Cost-Sensitive Online Classification

Jialei Wang, Peilin Zhao, Steven C.H. Hoi
School of Computer Engineering, Nanyang Technological University, Singapore 639798
Email: {jl.wang, zhao0106, chhoi}@ntu.edu.sg

***Abstract*—Both *cost-sensitive classification* and *online learning* have been studied extensively in data mining and machine learning communities, respectively. It is a bit surprising that there was very limited comprehensive study for addressing an important intersecting problem, that is, cost-sensitive online classification. In this paper, we formally study this problem, and propose a new framework for cost-sensitive online classification by exploiting the idea of online gradient descent techniques. Based on the framework, we propose a family of cost-sensitive online classification algorithms, which are designed to directly optimize two well-known cost-sensitive measures: (i) maximization of weighted sum of *sensitivity* and *specificity*, and (ii) minimization of weighted *misclassification cost*. We analyze the theoretical bounds of the cost-sensitive measures made by the proposed algorithms, and extensively examine their empirical performance on a variety of cost-sensitive online classification tasks.**

***Keywords*-cost-sensitive learning, online learning, classification**

I. Introduction

Online learning represents a family of efficient and scalable machine learning methods, which has been extensively studied in machine learning and data mining literature [5], [15], [21], [25], [27], [29]. In general, the goal of online learning is to incrementally learn some prediction models to make correct predictions on a stream of examples that arrive sequentially. Online learning is advantageous for its high efficiency and scalability for large-scale applications, and has been applied to solve online classification tasks in a variety of real-world data mining applications.

Despite being studied extensively in machine learning, most existing online learning techniques are unsuitable and potentially would not be effective enough to solve a *cost-sensitive classification* task, an important data mining problem which takes the misclassification costs into consideration [10], [7]. This is because most existing online learning studies often concern the performance of an online classification algorithm in terms of prediction *mistake rate* or *accuracy*, which is obviously *cost-insensitive* and thus *inappropriate* for many real applications in data mining, especially for cost-sensitive classification tasks where datasets are often class-imbalanced and the misclassification costs of instances from different classes can be very different [24], [4], [9], [20].

To address the above challenge of cost-sensitive classification, researchers especially in data mining literature have proposed more meaningful metrics, such as the weighted sum of *sensitivity* and *specificity* [13], [3] and the weighted *misclassification cost* [10], [1]. Over the past decades, substantial research efforts have been devoted to developing batch classification algorithms to improve the cost-sensitive measures, including the weighted sum of sensitivity and specificity and the weighted misclassification cost metrics [10], [1]. However, these batch classification algorithms often suffer from low efficiency and poor scalability when solving large-scale problems, making them unsuitable for online classification applications.

Although both *cost-sensitive classification* and *online learning* have been studied extensively in data mining and machine learning communities, respectively, there were very few comprehensive studies on cost-sensitive online classification in both data mining and machine learning literature. In this paper, we formally investigate this problem by attempting to develop cost-sensitive algorithms for solving an online cost-sensitive classification task. As the first comprehensive study, in this paper, we propose a new framework of Cost-Sensitive Online Classification to resolve this challenging open problem. The key challenge of our framework is how to develop an effective cost-sensitive online algorithm which can directly optimize a predefined cost-sensitive measure (e.g., balanced accuracy or weighted misclassification cost) for an online classification task, and further offer theoretical guarantee of the proposed algorithm.

To this end, we summarize the major contributions in this work as follows: (i) we propose a family of cost-sensitive online algorithms using online gradient descent learning technique to tackle the online optimization task of maximizing the weighted sum or minimizing the weighted misclassification cost; (ii) we theoretically analyze the cost-sensitive measure bounds of the proposed algorithms, and extensively examine their empirical performance of cost-sensitive online classification tasks.

The rest of the paper is organized as follows. Section II briefs the related works. Section III formulates the problem and presents the proposed algorithms. Section IV theoretically analyzes the bounds of the proposed algorithms. Section V discusses our experimental results, and finally Section VI concludes this work.

II. Related Work

Our work is mainly related to two groups of research in data mining and machine learning: (i) cost-sensitive classification in data mining literature, and (ii) online learning in machine learning literature.

