Cost-Sensitive Online Classification with Adaptive Regularization and Its Applications

Peilin Zhao*, Furen Zhuang*, Min Wu*, Xiao-Li Li* and Steven C.H. Hoi†
*Data Analytics Department, Institute for Infocomm Research, A*STAR, Singapore 138632
†School of Information Systems, Singapore Management University, Singapore 178902
Email: {zhaop, zhuangf, wumin, xlli}@i2r.a-star.edu.sg, chhoi@smu.edu.sg

***Abstract*—Cost-Sensitive Online Classification is recently proposed to directly online optimize two well-known cost-sensitive measures: (i) maximization of weighted sum of sensitivity and specificity, and (ii) minimization of weighted misclassification cost. However, the previous existing learning algorithms only utilized the first order information of the data stream. This is insufficient, as recent studies have proved that incorporating second order information could yield significant improvements on the prediction model. Hence, we propose a novel cost-sensitive online classification algorithm with adaptive regularization. We theoretically analyzed the proposed algorithm and empirically validated its effectiveness with extensive experiments. We also demonstrate the application of the proposed technique for solving several online anomaly detection tasks, showing that the proposed technique could be an effective tool to tackle cost-sensitive online classification tasks in various application domains.**

***Keywords*—Cost-Sensitive Classification; Online Learning; Adaptive Regularization;**

I. Introduction

Online learning has been extensively studied for years in machine learning and data mining literature [1], [2], [3], [4], [5], [6], [7], whose goal in general is to incrementally learn prediction models to make correct predictions on a stream of examples that arrive sequentially. Online learning enjoys many advantages for real-world large-scale applications. For example, in some real applications, data often arrives sequentially while prediction must be made immediately, such as, malicious URL detection [8] and portfolio selection [9]. Moreover, online learning is very attractive for large-scale learning task, e.g., training SVM from billions of data [10]. Although being studied extensively in the literature, most of the existing online learning algorithms are unsuitable to solving cost-sensitive classification tasks. Cost-sensitive classification is an important task for data mining, which differs with traditional classification by taking the misclassification costs into consideration [11], [12]. Most traditional online learning algorithms often concern the performance in terms of prediction mistake rate or accuracy, which is clearly cost-insensitive and thus inappropriate for quite a few real-world applications in data mining, where datasets are often class-imbalanced and the misclassification costs of instances from different classes can be significantly different [13], [14], [15], [16].

To address this issue, researchers have proposed more meaningful metrics for cost-sensitive classifications, including: the weighted sum of sensitivity and specificity [17], [18] and the weighted misclassification cost [11], [19]. Given these cost-sensitive measures, many batch classification algorithms are developed to optimize these performance during the past decades [11], [19]. However, these batch algorithms often suffer from poor efficiency and scalability for large-scale tasks, which makes them unsuitable to 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. As an attempt to fill the gap between cost-sensitive classification and online learning in machine learning and data mining, a new framework of Cost-Sensitive Online Classification [7] is recently proposed and investigated, which aims to directly optimize cost-sensitive measures for online classification tasks. Under this framework, a family of effective Cost-Sensitive Online Classification algorithms are proposed based on online gradient descent, which are termed as Cost-Sensitive Online Gradient Descent (CSOGD). Compared with many traditional online learning algorithms, encouraging results show that the CSOGD algorithms considerably outperform the traditional online learning algorithms for cost-sensitive online classification tasks [7].

Although CSOGD can solve CSOC better than traditional online learning algorithms, it only utilizes the first order information of the examples, i.e., weighted mean of the gradient. This is obviously insufficient, because recent studies [20], [21], [22], [5] have shown that the usage of second order information, i.e., the correlations between features, can significantly improve the performance of online learning. Hence, we propose Adaptively Regularized Cost-Sensitive Online Gradient Descent (ARCSOGD) based on the state-of-the-art Confidence Weighted [20], [21], [22], [5] strategy, which not only updates the model using the first order information but also the second order information, to further improve the learning efficacy. Furthermore, we theoretically analyzed its regret bound, which measures the difference between its cumulative loss and the one of the best model. To empirically evaluate the proposed algorithm, we conduct an extensive set of experiments on some benchmark datasets and several online anomaly detection tasks from various real-world application domains. Promising experimental results demonstrate the effectiveness and efficiency of the proposed algorithm, compared with many stat-of-the-art online learning algorihtms.

The rest of this paper is organized as follows: We first review related work in section 2, and then present the proposed algorithm and its theoretical analysis in section 3; we further

discuss the experiments in section 4. Section 5 shows an application to online anomaly detection tasks, and finally section 6 concludes the paper.

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, (ii) online learning in machine learning literature.

A. Cost-sensitive Classification

Cost-sensitive classification has been extensively studied in data mining and machine learning [23], [24], [25]. Classification problems such as fraud detection, medical diagnosis, are naturally cost sensitive. In these problems the cost of missing 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* [17], [18], and the weighted *misclassification cost* that takes cost into consideration when measuring classification performance [11], [19]. 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* [18], 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 [26], [27], [12], [11], [28], [29], [16]. However, few studies emphasis the case when data arrives sequentially, except Perceptron Algorithms with Uneven Margin(PAUM) [30], the Cost-sensitive Passive Aggressive(CPA) [3], and the CSOGD algorithm [7].

B. Online Learning

Online learning has been actively studied in machine learning community [1], [31], [32], [3], [33], [34], [35], [36], [37], in which a variety of online learning algorithms have been proposed, including a number of first-order algorithms [38], [3]. One of the most well-known first-order online approaches is the Perceptron algorithm [1], [39], which updates the learning function by adding the misclassified example with a constant weight to the current set of support vectors. Recently a number of online learning algorithms have been developed based on the criterion of maximum margin [31], [40], [41], [3], [2]. One example is the Relaxed Online Maximum Margin algorithm (ROMMA) [2], which repeatedly chooses the hyperplanes that correctly classify the existing training examples with a large margin. Another representative example is the Passive-Aggressive (PA) algorithm [3]. It updates the classification function when a new example is misclassified or its classification score does not exceed the predefined margin. Empirical studies showed that the maximum margin based online learning algorithms are generally more effective than the Perceptron algorithm. Despite the difference, these online learning algorithms only update the algorithm based the first-order information, such as the gradient of the loss. This constraint could significantly limit the performance of online learning.

Recent years have seen a surge of studies on the second-order online learning algorithms [42], [20], [22], [5], which have shown that parameter confidence information can be explored to guide and improve online learning performance [42]. For example, Second Order Perceptron (SOP) [42] is the first second-order online learning algorithm, which can be viewed as an online variant of the whitened Perceptron algorithm, where the whitened effect is achieved by using online correlation matrices of the previously seen instances. Later, some second order online learning algorithms with large margin are proposed. For example, Confidence-weighted (CW) learning [20], [43] maintains a Gaussian distribution over some linear classifier hypotheses and applies it to control the direction and scale of parameter updates [20]. Although CW learning has formal guarantees in the mistake-bound model [43], it can overfit in certain situations due to its aggressive update rules based upon a separable data assumption. Recently, an improved online algorithm, i.e., Adaptive Regularization of Weights (AROW) [22], relaxes such separable assumption by employing an adaptive regularization for each training example based upon its current confidence. This regularization comes in the form of minimizing a combination of the Kullback-Leibler divergence between Gaussian distributed weight vectors and a confidence penalty of vectors. Although AROW [22] is able to improve the original CW [43] learning by handling noisy and non-separable cases, it is not the exact corresponding soft extending part of CW (Like PA with PA-I and PA-II). In particular, the directly added loss and confidence regularization make AROW lose an important property of Confidence-weighted learning, i.e., Adaptive Margin property [43]. Following the similar idea of soft margin support vector machines, Soft Confidence-Weighted algorithms [5] algorithms are proposed to assign adaptive margins for different instances via a probability formulation, which enables CW to gain extra efficiency and effectiveness. In general, the second order algorithms are more accurate, converge faster.

Most online learning algorithms are cost-insensitive, with notable exceptions such as the perceptron algorithm with uneven margin ('PAUM') [30], the prediction-based PA algorithm ('CPA$_{PB}$') [3], and the CSOGD algorithm [7].

