Cost-Sensitive Online Classification

Jialei Wang, Peilin Zhao, and Steven C.H. Hoi, *Member, IEEE*

Abstract—Both *cost-sensitive classification* and *online learning* have been extensively studied in data mining and machine learning communities, respectively. However, very limited study addresses 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 directly optimizing cost-sensitive measures using online gradient descent techniques. Specifically, we propose two novel 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. Finally, we demonstrate the application of the proposed technique for solving several online anomaly detection tasks, showing that the proposed technique could be a highly efficient and effective tool to tackle cost-sensitive online classification tasks in various application domains.

Index Terms—Cost-sensitive classification; online learning; online gradient descent; online anomaly detection

1 Introduction

In the era of big data, an urgent need in data mining and machine learning is to develop efficient and scalable algorithms for mining massive rapidly growing data. A promising direction is to investigate *Online Learning*, a family of efficient and scalable machine learning methods, which has been actively studied in literature [6], [20], [30]. 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. Various online learning methods have been actively proposed in literature [6], [20], [30]. Examples include the well-known Perceptron algorithm [14], [30], Passive-aggressive (PA) learning [6], and many other recently proposed algorithms [10], [15], [16], [19], [40], [47].

Despite being studied extensively, most existing online learning techniques are unsuitable for *cost-sensitive classification* tasks, an important problem for data mining which has to address varied misclassification costs [9], [12]. The existing online learning techniques potentially might not be effective enough primarily 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 diverse [5], [11], [29], [38].

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* [32] and the weighted *misclassification cost* [1], [12]. 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 [1], [12]. However, these batch classification algorithms often suffer poor efficiency and scalability when solving large-scale problems, which thus are 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 a comprehensive study to address this open challenge, 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 two cost-sensitive

- *S. C. H. Hoi is with the School of Information Systems, Singapore Management University, Singapore. E-mail: stevenhoi@gmail.com.*
- *P. Zhao and J. Wang are with the School of Computer Engineering, Nanyang Technological University, Singapore 639798. E-mail: {jl.wang, zhao0106}@ntu.edu.sg.*

Manuscript received 25 Nov. 2012; revised 23 Aug. 2013; accepted 25 Aug. 2013. Date of publication 26 Sep. 2013; date of current version 29 Aug. 2014. Recommended for acceptance by H. Zha.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier 10.1109/TKDE.2013.157

online learning algorithms using online gradient descent 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 for cost-sensitive online classification tasks; (ii) we apply the proposed technique to solve a data mining application, i.e., online anomaly detection tasks. We note that a short version of this journal had been presented in the ICDM'12 conference [39]. This journal manuscript has been significantly extended by including a substantial amount of new contents and results.

The rest of the paper is organized as follows. Section 2 briefs the related works. Section 3 formulates the problem and presents the proposed algorithms. Section 4 theoretically analyzes the bounds of the proposed algorithms. Section 5 discusses our experimental results. Section 6 shows an application to online anomaly detection tasks, and finally Section 7 concludes this work.

2 Related Work and Background

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

2.1 Cost-sensitive Classification

Cost-sensitive classification has been extensively studied in data mining and machine learning [13], [23], [25], [26], [42], [49], [50]. Many real-world classification problems, such as fraud detection and medical diagnosis, are naturally cost-sensitive. For these problems, the cost of misclassifying a target is much higher than that of a false-positive, and classifiers that are optimal under symmetric costs tend to under perform. 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* [32], and the weighted *misclassification cost* that takes cost into consideration when measuring classification performance [1], [12]. 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* [32], which is widely used in anomaly detection tasks. Over the past decades, various batch learning algorithms have been proposed for cost-sensitive classification in literature [9], [12], [22], [27], [29], [34], [36]. However, few studies emphasis the case when data arrives sequentially, except the Cost-sensitive Passive Aggressive(CPA) [6] and Perceptron Algorithms with Uneven Margin(PAUM) [21].

2.2 Online Learning

Online learning operates on a sequence of data examples with time stamps. At time step t, the algorithm processes an incoming example $\mathbf{x}_t \in \mathbb{R}^d$ by first predicting its label $\hat{y}_t \in \{-1, +1\}$. After the prediction, the true label $y_t \in \{-1, +1\}$ is revealed and then the loss $\ell(y_t, \hat{y}_t)$, which is the difference between its prediction and the revealed true label y_t, is suffered. Finally, the loss is used to update the weights of the model based on some criterion. Overall, the goal of online learning is to minimize the cumulative mistake over the entire sequence of data examples [17].

The most well-known online learning algorithm perhaps is Perceptron [30]. Specifically, whenever the online learner makes a wrong classification, the perceptron algorithm simply updates the classifier as follows:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + y_t \mathbf{x}_t.$$

Passive Aggressive (PA) learning [6] attempts to improve Perceptron by introducing the idea of margin maximization into the online learning framework. PA algorithms update the classifier whenever the online classifier does not produce a large margin on the current received example. Specifically, the loss of PA algorithms is based on the hinge loss: $\ell(\mathbf{w}_t; (\mathbf{x}_t, y_t)) = \max\{0, 1 - y_t(\mathbf{w}_t \cdot \mathbf{x}_t)\}$. The optimization of the PA learning is formulated as:

$$\begin{aligned} \mathbf{w}_{t+1} = \arg\min_{w \in \mathbb{R}^d} \frac{1}{2}\|\mathbf{w} - \mathbf{w}_t\|^2 \\ s.t. \quad \ell(\mathbf{w}; (\mathbf{x}_t, y_t)) = 0. \end{aligned}$$

The closed-form solution to the above is expressed as:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \eta_t y_t \mathbf{x}_t, \tag{1}$$

where the optimal value of parameter $\eta_t = \frac{\ell(\mathbf{w}_t; (\mathbf{x}_t, y_t))}{\|\mathbf{x}_t\|^2}$.

To further make PA being able to handle non-separable instances, one can introduce a slack variable ξ into the optimization problem in (1):

$$\begin{aligned} \mathbf{w}_{t+1} = \arg\min_{w \in \mathbb{R}^d} \frac{1}{2}\|\mathbf{w} - \mathbf{w}_t\|^2 + C\xi \\ s.t. \quad \ell(\mathbf{w}; (\mathbf{x}_t, y_t)) \le \xi \text{ and } \xi \ge 0. \end{aligned}$$

The solution to the above soft-margin problem shares the same form as that of (1), but with different coefficient η_t as follows:

$$\eta_t = \min\left\{C, \frac{\ell(\mathbf{w}_t; (\mathbf{x}_t, y_t))}{\|\mathbf{x}_t\|^2}\right\}.$$

The above two variants of PA algorithms are called "PA" and "PA-I", respectively.

Unlike traditional first-order online learning algorithms (e.g., Perceptron and PA), Confidence-Weighted (CW) online learning [7], [10] assumes the weight vector follows a Gaussian distribution and updates the mean and covariance of the distribution for each received example. Specifically, assume the weight vector $\mathbf{w}_t$ has the mean vector $\boldsymbol{\mu} \in \mathbb{R}^d$ and covariance matrix $\Sigma \in \mathbb{R}^{d \times d}$, the CW learning performs the distribution update by minimizing the Kullback-Leibler divergence between the distributions of the new and old weight vectors, and meanwhile ensuring that the probability of a correct classification on the training instance is large enough. Adaptive Regularization of Weights Learning (AROW) [8] was proposed to overcome this limitation.

However, to the best of our knowledge, very few existing work in this area had attempted to directly optimize the two cost-sensitive metrics in an online learning setting, except [43] which is based on online Naive Bayes approach. The work in [43] assumes that variables are independent with each other, which is not suitable for some applications and lacks theoretical guarantee. Also we note that our work is very different from another recent online learning study [48], which aims to optimize AUC, but cannot be guaranteed to optimize the cost-sensitive measures in our study. Finally, we note that this work is focused on investigating online learning methodology for learning linear models, and thus exclude the direct comparison to other nonlinear online learning methods [16], [19], [45], [47].

2.3 Anomaly Detection

Anomaly detection, also referred to as outlier detection or novelty detection, aims to find abnormal patterns ("anomalies") in data that do not accord with normal patterns/expected behaviors. It has been extensively studied over the past decades in a variety of research areas and application domains [4]. Anomaly detection techniques have been widely applied to tackle problems in a wide range of real-world applications [4], such as detection of credit card fraud transactions, network intrusion detection, detection of abnormal jet engine operation, detection of malignant tumors from medical images, and so on.

In literature, a variety of techniques have been proposed to solve anomaly detection in different application domains [3], [31], [35]. One major category of techniques formulates anomaly detection as a classical supervised classification task by training a binary classification model in a batch/offline learning fashion to distinguish between anomalies and normal patterns. These techniques usually require to collect a considerable amount of training data in order to build a good classification model for anomaly detection. In contrast, another category of techniques formulates it as an online unsupervised/semi-supervised learning task to detect anomalies without requiring label information of anomalies [24], [28], [33]. These techniques however may suffer from poor detection performance without exploring any label/supervised information.