Cost-sensitive classification has been extensively studied in data mining and machine learning [18], [26], [30]. To address this problem, researchers have proposed a variety of cost-sensitive metrics. The well-known examples include the weighted sum of *sensitivity* and *specificity* [13], [3], and the weighted *misclassification cost* that takes cost into consideration when measuring classification performance [10], [1]. As a special case, when the weights are both equal to 0.5, the weighted sum of sensitivity and specificity is reduced to the well-known *balanced accuracy* [3]. Over the past decades, various batch learning algorithms have been proposed for cost-sensitive classification in literature [22], [23], [7], [10], [19], [17], [20].

Online learning has been extensively studied in machine learning community. Various online learning methods have been actively proposed in literature [21], [15], [5]. Examples include the well-known Perceptron algorithm [21], [11], the recent Passive-aggressive (PA) learning [5], and many other recently proposed algorithms, many of which usually follow the principle of large margin learning [6], [12], [14], [8]. Most online learning algorithms are cost-insensitive, except the prediction-based PA algorithm ('CPA$_{PB}$') [5] and the perceptron algorithm with uneven margin ('PAUM') [16]. However, to the best of our knowledge, no existing work in this area had attempted to directly optimize the two cost-sensitive metrics in an online learning setting. Finally, we note that our work is very different from another recent online learning study [28], which aims to optimize AUC, but cannot be guaranteed to optimize the cost-sensitive measures in our study.

III. Cost-Sensitive Online Classification

A. Problem Formulation

Without loss of generality, let us consider an online binary classification problem. At each learning round, the learner receives an instance and predicts its class label as "+1" or "-1". After making the prediction, the learner receives the true label of the instance and suffers a loss if the prediction is incorrect. At the end of each round, the learner makes use of the received training example and it class label to update the prediction model.

Formally, let us denote by $\mathbf{x}_t \in \mathbb{R}^n$ the instance received at the t-th learning step, and $\mathbf{w}_t \in \mathbb{R}^n$ a linear prediction model learned from the previous $t-1$ training examples. We also denote the prediction for the t-th instance as $\hat{y}_t = sign(\mathbf{w}_t \cdot \mathbf{x}_t)$, while the value $|\mathbf{w}_t \cdot \mathbf{x}_t|$, known as the "margin", is used as the confidence of the learner on the prediction. The true label for instance $\mathbf{x}_t$ is denoted as $y_t \in \{-1, +1\}$. If $\hat{y}_t \neq y_t$, the learner made a mistake; otherwise it made a correct prediction.

For binary classification, the result of each prediction for an instance can be classified into four cases: (1) *True Positive* (TP) if $\hat{y}_t = y_t = +1$; (2) *False Positive* (FP) if $\hat{y}_t = +1$ and $y_t = -1$; (3) *True Negative* (TN) if $\hat{y}_t = y_t = -1$; and (4) *False Negative* (FN) if $\hat{y}_t = -1$ and $y_t = +1$.

We now consider a sequence of training examples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ for online learning. Then, for convenience, we denote by $\mathcal{M}$ the set of indexes that correspond to the trials of misclassification:

$$\mathcal{M} = \{t \mid y_t \neq \text{sign}(\mathbf{w}_t \cdot \mathbf{x}_t),\ \forall t \in [T]\}$$

where $[T] = \{1, \ldots, T\}$. Similarly, we denote by $\mathcal{M}_p = \{t \mid t \in \mathcal{M}\ and\ y_t = +1\}$ the set of indexes for false negatives, and $\mathcal{M}_n = \{t \mid t \in \mathcal{M}\ and\ y_t = -1\}$ the set of indexes for false positives.

Further, we introduce notation $M = |\mathcal{M}|$ to denote the number of mistakes, $M_p = |\mathcal{M}_p|$ to denote the number of false negatives, and $M_n = |\mathcal{M}_n|$ to denote the number of false positives. Also we use notation $\mathcal{I}_T^p = \{i \in [T] | y_i = +1\}$ to denote the set of indexes of the positive examples, $\mathcal{I}_T^n = \{i \in [T] | y_i = -1\}$ to denote the set of indexes of negative examples, $T_p = |\mathcal{I}_T^p|$ to denote the number of positive examples, and $T_n = |\mathcal{I}_T^n|$ to denote the number of negative examples.