III. Adaptively Regularized Cost-Sensitive Online Classification

In this section, we first introduce the Cost-Sensitive Online Classification (CSOC) problem settings, and then present our proposed Adaptively Regularized Cost-Sensitive Online Gradient Descent Algorithm (ARCSOGD).

A. Problem Settings

Without loss of generality, let us consider an online binary classification problem. Our goal is to learn a linear model $\mathbf{w} \in \mathbb{R}^d$ based on a sequence of training examples $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)\}$, where $\mathbf{x}_t \in \mathbb{R}^d$ is a d-dimensional instance and $y_t \in \mathcal{Y} = \{-1, +1\}$ is the class label assigned to $\mathbf{x}_t$. We use $\mathrm{sign}(\mathbf{w}^\top \mathbf{x})$ to predict the class assignment/label for any instance $\mathbf{x}$.

Online binary classification algorithm learns the model in rounds. Formally, at the t-th round, the algorithm will receive the instance $\mathbf{x}_t$, and make a prediction $\hat{y}_t = \mathrm{sign}(\mathbf{w}_t^\top \mathbf{x}_t)$, where $\mathbf{w}_t$ is a model learnt using the previous $t-1$ examples. Then the true label $y_t \in \{-1, +1\}$ will be revealed for

comparison. If $\hat{y}_t \neq y_t$, the learner made a mistake; otherwise it made a correct prediction. For convenience, we denote $\mathcal{M} = \{t \mid y_t \neq \text{sign}(\mathbf{w}_t \cdot \mathbf{x}_t),\ \forall t \in [T]\}$, $\mathcal{M}_p = \{t \mid t \in \mathcal{M}\ and\ y_t = +1\}$ and $\mathcal{M}_n = \{t \mid t \in \mathcal{M}\ and\ y_t = -1\}$, where $[T] = \{1, \ldots, T\}$. In addition, we introduce notation $M = |\mathcal{M}|$, $M_p = |\mathcal{M}_p|$ and $M_n = |\mathcal{M}_n|$ to denote the number of mistakes, false negatives and false positives. Also we use $\mathcal{I}_T^p = \{i \in [T] | y_i = +1\}$, $\mathcal{I}_T^n = \{i \in [T] | y_i = -1\}$ and $T_p = |\mathcal{I}_T^p|$ and $T_n = |\mathcal{I}_T^n|$ to denote the number of positive examples and negative examples.

We assume the positive class is the rare class, i.e., $T_p \leq T_n$. Traditional online learning tries to maximize accuracy but this may be inappropriate for imbalanced data because a trivial learner which simply classifies all examples as negative could still achieve a high accuracy. Thus, a more appropriate metric is to measure the *sum* of weighted *sensitivity* and *specificity*, i.e.,

$$sum = \alpha_p \times \frac{T_p - M_p}{T_p} + \alpha_n \times \frac{T_n - M_n}{T_n}, \tag{1}$$

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

$$cost = c_p \times M_p + c_n \times M_n, \tag{2}$$

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.

Our objective is to either maximize sum or minimize $cost$. As shown in [7], both of these are equivalent to minimizing the following objective:

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

where $\rho = \frac{\alpha_p T_n}{\alpha_n T_p}$ for the maximization of the weighted sum, and $\rho = \frac{c_p}{c_n}$ for the minimization of the weighted misclassification cost. As the indicator function is not convex, we replace the indicator function by its convex surrogate:

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

We could see that for $\ell(\mathbf{w}; (\mathbf{x}, y))$, the required margin for specific class changed compared to the traditional hinge loss, causing more "frequent" updating. Now our goal is to find an online learning solution to minimize the regret of the learning process:

$$Regret := \sum_{t=1}^{T} \ell(\mathbf{w}_t; (\mathbf{x}_t, y_t)) - \sum_{t=1}^{T} \ell(\mathbf{w}^*; (\mathbf{x}_t, y_t)),$$

where $\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_{t=1}^{T} \ell(\mathbf{w}; (\mathbf{x}_t, y_t))$. To solve this problem, CSOGD [7] was proposed, i.e.,$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla \ell_t(\mathbf{w}_t)$ where η is the learning rate and $\ell_t(\mathbf{w}) = \ell(\mathbf{w}; (\mathbf{x}_t, y_t))$. However, this algorithm only adopts the first order information of the data stream to update the model. This is clearly insufficient, since recent studies have shown the importance of incorporating the second order information [20], [43], [22]. Motivated by this observation, we propose to use adaptive regularization to improve the cost-sensitive online classification.

B. Algorithms

To solve this cost-sensitive online classification task, we assume the online model satisfies a Gaussian distribution, i.e., $\mathbf{w} \sim \mathcal{N}(\mu, \Sigma)$. Given a Gaussian distribution, we would like to predict the label of an instance $\mathbf{x}$ according to $\text{sign}(\mathbf{w}^\top \mathbf{x})$. However, it is more practical to simply use the mean of the distribution $\mathbb{E}[\mathbf{w}] = \mu$ to make predictions for real-world tasks. The mean values μ_i represents the model's knowledge of the weight for feature i, while $\Sigma_{i,i}$ encodes the confidence in feature i. Generally, the smaller $\Sigma_{i,i}$, the more confidence the learner has in the mean weight value μ_i. The covariance terms $\Sigma_{i,j}$ keeps the correlations between weights i and j.

At the t-th round, when receiving $(\mathbf{x}_t, y_t)$, a natural rule to update the model is to minimize the following objective:

$$D_{KL}(\mathcal{N}(\mu, \Sigma) \| \mathcal{N}(\mu_t, \Sigma_t)) + \eta \ell_t(\mu) + \frac{1}{2\gamma} \mathbf{x}_t^\top \Sigma \mathbf{x}_t,$$

where D_{KL} is Kullback-Leibler divergence, i.e.,

$$D_{KL}(\mathcal{N}(\mu, \Sigma) \| \mathcal{N}(\mu_t, \Sigma_t)) = \frac{1}{2} \log \left(\frac{\det \Sigma_t}{\det \Sigma} \right) + \frac{1}{2} \text{Tr}(\Sigma_t^{-1} \Sigma) + \frac{1}{2} \|\mu_t - \mu\|^2_{\Sigma_t^{-1}} - \frac{d}{2}.$$

Generally, this objective would like to make the least adjustment, such that the loss on the current example is minimized and the confidence of the model is optimized. However, this optimization dose not have closed-form solution. To solve this issue, we replace the loss $\ell(\mu)$ with its first-order Taylor expansion $\ell(\mu_t) + \mathbf{g}_t^\top (\mu - \mu_t)$, where $\mathbf{g}_t = \partial \ell_t(\mu_t)$, to get the following optimization objective:

$$f_t(\mu, \Sigma) = D_{KL}(\mathcal{N}(\mu, \Sigma) \| \mathcal{N}(\mu_t, \Sigma_t)) + \eta \mathbf{g}_t^\top \mu + \frac{1}{2\gamma} \mathbf{x}_t^\top \Sigma \mathbf{x}_t,$$

which is much easier to be solved.

A simple approach to solve this objective is to solve it in following two steps:

- Update the mean parameter:
$$\mu_{t+1} = \arg\min_{\mu} f_t(\mu, \Sigma);$$
- If $\ell_t(\mu_t) \neq 0$, update the covariance matrix:
$$\Sigma_{t+1} = \arg\min_{\Sigma} f_t(\mu, \Sigma);$$

For the first step, setting the derivative of $\partial_\mu f_t(\mu_{t+1}, \Sigma)$ as zero will give

$$\Sigma_t^{-1}(\mu_{t+1} - \mu_t) + \eta \mathbf{g}_t = 0 \Rightarrow \mu_{t+1} = \mu_t - \eta \Sigma_t \mathbf{g}_t,$$

and for the second step, setting the derivative of $\partial_\Sigma f_t(\mu, \Sigma_{t+1})$ as zero will give

$$-\Sigma_{t+1}^{-1} + \Sigma_t^{-1} + \frac{\mathbf{x}_t \mathbf{x}_t^\top}{\gamma} = 0 \Rightarrow \Sigma_{t+1} = \Sigma_t - \frac{\Sigma_t \mathbf{x}_t \mathbf{x}_t^\top \Sigma_t}{\gamma + \mathbf{x}_t^\top \Sigma_t \mathbf{x}_t},$$

where the Woodbury identity is used. Furthermore, since the update of the mean relies on the confidence parameter, we

propose to update the mean based on the updated covariance matrix, which should be more accurate than the old covariance matrix, i.e.,

$$\mu_{t+1} = \mu_t - \eta \Sigma_{t+1} \mathbf{g}_t.$$

This is different from AROW, where the updating rule for μ_t relies on the old Σ_t. To intuitively explain the above update, let us assume Σ_{t+1} is a diagonal matrix. Then, this update actually assigns different dimensions with different learning rates, so that more unconfident weights will be updated more aggressively.

