8-2020

A unified framework for sparse online learning

Peilin ZHAO
Tencent AL Lab

Dayong WONG
PathAI

Pengcheng WU
DeepIR

Steven C. H. HOI
Singapore Management University, chhoi@smu.edu.sg

Citation

ZHAO, Peilin; WONG, Dayong; WU, Pengcheng; and HOI, Steven C. H.. A unified framework for sparse online learning. (2020). *ACM Transactions on Knowledge Discovery from Data*. 14, (5), 1-18.
Available at: https://ink.library.smu.edu.sg/sis_research/5957

Published in ACM Transactions on Knowledge Discovery from Data, 2020 August, 14 (5), Article number 3361559.
DOI: 10.1145/3361559

A Unified Framework for Sparse Online Learning

PEILIN ZHAO, Tencent AI Lab
DAYONG WANG, PathAI
PENGCHENG WU, DeepIR
STEVEN C. H. HOI, Singapore Management University

The amount of data in our society has been exploding in the era of big data. This article aims to address several open challenges in big data stream classification. Many existing studies in data mining literature follow the batch learning setting, which suffers from low efficiency and poor scalability. To tackle these challenges, we investigate a unified online learning framework for the big data stream classification task. Different from the existing online data stream classification techniques, we propose a unified Sparse Online Classification (SOC) framework. Based on SOC, we derive a second-order online learning algorithm and a cost-sensitive sparse online learning algorithm, which could successfully handle online anomaly detection tasks with the extremely unbalanced class distribution. As the performance evaluation, we analyze the theoretical bounds of the proposed algorithms and conduct an extensive set of experiments. The encouraging experimental results demonstrate the efficacy of the proposed algorithms over the state-of-the-art techniques on multiple data stream classification tasks.

CCS Concepts: • **Computing methodologies → Machine learning**;

Additional Key Words and Phrases: Online learning, sparse learning, classification, cost-sensitive learning

1 INTRODUCTION

In the era of big data, the amount of data in our society has been exploding, which has raised many opportunities and challenges for data analytic/mining research. In this article, we aim to tackle the emerging real-world big data stream classification problem, e.g., web-scale spam email classification. The big data stream classification task has the following five "high" characteristics:

> **High volume:** one has to deal with a huge amount of existing training data, in millions or even billion scales.

Authors' addresses: P. Zhao, Tencent AI Lab, 10 Gaoxin 6th Road, Nanshan District, Shenzhen, China; email: masonzhao@tencent.com; D. Wang (corresponding author), PathAI, 120 Brookline Ave, Boston, MA, USA 02481; email: dayong.wang@pathai.com; P. Wu (corresponding author), DeepIR, #910, Jordan Center, 86 Anling 2nd St. Huli District, Xiamen, China 361006; email: tim@deepir.com; S. C. H. Hoi (corresponding author), School of Information Systems, Singapore Management University, 80 Stamford Road, Singapore 178902; email: chhoi@smu.edu.sg.

High velocity: new data inputs arrive sequentially and rapidly, e.g., around 182.9 billion emails are sent all over the worldwide in one day according to the email statistic report, released by the Radicati Group [25].
High dimensionality: there are many features, e.g., for the spam email classification task, the length of the vocabulary list can go up from 10,000 to 50,000 or even to million scales.
High sparsity: in the high-dimensional feature, many elements are zero. Hence the fraction of active features is relatively small, e.g., according to the spam email classification study [35], accuracy saturates with dozens of features out of tens of thousands of features.
High class-imbalance: in many real-world applications, some class considerably dominates the others, e.g., for spam email classification, the number of non-spam emails is much larger than the number of spam emails.

These five characteristics present enormous challenges for big data stream classification tasks when using conventional data stream classification techniques. In general, the conventional algorithms are batch-learning based, which suffers a series of critical drawbacks: (i). They need a large memory capacity to cache examples. (ii). It is expensive to train/retrain the classification model over the entire dataset. (iii). They need all training instances available in advance, which is unpractical in many real-world data stream applications where data instances come rapidly in a sequential manner.

The online learning algorithm is a promising way to tackle those challenges, which conducts incremental training with streaming data in a sequential manner. Typically, an online learning algorithm processes one receiving instance at a time and makes a minor update repeatedly. The online algorithms are more efficient and comfortable, comparing with the batch learning algorithms, to re-train any existing model with new receiving data instances.

In literature, a large variety of online learning algorithms have been proposed, including the first-order algorithms [7, 26] and the second-order algorithms [3, 5, 27]. However, traditional online learning algorithms are limited for high-dimensional data, since they usually will learn non-sparse models, which suffer from low efficiency and take expensive computational costs for both training and test phases. Hence, sparse online learning (SOL) [19] is proposed to tackle this problem by using the sparsity penalty regularizer.

In this article, we introduce a unified framework of SOL for solving large-scale high-dimensional data stream classification.[1] We demonstrate that using the proposed framework, we can easily derive an existing first-order sparse online classification algorithm and further derive a new second-order algorithm. We provide a theoretical analysis of the proposed algorithm and conduct an extensive set of experiments. All evaluation results show that the proposed algorithm can achieve state-of-the-art performance. We organize the rest of this article as follows: Section 2 reviews related works; Section 3 drives the problem formulation; Section 4 presents the unified framework; and Section 5 discusses the experimental results. Our main contributions are summarized as follows:

- We propose a unified online learning framework, which can easily derive first-order and second-order algorithms.
- We provide a series of theoretical analyses, e.g., general regret and mistake bounds.

[1] Our preliminary work on this topic appeared in IEEE International Conference on Data Mining (ICDM), 2014 [30]. Adequate new contributions have been augmented into this article, including but not limited to (i) providing comprehensive theoretical bound analysis for the unified SOL framework, (ii) driving cost-sensitive algorithm, and (iii) doing additional experiments to demonstrate the performances of the proposed new algorithms.

—We evaluate the proposed algorithms on several high-dimensional and large-scale benchmark databases, in which we achieved the state-of-the-art performances.

2 RELATED WORK

2.1 Online Learning

Online learning represents a family of efficient and scalable machine learning algorithms [16]. Unlike batch learning methods that suffer from expensive re-training cost, online learning works sequentially by performing highly efficient (typically constant) updates for each new training data, making it highly scalable for data stream classification. In literature, various techniques [6, 7, 9, 10, 26, 31, 39] have been proposed for online learning. The well-known first-order online learning algorithms include Perceptron [14, 26], Passive-Aggressive (PA) algorithms [7], and the like.

One well-known method is the Perceptron algorithm [14, 26], which updates the model by adding a new example as a support vector with some constant weight. Recently, a series of sophisticated online learning algorithms have been proposed by following the criterion of maximum margin learning principle [7, 15, 18]. One popular algorithm is the PA algorithm [7], which evolves a classifier by suffering less loss on the current instance without moving far from the previous function.