For performance metrics, *sensitivity* is defined as the ratio between the number of true positives $T_p - M_p$ and the number of positive examples; *specificity* is defined as the ratio between $T_n - M_n$ and the number of negative examples; and *accuracy* is defined as the ratio between the number of correctly classified examples and the total number of examples. These can be summarized as:

$$\begin{aligned} sensitivity &= \frac{T_p - M_p}{T_p}, \\ specificity &= \frac{T_n - M_n}{T_n}, \\ accuracy &= \frac{T - M}{T} \end{aligned}$$

Consider an online binary classification task, without loss of generality, we assume positive class is the rare class, i.e., $T_p \leq T_n$, the number of positive examples is smaller than the number of negative examples. For simplicity, we also assume that $\|\mathbf{x}_t\| \leq 1$. For traditional online learning, the performance is measured by the prediction accuracy (or mistake rate equivalently) over the sequence of examples. This is inappropriate for imbalanced data because a trivial learner that simply classifies any example as negative could achieve a quite high accuracy for a highly imbalanced dataset. Thus, a more appropriate metric is to measure the *sum* of weighted *sensitivity* and *specificity*, i.e.,

$$sum = \eta_p \times sensitivity + \eta_n \times specificity \qquad (1)$$

where $\eta_p + \eta_n = 1$ and $0 \leq \eta_p, \eta_n \leq 1$ are two parameters to trade off between sensitivity and specificity. Notably, when $\eta_p = \eta_n = 0.5$, the corresponding sum is the well known balanced accuracy. In general, the higher the *sum* value, the better the classification performance. Besides, another approach is to measure the total misclassification cost suffered by the algorithm, which is defined as:

$$cost = c_p \times M_p + c_n \times M_n \quad (2)$$

where $c_p + c_n = 1$ and $0 \leq c_p, c_n \leq 1$ are the misclassification cost parameters for positive and negative classes, respectively. The lower the *cost* value, the better the classification performance.

B. Algorithms

In this section, we propose a framework of Cost-Sensitive Online Classification for cost-sensitive classification by optimizing two cost-sensitive measures. Before presenting our algorithms, we first prove the following important proposition that motivates our solution.

Proposition 1: Consider a cost-sensitive classification problem, the goal of maximizing the weighted sum in (1) or minimizing the weighted cost in (2) is equivalent to minimizing the following objective:

$$\sum_{y_t=+1} \rho \mathrm{I}_{(y_t \mathbf{w} \cdot \mathbf{x}_t < 0)} + \sum_{y_t=-1} \mathrm{I}_{(y_t \mathbf{w} \cdot \mathbf{x}_t < 0)} \quad (3)$$

where $\rho = \frac{\eta_p T_n}{\eta_n T_p}$ for the maximization of the weighted sum, and $\rho = \frac{c_p}{c_n}$ for the minimization of the weighted misclassification cost.

Proposition 1 gives the explicit objective function for optimization, but the indicator function is not convex. To facilitate the online optimization task, we replace the indicator function by its convex surrogate, i.e., either one of the following modified hinge loss functions:

$$\ell^{I}(\mathbf{w}; (\mathbf{x}, y)) = \max(0, (\rho * \mathrm{I}_{(y=1)} + \mathrm{I}_{(y=-1)}) - y(\mathbf{w} \cdot \mathbf{x})) \quad (4)$$

$$\ell^{II}(\mathbf{w}; (\mathbf{x}, y)) = (\rho * \mathrm{I}_{(y=1)} + \mathrm{I}_{(y=-1)}) * \max(0, 1 - y(\mathbf{w} \cdot \mathbf{x})) \quad (5)$$

As a result, we can formulate the optimization problem for cost-sensitive classification as follows:

$$\mathcal{F}_T^*(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{t=1}^{T} \ell^*(\mathbf{w}; (\mathbf{x}_t, y_t)) \quad * \in \{I, II\} \quad (6)$$

where $\|\mathbf{w}\|^2$ is introduced to regularize the complexity of the linear classifier and C is a positive penalty parameter of the cumulative loss. The idea of the above formulation is somewhat similar to the biased formulation of batch SVM for learning with imbalanced datasets [1].

Now our goal is to find an online learning solution to tackle the above convex optimization (6). To this end, we propose to solve the problem using the online gradient descent approach [31], [2], that is,

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \lambda \nabla \ell_t(\mathbf{w}_t)$$

where λ is a learning rate parameter and $\ell_t(\mathbf{w}) = \ell^*(\mathbf{w}; (x_t, y_t))$, $\forall * \in \{I, II\}$. Specifically, when using the loss function (4), the update rule can be expressed as:

$$\mathbf{w}_{t+1} = \begin{cases} \mathbf{w}_t + \lambda y_t \mathbf{x}_t & \text{if } \ell_t(\mathbf{w}_t) > 0 \\ \mathbf{w}_t & \text{otherwise} \end{cases}$$

We refer to the above resulting cost-sensitive online classification algorithm as "CSOGD-I" for short.