Although anomaly detection has been well studied for a few decades, it remains a very challenging research problem today, which is primarily due to several reasons. First of all, it is often a highly class-imbalanced learning problem as the number of anomalies is significantly smaller than that of normal patterns, which brings a critical challenge to many schemes using regular classification techniques. Second, it is usually very expensive to collect labeled data, especially the positive training data ("anomalies"), which limits the application of some classical supervised classification approaches. Moreover, in a real-world application, data usually arrives in a sequential/online fashion and the size of data patterns can be very large, leading to a big challenge for developing efficient and scalable algorithms for anomaly detection.

3 Cost-Sensitive Online Classification

In this section, we present our proposed Cost-Sensitive Online Classification(CSOC) framework, we first introduce the problem formulation and then present the proposed algorithms.

3.1 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:

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

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

$$sensitivity = \frac{T_p - M_p}{T_p}, \quad specificity = \frac{T_n - M_n}{T_n},$$
$$accuracy = \frac{T - M}{T}.$$

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.,

$$sum = \eta_p \times sensitivity + \eta_n \times specificity, \tag{2}$$

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:

$$cost = c_p \times M_p + c_n \times M_n, \tag{3}$$

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.

3.2 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 (2) or minimizing the weighted cost in (3) is equivalent to minimizing the following objective:*

$$\sum_{y_t=+1} \rho \mathrm{I}_{(y_t \mathbf{w}\cdot\mathbf{x}_t<0)} + \sum_{y_t=-1} \mathrm{I}_{(y_t \mathbf{w}\cdot\mathbf{x}_t<0)} \tag{4}$$

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.

Proof. Firstly, by analyzing the function of the weighted sum in (2), we can derive the following:

$$\begin{aligned} sum &= \eta_p \frac{T_p - M_p}{T_p} + \eta_n \frac{T_n - M_n}{T_n} \\ &= 1 - \frac{\eta_n}{T_n}\Big[\frac{\eta_p T_n}{\eta_n T_p} \sum_{y_t=+1} \mathrm{I}_{(y_t \mathbf{w}\cdot\mathbf{x}_t<0)} + \sum_{y_t=-1} \mathrm{I}_{(y_t \mathbf{w}\cdot\mathbf{x}_t<0)}\Big], \end{aligned}$$

where I_π is the indicator function that outputs 1 if the statement π holds and 0 otherwise. Thus, maximizing *sum* is equivalent to minimizing

$$\frac{\eta_p T_n}{\eta_n T_p} \sum_{y_t=+1} \mathrm{I}_{(y_t \mathbf{w}\cdot\mathbf{x}_t<0)} + \sum_{y_t=-1} \mathrm{I}_{(y_t \mathbf{w}\cdot\mathbf{x}_t<0)}.$$

Secondly, by analyzing the function of the weighted cost in (3), we can also derive the following:

$$\begin{aligned} cost &= c_p M_p + c_n M_n \\ &= c_n \Big[\frac{c_p}{c_n} \sum_{y_t=+1} \mathrm{I}_{(y_t \mathbf{w}\cdot\mathbf{x}_t<0)} + \sum_{y_t=-1} \mathrm{I}_{(y_t \mathbf{w}\cdot\mathbf{x}_t<0)}\Big]. \end{aligned}$$

Thus, minimizing *cost* is equivalent to minimizing

$$\frac{c_p}{c_n} \sum_{y_t=+1} \mathrm{I}_{(y_t \mathbf{w}\cdot\mathbf{x}_t<0)} + \sum_{y_t=-1} \mathrm{I}_{(y_t \mathbf{w}\cdot\mathbf{x}_t<0)}.$$

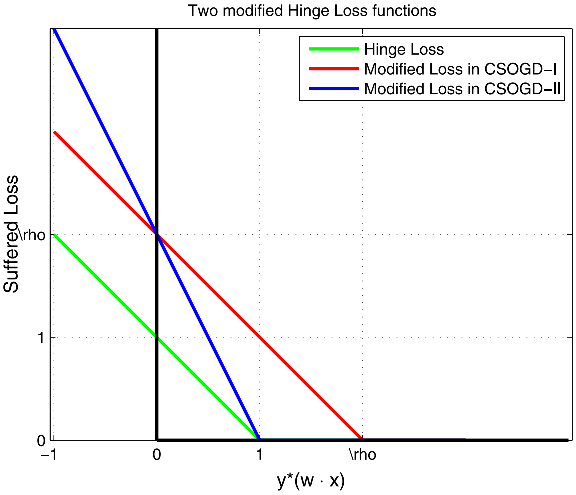


Fig. 1. Illustration of the modified hinge loss functions for CSOGD, where the value of ρ is set to 2.

Thus, the proposition holds by setting $\rho = \frac{\eta_p T_n}{\eta_n T_p}$ for sum, and $\rho = \frac{c_p}{c_n}$ for 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:

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

$$\ell^{II}(\mathbf{w}; (\mathbf{x}, y)) = (\rho * \mathrm{I}_{(y=1)} + \mathrm{I}_{(y=-1)}) * \max(0, 1 - y(\mathbf{w}\cdot\mathbf{x})). \tag{6}$$

We could see that for $\ell^I(\mathbf{w}; (\mathbf{x}, y))$, the required margin for specific class changed compared to the traditional hinge loss, cause to more "frequent" updating; while for $\ell^{II}(\mathbf{w}; (\mathbf{x}, y))$, the slope of the loss function changed for specific class, leading to more "aggressive" updating. Fig. 1 illustrates the differences of the modified hinge loss functions.

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

$$\mathcal{F}_T^*(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{t=1}^{T} \ell^*(\mathbf{w}; (\mathbf{x}_t, y_t)) \text{ here } * \in \{I, I\!I\}, \tag{7}$$

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 (7). To this end, we propose to solve the problem using the online gradient descent approach [51] as follows:

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

where λ is a learning rate parameter and $\ell_t(\mathbf{w}) = \ell^*(\mathbf{w}; (x_t, y_t))$, $\forall * \in \{I, I\!I\}$. Specifically, when using the loss function (5), the update rule can be expressed as:

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

Algorithm 1 The proposed CSOGD algorithms.

```
INPUT: learning rate $\lambda$; 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: $\mathbf{w}_{T+1}$.
```

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

When using the loss function (6), 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. It is clear that the overall time complexity of the algorithm is $\mathcal{O}(T \times n)$, which is linear with respect to the total number of received instances T and the dimensionality of the data n.

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, one may argue the values of T_n and T_p might be unknown in a real-world online classification task. To address this issue, a practical yet fairly effective approach is to estimate the ratio $\frac{T_n}{T_p}$ according to the distribution of online received training data instances over the historical sequence, and adaptively update this ratio during the online learning process. We will empirically examine this issue in the experimental section.

4 Theoretical 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 prove the following lemma that gives the loss bound achieved by the online learning algorithm to facilitate subsequent theoretical analysis, which was inspired by the work in [51].

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 $\|\mathbf{x}_t\| \le 1$ for all t. Then for any $\mathbf{w} \in \mathbb{R}^n$, by setting $\lambda = \frac{\|w\|}{\sqrt{S_p + S_n}}$, the following holds for CSOGD-I:*

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

and by setting $\lambda = \frac{\|w\|}{\sqrt{\rho^2 S_p + S_n}}$, the following holds for CSOGD-II:

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

Proof.

$$\begin{aligned} \|\mathbf{w}_{t+1} - \mathbf{w}\|^2 &= \|\mathbf{w}_t - \lambda \nabla \ell_t(\mathbf{w}_t) - \mathbf{w}\|^2 \\ &= \|\mathbf{w}_t - \mathbf{w}\|^2 + \lambda^2 \|\nabla \ell_t(\mathbf{w}_t)\|^2 \\ &\quad - 2\lambda \nabla \ell_t(\mathbf{w}_t)(\mathbf{w}_t - \mathbf{w}). \end{aligned}$$

For the convexity of the loss function,

$$\ell_t(\mathbf{w}_t) - \ell_t(\mathbf{w}) \le \nabla \ell_t(\mathbf{w}_t)(\mathbf{w}_t - \mathbf{w}).$$

We have the following:

$$\ell_t(\mathbf{w}_t) - \ell_t(\mathbf{w}) \le \frac{\|\mathbf{w}_t - \mathbf{w}\|^2 - \|\mathbf{w}_{t+1} - \mathbf{w}\|^2}{2\lambda} + \frac{\lambda}{2} \|\nabla \ell_t(\mathbf{w}_t)\|^2.$$

Summing over $t = 1, \ldots T$,

$$\begin{aligned} &\sum_{t=1}^{T} (\ell_t(\mathbf{w}_t) - \ell_t(\mathbf{w})) \\ &\le \frac{\|\mathbf{w}_1 - \mathbf{w}\|^2 - \|\mathbf{w}_{T+1} - \mathbf{w}\|^2}{2\lambda} + \frac{\lambda}{2} \sum_{t=1}^{T} \|\nabla \ell_t(\mathbf{w}_t)\|^2 \\ &\le \frac{\|\mathbf{w}\|^2}{2\lambda} + \frac{\lambda}{2} \sum_{t=1}^{T} \|\nabla \ell_t(\mathbf{w}_t)\|^2. \end{aligned}$$

For CSOGD-I, $\|\nabla \ell_t(\mathbf{w}_t)\| \le 1$ if $t \in \mathcal{S}$ and $\|\nabla \ell_t(\mathbf{w}_t)\| = 0$ otherwise. Thus,

$$\frac{\|\mathbf{w}\|^2}{2\lambda} + \frac{\lambda}{2} \sum_{t=1}^{T} \|\nabla \ell_t(\mathbf{w}_t)\|^2 \le \frac{\|\mathbf{w}\|^2}{2\lambda} + \frac{\lambda(S_p + S_n)}{2}.$$

We can obtain the bound by setting $\lambda = \frac{\|\mathbf{w}\|}{\sqrt{S_p + S_n}}$.