Finally, we can summarize the proposed Adaptive Regularized Cost-Sensitive Online Gradient Descent (ARCSOGD) in Algorithm 1.

Algorithm 1 Adaptive Regularized Cost-Sensitive Online Gradient Descent (**ARCSOGD**) algorithm.

Input: learning rate η; regularization parameter γ, bias parameter $\rho = \frac{\alpha_p T_n}{\alpha_n T_p}$ for "sum" and $\rho = \frac{c_p}{c_n}$ for "cost"
Initialize: $\mu_1 = 0$, $\Sigma_1 = I$.
for $t = 1, \ldots, T$ **do**
  Compute $\rho_t = \rho * \mathbb{I}_{(y_t=1)} + \mathbb{I}_{(y_t=-1)}$;
  Compute $\ell_t(\mu_t) = [\rho_t - y_t \mathbf{x}_t^\top \mu_t]_+$;
  if $\ell_t(\mu_t) > 0$ **then**
    $\Sigma_{t+1} = \Sigma_t - \frac{\Sigma_t \mathbf{x}_t \mathbf{x}_t^\top \Sigma_t}{\gamma + \mathbf{x}_t^\top \Sigma_t \mathbf{x}_t}$;
    $\mu_{t+1} = \mu_t - \eta \Sigma_{t+1} \mathbf{g}_t$, where $\mathbf{g}_t = \partial \ell_t(\mu_t)$;
  else
    $\mu_{t+1} = \mu_t$, $\Sigma_{t+1} = \Sigma_t$;
  end if
end for

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 $\rho = \frac{\alpha_p T_n}{\alpha_n T_p}$. 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. Another concern is the time complexity for the update of Σ_{t+1} and μ_{t+1}, which is $O(d^2)$. To reduce this time complexity, we can make the algorithm keep and maintain a diagonal version of Σ_t so that the time complexity decrease to $O(d)$.

C. Theoretical Analysis

In this subsection, we theoretically analyze the proposed algorithm in terms of two types of cost-sensitive measures. To this end, we first prove a key theorem, which gives the regret bound of the proposed algorithm and will facilitate later theoretical analysis.

Theorem 1. *Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ be a sequence of examples, where $\mathbf{x}_t \in \mathbb{R}^d$, $y_t \in \{-1, +1\}$. Then for any $\mu \in \mathbb{R}^d$, the proposed ARCSOGD satisfies*

$$Regret \leq \frac{1}{2\eta}(D_\mu)^2 \mathrm{Tr}(\Sigma_{T+1}^{-1}) + \frac{\eta\gamma}{2} \log(|\Sigma_{T+1}^{-1}|).$$

Setting $\eta = \sqrt{\frac{\max_{t \leq T} \|\mu_t - \mu\|^2 \mathrm{Tr}(\Sigma_{T+1}^{-1})}{\gamma \log(|\Sigma_{T+1}^{-1}|)}}$, we will get

$$Regret \leq D_\mu \sqrt{\gamma \mathrm{Tr}(\Sigma_{T+1}^{-1}) \log(|\Sigma_{T+1}^{-1}|)}.$$

where $D_\mu = \max_t \|\mu_t - \mu\|$.

Remark: Suppose $\|\mathbf{x}_t\| \leq 1$, it is easy to observe $\mathrm{Tr}(\Sigma_{T+1}^{-1}) \leq O(T/\gamma)$, so the regret is in the order of $O(\sqrt{T})$. This order is optimal, since the loss function is not strongly convex [44].

Thus, by our proposed method, we can guarantee the following bound on the sum of $\alpha_p \times sensitive + \alpha_n \times specificity$.

Theorem 2. *Under the same assumptions in the Theorem 1, by setting $\rho = \frac{\alpha_p T_n}{\alpha_n T_p}$, the proposed ARCSOGD satisfies for any $\mu \in \mathbb{R}^d$,*

$$sum \geq 1 - \frac{\alpha_n}{T_n}\Big[\sum_{t=1}^{T} \ell_t(\mu) + D_\mu \sqrt{\gamma \mathrm{Tr}(\Sigma_{T+1}^{-1}) \log(|\Sigma_{T+1}^{-1}|)}\Big].$$

Remark: It is easy to observe $\sum_{t=1}^{T} \ell_t(\mu)$ is an convex estimate of $\rho M_p + M_n$ for μ, so $\frac{\alpha_n}{T_n}\sum_{t=1}^{T} \ell_t(\mu)$ is an estimate of $\alpha_p \frac{M_p}{T_p} + \alpha_n \frac{M_n}{T_n}$. Moreover, please note α_n cannot be set as zero, since $\rho = \frac{\alpha_p T_n}{\alpha_n T_p}$. One limitation of the above algorithm is that we may not know the ratio $\frac{T_n}{T_p}$ in advance. To address this issue, an alternative is to consider the cost of the algorithm for performance evaluation, which does not need $\frac{T_n}{T_p}$ in advance since the bias ρ is set as $\frac{c_p}{c_n}$.

Theorem 3. *Under the same assumptions in the Theorem 1, by setting $\rho = \frac{c_p}{c_n}$, the proposed ARCSOGD satisfies for any $\mu \in \mathbb{R}^d$,*

$$cost \leq c_n\Big[\sum_{t=1}^{T} \ell_t(\mu) + D_\mu \sqrt{\gamma \mathrm{Tr}(\Sigma_{T+1}^{-1}) \log(|\Sigma_{T+1}^{-1}|)}\Big].$$

Remark: $\sum_{t=1}^{T} \ell_t(\mu)$ is an convex estimate of $\frac{c_p}{c_n} M_p + M_n$ for μ, so $c_n \sum_{t=1}^{T} \ell_t(\mu)$ is an estimate of $c_p M_p + c_n M_n$. Moreover, please note c_n cannot be set as zero, since $\rho = \frac{c_p}{c_n}$.

IV. Experiments

This section evaluates the empirical performance of the proposed algorithm ARCSOGD and its variant ARCSOGD$_{diag}$. ARCSOGD$_{diag}$ is a diagonalized version of ARCSOGD, where only a diagonal Σ_t is kept and updated online to save the memory cost and improve the scalability.

A. Experimental Testbed and Setup

We compare ARCSOGD with 2 standard and 3 well-known online learning algorithms: Perceptron; the Passive-Aggressive algorithm ("PA-I") [3]; cost-sensitive algorithms: prediction-based PA algorithm ('CPA$_{PB}$') [3]; perceptron algorithm with uneven margin ('PAUM') [30] and the 'CSOGD-I' algorithm, from which ARCSOGD was derived.

The algorithms were tested on 6 benchmark datasets as listed in Table I, obtained from LIBSVM [1]. For all datasets, the

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

instances are normalized, i.e., $\mathbf{x}_t \leftarrow \mathbf{x}_t/\|\mathbf{x}_t\|$, which is widely adopted for online learning, since the instances are received sequentially.

TABLE I. LIST OF BINARY DATASETS IN OUR EXPERIMENTS.