When using the loss function (5), the update rule can be expressed as:

$$\mathbf{w}_{t+1} = \begin{cases} \mathbf{w}_t + \lambda \rho_t y_t \mathbf{x}_t & \text{if } \ell_t(\mathbf{w}_t) > 0 \\ \mathbf{w}_t & \text{otherwise} \end{cases}$$

where $\rho_t = \rho * \mathrm{I}_{(y_t=1)} + \mathrm{I}_{(y_t=-1)}$. We refer to the above resulting algorithm as "CSOGD-II" for short.

Finally, Algorithm 1 summarizes the two proposed CSOGD algorithms.

Algorithm 1 The proposed CSOGD algorithms.

INPUT: learning rate λ; bias parameter $\rho = \frac{\eta_p T_n}{\eta_n T_p}$ for "sum" and $\rho = \frac{c_p}{c_n}$ for "cost"
INITIALIZATION: $\mathbf{w}_1 = 0$.
for $t = 1, \ldots, T$ **do**
  receive instance: $\mathbf{x}_t \in \mathbb{R}^n$;
  predict: $\hat{y}_t = sign(\mathbf{w}_t \cdot \mathbf{x}_t)$;
  receive correct label: $y_t \in \{-1, +1\}$;
  suffer loss $\ell_t(\mathbf{w}_t) = \ell^*(\mathbf{w}_t; (\mathbf{x}_t, y_t)); * \in \{I, II\}$
  if $(\ell_t(\mathbf{w}_t) > 0)$
    update classifier: $\mathbf{w}_{t+1} = \mathbf{w}_t - \lambda \nabla \ell_t(\mathbf{w}_t)$;
  end if
end for
OUTPUT: $\mathbf{w}_{T+1}$.

Remark. In Algorithm 1, one practical concern is about setting the value of ρ when the goal is to optimize the weighted sum performance. In the algorithm, ρ is formally defined as $\rho = \frac{\eta_p T_n}{\eta_n T_p}$. However, the values of T_n and T_p might be unknown in a real-world online learning task. In practice, one could try to approximate the ratio $\frac{T_n}{T_p}$ according to the distribution of online received training data instances over the past sequence, and adaptively update this ratio during the online learning process.

IV. Analysis of Cost-Sensitive Measure Bounds

Although the above proposed algorithm is simple, very limited existing study has formally investigated it for online learning tasks. Below we theoretically analyze its performance for classification tasks in terms of two types of cost-sensitive measures.

To ease our discussion, we denote by $\mathcal{S}$ the set of indexes that correspond to the trials when a margin error happens, $\mathcal{S} = \{t \mid \ell_t(\mathbf{w}_t) > 0\}$. Similarly, we denote by $\mathcal{S}_p = \{t \mid \ell_t(\mathbf{w}_t) > 0 \text{ and } y_t = +1\}$, $\mathcal{S}_n = \{t \mid \ell_t(\mathbf{w}_t) > 0 \text{ and } y_t = -1\}$, $S_p = |\mathcal{S}_p|$, and $S_n = |\mathcal{S}_n|$.

Firstly, we will prove the following lemma, which gives the loss regret bound achieved by the online learning algorithm, and will facilitate later theoretical analysis.

Lemma 1: Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ be a sequence of examples, where $\mathbf{x}_t \in \mathbb{R}^n$, $y_t \in \{-1, +1\}$ and $\|\mathbf{x}_t\| \leq 1$ for all t. Then for any $\mathbf{w} \in \mathbb{R}^n$, for CSOGD-I:

$$\sum_{t=1}^{T} \ell_t(\mathbf{w}_t) \leq \sum_{t=1}^{T} \ell_t(\mathbf{w}) + \|\mathbf{w}\| \sqrt{S_p + S_n}$$

and for CSOGD-II:

$$\sum_{t=1}^{T} \ell_t(\mathbf{w}_t) \leq \sum_{t=1}^{T} \ell_t(\mathbf{w}) + \|\mathbf{w}\| \sqrt{\rho^2 S_p + S_n}$$

Thus, by our proposed method, we can guarantee the following bound on the sum of $\eta_p \times sensitive + \eta_n \times specificity$, where $\eta_p + \eta_n = 1$ and $\eta_p, \eta_n > 0$.