For CSOGD-II, $\|\nabla \ell_t(\mathbf{w}_t)\| \le 1$ if $t \in \mathcal{S}_n$ and $\|\nabla \ell_t(\mathbf{w}_t)\| \le \rho$ if $t \in \mathcal{S}_p$ and $\|\nabla \ell_t(\mathbf{w}_t)\| = 0$ otherwise. So,

$$\frac{\|\mathbf{w}\|^2}{2\lambda} + \frac{\lambda}{2} \sum_{t=1}^{T} \|\nabla \ell_t(\mathbf{w}_t)\|^2 \le \frac{\|\mathbf{w}\|^2}{2\lambda} + \frac{\lambda(\rho^2 S_p + S_n)}{2}.$$

We can obtain the bound by setting $\lambda = \frac{\|\mathbf{w}\|}{\sqrt{\rho^2 S_p + S_n}}$. □

Remark. Firstly, because $S_p + S_n \le T$, we get a regret bound at most achieving $\sqrt{T}$ regret. Secondly, although when $\rho > 1$, CSOGD-I obtains a better bound than CSOGD-II on mathematical formulation (CSOGD-II has a constant ρ), since CSOGD-I has a more passive margin on positive examples, the number of support vectors should be larger than CSOGD-II. Finally, we could further improve the bounds by introducing strong convexity with regularization and adaptive learning rate, however it is not our main goal, so we just keep a constant learning rate here for simplicity.

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* $\|\mathbf{x}_t\| \le 1$ *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:*

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

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

Proof. For these two algorithms, if $t \in \mathcal{M}_p$, $\ell_t(\mathbf{w}_t) \ge \rho$, and if $t \in \mathcal{M}_n$, $\ell_t(\mathbf{w}_t) \ge 1$. Thus, we have

$$\rho M_p + M_n \le \sum_{t=1}^{T} \ell_t(\mathbf{w}_t). \tag{8}$$

From the definition of *sum*, we know that

$$sum = 1 - \frac{\eta_n}{T_n}\Big[\frac{\eta_p T_n}{\eta_n T_p} \sum_{y_t=+1} \mathrm{I}_{(y_t \mathbf{w}\cdot\mathbf{x}_t < 0)} + \sum_{y_t=-1} \mathrm{I}_{(y_t \mathbf{w}\cdot\mathbf{x}_t < 0)}\Big]$$
$$= 1 - \frac{\eta_n}{T_n}\left(\frac{\eta_p T_n}{\eta_n T_p} M_p + M_n\right)$$

letting $\rho = \frac{\eta_p T_n}{\eta_n T_p}$, and from Lemma 1 we know that for CSOGD-I

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

and for CSOGD-II

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

Combining above inequalities proves our conclusion. □

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* $\|\mathbf{x}_t\| \le 1$ *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:*

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

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

TABLE 1
List of Binary Datasets in Our Experiments

dataset	#Examples	#Features	#Pos:#Neg
covtype	581012	54	1:1
spambase	4601	57	1:1.5
german	1000	24	1:2.3
svmguide3	1243	21	1:3
a9a	48842	123	1:3.2
w8a	64700	300	1:32.5

Proof. From the definition of *cost*, we know that

$$cost = c_n\Big[\frac{c_p}{c_n} \sum_{y_t=+1} \mathrm{I}_{(y_t \mathbf{w}\cdot\mathbf{x}_t < 0)} + \sum_{y_t=-1} \mathrm{I}_{(y_t \mathbf{w}\cdot\mathbf{x}_t < 0)}\Big]$$
$$= c_n\left(\frac{c_p}{c_n} M_p + M_n\right).$$

Setting $\rho = \frac{c_p}{c_n}$, and combining it with (8), we have

$$c_n(\rho M_p + M_n) \le c_n \sum_{t=1}^{T} \ell_t(\mathbf{w}_t).$$

Combining the above inequality with Lemma 1 can easily prove this theorem. □

5 Experiments

This section aims to evaluate the empirical performance of the proposed algorithms (CSOGD-I and CSOGD-II) for cost-sensitive online classification tasks. To ease our discussions, we denote by CSOC$_{sum}$ the proposed CSOC algorithm for maximizing the weighted sum of sensitivity and specificity, and CSOC$_{cos}$ the proposed CSOC algorithm for minimizing the misclassification cost. The data sets and implementations of this work can be found in our project website http://CSOC.stevenhoi.org/.

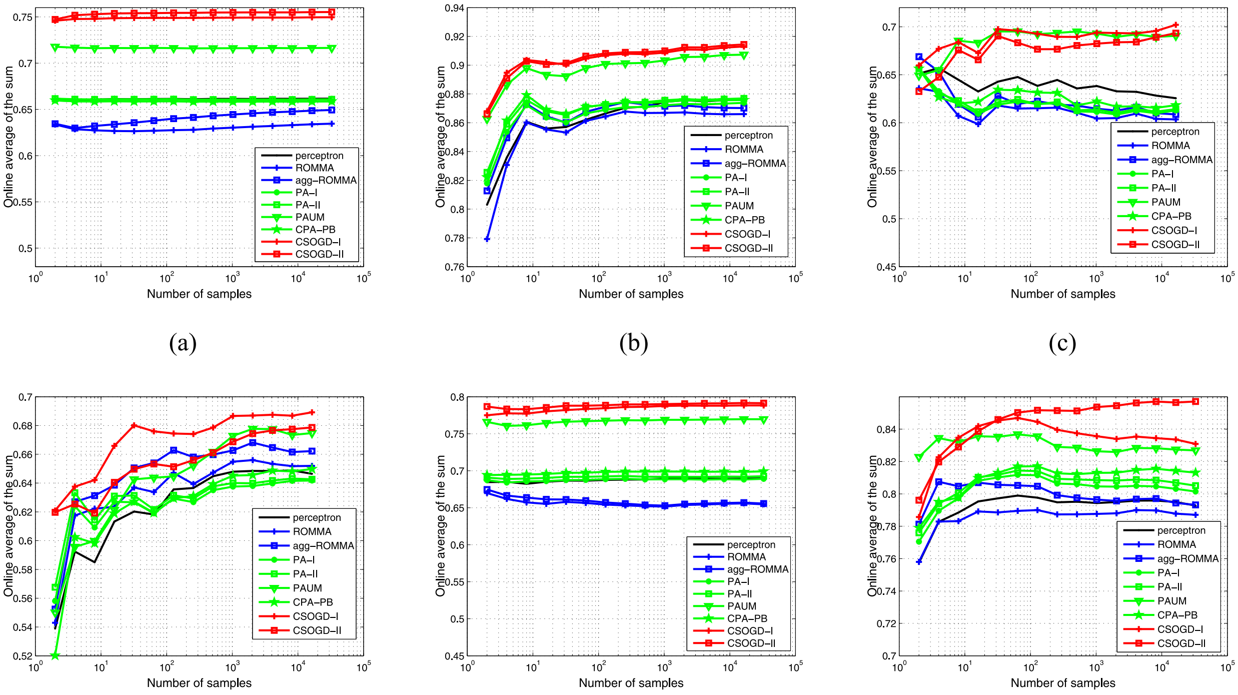
5.1 Experimental Testbed and Setup

We compare our CSOGD algorithms with various state-of-the-art online learning algorithms [17], including Perceptron, "ROMMA" and its aggressive version "agg-ROMMA", and two versions of the PA algorithms [6], i.e., PA-I and PA-II. We also compare with two existing cost-sensitive online algorithms: prediction-based PA algorithm ('CPA$_{PB}$') [6] and the perceptron algorithm with uneven margin ('PAUM') [21].

To examine the performance, we test all the algorithms on various benchmark datasets from web machine learning repositories. For space limitation, we randomly choose a few for discussion, as listed in Table 1. All of them can be downloaded from LIBSVM website[1].

