples), which is computed according to Eq. 5.

$$w_i = \frac{(n \times \varepsilon_i)}{\sum_{k=1}^{c} (n_k \times \varepsilon_k)} \quad (5)$$

Reminding the assumption that $n_i = n/c$, Eq. 5 becomes:

$$w_i = \frac{(c \times \varepsilon_i)}{\sum_{k=1}^{c} \varepsilon_k} \quad (6)$$

So, it is evident that in the traditional rescaling approach, the rescaling ratio of the i-th class against the j-th class is:

$$\tau_{old}(i,j) = \frac{w_i}{w_j} = \frac{(c \times \varepsilon_i) \,/\, \sum_{k=1}^{c} \varepsilon_k}{(c \times \varepsilon_j) \,/\, \sum_{k=1}^{c} \varepsilon_k} = \frac{\varepsilon_i}{\varepsilon_j} \quad (7)$$

When $c = 2$,

$$\tau_{old}(i,j) = \frac{\varepsilon_i}{\varepsilon_j} = \frac{\sum_{k=1}^{2} \varepsilon_{ik}}{\sum_{k=1}^{2} \varepsilon_{jk}} = \frac{\varepsilon_{11} + \varepsilon_{12}}{\varepsilon_{21} + \varepsilon_{22}}$$

$$= \frac{\varepsilon_{12}}{\varepsilon_{21}} = \frac{\varepsilon_{ij}}{\varepsilon_{ji}} = \tau_{opt}(i,j)$$

This explains that why the traditional rescaling approach can be effective in dealing with the unequal misclassification costs on binary-class problems, as previous research shows (Breiman *et al.* 1984; Domingos 1999; Ting 2002; Drummond & Holte 2003; Maloof 2003).

Unfortunately, when $c > 2$, $\tau_{old}(i,j)$ becomes Eq. 8, which is usually unequal to $\tau_{opt}(i,j)$. This explains that why the traditional rescaling approach is often not effective in dealing with the unequal misclassification costs on multi-class problems.

$$\tau_{old}(i,j) = \frac{\varepsilon_i}{\varepsilon_j} = \frac{\sum_{k=1}^{c} \varepsilon_{ik}}{\sum_{k=1}^{c} \varepsilon_{jk}} \tag{8}$$

The RESCALE$_{new}$ Approach

Suppose each class can be assigned with a weight w_i ($w_i > 0$) after rescaling (via weighting the training examples). In order to appropriately rescale all the classes simultaneously, according to the analysis presented in the previous section, it is desired that the weights satisfy $\frac{w_i}{w_j} = \tau_{opt}(i,j)$ ($i,j \in \{1..c\}$), which implies the following $\binom{c}{2}$ number of constraints:

$$\begin{aligned}
&\frac{w_1}{w_2} = \frac{\varepsilon_{12}}{\varepsilon_{21}}, \quad \frac{w_1}{w_3} = \frac{\varepsilon_{13}}{\varepsilon_{31}}, \quad \cdots, \quad \frac{w_1}{w_c} = \frac{\varepsilon_{1c}}{\varepsilon_{c1}} \\
&\qquad\qquad \frac{w_2}{w_3} = \frac{\varepsilon_{23}}{\varepsilon_{32}}, \quad \cdots, \quad \frac{w_2}{w_c} = \frac{\varepsilon_{2c}}{\varepsilon_{c2}} \\
&\qquad\qquad\quad \cdots \qquad\quad \cdots \qquad\quad \cdots \\
&\qquad\qquad\qquad\qquad\qquad \frac{w_{c-1}}{w_c} = \frac{\varepsilon_{c-1,c}}{\varepsilon_{c,c-1}}
\end{aligned}$$

These constraints can be transformed into the equations shown in Eq. 9. If non-trivial solution $\boldsymbol{w} = [w_1, w_2, \cdots, w_c]^{\mathbf{T}}$ can be solved from Eq. 9, then the classes can be appropriately rescaled simultaneously, which implies that the multi-class cost-sensitive learning problem can be solved with rescaling directly.