Dataset	#Examples	#Features	#Pos:#Neg
covtype	581012	54	1:1
spambase	4601	57	1:1.5
svmguide3	1243	21	1:3
a9a	48842	123	1:3.2
ijcnn1	141691	22	1:9.4
w8a	64700	300	1:32.5

To make a valid comparison, all algorithms adopted the same experimental setup. For *sum*, we set $\alpha_p = \alpha_n = 1/2$ for all cases, while for *cost*, we set $c_p = 0.9$ and $c_n = 0.1$; for PAUM, the uneven margin was set to ρ; for CPA$_{PB}$, $\rho(-1, 1)$ was set to 1 and $\rho(1, -1)$ was set to ρ. The parameter C for PA-I, learning rates λ of CSOGD-I and η of ARCSOGD and ARCSOGD$_{diag}$ were selected by cross validation from $[10^{-5}, 10^{-4}, \ldots, 10^{5}]$ for each dataset. The γ for ARCSOGD and ARCSOGD$_{diag}$ was set as 1. The value of ρ was set to $\frac{c_p}{c_n}$ for *cost* and $\frac{\alpha_p T_n}{\alpha_n T_p}$ for *sum*, respectively. All algorithms were implemented in MATLAB and run on a 2.00GHz Windows machine.

All experiments were conducted over 20 random permutations for each dataset. Results are reported by averaging over these 20 runs. Performance was evaluated by 4 metrics: *sensitivity*, *specificity*, the weighted *sum* of sensitivity and specificity, and the weighted *cost* of misclassification.

B. Evaluation of Cost-Sensitive Performance

The left and right parts of Table II summarizes the experimental results on *sum* and *cost* on three datasets, respectively.

By examining the *sum* and *cost* performance, we can see that our two proposed second order algorithms (i.e., ARCSOGD and ARCSOGD$_{diag}$) significantly outperform all the other online learning algorithms on all the datasets, which validates the effectiveness of introducing second order information.

Furthermore, the proposed algorithms usually result in the best sensitivity, and produce good specificity performance under both cost-sensitive measures. This shows that the proposed algorithms are effective in improving the prediction accuracy for the rare class.

Finally, while ARCSOGD$_{diag}$ achieves marginally smaller *sum* and larger *cost* than ARCSOGD, its computational complexity is similar to the first order algorithms' complexities, indicating that ARCSOGD$_{diag}$ is able to achieve a better trade-off between effectiveness and efficiency.

C. Performance Evaluation with Different Cost-Sensitive Weights

In this subsection, we aim to evaluate the performance of the proposed algorithms under varying cost-sensitive weights for both metrics.

Figure 3 shows the evaluation results of the weighted sum performance under varying weights of α_n, and Figure 4 shows the evaluation results of the weighted cost under varying weights of c_n. From the results, it is clear that the proposed algorithms consistently outperform all of the other algorithms for both metrics under varying weight values. These promising results further validate the efficacy of the proposed algorithms.

V. APPLICATION TO ONLINE ANOMALY DETECTION

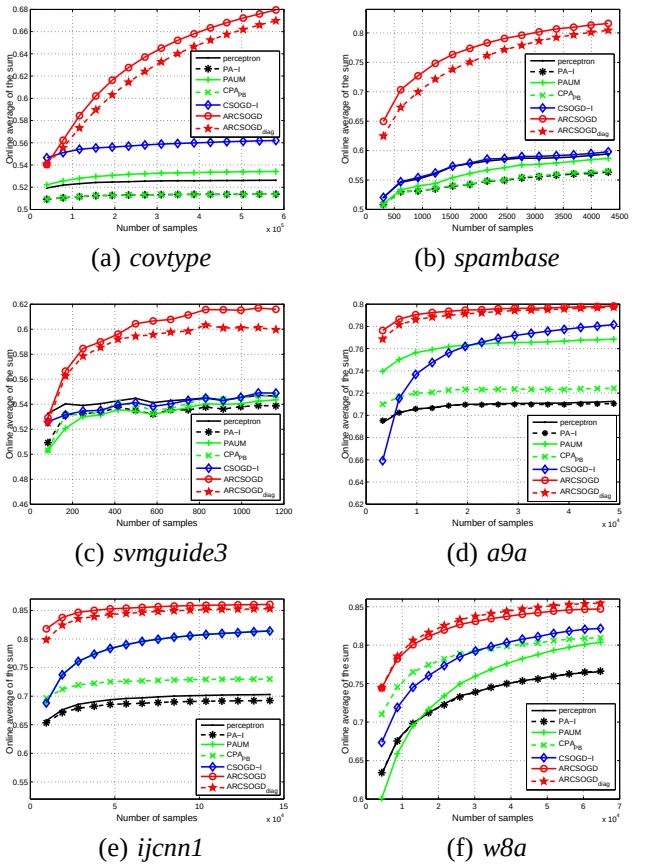
The proposed cost-sensitive online classification technique can potentially be applied to a wide range of real-world applications in data mining. In this section, we demonstrate an application of the proposed cost-sensitive online classification algorithms to tackle online anomaly detection tasks. Below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

A. Application Domains and Testbeds.

We apply the proposed algorithms to solve problems in the following domains:

- Medical Imaging: We apply our algorithms to solve a medical image anomaly detection problem using the "KDDCUP08" breast cancer dataset [2]. For this dataset, the task is to develop a computational method for early detection of breast cancer from X-ray images of the

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



(a) *covtype* (b) *spambase*

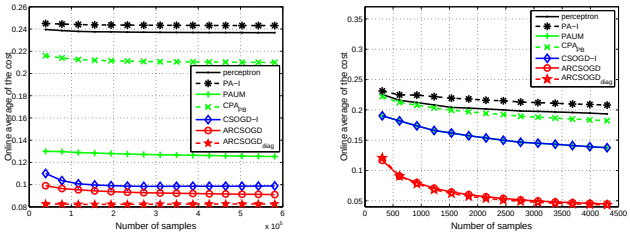
(c) *svmguide3* (d) *a9a*

(e) *ijcnn1* (f) *w8a*

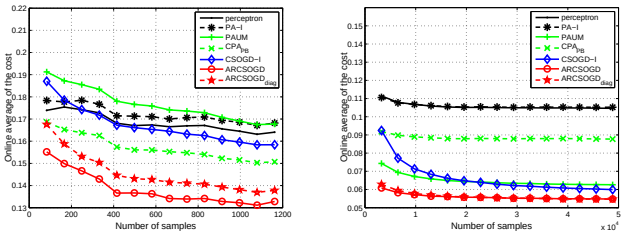
Fig. 1. Evaluation of online "*sum*" performance of the proposed algorithms on public datasets.

TABLE II. EVALUATION OF THE COST-SENSITIVE CLASSIFICATION PERFORMANCE OF ARCSOGD AND OTHER EXISTING ALGORITHMS.