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 CSOC$_{sum}$ algorithms, we set $\eta_p = \eta_n = 1/2$ for all cases, while for CSOC$_{cos}$, 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

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



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 2. Evaluation of online "sum" performance of the proposed CSOC$_{sum}$ algorithms on class-imbalance datasets. (a) *covtype*. (b) *spambase*. (c) *german*. (d) *svmguide3*. (e) *a9a*. (f) *w8a*.

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 CSOC$_{cos}$ and $\frac{\eta_p T_n}{\eta_n T_p}$ for CSOC$_{sum}$, respectively. We also evaluate the parameter sensitivity about the cost-sensitive weights in our experiments. All the algorithms were implemented in MATLAB and run in a Windows machine with 2.33GHz.

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**.

5.2 Evaluation of Weighted Sum Performance

We first evaluate the weighted sum performance. The first three columns of Table 2 summarize the results, and Fig. 2 shows the changes of online average *sum* performance. 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, 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.

Finally, from Fig. 1, we observe that the CSOGD algorithms consistently outperform the other algorithms in the entire online learning process.

5.2.1 Evaluation of Online Estimation of $\frac{T_n}{T_p}$

In our previous theoretical analysis section, we the parameter ρ to set as $\frac{\eta_p T_n}{\eta_n T_p}$ for CSOC$_{sum}$ algorithms. However, the value of $\frac{T_n}{T_p}$ is not always known in advance for online learning. In this section, we evaluate the performance of online estimation of $\frac{T_n}{T_p}$ compared with the original algorithm. We adopt a widely used laplace estimation which use $\frac{t_n+1}{t_p+1}$ to estimate $\frac{T_n}{T_p}$, where t_n and t_p are the number of received negative instances and positive instances at time t, respectively. Fig. 3 shows the performance of the online estimation, we can see that the online estimation approach performs very similar to the original approach, which validates the practical value of the CSOC$_{sum}$ algorithms.

5.3 Evaluation of Weighted Cost Performance

We further evaluate the performance of the CSOC$_{cos}$ algorithm in terms of the cost metric. The last three columns of Table 2 summarize the results of total cost evaluation, and Fig. 4 illustrates the changes of online average cost at each period. From the results, we can also draw several observations.

First, we found that the two existing cost-sensitive algorithms (PAUM and CPA$_{PB}$) usually outperform the other

TABLE 2
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	71.645 ± 0.010	73.277 ± 0.002	70.014 ± 0.023	76384.325 ± 3.359	73.277 ± 0.002	70.014 ± 0.023
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 spambase			”cost” on spambase		
	Sum(%)	Sensitivity(%)	Specificity (%)	Cost	Sensitivity(%)	Specificity (%)
Perceptron	87.349 ± 0.335	87.675 ± 0.533	87.023 ± 0.264	235.683 ± 5.838	87.372 ± 0.333	86.958 ± 0.355
ROMMA	86.343 ± 0.334	87.606 ± 0.772	85.081 ± 0.680	236.985 ± 13.553	87.463 ± 0.801	84.900 ± 0.688
agg-ROMMA	86.990 ± 0.359	87.794 ± 0.598	86.187 ± 0.462	232.582 ± 14.607	87.623 ± 0.841	86.081 ± 0.630
PA-I	87.515 ± 0.362	87.416 ± 0.390	87.615 ± 0.502	238.428 ± 7.658	87.154 ± 0.444	87.681 ± 0.403
PA-II	87.744 ± 0.373	87.601 ± 0.436	87.887 ± 0.485	233.422 ± 7.301	87.421 ± 0.429	87.966 ± 0.427
PAUM	89.916 ± 0.274	88.414 ± 0.511	91.417 ± 0.347	119.077 ± 5.902	94.788 ± 0.332	78.980 ± 0.452
CPA$_{PB}$	90.843 ± 0.155	91.065 ± 0.234	90.621 ± 0.076	164.800 ± 11.243	91.202 ± 0.663	90.477 ± 0.127
CSOGD-I	91.460 ± 0.177	91.219 ± 0.334	91.700 ± 0.294	163.790 ± 4.820	91.462 ± 0.289	87.999 ± 0.283
CSOGD-II	**91.473 ± 0.166**	91.577 ± 0.244	91.368 ± 0.315	**86.235 ± 3.652**	97.579 ± 0.222	68.056 ± 0.919

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	69.560 ± 0.657	75.333 ± 1.414	63.786 ± 0.101	82.975 ± 3.995	75.333 ± 1.414	63.786 ± 0.101
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 svmguide3			”cost” on svmguide3		
	Sum(%)	Sensitivity(%)	Specificity (%)	Cost	Sensitivity(%)	Specificity (%)
Perceptron	64.827 ± 0.598	60.980 ± 1.290	68.675 ± 1.148	124.558 ± 3.375	60.980 ± 1.290	68.675 ± 1.148
ROMMA	64.836 ± 1.484	59.831 ± 2.762	69.842 ± 2.680	127.235 ± 7.344	59.831 ± 2.762	69.842 ± 2.680
agg-ROMMA	65.264 ± 1.404	60.270 ± 2.776	70.259 ± 2.381	125.802 ± 7.408	60.270 ± 2.776	70.259 ± 2.381
PA-I	64.215 ± 0.983	60.220 ± 1.550	68.210 ± 1.056	126.915 ± 4.438	60.220 ± 1.550	68.210 ± 1.056
PA-II	64.507 ± 1.107	60.541 ± 1.894	68.474 ± 1.061	125.888 ± 5.373	60.541 ± 1.894	68.474 ± 1.061
PAUM	68.014 ± 0.709	61.318 ± 1.194	74.710 ± 0.224	120.750 ± 3.465	61.318 ± 1.194	74.710 ± 0.224
CPA$_{PB}$	64.106 ± 1.035	61.976 ± 1.796	66.235 ± 1.138	120.290 ± 4.978	63.345 ± 1.783	63.643 ± 1.434
CSOGD-I	**69.090 ± 0.743**	63.345 ± 1.410	74.836 ± 0.889	115.520 ± 4.231	63.176 ± 1.511	74.720 ± 0.774
CSOGD-II	68.654 ± 0.687	69.848 ± 1.462	67.460 ± 1.231	**93.523 ± 6.051**	74.730 ± 2.166	52.561 ± 1.944

Algorithm	”sum” on a9a			”cost” on a9a		
	Sum(%)	Sensitivity(%)	Specificity (%)	Cost	Sensitivity(%)	Specificity (%)
Perceptron	68.934 ± 0.180	69.277 ± 0.221	68.590 ± 0.290	3988.918 ± 25.498	69.308 ± 0.245	68.711 ± 0.272
ROMMA	64.835 ± 1.028	75.709 ± 3.202	53.962 ± 5.170	3577.020 ± 429.139	75.421 ± 5.299	54.351 ± 8.594
agg-ROMMA	65.171 ± 0.847	75.690 ± 2.864	54.653 ± 4.476	3538.443 ± 376.953	75.705 ± 4.682	54.730 ± 7.738
PA-I	68.958 ± 0.188	70.940 ± 0.283	66.976 ± 0.209	3850.168 ± 35.910	70.830 ± 0.351	67.083 ± 0.290
PA-II	69.286 ± 0.168	71.327 ± 0.272	67.245 ± 0.204	3802.995 ± 32.872	71.216 ± 0.321	67.316 ± 0.262
PAUM	76.766 ± 0.183	67.982 ± 0.373	85.551 ± 0.114	2445.478 ± 24.051	81.447 ± 0.219	79.244 ± 0.072
CPA$_{PB}$	69.864 ± 0.187	73.860 ± 0.267	65.867 ± 0.222	3289.655 ± 29.664	76.313 ± 0.274	64.488 ± 0.222
CSOGD-I	78.878 ± 0.101	85.670 ± 0.202	72.085 ± 0.103	1730.895 ± 16.698	89.446 ± 0.166	69.901 ± 0.139
CSOGD-II	**79.130 ± 0.059**	91.234 ± 0.164	67.026 ± 0.163	**1385.223 ± 17.186**	94.527 ± 0.143	58.143 ± 0.273

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	82.685 ± 0.396	67.822 ± 0.878	97.549 ± 0.087	667.825 ± 13.400	67.822 ± 0.878	97.549 ± 0.087
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