$$\begin{cases}
w_1 \times \varepsilon_{21} - w_2 \times \varepsilon_{12} + w_3 \times 0 + \cdots + w_c \times 0 = 0 \\
w_1 \times \varepsilon_{31} + w_2 \times 0 - w_3 \times \varepsilon_{13} + \cdots + w_c \times 0 = 0 \\
\cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots = 0 \\
w_1 \times \varepsilon_{c1} + w_2 \times 0 + w_3 \times 0 + \cdots - w_c \times \varepsilon_{1c} = 0 \\
w_1 \times 0 + w_2 \times \varepsilon_{32} - w_3 \times \varepsilon_{23} + \cdots + w_c \times 0 = 0 \\
\cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots = 0 \\
w_1 \times 0 + w_2 \times \varepsilon_{c2} + w_3 \times 0 + \cdots - w_c \times \varepsilon_{2c} = 0 \\
\cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots = 0 \\
\cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots = 0 \\
w_1 \times 0 + w_2 \times 0 + w_3 \times 0 + \cdots - w_c \times \varepsilon_{c-1,c} = 0
\end{cases} \tag{9}$$

Eq. 9 has non-trivial solution if and only if the rank of its coefficient matrix (which is a $\frac{c(c-1)}{2} \times c$ matrix) shown in Eq. 10 is smaller than c, which is equivalent to the condition that the determinant $|A|$ of any $c \times c$ sub-matrix A of Eq. 10 is zero. Note that for a $(\frac{c(c-1)}{2} \times c)$ matrix ($c > 2$), the rank is at most c.

$$\begin{bmatrix}
\varepsilon_{21} & -\varepsilon_{12} & 0 & \cdots & 0 \\
\varepsilon_{31} & 0 & -\varepsilon_{13} & \cdots & 0 \\
\cdots & \cdots & \cdots & \cdots & 0 \\
\varepsilon_{c1} & 0 & 0 & \cdots & -\varepsilon_{1c} \\
0 & \varepsilon_{32} & -\varepsilon_{23} & \cdots & 0 \\
\cdots & \cdots & \cdots & \cdots & 0 \\
0 & \varepsilon_{c2} & 0 & \cdots & -\varepsilon_{2c} \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
0 & \cdots & 0 & \cdots & -\varepsilon_{c-1,c}
\end{bmatrix} \tag{10}$$

For example, when all the classes are with equal costs, unit vector can be solved from Eq. 9 as a non-trivial solution of $\boldsymbol{w}$, and thus the classes should be equally rescaled (in this case the problem degenerates to a common equal-cost multi-class learning problem).

It is noteworthy that when the rank of the co-efficient matrix is c, Eq. 9 does not have non-trivial solution, which implies that there will be no proper weight assignment for rescaling all the classes simultaneously. Therefore, rescaling can hardly be applied directly, and in order to use rescaling, the multi-class problem has to be decomposed to many sub-problems (usually a series of binary-class problems by pairwise coupling) to address.

Based on the above analysis, the RESCALE$_{new}$ approach is proposed and summarized in Table 1.

Table 1: The RESCALE$_{new}$ approach

For a given cost matrix, generate the co-efficient matrix in the form of Eq. 10 and see whether its rank is smaller than c:
If yes (the cost matrix is called as a *consistent* cost matrix), solve $\boldsymbol{w}$ from Eq. 9, use $\boldsymbol{w}$ to rescale the classes simultaneously, and pass the rescaled data set to any cost-blind classifier.
Otherwise (the cost matrix is called as an *inconsistent* cost matrix), decompose the multi-class problem to $\binom{c}{2}$ number of binary-class problems by pairwise coupling (that is, every equation in Eq. 9 corresponds to a binary-class problem), rescale each binary-class data set and pass it to any cost-blind classifier, while the final prediction is made by voting the class labels predicted by the binary-class classifiers.