Theorem 1: Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ be a sequence of examples, where $\mathbf{x}_t \in \mathbb{R}^n$, $y_t \in \{-1, +1\}$ and $\|\mathbf{x}_t\| \leq 1$ for all t. By setting $\rho = \frac{\eta_p T_n}{\eta_n T_p}$, for any $\mathbf{w} \in \mathbb{R}^n$, we then have the bounds of the proposed algorithms:

$$\text{sum of CSOGD}_{\text{I}} \geq 1 - \frac{\eta_n}{T_n} \Big(\sum_{t=1}^{T} \ell_t(\mathbf{w}) + \|\mathbf{w}\| \sqrt{S_p + S_n} \Big)$$

$$\text{sum of CSOGD}_{\text{II}} \geq 1 - \frac{\eta_n}{T_n} \Big(\sum_{t=1}^{T} \ell_t(\mathbf{w}) + \|\mathbf{w}\| \sqrt{\rho^2 S_p + S_n} \Big)$$

One limitation of the above algorithm is that for a real online learning task, we may not know the ratio $\frac{T_n}{T_p}$ in advance. To address this issue, an alternative way is to consider the cost of the algorithm for performance evaluation, which does not need to know the ratio $\frac{T_n}{T_p}$ in advance. Specifically, instead of setting $\rho = \frac{\eta_p T_n}{\eta_n T_p}$, we propose to set $\rho = \frac{c_p}{c_n}$, where c_p and c_n are the cost of false negative and the cost of false positive, respectively. We assume $c_p + c_n = 1$, and $c_n, c_p > 0$. Finally, the following theorem gives the cost bound of the proposed cost based algorithm.

Theorem 2: Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ be a sequence of examples, where $\mathbf{x}_t \in \mathbb{R}^n$, $y_t \in \{-1, +1\}$ and $\|\mathbf{x}_t\| \leq 1$ for all t. By setting $\rho = \frac{c_p}{c_n}$, for any $\mathbf{w} \in \mathbb{R}^n$, we then have the bounds of the proposed algorithms:

$$\text{cost of CSOGD}_{\text{I}} \leq c_n \Big[\sum_{t=1}^{T} \ell_t(\mathbf{w}) + \|\mathbf{w}\| \sqrt{S_p + S_n} \Big]$$

$$\text{cost of CSOGD}_{\text{II}} \leq c_n \Big[\sum_{t=1}^{T} \ell_t(\mathbf{w}) + \|\mathbf{w}\| \sqrt{\rho^2 S_p + S_n} \Big]$$

V. Experiments of Cost-Sensitive Online Classification

This section is to evaluate the empirical performance of the two proposed algorithms (CSOGD-I and CSOGD-II). To ease our discussions, we denote by CSOL$_{sum}$ the proposed CSOL algorithm that aims to maximize the weighted sum of sensitivity and specificity, and CSOL$_{cos}$ the proposed CSOL algorithm that aims to minimize the overall misclassification cost.

A. Experimental Testbed and Setup

We compare two CSOGD algorithms with a number of state-of-the-art online learning algorithms, including Perceptron, "ROMMA" and its aggressive version "agg-ROMMA", and two versions of the Passive-Aggressive algorithms ("PA") [5], i.e., PA-I and PA-II. Besides, we also compare with the existing cost-sensitive online algorithms: prediction-based PA algorithm ('CPA$_{PB}$') [5] and the perceptron algorithm with uneven margin ('PAUM') [16].

To examine the performance, we test all the algorithms on a number of benchmark datasets from web machine learning repositories. All of them can be downloaded from LIBSVM website [1]. For space limitation, we randomly choose some of them for our following discussions, which are listed in Table 1.

Table I
List of binary datasets in our experiments.

dataset	#Examples	#Features	#Pos:#Neg
covtype	581012	54	1:1
german	1000	24	1:2.3
w8a	64700	300	1:32.5

To make a fair comparison, all algorithms adopt the same experimental setup. In particular, for all the compared algorithms, the penalty parameter C was set to 10; for the proposed CSOL$_{sum}$ algorithms, we set $\eta_p = \eta_n = 1/2$ for all cases, while for CSOL$_{cos}$, we set $c_p = 0.95$ and $c_n = 0.05$; for PAUM, the uneven margin was set to ρ; for PB-CPA, $\rho(-1, 1)$ was set to 1 and $\rho(1, -1)$ was set to ρ. The learning rate λ of CSOGD-I was set to 0.2, and the learning rate λ of CSOGD-II was set to 0.1. The value of ρ was set to $\frac{c_p}{c_n}$ for CSOL$_{cos}$ and $\frac{\eta_p T_n}{\eta_n T_p}$ for CSOL$_{sum}$, respectively. We also evaluate the parameter sensitivity about the cost-sensitive weights in our experiments.