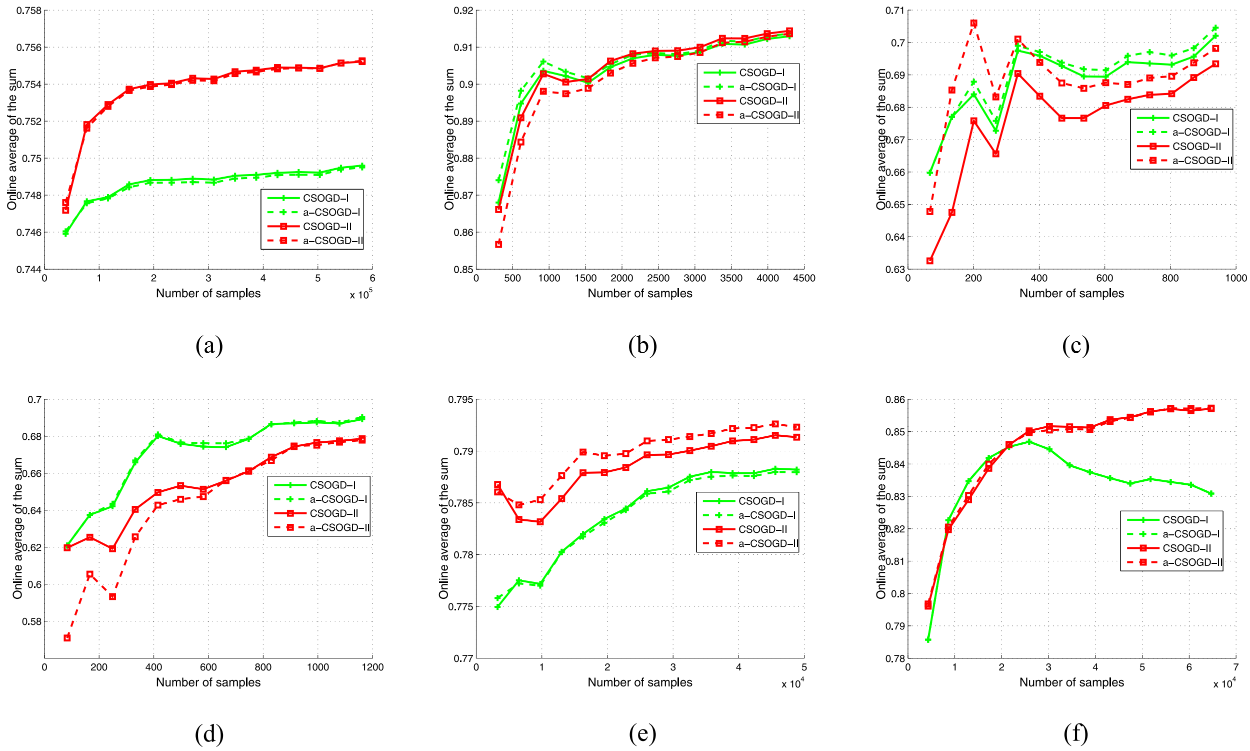


Fig. 3. Evaluation of performance impact using the online estimation of $\frac{T_n}{T_p}$. (a) *covtype*. (b) *spambase*. (c) *german*. (d) *svmguide3*. (e) *a9a*. (f) *w8a*.

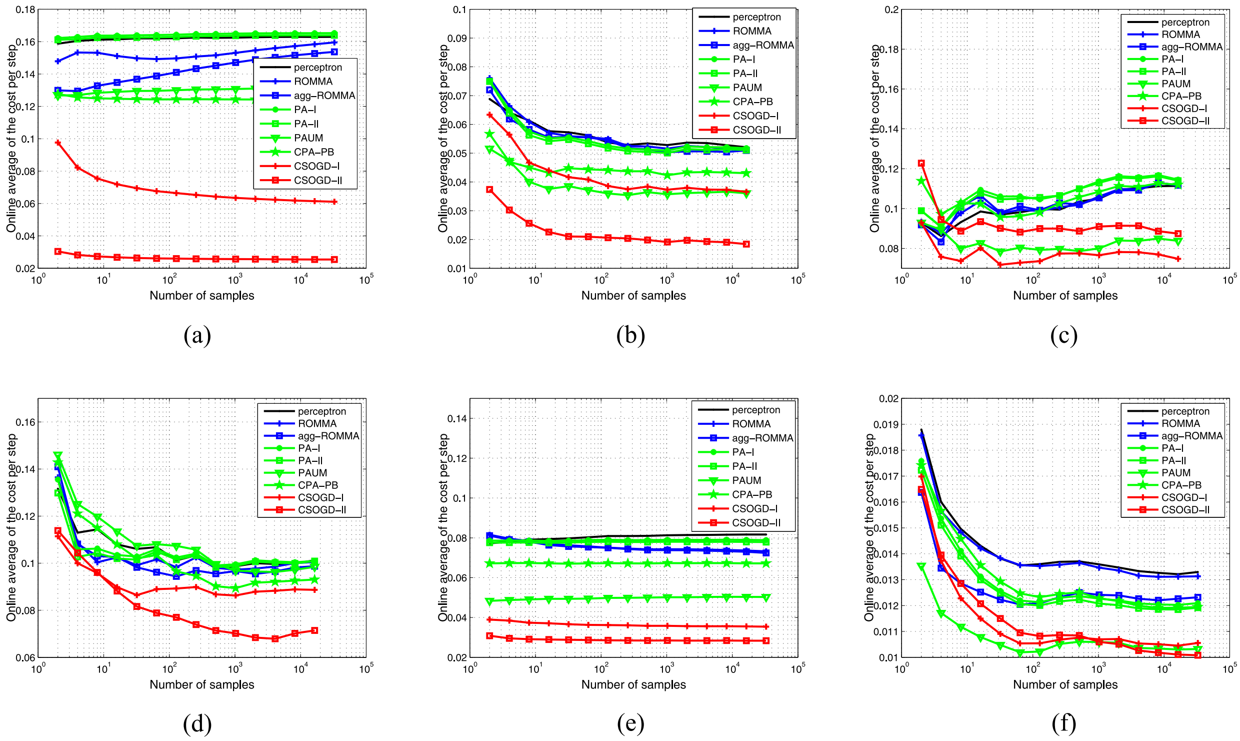


Fig. 4. Evaluation of online average "cost" of the proposed CSOC$_{cos}$ algorithms on class-imbalance datasets. (a) *covtype*. (b) *spambase*. (c) *german*. (d) *svmguide3*. (e) *a9a*. (f) *w8a*.

non-cost-sensitive algorithms, in which PAUM seems to be more effective than CPA$_{PB}$ for most cases.

Second, among all the algorithms, we found that the CSOGD algorithms achieve significantly less total misclassification *cost* than the others for most cases. For example, on "a9a", the total misclassification cost made by CSOGD-II is about one-third of those made by PA algorithms, and half of that made by 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

TABLE 3
Evaluation of Time Efficiency of Various Online Algorithms (seconds)

Algorithm	german	spambase	svmguide3	a9a	w8a	covtype
Perceptron	0.009	0.037	0.010	0.587	1.154	5.724
ROMMA	0.019	0.078	0.022	1.169	1.699	11.849
agg-ROMMA	0.020	0.082	0.024	1.284	2.108	13.650
PA-I	0.017	0.057	0.020	0.991	1.665	10.112
PA-II	0.017	0.058	0.020	0.999	1.658	10.182
PAUM	0.009	0.040	0.011	0.603	1.149	6.030
CPA$_{PB}$	0.019	0.068	0.022	1.094	1.845	11.656
CSOGD-I	0.009	0.038	0.011	0.581	1.152	5.880
CSOGD-II	0.009	0.038	0.011	0.601	1.170	5.848

algorithms themselves, we found that CSOGD-II tends to perform sightly better than CSOGD-I (except on the dataset "german").

5.4 Evaluation of Time Efficiency

In this subsection we evaluate the time efficiency of our proposed CSOGD methods compared with other online learning algorithms. Table 3 shows the results. We can see that the CSOGD algorithms are generally very efficient as other online learning approaches. For example, on "covtype" dataset which contains more than 500,000 data instances, CSOGD algorithms only require less than 6 seconds to finish the whole online learning processes in a regular computing machine.

5.5 Evaluation with Varied Cost-Sensitive Weights

In the previous experiments, the weights in both "cost" and "sum" metrics are fixed, which usually can be chosen empirically by different approaches. Despite the promising results achieved in the previous experiments, it is unknown how the algorithms are affected by different cost-sensitive weights. In this section, we aim to evaluate the performance of the proposed algorithms under varying cost-sensitive weights for both metrics.

Fig. 5 shows the evaluation results of the weighted sum performance under varying weights of η_n, and Fig. 6 shows the evaluation results of the weighted cost under varying weights of c_n. From the results, it is clear to see that the proposed algorithms consistently outperform most of the other algorithms for both metrics under varying settings of the weight values. These promising results further validate the efficacy of the proposed algorithms.

5.6 Evaluation of Parameter Sensitivity

We also examine the parameter sensitivity of the learning rate parameter λ. In particular, we set the learning rate as a factor in $[2^{-4}, 2^{-3}, \ldots, 2^{4}]$ times the original learning rate used in above section, as report the performance under the varied learning rate settings. Fig. 7 shows the evaluation results. We observe that our algorithms perform quite well on a relatively large parameter space of the learning rate.