Experiments

The proposed RESCALE$_{new}$ approach (denoted by NEW) is compared with the traditional rescaling approach (denoted by OLD) used in previous works (Breiman *et al.* 1984; Domingos 1999; Ting 2002; Drummond & Holte 2003; Maloof 2003). Here the scheme of weighting the training examples is used to realize the rescaling process. Since C4.5 decision tree can deal with weighted examples, it is used as the cost-blind learner (denoted by BLIND) in the experiments. Note that in this way, the OLD approach reassembles the C4.5CS method (Ting 2002).

Twenty multi-class data sets are used in the experiments, where the first ten data sets are without class imbalance

Table 2: Experimental data sets (A: # attributes, C: # classes)

Data set	Size	A	C	Class distribution
mfeat-fouri	2,000	76	10	[200*10]
segment	2,310	19	7	[330*7]
syn-a	1,500	2	3	[500*3]
syn-b	3,000	2	3	[1,000*3]
syn-c	6,000	2	3	[2,000*3]
syn-d	2,500	2	5	[500*5]
syn-e	5,000	2	5	[1,000*5]
syn-f	10,000	2	5	[2,000*5]
vowel	990	13	11	[90*11]
waveform	3,000	40	3	[1,000*3]
abalone	4,177	8	3	[1,307; 1,342; 1,528]
ann	7,200	21	3	[166; 368; 6,666]
balance	625	4	3	[49; 288; 288]
car	1,728	6	4	[65; 69; 384; 1,210]
cmc	1,473	9	3	[333; 511; 629]
connect4	67,557	42	3	[6,449; 16,635; 44,473]
page	5,473	10	5	[28; 88; 115; 329; 4,913]
satellite	6,435	36	6	[626; 703; 707; 1358; 1508; 1533]
solarflare2	1,066	11	6	[43; 95; 147; 211; 239; 331]
splice	3,190	60	3	[767; 768; 1,655]

while the remaining ten are imbalanced. There are 14 UCI data sets (Blake, Keogh, & Merz 1998) and 6 synthetic data sets. The synthetic ones are generated as follows. Each synthetic data set has two attributes, three or five classes, and its examples are generated randomly from normal distributions under the following constraints: the mean value and standard deviation of each attribute are random real values in $[0, 10]$, and the coefficients are random real values in $[-1, +1]$. Information on the experimental data sets are summarized in Table 2.

On each data set, two series of experiments are performed. The first series of experiments deal with consistent cost matrices while the second series deal with inconsistent ones. Here the consistent matrices are generated as follows: a c-dimensional real value vector is randomly generated and regarded as the root of Eq. 9, then a real value is randomly generated for ε_{ij} ($i, j \in [1, c]$ and $i \neq j$) such that ε_{ji} can be solved from Eq. 9. All these real values are in $[1, 10]$, $\varepsilon_{ii} = 0$ and at least one ε_{ij} is 1.0. Note that in generating cost matrices for imbalanced data sets, it is constrained that the cost of misclassifying the smallest class to the largest class is the biggest while the cost of misclassifying the largest class to the smallest class is the smallest. This owes to the fact that when the largest class is with the biggest misclassification cost, classical machine learning approaches are good enough and therefore this situation is not concerned in the research of cost-sensitive learning and class-imbalance learning. The inconsistent matrices are generated in a similar way except that one ε_{ji} solved from Eq. 9 is replaced by a random value. The ranks of the co-efficient matrices corresponding to these cost matrices have been examined to guarantee that they are smaller or not smaller than c, respectively.

In each series of experiments, ten times 10-fold cross validation are performed. Concretely, 10-fold cross validation is repeated for ten times with randomly generated cost matrices belonging to the same type (i.e. consistent or inconsistent), and the average results are recorded.