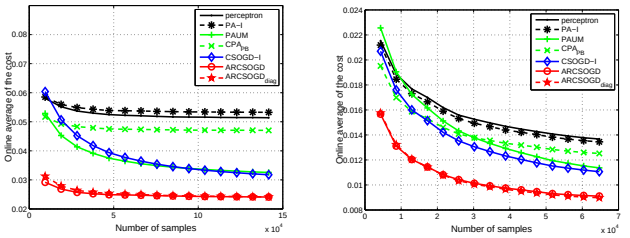
Algorithm	"sum" on covtype				"cost" on covtype			
	Sum(%)	Sensitivity(%)	Specificity (%)	Time (s)	Cost(%)	Sensitivity(%)	Specificity (%)	Time (s)
Perceptron	52.626 ± 0.075	51.451 ± 0.077	53.801 ± 0.073	17.470	23.670 ± 0.028	51.457 ± 0.058	**53.807 ± 0.055**	17.375
PA-I	51.370 ± 0.059	50.115 ± 0.055	52.625 ± 0.072	31.576	24.313 ± 0.022	50.127 ± 0.044	52.640 ± 0.058	29.771
PAUM	53.418 ± 0.064	52.069 ± 0.100	54.767 ± 0.070	18.093	12.532 ± 0.030	79.855 ± 0.071	27.949 ± 0.069	17.682
CPA$_{PB}$	51.373 ± 0.059	50.299 ± 0.054	52.448 ± 0.071	29.447	20.996 ± 0.035	58.522 ± 0.079	45.474 ± 0.068	27.159
CSOGD-I	56.200 ± 0.047	41.878 ± 0.275	**70.522 ± 0.210**	15.112	9.883 ± 0.048	86.648 ± 0.126	21.489 ± 0.155	18.074
ARCSOGD	**67.950 ± 0.053**	**70.318 ± 0.069**	65.583 ± 0.087	103.222	9.107 ± 0.073	88.248 ± 0.187	22.926 ± 0.189	141.132
ARCSOGD$_{diag}$	66.981 ± 0.382	68.710 ± 0.550	65.252 ± 0.229	24.442	**8.258 ± 0.323**	**90.889 ± 0.902**	16.862 ± 1.442	37.207
Algorithm	"sum" on spambase				"cost" on spambase			
	Sum(%)	Sensitivity(%)	Specificity (%)	Time (s)	Cost(%)	Sensitivity(%)	Specificity (%)	Time (s)
Perceptron	59.766 ± 0.837	51.269 ± 1.006	68.264 ± 0.668	0.128	19.231 ± 0.565	51.202 ± 1.434	**68.228 ± 0.942**	0.115
PA-I	56.415 ± 0.692	47.303 ± 0.916	65.527 ± 0.524	0.232	20.718 ± 0.358	47.441 ± 0.932	65.707 ± 0.543	0.216
PAUM	58.819 ± 0.775	50.292 ± 0.780	67.346 ± 1.088	0.143	13.655 ± 0.416	70.177 ± 1.186	49.204 ± 0.769	0.132
CPA$_{PB}$	56.611 ± 0.738	48.958 ± 1.149	64.265 ± 0.397	0.235	18.086 ± 0.404	55.888 ± 1.084	59.695 ± 0.523	0.200
CSOGD-I	60.055 ± 0.820	51.627 ± 0.981	68.483 ± 0.659	0.154	13.655 ± 0.416	70.177 ± 1.186	49.204 ± 0.769	0.135
ARCSOGD	**81.860 ± 0.357**	**86.751 ± 0.780**	76.969 ± 0.769	0.739	4.402 ± 0.356	94.647 ± 1.174	58.684 ± 1.760	0.966
ARCSOGD$_{diag}$	80.766 ± 0.598	84.435 ± 1.015	**77.098 ± 1.094**	0.211	**4.248 ± 0.192**	**95.761 ± 0.716**	54.709 ± 1.818	0.227
Algorithm	"sum" on svmguide3				"cost" on svmguide3			
	Sum(%)	Sensitivity(%)	Specificity (%)	Time (s)	Cost(%)	Sensitivity(%)	Specificity (%)	Time (s)
Perceptron	54.835 ± 1.354	31.149 ± 2.038	78.522 ± 0.672	0.030	16.401 ± 0.431	31.115 ± 1.811	78.506 ± 0.571	0.034
PA-I	53.902 ± 1.615	29.493 ± 2.533	78.310 ± 0.798	0.051	16.842 ± 0.464	29.172 ± 1.977	78.178 ± 0.685	0.065
PAUM	54.637 ± 1.253	25.811 ± 3.097	**83.464 ± 0.966**	0.032	16.665 ± 0.425	28.615 ± 2.123	82.075 ± 0.771	0.042
CPA$_{PB}$	54.802 ± 1.671	35.676 ± 2.442	73.928 ± 1.139	0.053	15.060 ± 0.442	40.220 ± 1.803	70.491 ± 1.098	0.063
CSOGD-I	54.986 ± 1.061	31.419 ± 1.593	78.553 ± 0.537	0.037	15.778 ± 0.364	34.848 ± 1.634	76.188 ± 0.828	0.042
ARCSOGD	**61.582 ± 1.167**	**40.101 ± 2.213**	83.062 ± 0.942	0.064	**13.244 ± 0.401**	**44.105 ± 2.008**	**83.400 ± 0.888**	0.087
ARCSOGD$_{diag}$	60.231 ± 1.358	39.122 ± 2.487	81.341 ± 0.952	0.048	13.697 ± 0.470	43.074 ± 2.385	80.359 ± 0.835	0.061
Algorithm	"sum" on a9a				"cost" on a9a			
	Sum(%)	Sensitivity(%)	Specificity (%)	Time (s)	Cost(%)	Sensitivity(%)	Specificity (%)	Time (s)
Perceptron	71.255 ± 0.177	56.269 ± 0.268	86.241 ± 0.085	1.389	10.462 ± 0.068	56.277 ± 0.285	**86.244 ± 0.090**	1.491
PA-I	71.048 ± 0.149	55.949 ± 0.224	86.147 ± 0.122	2.221	10.521 ± 0.064	56.031 ± 0.274	86.170 ± 0.108	2.504
PAUM	76.842 ± 0.157	68.121 ± 0.266	85.562 ± 0.101	1.485	6.260 ± 0.033	77.486 ± 0.150	81.441 ± 0.093	1.705
CPA$_{PB}$	72.432 ± 0.207	62.417 ± 0.350	82.446 ± 0.191	2.252	8.773 ± 0.082	66.485 ± 0.362	79.549 ± 0.119	2.409
CSOGD-I	78.161 ± 0.153	69.098 ± 0.391	**87.223 ± 0.134**	1.604	5.995 ± 0.061	78.832 ± 0.298	81.120 ± 0.109	1.756
ARCSOGD	**79.831 ± 0.096**	**73.385 ± 0.264**	86.277 ± 0.096	17.937	5.476 ± 0.055	81.163 ± 0.288	81.347 ± 0.152	19.509
ARCSOGD$_{diag}$	79.727 ± 0.086	73.236 ± 0.191	86.217 ± 0.114	2.122	**5.470 ± 0.058**	**81.617 ± 0.351**	80.141 ± 0.299	2.466
Algorithm	"sum" on ijcnn1				"cost" on ijcnn1			
	Sum(%)	Sensitivity(%)	Specificity (%)	Time (s)	Cost(%)	Sensitivity(%)	Specificity (%)	Time (s)
Perceptron	70.298 ± 0.123	46.285 ± 0.222	94.311 ± 0.024	3.285	5.139 ± 0.018	46.319 ± 0.189	94.314 ± 0.020	3.863
PA-I	69.241 ± 0.143	43.872 ± 0.271	94.610 ± 0.038	4.916	5.324 ± 0.027	43.865 ± 0.299	**94.607 ± 0.033**	6.104
PAUM	81.410 ± 0.120	68.851 ± 0.243	93.970 ± 0.038	3.710	3.256 ± 0.024	68.489 ± 0.281	94.022 ± 0.039	4.446
CPA$_{PB}$	73.003 ± 0.170	55.937 ± 0.286	90.070 ± 0.066	5.173	4.704 ± 0.026	55.718 ± 0.263	90.176 ± 0.056	5.870
CSOGD-I	81.410 ± 0.120	68.851 ± 0.243	93.970 ± 0.038	4.357	3.176 ± 0.030	68.837 ± 0.370	94.569 ± 0.043	4.795
ARCSOGD	**86.048 ± 0.132**	**77.298 ± 0.278**	**94.798 ± 0.072**	5.973	**2.410 ± 0.015**	**77.729 ± 0.207**	94.569 ± 0.051	6.604
ARCSOGD$_{diag}$	85.307 ± 0.300	76.304 ± 0.646	94.310 ± 0.068	4.632	2.414 ± 0.014	77.668 ± 0.213	94.584 ± 0.070	5.200
Algorithm	"sum" on w8a				"cost" on w8a			
	Sum(%)	Sensitivity(%)	Specificity (%)	Time (s)	Cost(%)	Sensitivity(%)	Specificity (%)	Time (s)
Perceptron	76.549 ± 0.314	54.501 ± 0.609	98.597 ± 0.019	1.886	1.367 ± 0.020	54.240 ± 0.676	98.589 ± 0.021	1.721
PA-I	76.622 ± 0.368	54.361 ± 0.737	**98.884 ± 0.036**	2.678	1.345 ± 0.024	54.027 ± 0.850	**98.878 ± 0.030**	2.342
PAUM	80.371 ± 0.416	62.297 ± 0.865	98.445 ± 0.047	2.141	1.137 ± 0.015	61.868 ± 0.567	98.849 ± 0.029	1.802
CPA$_{PB}$	80.949 ± 0.290	65.354 ± 0.586	96.544 ± 0.060	2.683	1.252 ± 0.022	62.636 ± 0.768	97.450 ± 0.032	2.184
CSOGD-I	82.170 ± 0.307	66.244 ± 0.617	98.095 ± 0.025	2.298	1.106 ± 0.014	64.315 ± 0.513	98.489 ± 0.036	1.842
ARCSOGD	84.692 ± 0.279	70.869 ± 0.566	98.515 ± 0.025	10.615	0.911 ± 0.013	70.173 ± 0.508	98.877 ± 0.039	9.959
ARCSOGD$_{diag}$	**85.456 ± 0.303**	**72.742 ± 0.627**	98.170 ± 0.045	2.236	**0.898 ± 0.014**	**70.846 ± 0.533**	98.826 ± 0.030	1.988