TABLE 4
Evaluation of Generalization Ability for the CSOGD Algorithms

Algorithm	w8a	a9a	covtype	german	spambase	svmguide3
Perceptron	76.973 ± 1.121	67.469 ± 5.979	67.350 ± 1.914	64.892 ± 1.567	86.502 ± 3.006	64.956 ± 3.426
ROMMA	78.476 ± 0.809	62.233 ± 0.565	63.337 ± 4.958	60.635 ± 3.390	87.907 ± 2.372	65.670 ± 7.949
agg-ROMMA	79.293 ± 0.994	62.065 ± 0.023	67.001 ± 2.129	60.173 ± 3.276	88.303 ± 2.652	64.385 ± 7.202
PA-I	79.326 ± 0.412	66.778 ± 0.469	67.783 ± 1.125	65.273 ± 2.510	86.635 ± 2.669	64.285 ± 7.378
PA-II	79.422 ± 0.510	67.184 ± 0.261	67.993 ± 1.161	65.437 ± 1.874	86.974 ± 2.681	64.212 ± 7.574
PAUM	80.348 ± 0.360	75.918 ± 1.146	70.938 ± 0.028	68.418 ± 1.497	91.705 ± 0.923	67.859 ± 2.803
CPA$_{PB}$	81.180 ± 0.581	68.259 ± 0.172	67.799 ± 1.131	65.030 ± 1.708	86.871 ± 2.226	62.641 ± 7.911
CSOGD-I	82.154 ± 0.355	78.229 ± 0.023	75.638 ± 0.315	71.256 ± 1.120	**92.133 ± 0.595**	**70.835 ± 2.984**
CSOGD-II	**85.695 ± 0.996**	**79.528 ± 0.463**	**75.847 ± 0.054**	**71.538 ± 0.855**	92.022 ± 0.036	70.090 ± 0.649

5.7 Evaluation of Generalization Ability

Finally, we examine the generalization ability of our algorithms, which could be an issue when converting an online algorithm to batch training purposes. We use half of the data set for training, and the rest for test. Table 4 summarizes the results, in which we found that our algorithms still achieved the best, indicating that our CSOC algorithms could be potentially a useful tool for training large-scale cost-sensitive models.

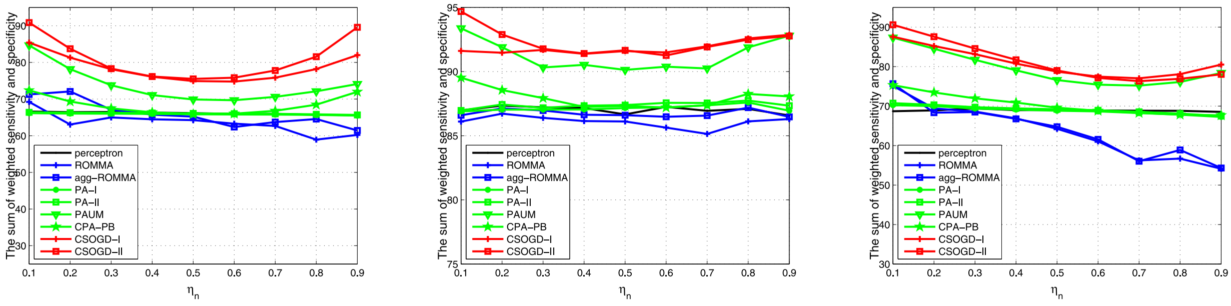
6 Application to Online Anomaly Detection

The proposed cost-sensitive online classification technique can be potentially applied to a wide range of real-world applications in data mining. In this section, we demonstrate an application of the proposed cost-sensitive online classification algorithms to tackle online anomaly detection tasks. Below we first introduce the related application domains, and then show our empirical evaluation results.

6.1 Application Domains and Testbeds

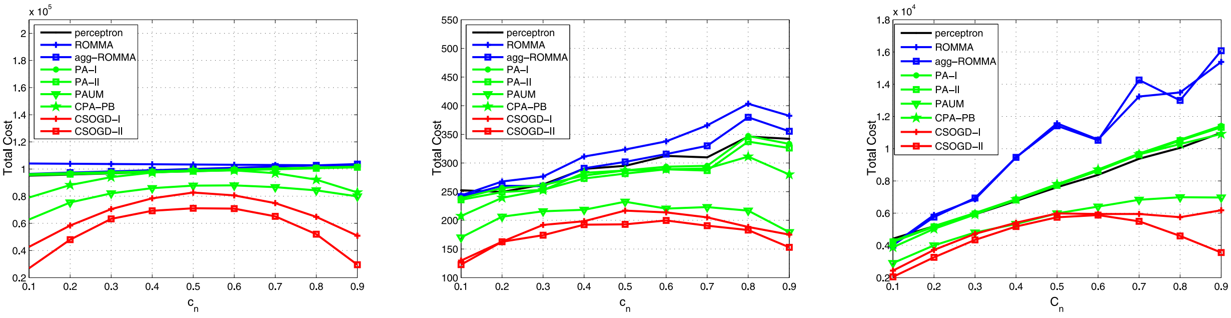
We address problems in the following domains:

- Bioinformatics: This is an anomaly detection task with the "Code-RNA" dataset [37]. The goal is to develop a computational method to detect novel non-coding RNAs from some large sequenced



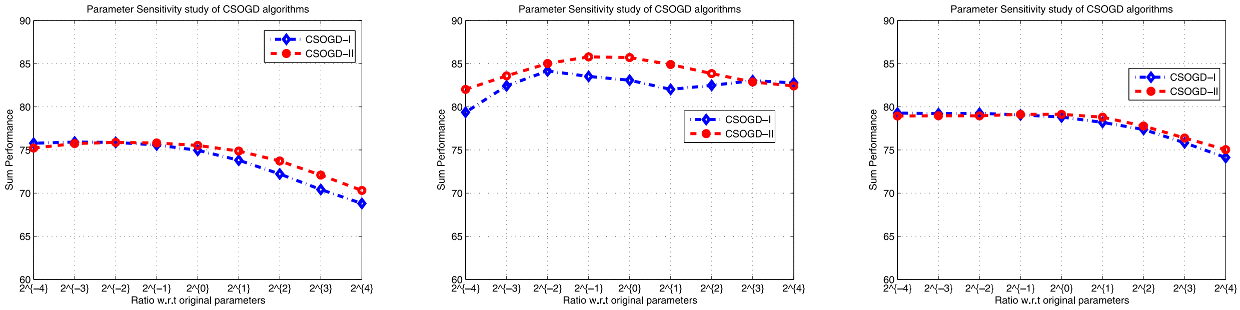
(a) (b) (c)

Fig. 5. Evaluation of the weighted “sum” under varying weights of sensitivity and specificity. (a) *covtype*. (b) *spambase*. (c) *a9a*.



(a) (b) (c)

Fig. 6. Evaluation of weighted “cost” measure under varying weights for FP and FN. (a) *covtype*. (b) *spambase*. (c) *a9a*.



(a) (b) (c)

Fig. 7. Evaluation of parameter sensitivity of the learning rate parameter in the proposed CSOGD algorithms. (a) *covtype*. (b) *w8a*. (c) *a9a*.

genomes. Non-coding RNAs are defined as anomalies and others are considered as normal instances.