In recent years, researchers frequently use convex optimization tools for the design of efficient online learning algorithms. Furthermore, most of previously proposed efficient online algorithms can be jointly analyzed with the following elegant model [28]:

ALGORITHM 1: Online Convex Optimization Scheme

```
INPUT: A convex set $\mathbb{R}^d$.
for $t = 1, \ldots, T$ do
    predict a vector $\mathbf{w}_t \in \mathbb{R}^d$;
    receive a convex loss function $\ell_t : S \rightarrow \mathbb{R}$;
    suffer loss $\ell_t(\mathbf{w}_t)$;
end for
```

Based on the previous framework, we can consider online learning as an algorithmic framework for convex online learning problem:

$$\min_{\mathbf{w}} f(\mathbf{w}) = \min_{\mathbf{w}} \sum_t \ell_t(\mathbf{w}),$$

where $f(\mathbf{w})$ is a convex empirical loss function for the sum of losses over a sequence of observations. The regret of the algorithm is defined as follows:

$$R_T = \sum_{t=1}^{T} \ell_t(\mathbf{w}_t) - \min_{\mathbf{w}} \sum_{t=1}^{T} \ell_t(\mathbf{w}),$$

where $\mathbf{w}$ is any vector in the convex space $\mathbb{R}^d$. The goal of online learning algorithm is to find a low regret scheme, in which the regret R_T grows sub-linearly with the number of iteration T. Thus, when the round number T goes to infinity, the difference between the *average* loss of the learner and the *average* loss of the best learner tends to zero.

Although the general online learning algorithms (e.g., Perceptron and PA) have solid theoretical guarantees and perform well on many applications, they are limited in several aspects. First, the general online learning algorithms exploit the full features, which is not suitable for the large-scale high-dimensional problem. *SOL* has been extensively studied recently to tackle this limitation. Second, the general online learning algorithms only exploit the first-order information, and all features are adopted the same learning rate. This problem can be addressed by *second order online learning* algorithms. Last but not least, the general online learning algorithms are not suitable for the imbalance input data streams, which can be efficiently solved by the *cost-sensitive online learning* algorithms. In the following parts, we will briefly introduce several representative algorithms in the previous three aspects.

2.2 Sparse Online Learning

SOL [12, 19] aims to learn a sparse linear classifier, which only contains limited size of active features. It has been actively studied [12, 29, 32, 34]. There are two groups of solutions for *SOL*. The first group study on SOL follows the general idea of subgradient descent with truncation. For example, Duchi and Singer propose the FOBOS algorithm [12], which extends the *Forward-Backward Splitting* method to solve the SOL problem in the following two phases: (i) an unconstrained subgradient descent step; and (ii) an instantaneous optimization for a trade-off between minimizing regularization term and keeping close to the result obtained in the first phase. We can solve the optimization problem in the second phase by adopting simple *soft-thresholding* operations that perform some truncation on the weight vectors. Following a similar scheme, Langford et al. [19] argued that the truncation in each iteration is too aggressive as each step modifies the coefficients by only a small amount. They proposed the *Truncated Gradient* (TG) method, which truncates coefficients every K steps when they are less than a predefined threshold of θ. The second group study on SOL mainly follows the dual averaging method of [24], which can explicitly exploit the regularization structure in an online setting. For example, One representative work is *Regularized Dual Averaging* (RDA) [34], which learns the variables by solving a simple optimization problem that involves the running average of all past subgradients of the loss functions, not just the subgradient in each iteration. Lee et al. [20] further extend the RDA algorithm by using a more aggressive truncation threshold and generates significantly more sparse solutions.

2.3 Second-order Online Learning

Second Order Online Learning aims to dynamically incorporate knowledge of observed data in the earlier iteration to perform more informative gradient-based learning. Unlike first-order algorithms that often adopt the same learning rate for all coordinates, the second-order online learning algorithms take different distills to the step size employed for each coordinate. Some new second-order online learning algorithms attempt to incorporate knowledge of the geometry of the data observed in earlier iterations to perform more effective online updates. For example, Balakrishnan et al. [2] proposed algorithms for sparse linear classifiers, which requires $O(d^2)$ time and $O(d^2)$ space in the worst case. Another family of second-order online learning algorithm is using confidence-weighted (CW) learning [8, 9, 10, 23, 31], which exploit confidence of weights when making updates in online learning processes. The second-order algorithms are more accurate and will converge faster. However, they fall short in the following two aspects: (i) they incur a higher computational cost, especially when dealing with high-dimensional data; and (ii) the learned weight vectors are dense. Recently, Duchi et al. address the sparsity and the second-order update in the same framework. They proposed the Adaptive Subgradient method [11] (Ada-RDA), which adaptively modifies the proximal function at each iteration to incorporate knowledge about the geometry of the data.

2.4 Cost-Sensitive Online Learning

The cost-sensitive classification has been extensively studied in data mining and machine learning: for example, the weighted sum of *sensitivity* and *specificity* [4], and the weighted *misclassification cost* [1, 13]. Although both cost-sensitive classification and online learning have been studied extensively in data mining and machine learning communities, respectively. There are only a few works on *cost-sensitive online learning*. For example, Wang et al. [33] proposed a family of cost-sensitive online classification framework, which directly optimized two well-known cost-sensitive measures. Zhao and Hoi [38] tackled the same problem by adopting the double updating technique and propose Cost-Sensitive Double Updating Online Learning (CSDUOL). Zhao et al. further adopt adaptive regularization on cost-sensitive online classification problem [40, 41], which can significantly reduce the regret bound.

For more related work on online learning, please refer to a survey [17], which provides more details for various online learning algorithms and their applications.

3 SPARSE ONLINE LEARNING FOR DATA STREAM CLASSIFICATION

In this section, we propose a general SOL framework for online data stream classification, then we derive the regret of the proposed framework, which is used to derive a family of first-order and second-order sparse online classification algorithms.

3.1 General Sparse Online Learning

Without losing the generality, we consider the SOL algorithm for binary classification problems. The sparse online classification algorithm generally works in rounds, where one instance $\mathbf{x}_t \in \mathbb{R}^d$ is provided at the t-th round and the online algorithm predicts its label as:

$$\hat{y}_t = \mathrm{sign}(\mathbf{w}_t^\top \mathbf{x}_t),$$

where $\mathbf{w}_t \in \mathbb{R}^d$ is a linear classifier. Given the prediction $\hat{y}_t$ and the ground truth label $y_t \in \{+1, -1\}$, the algorithm has a loss $\ell_t(\mathbf{w}_t)$. For example, the following hinge loss:

$$\ell_t(\mathbf{w}) = [1 - y_t \mathbf{w}^\top \mathbf{x}_t]_+, \text{ where } [a]_+ = \max(a, 0), \tag{1}$$

is the most popular loss function for binary classification problems. Then, the algorithm updates the classifier parameters $\mathbf{w}_t$. The optimal goal of online learning is to minimize the number of mistakes. Given a series of δ-strongly convex functions $\Phi_{t=1,\ldots,T}$, with respect to the norms $\|\cdot\|_{\Phi_t}$ and the dual norms $\|\cdot\|_{\Phi_t^*}$, we define a general sparse online classification (SOL) algorithm, as shown in Algorithm 2:

ALGORITHM 2: General Sparse Online Learning (SOL)

```
INPUT: sparse parameter $\lambda$ and learning rate $\eta$.
INITIALIZATION: $\theta_1 = 0$.
for $t = 1, \ldots, T$ do
    receive $\mathbf{x}_t \in \mathbb{R}^d$;
    $\mathbf{u}_t = \nabla \Phi_t^*(\theta_t)$;
    $\mathbf{w}_t = \arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{u}_t - \mathbf{w}\|_2^2 + \lambda_t \|\mathbf{w}\|_1$;
    predict $\hat{y}_t = \mathrm{sign}(\mathbf{w}_t^\top \mathbf{x}_t)$;
    receive $y_t$ and suffer $\ell_t(\mathbf{w}_t) = [1 - y_t \mathbf{w}_t^\top \mathbf{x}_t]_+$;
    if $\ell_t(\mathbf{w}_t) > 0$ then
        $\theta_{t+1} = \theta_t - \eta_t \nabla \ell_t(\mathbf{w}_t)$;
    end if
end for
```

3.2 Theoretical Bound

In this section, we derive the regret R_T of the general SOL framework in Algorithm 2.