All the experiments were conducted over 20 random permutations for each dataset. The results are reported by averaging over these 20 runs. We evaluate the online classification performance by several metrics: **sensitivity**, **specificity**, the weighted **sum** of sensitivity and specificity, and the weighted **cost**.

B. Evaluation of Weighted Sum Performance

We first evaluate the weighted sum performance. The first three columns of Table 2 summarize the results of the algorithms. Some observations can be drawn below.

First of all, by examining the *sum* results, we found that CSOGD always achieves the best among all the datasets, which significantly outperforms all the online algorithms,

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

Table II
EVALUATION OF THE COST-SENSITIVE CLASSIFICATION PERFORMANCE OF CSOGD AND OTHER EXISTING ALGORITHMS.

Algorithm	"sum" on covtype			"cost" on covtype		
	Sum(%)	Sensitivity(%)	Specificity (%)	Cost	Sensitivity(%)	Specificity (%)
Perceptron	66.149 ± 0.034	66.771 ± 0.056	65.528 ± 0.051	94563.580 ± 150.542	66.771 ± 0.056	65.528 ± 0.051
ROMMA	63.799 ± 0.562	66.266 ± 2.963	61.332 ± 4.064	96545.407 ± 7371.897	66.266 ± 2.963	61.332 ± 4.064
agg-ROMMA	64.833 ± 0.628	68.768 ± 2.936	60.897 ± 4.113	89876.875 ± 7293.558	68.768 ± 2.936	60.897 ± 4.113
PA-I	65.880 ± 0.044	66.263 ± 0.045	65.498 ± 0.057	95934.380 ± 125.245	66.263 ± 0.045	65.498 ± 0.057
PA-II	66.103 ± 0.043	66.550 ± 0.047	65.656 ± 0.055	95137.125 ± 130.178	66.550 ± 0.047	65.656 ± 0.055
PAUM	69.867 ± 0.035	69.825 ± 0.050	69.908 ± 0.048	33239.145 ± 85.815	90.414 ± 0.031	50.013 ± 0.022
CPA$_{PB}$	65.891 ± 0.044	66.484 ± 0.046	65.298 ± 0.056	72060.113 ± 129.526	75.765 ± 0.047	54.081 ± 0.064
CSOGD-I	74.947 ± 0.022	77.543 ± 0.051	72.351 ± 0.052	35544.630 ± 80.287	89.366 ± 0.030	53.475 ± 0.034
CSOGD-II	**75.526 ± 0.018**	78.960 ± 0.041	72.091 ± 0.048	**14752.020 ± 31.166**	99.245 ± 0.010	14.547 ± 0.074
Algorithm	"sum" on german			"cost" on german		
	Sum(%)	Sensitivity(%)	Specificity (%)	Cost	Sensitivity(%)	Specificity (%)
Perceptron	62.001 ± 1.259	64.967 ± 2.229	59.036 ± 1.483	114.182 ± 6.309	64.967 ± 2.229	59.036 ± 1.483
ROMMA	60.504 ± 1.496	64.400 ± 2.588	56.607 ± 2.202	116.647 ± 7.239	64.400 ± 2.588	56.607 ± 2.202
agg-ROMMA	61.012 ± 1.386	65.517 ± 3.012	56.507 ± 2.156	113.500 ± 8.260	65.517 ± 3.012	56.507 ± 2.156
PA-I	61.654 ± 1.495	65.000 ± 2.372	58.307 ± 1.472	114.342 ± 6.863	65.000 ± 2.372	58.307 ± 1.472
PA-II	61.893 ± 1.467	65.300 ± 2.420	58.486 ± 1.390	113.425 ± 6.974	65.300 ± 2.420	58.486 ± 1.390
PAUM	65.019 ± 1.144	52.367 ± 2.173	77.671 ± 0.980	102.045 ± 6.052	68.367 ± 2.171	66.029 ± 1.243
CPA$_{PB}$	61.850 ± 1.601	65.500 ± 2.218	58.200 ± 1.858	112.612 ± 7.229	65.650 ± 2.514	57.957 ± 1.338
CSOGD-I	**70.690 ± 0.846**	77.367 ± 1.284	64.014 ± 1.039	**77.313 ± 3.514**	77.283 ± 1.244	64.086 ± 1.068
CSOGD-II	70.619 ± 0.824	77.667 ± 1.475	63.571 ± 0.703	84.747 ± 4.635	75.067 ± 1.603	60.893 ± 1.278
Algorithm	"sum" on w8a			"cost" on w8a		
	Sum(%)	Sensitivity(%)	Specificity (%)	Cost	Sensitivity(%)	Specificity (%)
Perceptron	79.011 ± 0.319	65.717 ± 0.614	92.305 ± 0.079	871.072 ± 12.103	65.717 ± 0.614	92.305 ± 0.079
ROMMA	78.559 ± 0.267	62.230 ± 0.440	94.888 ± 0.204	854.022 ± 11.630	62.230 ± 0.440	94.888 ± 0.204
agg-ROMMA	79.090 ± 0.191	61.094 ± 0.381	97.086 ± 0.115	805.900 ± 7.383	61.094 ± 0.381	97.086 ± 0.115
PA-I	79.703 ± 0.300	63.621 ± 0.596	95.785 ± 0.100	800.330 ± 11.264	63.621 ± 0.596	95.785 ± 0.100
PA-II	79.998 ± 0.312	64.307 ± 0.633	95.689 ± 0.099	790.747 ± 11.521	64.307 ± 0.633	95.689 ± 0.099
PAUM	80.849 ± 0.344	63.011 ± 0.694	98.686 ± 0.024	723.015 ± 11.433	62.646 ± 0.632	98.819 ± 0.021
CPA$_{PB}$	80.933 ± 0.304	70.998 ± 0.613	90.868 ± 0.183	798.985 ± 11.668	70.031 ± 0.601	92.077 ± 0.150
CSOGD-I	83.159 ± 0.258	71.128 ± 0.533	95.191 ± 0.058	681.158 ± 9.100	71.136 ± 0.525	95.185 ± 0.059
CSOGD-II	**85.619 ± 0.254**	89.289 ± 0.330	81.949 ± 0.355	**652.142 ± 8.337**	85.331 ± 0.429	87.803 ± 0.285