Table 3: Misclassification costs on consistent cost matrices

Data set	BLIND	OLD	NEW
mfeat-fouri	519.23 ± 149.11	$.894 \pm .136$	$\mathbf{.821 \pm .183}$
segment	$\mathbf{49.71 \pm 21.60}$	$1.030 \pm .170$	$1.025 \pm .123$
syn-a	166.41 ± 120.75	$.934 \pm .124$	$\mathbf{.831 \pm .206}$
syn-b	357.44 ± 277.53	$.879 \pm .243$	$\mathbf{.802 \pm .189}$
syn-c	245.51 ± 264.63	$.913 \pm .120$	$\mathbf{.889 \pm .154}$
syn-d	302.07 ± 126.70	$.949 \pm .061$	$\mathbf{.868 \pm .147}$
syn-e	690.71 ± 252.04	$.914 \pm .100$	$\mathbf{.854 \pm .116}$
syn-f	1511.93 ± 539.57	$.890 \pm .088$	$\mathbf{.813 \pm .086}$
vowel	123.17 ± 30.76	$1.000 \pm .096$	$\mathbf{.979 \pm .120}$
waveform	211.40 ± 98.57	$.926 \pm .132$	$\mathbf{.884 \pm .143}$
ave.	417.76 ± 188.13	$.933 \pm .127$	$\mathbf{.877 \pm .147}$
abalone	638.83 ± 191.03	$.788 \pm .220$	$\mathbf{.728 \pm .212}$
ann	4.806 ± 1.517	$\mathbf{.975 \pm .106}$	$.991 \pm .069$
balance	71.05 ± 43.06	$1.016 \pm .231$	$\mathbf{.826 \pm .171}$
car	43.29 ± 13.97	$.888 \pm .206$	$\mathbf{.766 \pm .172}$
cmc	243.01 ± 137.35	$.888 \pm .165$	$\mathbf{.798 \pm .179}$
connect4	5032.21 ± 3447.36	$.928 \pm .143$	$\mathbf{.895 \pm .152}$
page	122.13 ± 106.11	$.983 \pm .100$	$\mathbf{.972 \pm .085}$
satellite	502.15 ± 150.72	$.970 \pm .056$	$\mathbf{.950 \pm .064}$
solarflare2	194.90 ± 91.46	$.876 \pm .236$	$\mathbf{.819 \pm .202}$
splice	56.12 ± 26.99	$.983 \pm .093$	$\mathbf{.929 \pm .091}$
ave.	690.85 ± 1460.57	$.930 \pm .066$	$\mathbf{.867 \pm .087}$

Table 4: Misclassification costs on inconsistent cost matrices

Data set	BLIND	OLD	NEW
mfeat-fouri	263.16 ± 33.88	$.999 \pm .036$	$\mathbf{.833 \pm .077}$
segment	$\mathbf{36.97 \pm 9.98}$	$1.058 \pm .100$	$1.046 \pm .176$
syn-a	300.59 ± 74.17	$.979 \pm .109$	$\mathbf{.911 \pm .132}$
syn-b	625.74 ± 134.10	$.860 \pm .131$	$\mathbf{.715 \pm .178}$
syn-c	318.86 ± 99.70	$1.009 \pm .072$	$\mathbf{.945 \pm .045}$
syn-d	313.35 ± 29.72	$1.002 \pm .038$	$\mathbf{.939 \pm .049}$
syn-e	940.71 ± 98.86	$.968 \pm .052$	$\mathbf{.912 \pm .087}$
syn-f	1928.98 ± 269.45	$1.041 \pm .081$	$\mathbf{.924 \pm .079}$
vowel	113.30 ± 9.39	$1.008 \pm .090$	$\mathbf{.978 \pm .073}$
waveform	378.27 ± 52.86	$.933 \pm .120$	$\mathbf{.850 \pm .099}$
ave.	521.99 ± 81.21	$.986 \pm .083$	$\mathbf{.905 \pm .099}$
abalone	$1,035.28 \pm 137.58$	$.643 \pm .114$	$\mathbf{.548 \pm .114}$
ann	7.760 ± 2.606	$\mathbf{.927 \pm .199}$	$1.234 \pm .364$
balance	72.77 ± 10.37	$\mathbf{.919 \pm .125}$	$.930 \pm .081$
car	71.82 ± 16.63	$.974 \pm .143$	$\mathbf{.769 \pm .084}$
cmc	369.41 ± 80.94	$.882 \pm .091$	$\mathbf{.870 \pm .130}$
connect4	7135.72 ± 716.50	$\mathbf{.942 \pm .074}$	$.952 \pm .081$
page	99.15 ± 13.52	$1.007 \pm .061$	$\mathbf{.902 \pm .084}$
satellite	479.25 ± 63.00	$.982 \pm .038$	$\mathbf{.870 \pm .060}$
solarflare2	153.46 ± 21.65	$.995 \pm .063$	$\mathbf{.940 \pm .090}$
splice	71.21 ± 20.97	$.983 \pm .042$	$\mathbf{.913 \pm .068}$
ave.	$949.58 \pm 2,082.99$	$.925 \pm .101$	$\mathbf{.893 \pm .161}$