Lemma 3.1. *Let $\Phi_{t,t=1,\dots,T}$ be a set of δ-strongly convex functions with respect to the norm $\|\cdot\|_{\Phi_t}$ and dual norm $\|\cdot\|_{\Phi_t^*}$. Let $\Phi_0(\cdot) = 0$ and $\mathbf{x}_1, \dots, \mathbf{x}_T$ be an arbitrary sequence of vectors in $\mathbb{R}^d$. Assuming that Algorithm 2 is adopted using the aforementioned sequence with function Φ_t, for any $\mathbf{w}$ and any $\lambda > 0$ we have*

$$\sum_{t=1}^{T} \eta_t (\mathbf{w}_t - \mathbf{w})^\top \mathbf{z}_t \leq \Phi_T(\mathbf{w}) + \sum_{t=1}^{T} \left[\Phi_t^*(\theta_t) - \Phi_{t-1}^*(\theta_t) + \frac{\eta_t^2}{2\delta} \|\mathbf{z}_t\|_{\Phi_t^*}^2 + \eta_t \lambda_t \|\mathbf{z}_t\|_1 \right], \tag{2}$$

where $\mathbf{z}_t = \nabla \ell_t(\mathbf{w}_t)$ and η_t is the learning rate of t-th iteration.

Proof. First, define $\Delta_t = \Phi_t^*(\theta_{t+1}) - \Phi_{t-1}^*(\theta_t)$, then

$$\sum_{t=1}^{T} \Delta_t = \Phi_T^*(\theta_{T+1}) - \Phi_0^*(\theta_1) = \Phi_T^*(\theta_{T+1}) \geq \mathbf{w}^\top \theta_{T+1} - \Phi_T(\mathbf{w}),$$

where the final inequality is according to Fenchel's inequality. Besides,

$$\Delta_t = \Phi_t^*(\theta_{t+1}) - \Phi_t^*(\theta_t) + \Phi_t^*(\theta_t) - \Phi_{t-1}^*(\theta_t) \leq \Phi_t^*(\theta_t) - \Phi_{t-1}^*(\theta_t) - \eta_t (\nabla \Phi_t^*(\theta_t))^\top \mathbf{z}_t + \frac{\eta_t^2}{2\delta} \|\mathbf{z}_t\|_{\Phi_t^*}^2.$$

Second, by combining the above two inequalities, we derive the inequality

$$-\sum_{t=1}^{T} \eta_t \mathbf{w}^\top \mathbf{z}_t - \Phi_T(\mathbf{w}) \leq \sum_{t=1}^{T} \Delta_t \leq \sum_{t=1}^{T} \left[\Phi_t^*(\theta_t) - \Phi_{t-1}^*(\theta_t) - \eta_t \mathbf{u}_t^\top \mathbf{z}_t + \frac{\eta_t^2}{2\delta} \|\mathbf{z}_t\|_{\Phi_t^*}^2 \right],$$

where $\mathbf{u}_t = \nabla \Phi_t^*(\theta_t)$. By rearranging the above inequalities, we achieve the inequality

$$\sum_{t=1}^{T} \eta_t (\mathbf{u}_t - \mathbf{w})^\top \mathbf{z}_t \leq \Phi_T(\mathbf{w}) + \sum_{t=1}^{T} \left[\Phi_t^*(\theta_t) - \Phi_{t-1}^*(\theta_t) + \frac{\eta_t^2}{2\delta} \|\mathbf{z}_t\|_{\Phi_t^*}^2 \right]. \tag{3}$$

Third, given Algorithm 2, we have

$$\begin{aligned}
\mathbf{w}_t^\top \mathbf{z}_t &= \sum_{i=1}^{d} w_{t,i} z_{t,i} = \sum_{i=1}^{d} \operatorname{sign}(u_{t,i}) [|u_{t,i}| - \lambda_t]_+ z_{t,i} \\
&= \sum_{u_{t,i} z_{t,i} \geq 0} [|u_{t,i}| - \lambda_t]_+ |z_{t,i}| - \sum_{u_{t,i} z_{t,i} < 0} [|u_{t,i}| - \lambda_t]_+ |z_{t,i}| \\
&\leq \sum_{u_{t,i} z_{t,i} \geq 0} |u_{t,i}||z_{t,i}| + \sum_{u_{t,i} z_{t,i} < 0} (-|u_{t,i}||z_{t,i}| + \lambda_t |z_{t,i}|) \\
&\leq \sum_{u_{t,i} z_{t,i} \geq 0} u_{t,i} z_{t,i} + \sum_{u_{t,i} z_{t,i} < 0} (u_{t,i} z_{t,i} + \lambda_t |z_{t,i}|) \leq \mathbf{u}_t^\top \mathbf{z}_t + \lambda_t \|\mathbf{z}_t\|_1.
\end{aligned} \tag{4}$$

Finally, we derive the Lemma 3.1 by combining the above inequalities (3) and (4). □

Given Lemma 3.1, we derive a general corollary, which provides a upper bound of the regret suffered by the proposed framework. Given the property of convex function, we achieve a lower bound of the left-hand side component of inequality (2) based on $\ell_t(\mathbf{w}_t) - \ell_t(\mathbf{w}) \leq (\mathbf{w}_t - \mathbf{w})^\top \mathbf{z}_t$.

Corollary 1. *Based on Lemma 3.1, by assuming ℓ is convex and $\eta_t = \eta$, the regret of the proposed framework (2), R_T, satisfies the following inequality:*

$$R_T = \sum_{t=1}^{T} \ell_t(\mathbf{w}_t) - \min_{\mathbf{w}} \sum_{t=1}^{T} \ell_t(\mathbf{w}) \leq \frac{\Phi_T(\mathbf{w})}{\eta} + \sum_{t=1}^{T} \left[\frac{\eta}{2\delta} \|\mathbf{z}_t\|_{\Phi_t^*}^2 + \lambda_t \|\mathbf{z}_t\|_1 \right] + \frac{\sum_{t=1}^{T} \Delta_t^*}{\eta}, \tag{5}$$

where $\Delta_t^* = \Phi_t^*(\theta_t) - \Phi_{t-1}^*(\theta_t)$.

Using the proposed general framework and the derived Corollary 1, in the following section, we derive a series of specified algorithms and provide corresponding regret bounds, respectively.