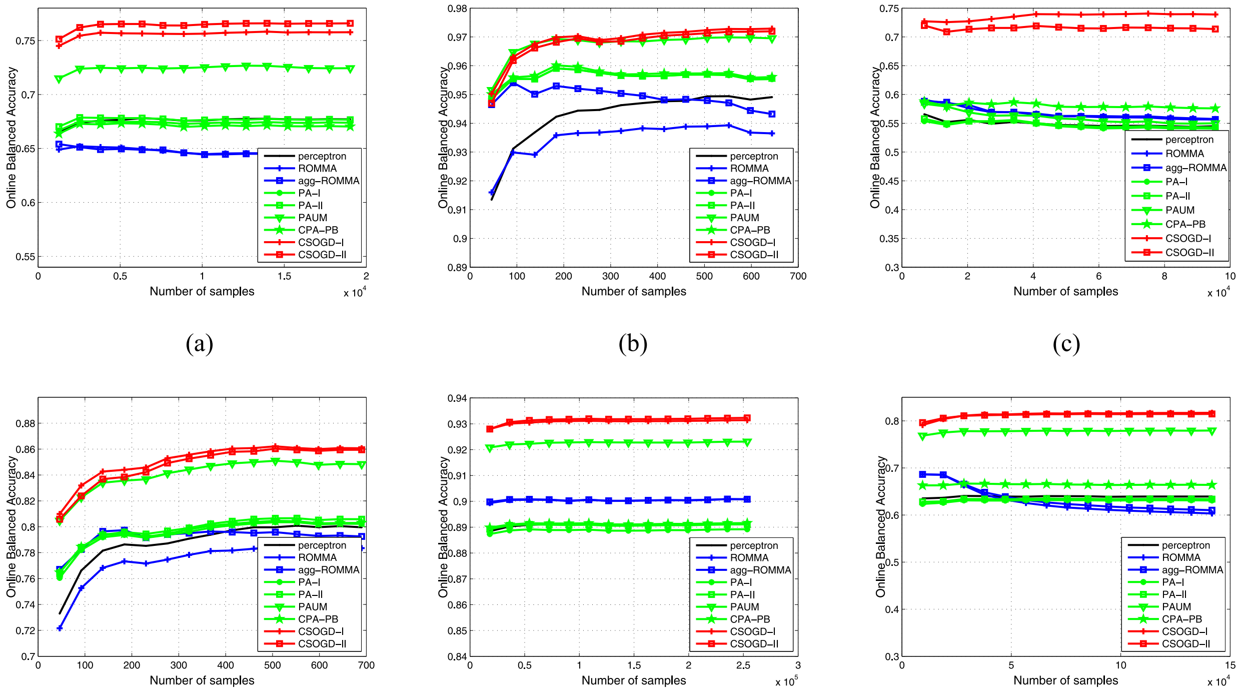
- Medical Imaging: We address medical image anomaly detection using two datasets: (i) the “Wisconsin Breast Cancer" [41] for detecting breast cancer from medical images of a fine needle aspirate (FNA) of a breast mass; and (ii) the “KDDCUP08" breast cancer dataset[2] for early detection of breast cancer from X-ray images of the breast. For both tasks, the “benign" class is treated as normal, and the “malignant" class is treated as anomaly.
- Finance: We address a credit card approval task in finance domain using the well-known Australia credit card data set with 690 instances from an Australian credit company, which is to distinguish credit-worthy from non credit-worthy customers.
- Nuclear: The “magic04” dataset [2] are MC generated to simulate registration of high energy gamma particles in a ground-based atmospheric Cherenkov gamma telescope using the imaging technique. The gamma signal instances are treated as normal and the hadron are outliers.

Table 5 summarizes the details of the data sets for online anomaly detection.

TABLE 5
Data Sets for Online Anomaly Detection

Dataset Name	#Examples	#Features	#Outlier:#Normal
Magic04	19020	10	1:1.8
Breast Cancer	683	10	1:1.86
KDDCUP08	102294	117	1:163.19
Australian	690	14	1:1.25
Cod-RNA	271617	8	1:2.00
ijcnn1	141691	22	1:9.4

[2] http://www.sigkdd.org/kddcup/



(a) (b) (c)

(d) (e) (f)

Fig. 8. Evaluation of online anomaly detection for the proposed CSOC algorithms. (a) *Magic04*. (b) *Wisconsin Breast Cancer*. (c) *KDDCUP08*. (d) *Australian*. (e) *Cod-RNA*. (f) *ijcnn1*.

6.2 Empirical Evaluation Results.

We apply our algorithms to solve real-world anomaly detection tasks as shown in Table 5. For performance metric, we evaluate the anomaly detection performance using the *balanced accuracy*, which are very common in anomaly detection tasks in order to avoid inflated performance estimates on imbalanced datasets.

Table 6 and Fig. 8 summarize the experimental results, in which we can draw some observations as follows. First of all, among all the existing algorithms, the two cost-sensitive algorithms (PAUM and CPA$_{PB}$) generally perform better than the other regular algorithms. However, the improvements are not always consistent and significant over different datasets. Such observations indicate the importance of studying more effective cost-sensitive algorithms. Second, among all the compared algorithms, it is obvious to see that the two proposed cost-sensitive algorithms significantly outperform the other algorithms for all the datasets. Moreover, we found that the improvements are particularly more significant when the dataset is highly class-imbalanced, such as the KDDCUP08 dataset where the proposed CSOGD algorithms achieved the balanced accuracy of over 70%, which is much higher than the other existing algorithms. This promising result validates the advantage of the proposed algorithms for solving a real-world online anomaly detection task which is often highly class-imbalanced.

7 Conclusion

As an attempt to fill the gap between *cost-sensitive classification* and *online learning*, this paper investigated a new framework of Cost-Sensitive Online Classification to solve large-scale online classification tasks in real-world applications. We proposed two cost-sensitive online learning algorithms by directly optimizing cost-sensitive measures

TABLE 6
Evaluation of Balanced Accuracy for Online Anomaly Detection

Algorithm	Breast	KDDCUP08
Perceptron	94.897 ± 0.552	54.347 ± 1.036
ROMMA	93.638 ± 0.553	54.618 ± 2.313
agg-ROMMA	94.280 ± 0.630	54.698 ± 2.105
PA-I	95.496 ± 0.538	53.936 ± 0.746
PA-II	95.541 ± 0.564	54.128 ± 0.696
PAUM	96.954 ± 0.409	54.886 ± 0.448
CPA$_{PB}$	95.537 ± 0.677	57.282 ± 1.187
CSOGD-I	**97.286** ± **0.301**	**73.852** ± **0.301**
CSOGD-II	97.180 ± 0.217	71.461 ± 0.576
Algorithm	Australian	Cod-RNA
Perceptron	79.962 ± 0.981	89.164 ± 0.037
ROMMA	78.352 ± 1.250	90.070 ± 0.033
agg-ROMMA	79.253 ± 1.285	90.071 ± 0.0333
PA-I	80.228 ± 1.105	88.918 ± 0.043
PA-II	80.582 ± 1.043	89.106 ± 0.041
PAUM	84.834 ± 0.603	92.315 ± 0.029
CPA$_{PB}$	80.296 ± 1.140	89.164 ± 0.045
CSOGD-I	**86.060** ± **0.425**	93.121 ± 0.016
CSOGD-II	85.949 ± 0.467	**93.220** ± **0.015**
Algorithm	Magic04	ijcnn1
Perceptron	67.700 ± 0.319	63.930 ± 0.204
ROMMA	64.411 ± 0.425	60.318 ± 1.136
agg-ROMMA	64.407 ± 0.365	61.025 ± 0.219
PA-I	67.381 ± 0.370	63.078 ± 0.089
PA-II	67.660 ± 0.314	63.378 ± 0.123
PAUM	72.437 ± 0.201	77.932 ± 0.123
CPA$_{PB}$	67.025 ± 0.373	66.388 ± 0.110
CSOGD-I	75.769 ± 0.162	81.701 ± 0.059
CSOGD-II	**76.591** ± **0.127**	**81.462** ± **0.075**

based on online gradient descent techniques. We then theoretically analyzed their cost-sensitive bounds, further examined their empirical performance, and finally demonstrated their applications to tackle real-world online anomaly detection tasks. Our encouraging results showed that our method achieved the state-of-the-art performance for cost-sensitive online classification tasks. Future work can further explore in-depth theory of cost-sensitive online classification and new algorithms to tackle emerging big data mining challenges, such as online feature selection [18], domain adaptation [44], and online active learning [46].

REFERENCES

[1] R. Akbani, S. Kwek, and N. Japkowicz, "Applying support vector machines to imbalanced datasets," in *Proc. 15th ECML*, Pisa, Italy, 2004, pp. 39–50.

[2] B. R. Bocka, "Methods for multidimensional event classification: A case study using images from a Cherenkov gamma-ray telescope," *Nucl. Instrum. Meth.*, vol. 516, no. 2–3, pp. 511-528, 2004.

[3] G. Blanchard, G. Lee, and C. Scott, "Semi-supervised novelty detection," *J. Mach. Learn. Res.*, vol. 11, pp. 2973–3009, Nov. 2010.

[4] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM CSUR*, vol. 41, no. 3, Article 15, 2009.

[5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, no. 1, pp. 321–357, 2002.

[6] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *J. Mach. Learn. Res.*, vol. 7, pp. 551–585, Mar. 2006.

[7] K. Crammer, M. Dredze, and F. Pereira, "Exact convex confidence-weighted learning," in *Proc. NIPS*, 2008, pp. 345–352.

[8] K. Crammer, A. Kulesza, and M. Dredze, "Adaptive regularization of weight vectors," in *Proc. NIPS*, 2009, pp. 345–352.

[9] P. Domingos, "Metacost: A general method for making classifiers cost-sensitive," in *Proc. 5th ACM SIGKDD Int. Conf. KDD*, San Diego, CA, USA, 1999, pp. 155–164.

[10] M. Dredze, K. Crammer, and F. Pereira, "Confidence-weighted linear classification," in *Proc. 25th ICML*, Helsinki, Finland, 2008, pp. 264–271.

[11] C. Drummond and R. C. Holte, "C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling," in *Proc. ICML*, 2003, pp. 1–8.

[12] C. Elkan, "The foundations of cost-sensitive learning," in *Proc. 17th IJCAI*, San Francisco, CA, USA, 2001, pp. 973–978.

[13] W. Fan, S. J. Stolfo, J. Zhang, and P. K. Chan, "Adacost: Misclassification cost-sensitive boosting," in *Proc. 16th ICML*, New York, NY, USA, 1999, pp. 97–105.