Table 5: Summary of the comparison (win/tie/loss) under pairwise two-tailed t-tests with 0.05 significance level (CCM: consistent cost matrices, ICM: inconsistent cost matrices)

	on CCM		on ICM	
	BLIND	OLD	BLIND	OLD
OLD	8/12/0	-	7/13/0	-
NEW	15/5/0	13/7/0	17/2/1	13/6/1

There are some powerful tools such as ROC and cost curves (Drummond & Holte 2000) for visually evaluating the performance of binary-class cost-sensitive learning approaches. Unfortunately, they can hardly be applied to multi-class problems. Therefore, here the *misclassification costs* are compared. The results in the form of "mean $\pm$ standard deviation" on consistent and inconsistent cost matrices are tabulated in Tables 3 and 4, respectively, where the best performance of each row is boldfaced. Note that for the cost-blind approach, the absolute misclassification costs are reported; while for the traditional rescaling approach and RESCALE$_{new}$, the ratios of their misclassification costs against that of the cost-blind approach are presented.

Tables 3 and 4 reveal that no matter on balanced or imbalanced data sets and on consistent or inconsistent cost matrices, the RESCALE$_{new}$ approach performs apparently better than the traditional rescaling approach.

In detail, pairwise two-tailed t-test with 0.05 significance level indicates that on consistent cost matrices, the traditional rescaling approach is effective on only 8 data sets, i.e. *mfeat-fouri*, *syn-b* to *syn-f*, *abalone* and *car*, while RESCALE$_{new}$ is effective on 15 data sets, i.e. *mfeat-fouri*, *syn-b* to *syn-f*, *waveform*, *abalone*, *balance*, *car*, *cmc*, *connect4*, *satellite*, *solarflare2* and *splice*; on inconsistent cost matrices, the traditional rescaling approach is effective on only 7 data sets, i.e. *syn-b*, *syn-e*, *abalone*, *ann*, *balance*, *cmc* and *connect4*, while RESCALE$_{new}$ is effective on 17 data sets, i.e. except on *segment*, *vowel* and *ann*. Moreover, on consistent cost matrices, RESCALE$_{new}$ performs significantly better than the traditional rescaling approach on 13 data sets, i.e. *mfeat-fouri*, *syn-b* to *syn-f*, *waveform*, *abalone*, *balance*, *cmc*, *connect4*, *solarflare2* and *splice*; on inconsistent cost matrices, RESCALE$_{new}$ also performs significantly better than the traditional rescaling approach on 13 data sets, i.e. *mfeat-fouri*, *syn-b* to *syn-f*, *waveform*, *abalone*, *car*, *page*, *satellite*, *solarflare2* and *splice*. These comparisons are summarized in Table 5, which presents the times of win/tie/loss of the approach on the row over the approach on the column.