(a) *covtype* (b) *spambase*

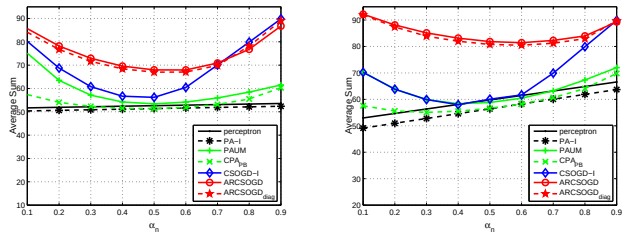


(c) *svmguide3* (d) *a9a*

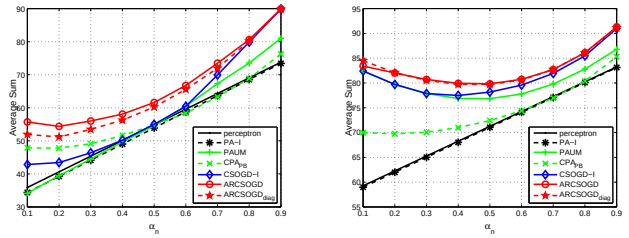


(e) *ijcnn1* (f) *w8a*

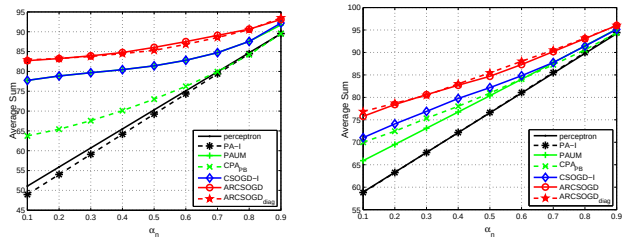
Fig. 2. Evaluation of online average "*cost*" of the proposed algorithms on public datasets.



(a) *covtype* (b) *spambase*



(c) *svmguide3* (d) *a9a*



(e) *ijcnn1* (f) *w8a*

Fig. 3. Evaluation of the weighted "*sum*" under varying weights of sensitivity and specificity.

breast. For this task, the class "benign" is assigned as the normal class, and the class "malignant" is the anomaly class.

- Finance: We apply our algorithms to a credit card approval problem in finance domain. In particular, we work on a data set with 690 instances from an Australian credit company, in which the task is to distinguish credit-worthy customers from non credit-worthy ones.
- Bioinformatics: We apply our algorithms to solve a bioinformatics problem using the "Code-RNA" dataset [45]. The goal of this task is to develop a computational method to detect novel non-coding RNAs from some large sequenced genomes. Non-coding RNAs are defined as anomalies and others are considered as normal instances.
- Nuclear: The "magic04" dataset [46] 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 data and the hadrons are seen as outliers.

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

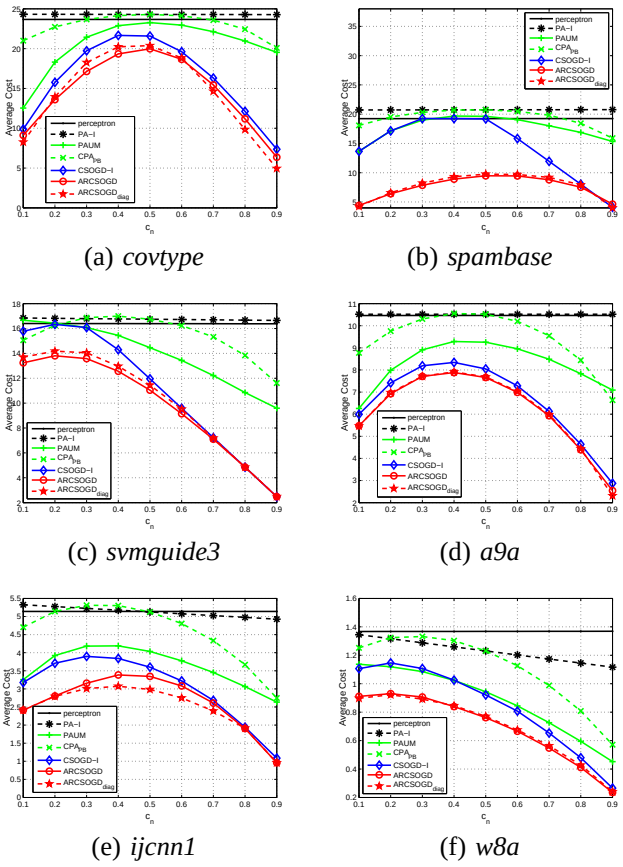
TABLE III. DATA SETS FOR ONLINE ANOMALY DETECTION.

Dataset Name	#Examples	#Features	#Outlier:#Normal
KDDCUP08	102294	117	1:163.19
Australian	690	14	1:1.25
Cod-RNA	271617	8	1:2.00
Magic04	19020	10	1:1.8

B. Empirical Evaluation Results.

We apply our algorithms to solve anomaly detection tasks on the real-world datasets as shown in Table III and evaluate the anomaly detection performance using *balanced accuracy*, which is able to avoid inflated performance estimates on imbalanced datasets. The experimental results are summarized in Table IV.

From the results, we can draw several observations as follows. First of all, among all the existing algorithms, the two cost-sensitive algorithms (PAUM and CPA$_{PB}$) generally perform better than their regular versions (Perceptron and PA-I, respectively), which implies the necessity of introducing cost-sensitiveness for online learning. In addition, all the first four algorithms are outperformed by the CSOGD algorithm on most of the datasets, which demonstrates that is effective to directly optimize cost-sensitive measures. Furthermore, the proposed ARCSOGD significantly outperforms the other algorithms for all the datasets. Because ARCSOGD is a variant of CSOGD with adaptive regularization using second order information,



(a) *covtype* (b) *spambase*

(c) *svmguide3* (d) *a9a*

(e) *ijcnn1* (f) *w8a*

Fig. 4. Evaluation of weighted "*cost*" measure under varying weights for False Positives and False Negatives.

TABLE IV. EVALUATION OF BALANCED ACCURACY PERFORMANCE FOR ONLINE ANOMALY DETECTION.

Algorithm	KDDCUP08	Australian
Perceptron	58.583 ± 0.586	57.543 ± 1.944
PA-I	56.810 ± 0.546	57.367 ± 2.182
PAUM	56.464 ± 0.686	60.657 ± 2.012
CPA$_{PB}$	65.382 ± 0.698	57.643 ± 2.311
CSOGD-I	61.980 ± 0.624	65.892 ± 0.570
ARCOSGD	**67.169 ± 0.581**	**68.163 ± 0.843**
ARCOSGD$_{diag}$	**66.639 ± 0.542**	**68.070 ± 0.956**
Algorithm	**Cod-RNA**	**Magic04**
Perceptron	75.742 ± 0.355	59.146 ± 0.260
PA-I	73.654 ± 0.147	57.336 ± 0.200
PAUM	80.943 ± 0.104	61.242 ± 0.254
CPA$_{PB}$	74.473 ± 0.115	57.906 ± 0.289
CSOGD-I	81.095 ± 0.149	65.869 ± 0.193
ARCOSGD	**86.539 ± 0.075**	**72.310 ± 0.187**
ARCOSGD$_{diag}$	86.118 ± 0.042	71.448 ± 0.589

this implies the effectiveness of introducing second order information for improving cost-sensitive online learning efficacy.