including two cost-sensitive online algorithms (PAUM and CPA). This shows that it is important to study effective cost-sensitive algorithms.

Second, by examining both *sensitivity* and *specificity* metrics, we found that CSOGD is not only guaranteed to achieve the best *sensitivity* for all cases, but also can produce a fairly good specificity performance for most cases. This shows that the proposed approach for CSOGD is effective in improving the accuracy of predicting the examples from the rare class.

Third, similar to the previous results, the two CSOGD algorithms in general achieved comparable sum performance, in which CSOGD-I tends to perform slightly better than CSOGD-II.

C. Evaluation of Weighted Cost Performance

We further evaluate the performance of the CSOL$_{cos}$ algorithm in terms of the cost metric. The last three columns of Table 2 summarize the results of total cost evaluation. From the experimental results, we can also draw several observations.

First of all, we found that the two existing cost-sensitive algorithms (PAUM and CPA$_{PB}$) usually outperform the other cost-insensitive algorithms, in which PAUM seems to be more effective than CPA$_{PB}$ for most cases.

Second, among all the algorithms, we found that the proposed CSOGD algorithms achieve significantly less total misclassification *cost* than the other algorithms for most cases. For example, on the dataset "w8a", the total misclassification cost of CSOGD-II is about 20% less than that of PA algorithms, and about 10% less than that of PAUM.

Further, by examining both *sensitivity* and *specificity* metrics, we found that CSOGD often achieves the best *sensitivity* result, but does not always guarantee the best results for *specificity*. Finally, by examining the two CSOGD algorithms themselves, we found that CSOGD-II tends to perform sightly better than CSOGD-I (except on the dataset "german").

VI. CONCLUSION

As an attempt to fill the gap between cost-sensitive classification and online learning in machine learning and data mining, this paper investigated a new framework of Cost-Sensitive Online Classification, which aims to directly optimize cost-sensitive measures for online classification tasks. We proposed a family of effective algorithms based on online gradient descent, theoretically analyzed their cost-sensitive bounds, and finally examined their empirical performance extensively. Our encouraging results show that

the proposed algorithms considerably outperform the traditional online learning algorithms for cost-sensitive online classification tasks. Through this study, we hope to inspire researchers in both data mining and machine learning to further explore in-depth theory of cost-sensitive online classification and the application of new cost-sensitive online learning techniques to tackle a variety of emerging challenges in real-world data mining applications.

ACKNOWLEDGMENTS

This work was supported by Singapore MOE tier 1 project (RG33/11) and Microsoft Research project (M4060936).