4 DERIVED ALGORITHMS

In this section, we first demonstrate the RDA algorithm [34] is a specialized case of the proposed general framework by setting $\Phi_t(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|_2^2$. Then, we derive a family of algorithms by modifying different components in the proposed general framework. In the following sections, we denote $L_t = \mathbb{I}_{(\ell_t(\mathbf{w}_t)>0)}$ as an indicator function, where $\mathbb{I}_v = 1$ if v is true, otherwise $\mathbb{I}_v = 0$.

4.1 First Order Algorithm

$\Phi_t(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|_2^2$ is a 1-strongly convex function with respect to the norm $\|\cdot\|_2$, which owns the dual norm of as itself: $\Phi_t^* = \Phi_t$. Adopting this 1-strongly convex function into the proposed SOL framework, we can directly derive a first-order sparse online learning (FSOL) algorithm, which is equivalent to the RDA algorithm with soft 1-norm regularization [34], shown in the following algorithm.

ALGORITHM 3: First Order Sparse Online Learning (FSOL)

```
INPUT: $\lambda, \eta \geq 0$.
INITIALIZATION: $\theta_1 = 0$.
for $t = 1, \ldots, T$ do
    receive $\mathbf{x}_t \in \mathbb{R}^d$;
    $\mathbf{w}_t = \mathrm{sign}(\theta_t) \odot [|\theta_t| - \lambda_t]_+$;
    predict $\hat{y}_t = \mathrm{sign}(\mathbf{w}_t^\top \mathbf{x}_t)$ and receive $y_t \in \{-1, 1\}$;
    suffer $\ell_t(\mathbf{w}_t) = [1 - y_t \mathbf{w}_t^\top \mathbf{x}_t]_+$;
    $\theta_{t+1} = \theta_t + \eta L_t y_t \mathbf{x}_t$, where $L_t = \mathbb{I}_{(\ell_t(\mathbf{w}_t)>0)}$;
end for
```

Theorem 2. *Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ be a sequence of training examples, where $\mathbf{x}_t \in \mathbb{R}^d$, $y_t \in \{-1, +1\}$ and $\|\mathbf{x}_t\|_1 \leq X$ for all t. By setting $\lambda_t = \eta\lambda$, the regret R_T of Algorithm 3 is bounded as follows:*

$$R_T \leq \frac{\frac{1}{2}\|\mathbf{w}\|_2^2}{\eta} + \frac{\eta}{2}\sum_{t=1}^{T} X^2 + \sum_{t=1}^{T} \eta\lambda X.$$

Proof. Firstly $\Delta_t^* = \Phi_t^*(\theta_t) - \Phi_{t-1}^*(\theta_t) = 0$, then according to corollary (1), we have

$$R_T \leq \frac{\frac{1}{2}\|\mathbf{w}\|_2^2}{\eta} + \sum_{t=1}^{T}\left[\frac{\eta}{2}\|L_t y_t \mathbf{x}_t\|_2^2 + \lambda_t \|L_t y_t \mathbf{x}_t\|_1\right] \leq \frac{\frac{1}{2}\|\mathbf{w}\|_2^2}{\eta} + \frac{\eta}{2}\sum_{t=1}^{T} X^2 + \sum_{t=1}^{T} \eta\lambda X. \qquad \square$$

By setting $\eta = \frac{\|\mathbf{w}\|_2}{\sqrt{(X^2+2\lambda X)T}}$, we achieve $R_T \leq \|\mathbf{w}\|_2\sqrt{(X^2 + 2\lambda X)T}$, which indicates that the regret of this derived algorithm has an upper bound with the order of $O(\sqrt{T})$. The same observation is presented by Xiao et al. [34].

4.2 Second Order Algorithm

Similarly, we can easily derive a second-order algorithm by setting $\Phi_t(\mathbf{w}) = \frac{1}{2}\mathbf{w}^\top A_t \mathbf{w}$, where $A_t = A_{t-1} + \frac{\mathbf{x}_t \mathbf{x}_t^\top}{r}$, $r > 0$ and $A_0 = I$. Φ_t is 1-strongly convex with respect to the norm $\|\mathbf{w}\|_{\Phi_t}^2 = \mathbf{w}^\top A_t \mathbf{w}$ and the dual function $\|\mathbf{w}\|_{\Phi_t^*}^2 = \mathbf{w}^\top A_t^{-1} \mathbf{w}$. By adopting the Woodbury identity, we can incrementally update the inverse of A_t with $A_t^{-1} = A_{t-1}^{-1} - \frac{A_{t-1}^{-1}\mathbf{x}_t\mathbf{x}_t^\top A_{t-1}^{-1}}{r+\mathbf{x}_t^\top A_{t-1}^{-1}\mathbf{x}_t}$. The derived second-order sparse online learning (SSOL) algorithm is shown in Algorithm 4:

ALGORITHM 4: Second Order Sparse Online Learning (SSOL)

```
INPUT: $\lambda, \eta \geq 0$.
INITIALIZATION: $\theta_1 = 0$.
for $t = 1, \dots, T$ do
    receive $\mathbf{x}_t \in \mathbb{R}^d$;
    $A_t^{-1} = A_{t-1}^{-1} - \frac{A_{t-1}^{-1}\mathbf{x}_t\mathbf{x}_t^\top A_{t-1}^{-1}}{r+\mathbf{x}_t^\top A_{t-1}^{-1}\mathbf{x}_t}$;
    $\mathbf{u}_t = A_t^{-1}\theta_t$;
    $\mathbf{w}_t = \text{sign}(\mathbf{u}_t) \odot [|\mathbf{u}_t| - \lambda_t]_+$;
    predict $\hat{y}_t = \text{sign}(\mathbf{w}_t^\top \mathbf{x}_t)$ and receive $y_t \in \{-1, 1\}$;
    suffer $\ell_t(\mathbf{w}_t) = [1 - y_t\mathbf{w}_t^\top \mathbf{x}_t]_+$;
    $\theta_{t+1} = \theta_t + \eta L_t y_t \mathbf{x}_t$;
end for
```

Theorem 3. *Let $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_T, y_T)$ be a sequence of examples, where $\mathbf{x}_t \in \mathbb{R}^d$, $y_t \in \{-1, +1\}$ and $\|\mathbf{x}_t\|_1 \leq X$ for all t. By setting $\lambda_t = \lambda/t$, the regret R_T of Algorithm 4 is bounded as follows:*

$$R_T \leq \frac{\frac{1}{2}(\|\mathbf{w}\|_2^2 + \frac{\sum_{t=1}^T(\mathbf{w}^\top\mathbf{x}_t)^2}{r})}{\eta} + \frac{\eta}{2}rd\log\left(\left(1 + \frac{X^2}{r}T\right)\right) + \lambda X[\log(T) + 1]. \tag{6}$$

Proof. First,

$$\Delta_t^* = \frac{1}{2}\theta_t^\top A_t^{-1}\theta_t - \frac{1}{2}\theta_t^\top A_{t-1}^{-1}\theta_t = -\frac{(\mathbf{x}_t^\top A_{t-1}^{-1}\theta_t)^2}{2(r + \mathbf{x}_t^\top A_{t-1}^{-1}\mathbf{x}_t)} \leq 0.$$