It can also be observed that the diagonal version of ARCSOGD performs comparably with ARCSOGD. Since the computational complexity of ARCSOGD$_{diag}$ is the same as those of the first order algorithms, it can be a good choice for high dimension problems, when it is too expensive to maintain and update a full matrix. In all, the promising results validate the advantages of the proposed algorithms for solving real-world online anomaly detection tasks which are often highly class-imbalanced.

VI. CONCLUSION

In this paper, to overcome the limitation of first order cost-sensitive online learning algorithms, we studied cost-sensitive online classification with adaptive regularization. Specifically, we proposed a second order cost-sensitive online classification algorithm, i.e., ARCSOGD, and theoretically analyzed its regret bound. We further empirically evaluate the proposed algorithm on several public real-world datasets. The promising experimental results demonstrate the effectiveness of the proposed algorithm.

APPENDIX

This section presents the proofs for all the theorems.

A. Proof of Theorem 1

Proof: It is easy to verify that $\mu_{t+1} = \arg\min_\mu h_t(\mu)$ where $h_t(\mu) = \frac{1}{2}\|\mu_t - \mu\|^2_{\Sigma^{-1}_{t+1}} + \eta \mathbf{g}_t^\top \mu$. Because h_t is convex, we have

$$\begin{aligned}&\partial h_t(\mu_{t+1})^\top(\mu - \mu_{t+1})\\ &= [(\mu_{t+1} - \mu_t)^\top \Sigma^{-1}_{t+1} + \eta \mathbf{g}_t^\top](\mu - \mu_{t+1}) \geq 0, \forall \mu.\end{aligned}$$

Re-arranging the above inequality will result in

$$\begin{aligned}(\eta \mathbf{g}_t)^\top(\mu_{t+1} - \mu) &\leq (\mu_{t+1} - \mu_t)^\top \Sigma^{-1}_{t+1}(\mu - \mu_{t+1})\\ &= \frac{1}{2}[\|\mu_t - \mu\|^2_{\Sigma^{-1}_{t+1}} - \|\mu_{t+1} - \mu\|^2_{\Sigma^{-1}_{t+1}}\\ &\quad -\|\mu_t - \mu_{t+1}\|^2_{\Sigma^{-1}_{t+1}}].\end{aligned}$$

Since $\ell_t(\mu)$ is convex, we have

$$\begin{aligned}\mathbf{g}_t^\top(\mu_{t+1} - \mu) &= \mathbf{g}_t^\top(\mu_t - \mu + \mu_{t+1} - \mu_t)\\ &\geq \ell_t(\mu_t) - \ell_t(\mu) + \mathbf{g}_t^\top(\mu_{t+1} - \mu_t).\end{aligned}$$

Combining the above two inequalities, will give the following important inequality

$$\begin{aligned}\ell_t(\mu_t) - \ell_t(\mu) \leq \frac{1}{2\eta}[&\|\mu_t - \mu\|^2_{\Sigma^{-1}_{t+1}} - \|\mu_{t+1} - \mu\|^2_{\Sigma^{-1}_{t+1}}\\ &-\|\mu_t - \mu_{t+1}\|^2_{\Sigma^{-1}_{t+1}}] - \mathbf{g}_t^\top(\mu_{t+1} - \mu_t).\end{aligned}$$

Summing the above inequality over $t = 1, 2, \ldots, T$, gives

$$\begin{aligned}&\sum_{t=1}^T [\ell_t(\mu_t) - \ell_t(\mu)]\\ &\leq \frac{1}{2\eta}\sum_{t=1}^T \left[\|\mu_t - \mu\|^2_{\Sigma^{-1}_{t+1}} - \|\mu_{t+1} - \mu\|^2_{\Sigma^{-1}_{t+1}}\right]\\ &-\frac{1}{2\eta}\sum_{t=1}^T \|\mu_t - \mu_{t+1}\|^2_{\Sigma^{-1}_{t+1}} - \sum_{t=1}^T \mathbf{g}_t^\top(\mu_{t+1} - \mu_t). \quad (4)\end{aligned}$$

Now, we would like to bound the right hand side of the above inequality. Firstly, we bound the first term as

$$\begin{aligned}
&\sum_{t=1}^{T}\Big[\|\mu_t-\mu\|^2_{\Sigma_{t+1}^{-1}}-\|\mu_{t+1}-\mu\|^2_{\Sigma_{t+1}^{-1}}\Big]\\
&\leq \|\mu_1-\mu\|^2_{\Sigma_2^{-1}}+\sum_{t=2}^{T}\Big[\|\mu_t-\mu\|^2_{\Sigma_{t+1}^{-1}}-\|\mu_t-\mu\|^2_{\Sigma_t^{-1}}\Big]\\
&=\|\mu_1-\mu\|^2_{\Sigma_2^{-1}}+\sum_{t=2}^{T}\Big[\|\mu_t-\mu\|^2_{(\Sigma_{t+1}^{-1}-\Sigma_t^{-1})}\Big]\\
&\leq \|\mu_1-\mu\|^2\lambda_{max}(\Sigma_2^{-1})+\sum_{t=2}^{T}\|\mu_t-\mu\|^2\lambda_{max}(\Sigma_{t+1}^{-1}-\Sigma_t^{-1})\\
&\leq \|\mu_1-\mu\|^2\mathrm{Tr}(\Sigma_2^{-1})+\sum_{t=2}^{T}\|\mu_t-\mu\|^2\mathrm{Tr}(\Sigma_{t+1}^{-1}-\Sigma_t^{-1})\\
&\leq \max_{t\leq T}\|\mu_t-\mu\|^2\mathrm{Tr}(\Sigma_2^{-1})+\sum_{t=2}^{T}\max_{t\leq T}\|\mu_t-\mu\|^2\mathrm{Tr}(\Sigma_{t+1}^{-1}-\Sigma_t^{-1})\\
&=\max_{t\leq T}\|\mu_t-\mu\|^2\mathrm{Tr}(\Sigma_{T+1}^{-1}), \qquad (5)
\end{aligned}$$

where $\lambda_{max}(\Sigma)$ is the largest eigenvalue of Σ.

Next, we will bound the remaining terms. To this end, we notice that the following inequality holds according to the update rule of μ

$$(\mu_{t+1}-\mu_t)^\top\Sigma_{t+1}^{-1}+\eta\mathbf{g}_t^\top=0,$$

so that

$$\begin{aligned}
\|\mu_t-\mu_{t+1}\|^2_{\Sigma_{t+1}^{-1}}&=(\mu_{t+1}-\mu_t)^\top\Sigma_{t+1}^{-1}\Sigma_{t+1}\Sigma_{t+1}^{-1}(\mu_{t+1}-\mu_t)\\
&=\eta^2\mathbf{g}_t^\top\Sigma_{t+1}\mathbf{g}_t,
\end{aligned}$$

and

$$\mathbf{g}_t^\top(\mu_{t+1}-\mu_t)=-\eta\mathbf{g}_t^\top\Sigma_{t+1}\mathbf{g}_t.$$

Combining the above two inequalities results in

$$\begin{aligned}
&-\frac{1}{2\eta}\sum_{t=1}^{T}\|\mu_t-\mu_{t+1}\|^2_{\Sigma_{t+1}^{-1}}-\sum_{t=1}^{T}\mathbf{g}_t^\top(\mu_{t+1}-\mu_t)\\
&=\sum_{t=1}^{T}\eta\mathbf{g}_t^\top\Sigma_{t+1}\mathbf{g}_t-\sum_{t=1}^{T}\frac{\eta}{2}\mathbf{g}_t^\top\Sigma_{t+1}\mathbf{g}_t\\
&=\frac{\eta}{2}\sum_{t=1}^{T}\mathbf{g}_t^\top\Sigma_{t+1}\mathbf{g}_t. \qquad (6)
\end{aligned}$$