REFERENCES

[1] R. Akbani, S. Kwek, and N. Japkowicz. Applying support vector machines to imbalanced datasets. In *ECML*, pages 39–50, 2004.

[2] P. L. Bartlett, E. Hazan, and A. Rakhlin. Adaptive online gradient descent. In *NIPS*, 2007.

[3] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann. The balanced accuracy and its posterior distribution. In *ICPR*, pages 3121–3124, 2010.

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *JAIR*, 16:321–357, 2002.

[5] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *JMLR*, 7:551–585, 2006.

[6] K. Crammer and Y. Singer. Ultraconservative online algorithms for multiclass problems. *JMLR*, 3:951–991, 2003.

[7] P. Domingos. Metacost: a general method for making classifiers cost-sensitive. In *KDD'99*, pages 155–164, San Diego, CA, USA, 1999. ACM.

[8] M. Dredze, K. Crammer, and F. Pereira. Confidence-weighted linear classification. In *ICML*, pages 264–271, 2008.

[9] C. Drummond and R. C. Holte. C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. In *ICML'03 Workshop on Learning from Imbalanced Data Sets*, pages 1–8, 2003.

[10] C. Elkan. The foundations of cost-sensitive learning. In *IJCAI*, pages 973–978, 2001.

[11] Y. Freund and R. E. Schapire. Large margin classification using the perceptron algorithm. *Mach. Learn.*, 37(3):277–296, 1999.

[12] C. Gentile. A new approximate maximal margin classification algorithm. *JMLR*, 2:213–242, 2001.

[13] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2000.

[14] J. Kivinen, A. J. Smola, and R. C. Williamson. Online learning with kernels. In *NIPS*, pages 785–792, 2001.

[15] Y. Li and P. M. Long. The relaxed online maximum margin algorithm. In *NIPS*, pages 498–504, 1999.

[16] Y. Li, H. Zaragoza, R. Herbrich, J. Shawe-Taylor, and J. S. Kandola. The perceptron algorithm with uneven margins. In *ICML*, pages 379–386, 2002.

[17] Y.-F. Li, J. T. Kwok, and Z.-H. Zhou. Cost-sensitive semi-supervised support vector machine. In *AAAI*, 2010.

[18] X.-Y. Liu and Z.-H. Zhou. The influence of class imbalance on cost-sensitive learning: An empirical study. In *Proceedings of the Sixth International Conference on Data Mining (ICDM'06)*, pages 970–974. IEEE Computer Society, 2006.

[19] A. C. Lozano and N. Abe. Multi-class cost-sensitive boosting with p-norm loss functions. In *KDD'08*, pages 506–514, Las Vegas, Nevada, USA, 2008. ACM.

[20] H. Masnadi-Shirazi and N. Vasconcelos. Risk minimization, probability elicitation, and cost-sensitive svms. In *ICML*, pages 759–766, 2010.

[21] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–407, 1958.

[22] M. Tan. Cost-sensitive learning of classification knowledge and its applications in robotics. *Mach. Learn.*, 13(1):7–33, Oct. 1993.

[23] P. D. Turney. Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. *JAIR*, 2:369–409, 1995.

[24] K. Veropoulos, C. Campbell, and N. Cristianini. Controlling the sensitivity of support vector machines. In *IJCAI*, pages 55–60, 1999.

[25] J. Wang, P. Zhao, and S. C. Hoi. Exact soft confidence-weighted learning. In *ICML*, 2012.

[26] B. Zadrozny, J. Langford, and N. Abe. Cost-sensitive learning by cost-proportionate example weighting. In *Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03)*, pages 435–, Washington, DC, USA, 2003.

[27] P. Zhao, S. C. H. Hoi, and R. Jin. Double updating online learning. *Journal of Machine Learning Research*, 12:1587–1615, 2011.

[28] P. Zhao, S. C. H. Hoi, R. Jin, and T. Yang. Online auc maximization. In *ICML*, pages 233–240, 2011.

[29] P. Zhao, J. Wang, P. Wu, R. Jin, and S. C. Hoi. Fast bounded online gradient descent algorithms for scalable kernel-based online learning. In *ICML*, 2012.

[30] X. Zhu and X. Wu. Class noise handling for effective cost-sensitive learning by cost-guided iterative classification filtering. *IEEE Trans. on Knowl. and Data Eng.*, 18(10):1435–1440, Oct. 2006.

[31] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In T. Fawcett and N. Mishra, editors, *ICML*, volume 20, 2003.