According to Corollary 1, we have the following inequality

$$R_T \leq \frac{\frac{1}{2}\mathbf{w}^\top A_T \mathbf{w}}{\eta} + \sum_{t=1}^T \left[\frac{\eta}{2}L_t\mathbf{x}_t^\top A_t^{-1}\mathbf{x}_t + \lambda_t\|L_t y_t \mathbf{x}_t\|_1\right] \leq \frac{\frac{1}{2}\mathbf{w}^\top A_T \mathbf{w}}{\eta} + \frac{\eta}{2}\sum_{t=1}^T \mathbf{x}_t^\top A_t^{-1}\mathbf{x}_t + X\sum_{t=1}^T \lambda_t.$$

Second,

$$\sum_{t=1}^T \mathbf{x}_t^\top A_t^{-1}\mathbf{x}_t = r\sum_{t=1}^T \left(1 - \frac{\det(A_{t-1})}{\det(A_t)}\right) \leq -r\sum_{t=1}^T \log\left(\frac{\det(A_{t-1})}{\det(A_t)}\right) = r\log(\det(A_T)).$$

Third,

$$R_T \leq \frac{\frac{1}{2}(\|\mathbf{w}\|_2^2 + \frac{\sum_{t=1}^T(\mathbf{w}^\top\mathbf{x}_t)^2}{r})}{\eta} + \frac{\eta}{2}r\log(\det(A_T)) + \lambda X[\log(T) + 1]. \tag{7}$$

Since $A_T = I + \sum_{t=1}^T \frac{\mathbf{x}_t\mathbf{x}_t^\top}{r}$, its eigenvalue μ_i satisfies

$$\mu_i \leq 1 + trace\left(\sum_{t=1}^T \frac{\mathbf{x}_t\mathbf{x}_t^\top}{r}\right) = 1 + \sum_{t=1}^T \frac{\|\mathbf{x}_t\|_2^2}{r}.$$

Hence,

$$\det(A_T) = \prod_{i=1}^d \mu_i \leq \left(1 + \frac{X^2}{r}T\right)^d.$$

Finally, the theorem is proved by plugging the above inequality into the inequality (7). □

According to Theorem 3, comparing with the first-order solution, the regret bound R_T can be further reduce to the order of $O(\log(T))$ by adopting the second order information.

4.3 Diagonal Algorithm

Although the proposed second-order algorithm in Algorithm 4 can significantly reduce the regret R_T, it requires the computational complexity with the order of $O(d^2)$. In order to reduce the computational complexity to the order of $O(d)$, we propose a diagonal algorithm, which only maintains a diagonal matrix instead of a full matrix A_t as shown in Algorithm 5.[2]

ALGORITHM 5: Diagonal Second Order Sparse Online Learning

```
INPUT: $\lambda, \eta \geq 0$.
INITIALIZATION: $\theta_1 = 0$.
for $t = 1, \ldots, T$ do
    receive $\mathbf{x}_t \in \mathbb{R}^d$;
    $A_t^{-1} = A_{t-1}^{-1} - \frac{A_{t-1}^{-1}\mathrm{diag}(\mathbf{x}_t\mathbf{x}_t^\top)A_{t-1}^{-1}}{r+\mathbf{x}_t^\top A_{t-1}^{-1}\mathbf{x}_t}$;
    $\mathbf{u}_t = A_t^{-1}\theta_t$;
    $\mathbf{w}_t = \mathrm{sign}(\mathbf{u}_t) \odot [|\mathbf{u}_t| - \lambda_t]_+$;
    predict $\hat{y}_t = \mathrm{sign}(\mathbf{w}_t^\top \mathbf{x}_t)$ and receive $y_t \in \{-1, 1\}$;
    suffer $\ell_t(\mathbf{w}_t) = [1 - y_t\mathbf{w}_t^\top \mathbf{x}_t]_+$;
    $\theta_{t+1} = \theta_t + \eta L_t y_t \mathbf{x}_t$;
end for
```

4.4 Cost-Sensitive Algorithm

All aforementioned algorithms are cost-insensitive, which suffer the same *loss* for misclassified positive and negative samples. However, in some real-world data stream classification problem, it is essential to penalize one kind of loss more seriously than others. For example, for online anomaly detection problem, the distribution of class is extremely imbalanced where abnormal events are rare, hence the algorithm should suffer more loss when an abnormal samples are misclassified.

In this section, we propose a cost-sensitive sparse online classification algorithm based on our general framework. Without loss of generality, we assume the positive class is the rare class, which means that there are more negative examples than positive samples. The online algorithm will suffer a higher *loss* when a positive sample is misclassified, comparing with misclassifying a negative sample.

We denote the number of positive samples and negative sample by T_+ and T_-, and denote the number of false negative and false positive by M_+ and M_-. We denote $T = T_+ + T_-$ and $M = M_+ + M_-$. Instead of using the cost-insensitive metric $accuracy = \frac{T-M}{T}$, researchers have proposed a variety of cost-sensitive metrics. In our framework, we adopt a cost-sensitive metric, named as weighted *sum* of *sensitivity* and *specificity*, to measure the classification performance:

$$sum = \mu_+ sensitivity + \mu_- specificity = \mu_+ \frac{T_+ - M_+}{T_+} + \mu_- \frac{T_- - M_-}{T_-},$$

where μ_+ and μ_- ($\mu_+ + \mu_- = 1, 0 \leq \mu_+, \mu_- \leq 1$), are two parameters to balance sensitivity and specificity. The higher the *sum* value is, the better the classification performance is. When $\mu_+ = \mu_- = 0.5$, the corresponding *sum* is known as *balanced accuracy* [4].

[2] In the whole article, we use the diagonal second-order SOL algorithm unless otherwise specified.

To maximize *sum*, we propose a cost-sensitive sparse online classification algorithm following the existing works from Wang et al. [33, 38]. In particular, we use a modified hinge loss function:

$$\ell_t(\mathbf{w}) = (\rho \mathbb{I}_{y_t=1} + \mathbb{I}_{y_t=-1})[1 - y_t \mathbf{w}^\top \mathbf{x}_t]_+$$

where $\rho = \frac{\mu_+ T_-}{\mu_- T_+}$, $\mathbb{I}_v$ is an indicator function. We use *balance accuracy* as the evaluation metric. In general, it is difficult to know the real value of T_+ and T_-, hence, we use two parameters c_+ and c_- to combine the positive and negative losses. The final loss function is reformulated as:

$$\ell_t(\mathbf{w}) = c_t[1 - y_t \mathbf{w}^\top \mathbf{x}_t]_+,$$

where $c_t = c_+ * \mathbb{I}_{y_t=1} + c_- * \mathbb{I}_{y_t=-1}$.

Given the above cost-sensitive loss functions, the first-order cost-sensitive sparse online learning algorithm (CS-FSOL) and second-order cost-sensitive sparse online classification (CS-SSOL) algorithm are derived as in Algorithms 6 and 7, respectively.