Note that the performance of RESCALE$_{new}$ is almost always significantly better than or at least comparable to that of the traditional rescaling approach, except that on inconsistent cost matrices RESCALE$_{new}$ degenerates the performance on *ann*. It can be found from Table 2 that the *ann* data set is seriously imbalanced, where the largest class is over 40 times bigger than the smallest one. This may suggest that in dealing with data sets with unequal misclassification costs and serious imbalance, using only the cost information to rescale the classes may not be sufficient. This issue will be investigated further in future work.

Table 6: MAUC values on pure class-imbalance learning

Data set	BLIND	RESCALE$_{new}$
abalone	$.707 \pm .005$	$\mathbf{.713 \pm .005}$
ann	$.995 \pm .001$	$\mathbf{.999 \pm .000}$
balance	$.752 \pm .013$	$\mathbf{.757 \pm .011}$
car	$\mathbf{.975 \pm .003}$	$.968 \pm .006$
cmc	$.679 \pm .010$	$\mathbf{.690 \pm .008}$
connect4	$\mathbf{.843 \pm .001}$	$.824 \pm .003$
page	$.969 \pm .005$	$\mathbf{.977 \pm .003}$
satellite	$.962 \pm .002$	$\mathbf{.964 \pm .002}$
solarflare2	$.878 \pm .003$	$\mathbf{.903 \pm .004}$
splice	$\mathbf{.975 \pm .001}$	$\mathbf{.975 \pm .001}$
ave.	$.874 \pm .116$	$\mathbf{.877 \pm .114}$

As mentioned before, cost-sensitive learning approaches have been deemed as good solutions to class-imbalance learning (Chawla *et al.* 2002; Weiss 2004). Therefore, it is interesting to see whether the RESCALE$_{new}$ approach can work well on learning from imbalanced multi-class data sets. Actually, although class-imbalance learning is a hot topic, few work has been devoted to the research of multi-class class-imbalance learning.

For this purpose, experiments are performed on the ten imbalance data sets shown in the second half of Table 2. Note that here equal misclassification costs are used. In other words, the experiments are conducted to evaluate the performance of RESCALE$_{new}$ on pure class-imbalance learning.

In the experiments C4.5 decision tree is still used as the baseline, which does not take into account the class imbalance information (still denoted by BLIND). For the RESCALE$_{new}$ approach, considering that the influences of the smaller classes should be increased while that of the larger classes should be decreased by the rescaling process, the reciprocals of the sizes of the classes are used as the rescaling information. For example, suppose the i-th class has n_i number of examples, then ε_{ij} ($j \in \{1..c\}$ and $j \neq i$) in Eq. 9 is set to $1/n_i$. Note that since $\varepsilon_{ij} = \varepsilon_{ik}$ ($j, k \in \{1..c\}$ and $j, k \neq i$), the resulting Eq. 10 always has non-trivial solutions, which is somewhat similar to the cases of cost-sensitive learning with consistent cost matrices.

The MAUC measure (Hand & Till 2001) is used to evaluate the performance, which is a variant of AUC designed for multi-class class-imbalance learning. The bigger the MAUC value, the better the performance. Ten times 10-fold cross validation are executed and the results in the form of "mean $\pm$ standard deviation" are tabulated in Table 6, where the best performance of each row is boldfaced.

Pairwise two-tailed t-test with 0.05 significance level indicates that the performance of RESCALE$_{new}$ is significantly better than that of the standard C4.5 decision tree on 6 data sets, i.e. *abalone*, *ann*, *cmc*, *page*, *satellite* and *solarflare2*, worse on *car* and *connect4*, and there is no significant differ-

ence on *balance* and *splice*. This suggests that RESCALE$_{new}$ can also be used to address pure class-imbalance learning on multi-class problems.