Plugging the above two upper bounds (5) and (6) into the inequality (4), we can get

$$\begin{aligned}
\sum_{t=1}^{T}[\ell_t(\mu_t)-\ell_t(\mu)]&\leq\frac{1}{2\eta}\max_{t\leq T}\|\mu_t-\mu\|^2\mathrm{Tr}(\Sigma_{T+1}^{-1})\\
&\quad+\frac{\eta}{2}\sum_{t=1}^{T}\mathbf{g}_t^\top\Sigma_{t+1}\mathbf{g}_t \qquad (7)
\end{aligned}$$

As we know

$$\mathbf{g}_t=L_t y_t\mathbf{x}_t$$

where $L_t=1$, if $\ell_t(\mu_t)>0$, and $L_t=0$, otherwise, we can bound $\sum_{t=1}^{T}\mathbf{g}_t^\top\Sigma_{t+1}\mathbf{g}_t$ as follows,

$$\begin{aligned}
\sum_{t=1}^{T}\mathbf{g}_t^\top\Sigma_{t+1}\mathbf{g}_t&=\sum_{t=1}^{T}L_t\mathbf{x}_t^\top\Sigma_{t+1}\mathbf{x}_t=\gamma\sum_{t=1}^{T}(1-\frac{|\Sigma_t^{-1}|}{|\Sigma_{t+1}^{-1}|})\\
&\leq-\gamma\sum_{t=1}^{T}\log(\frac{|\Sigma_t^{-1}|}{|\Sigma_{t+1}^{-1}|})\leq\gamma\log(|\Sigma_{T+1}^{-1}|), \qquad (8)
\end{aligned}$$

where we used

$$\Sigma_{t+1}^{-1}=\Sigma_t^{-1}+\frac{L_t\mathbf{x}_t\mathbf{x}_t^\top}{\gamma}\Rightarrow\frac{L_t}{\gamma}\mathbf{x}_t^\top\Sigma_{t+1}\mathbf{x}_t=1-\frac{|\Sigma_t^{-1}|}{|\Sigma_{t+1}^{-1}|}.$$

Plugging (8) into the inequality (7) concludes the proof. ■

B. Proof of Theorem 2

Proof: For ARCSOGD, if $t\in\mathcal{M}_p$, $\ell_t(\mu_t)\geq\rho$ and $t\in\mathcal{M}_n$, $\ell_t(\mu_t)\geq1$, we have

$$\rho M_p+M_n\leq\sum_{t=1}^{T}\ell_t(\mu_t). \qquad (9)$$

From the definition of sum, we know that

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

Setting $\rho=\frac{\eta_p T_n}{\eta_n T_p}$ and combining the above inequality with the regret bound in theorem 1 concludes the proof. ■

C. Proof of Theorem 3

Proof: From the definition of $cost$, we know that

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

Setting $\rho=\frac{c_p}{c_n}$ and combining it with inequality(9), we have

$$c_n(\rho M_p+M_n)\leq c_n\sum_{t=1}^{T}\ell_t(\mu_t)$$

Combining the above inequality with theorem 1 will prove this theorem. ■

REFERENCES

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

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

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

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

[5] J. Wang, P. Zhao, and S. C. H. Hoi, "Exact soft confidence-weighted learning," in *ICML*, 2012.

[6] P. Zhao and S. C. H. Hoi, "Cost-sensitive online active learning with application to malicious URL detection," in *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013*, 2013, pp. 919–927.

[7] J. Wang, P. Zhao, and S. C. H. Hoi, "Cost-sensitive online classification," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 10, pp. 2425–2438, 2014. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/TKDE.2013.157

[8] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Learning to detect malicious urls," *ACM TIST*, vol. 2, no. 3, p. 30, 2011. [Online]. Available: http://doi.acm.org/10.1145/1961189.1961202

[9] B. Li, S. C. H. Hoi, P. Zhao, and V. Gopalkrishnan, "Confidence weighted mean reversion strategy for online portfolio selection," *TKDD*, vol. 7, no. 1, p. 4, 2013.

[10] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter, "Pegasos: primal estimated sub-gradient solver for SVM," *Math. Program.*, vol. 127, no. 1, pp. 3–30, 2011.

[11] C. Elkan, "The foundations of cost-sensitive learning," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2001, pp. 973–978.

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

[13] K. Veropoulos, C. Campbell, and N. Cristianini, "Controlling the sensitivity of support vector machines," in *Proceedings of the International Joint Conference on AI*, 1999, pp. 55–60.

[14] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *J. Artif. Intell. Res. (JAIR)*, vol. 16, pp. 321–357, 2002.

[15] 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*, 2003, pp. 1–8.

[16] H. Masnadi-Shirazi and N. Vasconcelos, "Risk minimization, probability elicitation, and cost-sensitive svms," in *ICML*, 2010, pp. 759–766.

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

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

[19] R. Akbani, S. Kwek, and N. Japkowicz, "Applying support vector machines to imbalanced datasets," in *ECML*, 2004, pp. 39–50.

[20] M. Dredze, K. Crammer, and F. Pereira, "Confidence-weighted linear classification," in *ICML*, 2008, pp. 264–271.

[21] K. Crammer, M. Dredze, and F. Pereira, "Exact convex confidence-weighted learning," in *Advances in Neural Information Processing Systems (NIPS)*, 2008, pp. 345–352.

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

[23] 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*, ser. ICDM '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 970–974.

[24] 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*, ser. ICDM '03. Washington, DC, USA: IEEE Computer Society, 2003, pp. 435–. [Online]. Available: http://dl.acm.org/citation.cfm?id=951949.952181

[25] 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.*, vol. 18, no. 10, pp. 1435–1440, Oct. 2006.

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

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

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

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

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

[31] K. Crammer and Y. Singer, "Ultraconservative online algorithms for multiclass problems," *JMLR*, vol. 3, pp. 951–991, 2003.

[32] N. Cesa-Bianchi, A. Conconi, and C. Gentile, "On the generalization ability of on-line learning algorithms," *IEEE Trans. on Inf. Theory*, vol. 50, no. 9, pp. 2050–2057, 2004.

[33] M. Fink, S. Shalev-Shwartz, Y. Singer, and S. Ullman, "Online multiclass learning by interclass hypothesis sharing," in *ICML*, 2006, pp. 313–320.

[34] P. Zhao and S. C. H. Hoi, "OTL: A framework of online transfer learning," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, 2010, pp. 1231–1238. [Online]. Available: http://www.icml2010.org/papers/219.pdf

[35] P. Zhao, S. C. H. Hoi, R. Jin, and T. Yang, "Online AUC maximization," in *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, 2011, pp. 233–240.

[36] S. C. H. Hoi, J. Wang, and P. Zhao, "LIBOL: a library for online learning algorithms," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 495–499, 2014. [Online]. Available: http://dl.acm.org/citation.cfm?id=2627450

[37] D. Wang, P. Wu, P. Zhao, Y. Wu, C. Miao, and S. C. H. Hoi, "High-dimensional data stream classification via sparse online learning," in *2014 IEEE International Conference on Data Mining, ICDM 2014, Shenzhen, China, December 14-17, 2014*, 2014, pp. 1007–1012. [Online]. Available: http://dx.doi.org/10.1109/ICDM.2014.46

[38] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psych. Rev.*, vol. 7, pp. 551–585, 1958.

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

[40] C. Gentile, "A new approximate maximal margin classification algorithm," *JMLR*, vol. 2, pp. 213–242, 2001.

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

[42] N. Cesa-Bianchi, A. Conconi, and C. Gentile, "A second-order perceptron algorithm," *SIAM J. Comput.*, vol. 34, no. 3, pp. 640–668, 2005.

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

[44] J. Abernethy, P. L. Bartlett, A. Rakhlin, and A. Tewari, "Optimal stragies and minimax lower bounds for online convex games," in *21st Annual Conference on Learning Theory - COLT 2008, Helsinki, Finland, July 9-12, 2008*, 2008, pp. 415–424. [Online]. Available: http://colt2008.cs.helsinki.fi/papers/111-Abernethy

[45] 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, p. 173, 2006.

[46] B. R. B, "Methods for multidimensional event classification: a case study using images from a cherenkov gamma-ray telescope," *Nuclear Instruments and Methods*, 2004.