[14] Y. Freund and R. E. Schapire, "Large margin classification using the perceptron algorithm," *Mach. Learn.*, vol. 37, no. 3, pp. 277–296, 1999.

[15] C. Gentile, "A new approximate maximal margin classification algorithm," *J. Mach. Learn. Res.*, vol. 2, pp. 213–242, Dec. 2001.

[16] S. C. H. Hoi, R. Jin, P. Zhao, and T. Yang, "Online multiple kernel classification," *Mach. Learn.*, vol. 90, no. 2, pp. 289–316, 2013.

[17] S. C. H. Hoi, J. Wang, and P. Zhao, "LIBOL: A library for online learning algorithms," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 495–499, 2014.

[18] S. C. H. Hoi, J. Wang, P. Zhao, and R. Jin, "Online feature selection for mining big data," in *Proc. 1st ACM Int. Workshop BigMine*, Beijing, China, 2012, pp. 93–100.

[19] J. Kivinen, A. J. Smola, and R. C. Williamson, "Online learning with kernels," in *Proc. NIPS*, 2001, pp. 785–792.

[20] Y. Li and P. M. Long, "The relaxed online maximum margin algorithm," in *Proc. NIPS*, 1999, pp. 498–504.

[21] Y. Li, H. Zaragoza, R. Herbrich, J. Shawe-Taylor, and J. S. Kandola, "The perceptron algorithm with uneven margins," in *Proc. 19th ICML*, San Francisco, CA, USA, 2002, pp. 379–386.

[22] Y.-F. Li, J. T. Kwok, and Z.-H. Zhou, "Cost-sensitive semi-supervised support vector machine," in *Proc. 24th AAAI*, 2010.

[23] C. X. Ling, V. S. Sheng, and Q. Yang, "Test strategies for cost-sensitive decision trees," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 8, pp. 1055–1067, Aug. 2006.

[24] H. Liu, S. Shah, and W. Jiang, "On-line outlier detection and data cleaning," *Comput. Chem. Eng.*, vol. 28, no. 9, pp. 1635–1647, 2004.

[25] X.-Y. Liu and Z.-H. Zhou, "The influence of class imbalance on cost-sensitive learning: An empirical study," in *Proc. 6th ICDM*, Washington, DC, USA, 2006, pp. 970–974.

[26] H.-Y. Lo, J.-C. Wang, H.-M. Wang, and S.-D. Lin, "Cost-sensitive multi-label learning for audio tag annotation and retrieval," *IEEE Trans. Multimedia*, vol. 13, no. 3, pp. 518–529, Jun. 2011.

[27] A. C. Lozano and N. Abe, "Multi-class cost-sensitive boosting with p-norm loss functions," in *Proc. 14th ACM SIGKDD Int. Conf. KDD*, New York, NY, USA, 2008, pp. 506–514.

[28] J. Ma and S. Perkins, "Online novelty detection on temporal sequences," in *Proc. 9th ACM SIGKDD Int. Conf. KDD*, Washington, DC, USA, 2003, pp. 613–618.

[29] H. Masnadi-Shirazi and N. Vasconcelos, "Risk minimization, probability elicitation, and cost-sensitive svms," in *Proc. 27th ICML*, Haifa, Israel, 2010, pp. 759–766.

[30] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psych. Rev.*, vol. 65, no. 6, pp. 386–408, 1958.

[31] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt, "Support vector method for novelty detection," in *Proc. NIPS*, 1999, pp. 582–588.

[32] M. Sokolova, N. Japkowicz, and S. Szpakowicz, "Beyond accuracy, f-score and ROC: A family of discriminant measures for performance evaluation," in *Proc. AAAI*, Hobart, TAS, Australia, 2006, pp. 1015–1021.

[33] S. Subramaniam, T. Palpanas, D. Papadopoulos, V. Kalogeraki, and D. Gunopulos, "Online outlier detection in sensor data using non-parametric models," in *Proc. 32nd Int. Conf. VLDB*, Seoul, Korea, 2006, pp. 187–198.

[34] M. Tan, "Cost-sensitive learning of classification knowledge and its applications in robotics," *Mach. Learn.*, vol. 13, no. 1, pp. 7–33, Oct. 1993.

[35] D. M. J. Tax and R. P. W. Duin, "Support vector data description," *Mach. Learn.*, vol. 54, no. 1, pp. 45–66, 2004.

[36] P. D. Turney, "Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm," *J. Artif. Intell. Res.*, vol. 2, pp. 369–409, Mar. 1995.

[37] A. V. Uzilov, J. M. Keegan, and D. H. Mathews, "Detection of non-coding RNAs on the basis of predicted secondary structure formation free energy change," *BMC Bioinformatics*, vol. 7, Article 173, Mar. 2006.

[38] K. Veropoulos, C. Campbell, and N. Cristianini, "Controlling the sensitivity of support vector machines," in *Proc. IJCAI*, Stockholm, Sweden, 1999, pp. 55–60.

[39] J. Wang, P. Zhao, and S. C. H. Hoi, "Cost-sensitive online classification," in *Proc. 12th IEEE ICDM*, Brussels, Belgium, 2012.

[40] J. Wang, P. Zhao, and S. C. H. Hoi, "Exact soft confidence-weighted learning," in *Proc. 29th ICML*, Edinburgh, U.K., 2012.

[41] S. W. Wolberg, W. N. Street, and O. Mangasarian. "Machine learning techniques to diagnose breast cancer from fine-needle aspirates," *Cancer Lett.*, vol. 77, no. 2–3, pp. 163–171, 1994.

[42] B. Zadrozny, J. Langford, and N. Abe, "Cost-sensitive learning by cost-proportionate example weighting," in *Proc. 3rd IEEE ICDM*, Washington, DC, USA, 2003, pp. 435–442.

[43] J. H. Zhao, X. Li, and Z. Y. Dong, "Online rare events detection," in *Proc. PAKDD*, Nanjing, China, 2007, pp. 1114–1121.

[44] P. Zhao and S. C. H. Hoi, "OTL: A framework of online transfer learning," in *Proc. ICML*, Haifa, Israel, 2010, pp. 1231–1238.

[45] P. Zhao and S. C. H. Hoi, "Cost-sensitive double updating online learning and its application to online anomaly detection," in *Proc. SDM*, Austin, TX, USA, 2013.

[46] P. Zhao and S. C. H. Hoi, "Cost-sensitive online active learning with application to malicious URL detection," in *Proc. 19th ACM SIGKDD Int. Conf. KDD*, Chicago, IL, USA, 2013.

[47] P. Zhao, S. C. H. Hoi, and R. Jin, "Double updating online learning," *J. Mach. Learn. Res.*, vol. 12, pp. 1587–1615, May 2011.

[48] P. Zhao, S. C. H. Hoi, R. Jin, and T. Yang, "Online AUC maximization," in *Proc. 28th ICML*, 2011, Bellevue, WA, USA, pp. 233–240.

[49] Z.-H. Zhou and X.-Y. Liu, "On multi-class cost-sensitive learning," *Comput. Intell.*, vol. 26, no. 3, pp. 232–257, 2010.

[50] X. Zhu and X. Wu, "Class noise handling for effective cost-sensitive learning by cost-guided iterative classification filtering," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 10, pp. 1435–1440, Oct. 2006.

[51] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proc. 20th ICML*, Washington, DC, USA, 2003.

Jialei Wang is currently a Research Assistant with the School of Computer Engineering at the Nanyang Technological University, Singapore. He received the bachelor's degree from the University of Science and Technology of China, Hefei, China, in 2011. His current research interests include machine learning, statistics, and optimization with applications to data mining, computer vision, and natural language processing.

Peilin Zhao is currently a Ph.D. candidate with the School of Computer Engineering at Nanyang Technological University, Singapore. He received the bachelor's degree from Zhejiang University, Hangzhou, China, in 2008. His current research interests include statistical machine learning, and data mining.

Steven C.H. Hoi is an Associate Professor with the School of Computer Engineering at Nanyang Technological University, Singapore. He received the bachelor's degree from Tsinghua University, China, in 2002, and the Ph.D. degree in computer science and engineering from the Chinese University of Hong Kong, in 2006. His current research interests include machine learning and data mining and their applications to multimedia information retrieval (image and video retrieval), social media and web mining, and computational finance. He has published over 100 referred papers in top conferences and journals in related areas. He has served as a General Co-Chair for ACM SIGMM Workshops on Social Media (WSM'09, WSM'10, WSM'11), program Co-Chair for the fourth Asian Conference on Machine Learning (ACML'12), book editor for "*Social Media Modeling and Computing*", Guest Editor for *ACM TIST*, Technical PC Member for many international conferences, and external reviewer for many top journals and worldwide funding agencies, including NSF in U.S. and RGC in Hong Kong. He is a member of the IEEE and ACM.