Conclusion

This paper analyzes that why the traditional rescaling approach is helpful in binary-class cost-sensitive learning but is often helpless on multi-class cost-sensitive learning. The analysis shows that applying rescaling directly on multi-class tasks can obtain good performance only when the costs are consistent. Although costs in real-world applications are not random and consistent costs do appear in some practical tasks, the consistency of the costs should be examined before rescaling is used. Based on the analysis, this paper presents a new approach, which should be the choice if the user really wants to use rescaling, instead of other approaches (Zhou & Liu 2006), to deal with multi-class cost-sensitive learning. It is shown that the proposed approach is also helpful when unequal misclassification costs and class imbalance occur simultaneously, and can even be used to tackle pure class-imbalance learning. Thus, the proposed approach provides a unified framework for using rescaling to address multi-class cost-sensitive learning as well as multi-class class-imbalance learning.

It has been observed in the experiments reported in this paper that when unequal misclassification costs and class imbalance occur simultaneously, using the cost information to rescale the classes can work well on most data sets, but does not on seriously imbalanced data sets. Exploring the ground under this observation and designing strong rescaling schemes for such cases are important future issues. Moreover, in most studies on cost-sensitive learning, the cost matrices are usually fixed, while in some real-world tasks the costs might change due to many reasons. Designing effective methods for cost-sensitive learning with variable cost matrices is another interesting issue to be studied in the future. Furthermore, developing powerful tools for visually evaluating multi-class cost-sensitive learning approaches, such as the ROC and cost curves for binary-class cases, is also an interesting issue for future work.

Acknowledgments

This work was supported by the National Science Fund for Distinguished Young Scholars of China under Grant No. 60325207 and the Jiangsu Science Foundation under Grant No. BK2004001.

References

Abe, N.; Zadrozny, B.; and Langford, J. 2004. An iterative method for multi-class cost-sensitive learning. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 3–11.

Blake, C.; Keogh, E.; and Merz, C. J. 1998. UCI repository of machine learning databases. [http://www.ics.uci.edu/∼mlearn/MLRepository.html], Department of Information and Computer Science, University of California, Irvine, CA.

Brefeld, U.; Geibel, P.; and Wysotzki, F. 2003. Support vector machines with example dependent costs. In *Proceedings of the 14th European Conference on Machine Learning*, 23–34.

Breiman, L.; Friedman, J. H.; Olsen, R. A.; and Stone, C. J. 1984. *Classification and Regression Trees*. Belmont, CA: Wadsworth.

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

Domingos, P. 1999. MetaCost: A general method for making classifiers cost-sensitive. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 155–164.

Drummond, C., and Holte, R. C. 2000. Explicitly representing expected cost: An alternative to ROC representation. In *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 198–207.

Drummond, C., and Holte, R. C. 2003. C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. In *Working Notes of the ICML'03 Workshop on Learning from Imbalanced Data Sets*.

Elkan, C. 2001. The foundations of cost-senstive learning. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, 973–978.

Hand, D. J., and Till, R. J. 2001. A simple generalisation of the area under the ROC curve for multiple class classification problems. *Machine Learning* 45(2):171–186.

Maloof, M. A. 2003. Learning when data sets are imbalanced and when costs are unequal and unknown. In *Working Notes of the ICML'03 Workshop on Learning from Imbalanced Data Sets*.

Saitta, L., ed. 2000. *Machine Learning - A Technological Roadmap*.

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

Turney, P. D. 2000. Types of cost in inductive concept learning. In *Proceedings of the ICML'2000 Workshop on Cost-Sensitive Learning*, 15–21.

Weiss, G. M. 2004. Mining with rarity - problems and solutions: A unifying framework. *SIGKDD Explorations* 6(1):7–19.

Zadrozny, B., and Elkan, C. 2001. Learning and making decisions when costs and probabilities are both unknown. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 204–213.

Zadrozny, B.; Langford, J.; and Abe, N. 2002. A simple method for cost-sensitive learning. Technical report, IBM.

Zhou, Z.-H., and Liu, X.-Y. 2006. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge and Data Engineering* 18(1):63–77.