ALGORITHM 6: Cost-Sensitive First Order Sparse Online Learning (CS-FSOL)

```
INPUT: $\lambda, \eta, c_+, c_- \geq 0$.
INITIALIZATION: $\theta_1 = 0, A_0^{-1} = I$.
for $t = 1, \ldots, T$ do
    receive $\mathbf{x}_t \in \mathbb{R}^d$;
    $\mathbf{w}_t = \mathrm{sign}(\theta_t) \odot [|\theta_t| - \lambda_t]_+$;
    predict $\hat{y}_t = \mathrm{sign}(\mathbf{w}_t^\top \mathbf{x}_t)$ and receive $y_t \in \{-1, 1\}$;
    suffer $\ell_t(\mathbf{w}_t) = c_t[1 - y_t \mathbf{w}_t^\top \mathbf{x}_t]_+$, where $c_t = c_+ * \mathbb{I}_{y_t=1} + c_- * \mathbb{I}_{y_t=-1}$;
    $\theta_{t+1} = \theta_t + \eta c_t L_t y_t \mathbf{x}_t$;
end for
```

ALGORITHM 7: Cost-Sensitive Second Order Sparse Online Learning (CS-SSOL)

```
INPUT: $\lambda, \eta, c_+, c_- \geq 0$.
INITIALIZATION: $\theta_1 = 0, A_0^{-1} = I$.
for $t = 1, \ldots, T$ do
    receive $\mathbf{x}_t \in \mathbb{R}^d$;
    $A_t^{-1} = A_{t-1}^{-1} - \frac{A_{t-1}^{-1} \mathrm{diag}(\mathbf{x}_t \mathbf{x}_t^\top) A_{t-1}^{-1}}{r + \mathbf{x}_t^\top A_{t-1}^{-1} \mathbf{x}_t}$;
    $\mathbf{u}_t = A_t^{-1} \theta_t$;
    $\mathbf{w}_t = \mathrm{sign}(\mathbf{u}_t) \odot [|\mathbf{u}_t| - \lambda_t]_+$;
    predict $\hat{y}_t = \mathrm{sign}(\mathbf{w}_t^\top \mathbf{x}_t)$ and receive $y_t \in \{-1, 1\}$;
    suffer $\ell_t(\mathbf{w}_t) = c_t[1 - y_t \mathbf{w}_t^\top \mathbf{x}_t]_+$, where $c_t = c_+ * \mathbb{I}_{y_t=1} + c_- * \mathbb{I}_{y_t=-1}$;
    $\theta_{t+1} = \theta_t + \eta c_t L_t y_t \mathbf{x}_t$;
end for
```

5 EXPERIMENTS

5.1 Experimental Setup

In our experiments, we compared the proposed algorithms with the state-of-the-art algorithms. The methodology details of all compared algorithms are listed in Table 1. Three of them (CS-OGD, CPA, and PAUM) are non-sparse cost-sensitive online learning algorithms.

In addition to a synthetic dataset, we evaluate these algorithms with several public benchmark datasets, as shown in Table 2. Using these datasets, we could compare these algorithms in various aspects, where the number of training samples ranges from thousands to millions, the feature

Table 1. Compared Algorithms

Algorithm	First/Second-Order	Sparsity
STG [19]	First Order	Truncate Gradient
FOBOS [12]	First Order	Truncate Gradient
Ada-FOBOS [11]	Second Order	Truncate Gradient
Ada-RDA [11]	Second Order	Dual Averaging
FSOL, Algorithm 3	First Order	Dual Averaging
SSOL, Algorithm 5	Second Order	Dual Averaging
CS-OGD [33]	First Order	Non-Sparse
CPA [7]	First Order	Non-Sparse
PAUM [21]	First Order	Non-Sparse
CS-FSOL, Algorithm 6	First Order	Dual Averaging
CS-SSOL, Algorithm 7	Second Order	Dual Averaging

Table 2. List of Real-world Datasets in Our Experiments

DataSet	#Train	#Test	#Feature	#Nonzero Features	Sparsity(%)	$T_+ \setminus T_-$
AUT	40,000	22,581	20,707	1,969,407	3.07	$1 \setminus 0.33$
PCMAC	1,000	946	7,510	55,470	3.99	$1 \setminus 1.00$
NEWS	10,000	9,996	1,355,191	5,513,533	29.88	$1 \setminus 1.50$
RCV1	781,265	23,149	47,152	59,155,144	8.80	$1 \setminus 1.11$
URL	2,000,000	396,130	3,231,961	231,249,028	7.44	$1 \setminus 2.02$
WEBSPAM	300,000	50,000	16,071,971	1,118,027,721	95.82	$1 \setminus 0.64$
URL2	1,000,000	100,000	3,231,961	114,852,082	44.96	$1 \setminus 99$
WEBSPAM2	100,000	10,000	16,071,971	224,201,808	96.19	$1 \setminus 99$

dimentional ranges from hundreds to over 16-million, and the feature sparsity ranges from 3% to 96%.

We conducted our experiments following the standard online learning setting, where the online learner received one training example per iteration and updated the model sequentially. For fairness, the same experimental settings are adopted for all the compared algorithms. To identify the best parameters, For each algorithm and each dataset, we conducted a five-fold cross validation, with fixing the sparsity regularization parameter λ as 0. The searching range of learning rates was $[2^{-1}, 2^{0}, \ldots, 2^{9}]$ and the range of other parameters was $[2^{-5}, 2^{-4}, \ldots, 2^{5}]$. Using the best-tuned parameters, each algorithm was evaluated for 5 times with a random permutation of the training set. All the experiments were conducted using a Linux server with Intel Xeon CPU E5-2620 @ 2.00 GHz and 8 GB memory.

5.2 Experiment on Synthetic Dataset

First we evaluated the effectiveness of feature usage with a a synthetic dataset. In the synthetic dataset, we controlled the percentage of *effective* features. Following the scheme in [8, 9], we generated a high-dimensional and high-sparse synthetic dataset with a group of *effective* features that were correlated with the class labels and a group of *noisy* features that were uncorrelated with the labels.

We generated the synthetic dataset with 100,000 training examples and 10,000 test examples in 1,000-dimensional feature space. Given an instance, the first 100 features were sampled from a multivariate Gaussian distribution with diagonal covariance. For each feature, the mean value was

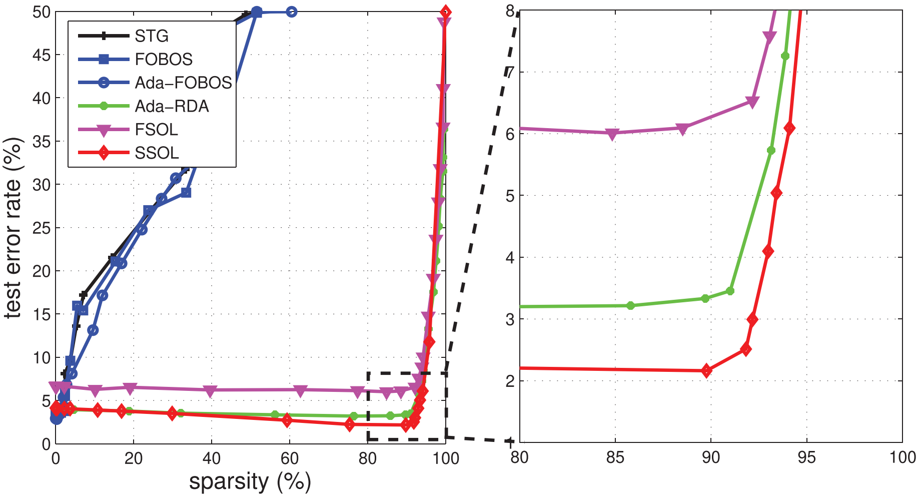


Fig. 1. Test error rate of sparse online classification on synthetic dataset.

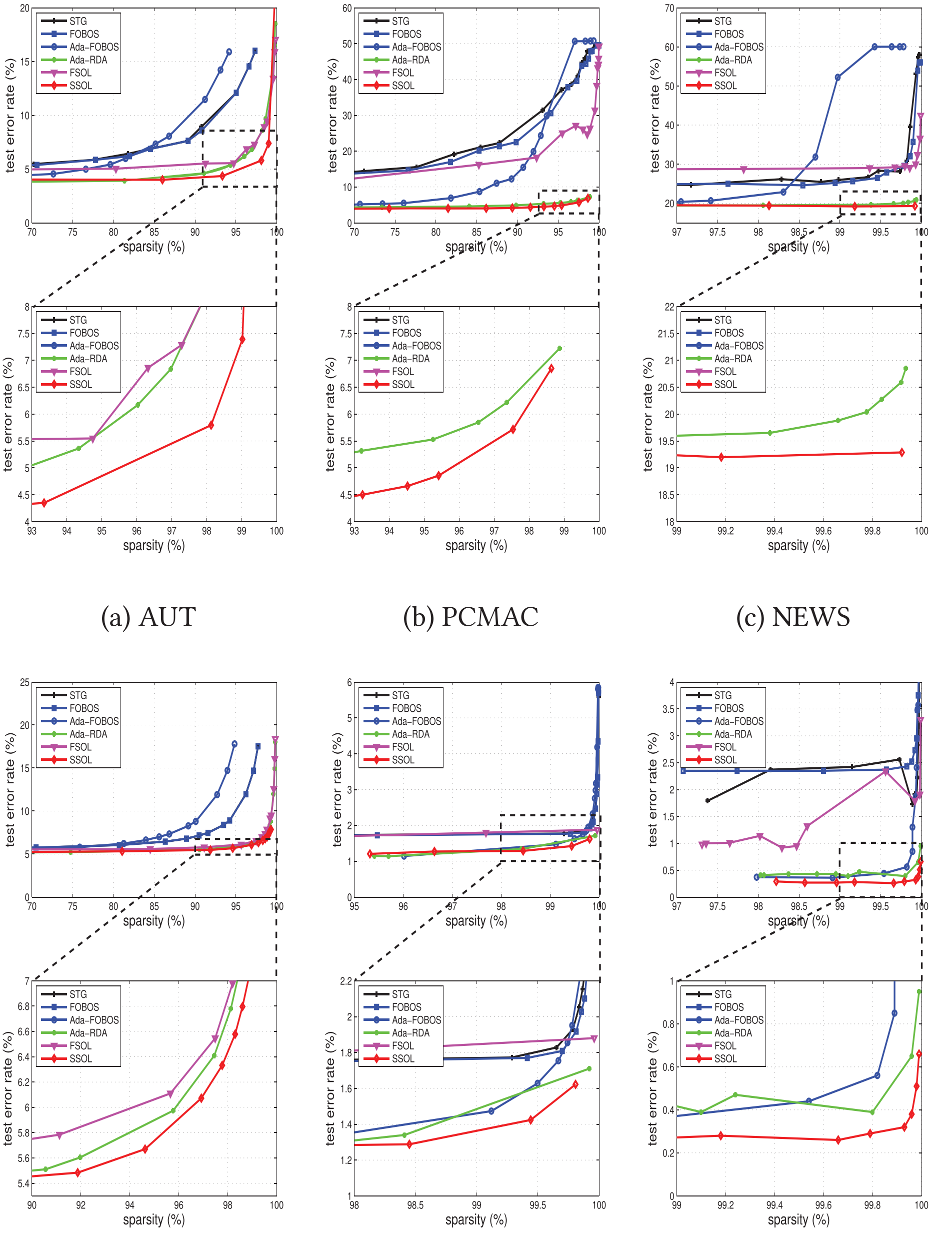
uniformly sampled from −1 to 1 and the corresponding covariance was uniformly sampled from 0.5 to 100. We generated the split plane the same as the mean vector. To introduce noisy feature dimensions, we randomly chose 200 dimensions out from the remaining 900 dimensions for each example, and sampled the noises from a Gaussian distribution of $\mathcal{N}(0, 100)$. We evaluated all the cost-insensitive sparse online classification algorithms using the synthetic dataset. Figure 1 presents the test error rates of all the compared algorithms, and the right figure is a sub-figure of the left one with sparsity from 80% to 100%. Several observations can be drawn from the experimental results.

First, the test error rates of the *truncate gradient* based algorithms (STG, FOBOS, and Ada-FOBOS) increase significantly when the sparsity increases. For the *dual averaging* based algorithms (FSOL, Ada-RDA, and SSOL), the test error rates keep stable or even decrease when the sparsity increases. However, the test error rate of *dual averaging* based algorithms will increase dramatically when the sparsity is larger than 90%, which is the actual sparsity used for constructing the synthetic dataset. The result indicates the dual averaging based algorithms can exploit the sparsity more effectively in the dataset. Similar observations were also reported in [34] who argued that the dual averaging based methods took more aggressive truncations and thus could generate significantly more sparse solutions. Second, the proposed second-order algorithm SSOL achieves the lowest error rate among all the compared algorithms, especially when the sparsity is high. The encouraging experimental results show that the proposed SSOL algorithm can effectively exploit the sparsity.

5.3 Test Error Rate on Large Real Datasets

In this experiment, we compared the proposed algorithms (FSOL and SSOL) with the other cost-insensitive algorithms on several datasets. Table 2 shows the information of six datasets in details. These six datasets can be grouped into the following two categories: the first two datasets (AUT and PCMAC) are general binary small-scale datasets and the corresponding experimental results are shown in Figure 2(a) and (b). The other four datasets (NEWS, RCV1, URL, and WEBSPAM) are large-scale high-dimensional sparse datasets and the corresponding experimental results are presented in Figure 2(c)–(f). We can draw several observations from these results.

First, we observe most of algorithms can learn an effective sparse classification model without suffering too much loss in accuracy. For example, in Figure 2(d), the performances of all the



(a) AUT (b) PCMAC (c) NEWS

(d) RCV1 (e) URL (f) WEBSPAM

Fig. 2. Test error rate on six datasets. (a)–(b) are two general datasets, (c)–(f) are four large-scale high-dimensional sparse datasets. The second and forth rows are the sub-figures of the first and the third rows at high sparsity, respectively.

algorithms are almost stable when sparsity is smaller than 80%. It indicates that all the compared sparse online classification algorithm can effectively explore the low-level sparsity information.

Second, for each algorithm, we observe that there is a threshold t for sparsity. When sparsity is smaller than t, the performance is almost stable, however, when sparsity is larger than t, the test error rate will get worse.

Third, the dual averaging based second order algorithms (Ada-RDA and SSOL) consistently outperform the other algorithms (STG, FOBOS, FSOL, and Ada-FOBOS), especially at high sparsity. It indicates that the dual averaging technique and second order updating rules are more helpful to boost the classification performance.

Finally, when the sparsity is high, the proposed SSOL algorithm consistently outperforms other compared algorithms on all evaluated datasets. For example, as shown in Figure 2(f), when the sparsity is 99.8% for the WEBSPAM dataset (the total feature dimensionality is 16,609,143), the test error rate of SSOL is about 0.3%, which is less than the those of Ada-RDA and Ada-FOBOS, 0.4% and 0.55%.

5.4 Running Time on Large Real Datasets

We evaluated the time costs of various sparse online classification algorithms. Figure 3 presents the experimental results. We can draw several observations from the results. First, we observe that when sparsity is low, the time cost is stable in general. When sparsity is high, the time cost of second other algorithms will slightly increase. One reason may be that when the sparsity is high, the model might not be informative enough for prediction and make more parameter updates. Since the second-order algorithms are more complicated than the first-order algorithms, they are more sensitive to the increasing number of parameter updates.

Second, we can see that the proposed SSOL algorithm runs more efficiently than other second-order based algorithms (Ada-RDA and Ada-FOBOS). It is even sometimes better than the first order based algorithm (e.g., FOBOS and STD). However, the first order FSOL algorithm is consistently faster than the second order SSOL algorithm. In summary, we find that the proposed SSOL algorithm can achieve comparable or better accuracy than the existing second-order algorithms with less time cost.

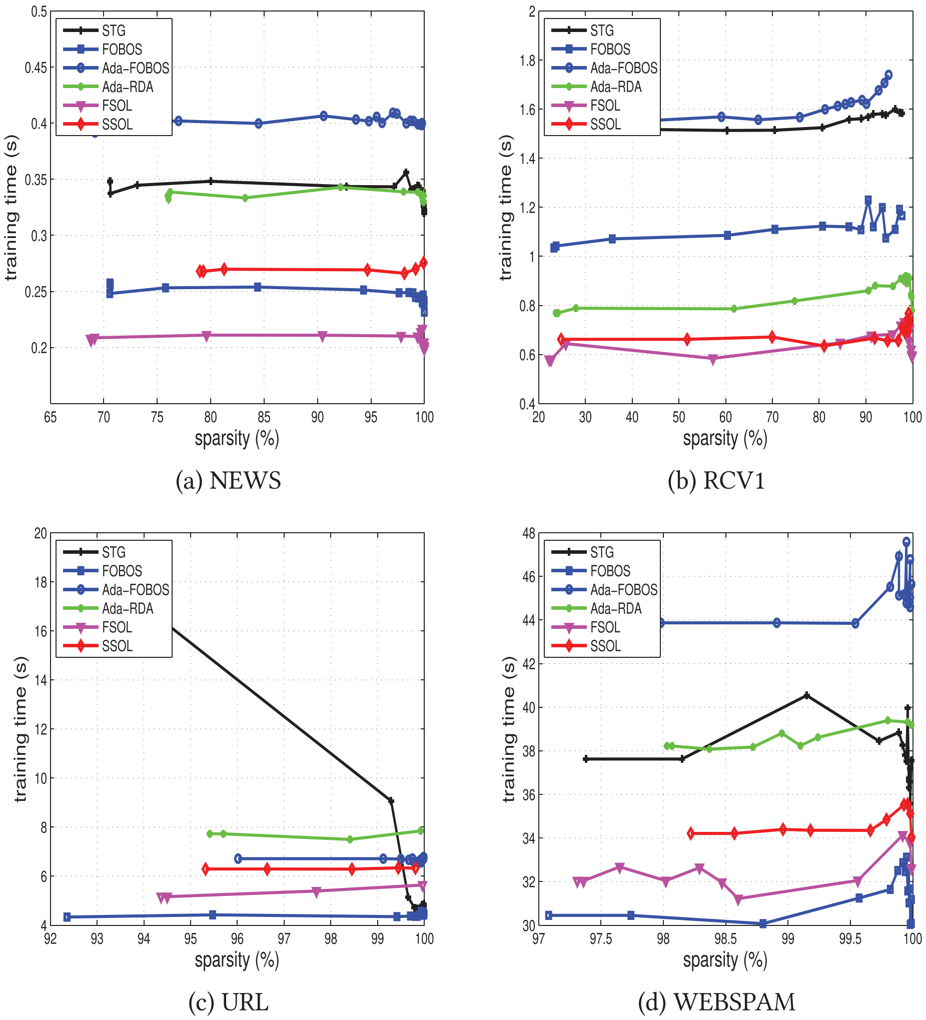
5.5 Applications on Online Anomaly Detection

The following two experiments aim to explore the proposed sparse online classification technique for the online anomaly detection task, i.e., malicious URL detection and web spam detection, where the class distribution is imbalanced.

5.5.1 Malicious URL Detection. We evaluated the cost-sensitive online learning algorithms for malicious URL detection task with the benchmark dataset.[3] The original URL dataset was created in purpose to have balanced classes. In our experiment, we created a subset (denoted as "ULR2") by sampling from the original dataset to make it similar to a realistic distribution scenario where the number of normal URLs was significantly larger than the number of malicious URLs. Following the experiment setting in [38], we chose 10,000 positive (malicious) instance and 990,000 negative (normal) instance. Hence, the ratio $T_+ \setminus T_- = 1 \setminus 99$. For test dataset, we collected 100,000 samples from the original test set with the same ratio. Table 2 shows more details of the unbalanced URL2 dataset.

We compared the proposed CS-FSOL and CS-SSOL with three other cost-sensitive algorithms (CS-OGD, CPA, and PAUM), as shown in Table 1. Besides, we evaluated all aforementioned

[3] http://sysnet.ucsd.edu/projects/url/.



(a) NEWS

(b) RCV1

(c) URL

(d) WEBSPAM

Fig. 3. Time cost on four large-scale datasets: NEWS, RCV1, URL, and WEBSPAM.

cost-insensitive algorithms. The experiment results are shown in Figure 4, where CS-OGD, CPA, and PAUM are non-sparse cost-sensitive online learning algorithms.

Several observations can be drawn from the results. First, all the cost-sensitive algorithms perform consistently better than cost-insensitive ones. Second, among all cost-insensitive algorithms, the second order online learning algorithms are better than the first-order algorithms. Third, the proposed CS-SSOL algorithm achieves the best performance. In this experiment, given the high-dimensional feature representation, in general, the output model of CS-SSOL is very sparse, which demonstrate the efficacy of our framework.

5.5.2 Web Spam Detection. In this experiment, we evaluated the proposed cost-sensitive online learning algorithms with web spam detection task. We constructed an unbalanced subset of the original web spam dataset used in Section 5.3. In particular, for the training dataset, we randomly chose 1,000 positive instances and 99,000 negative instances. Hence, the ratio $T_+ \setminus T_-$ of the training

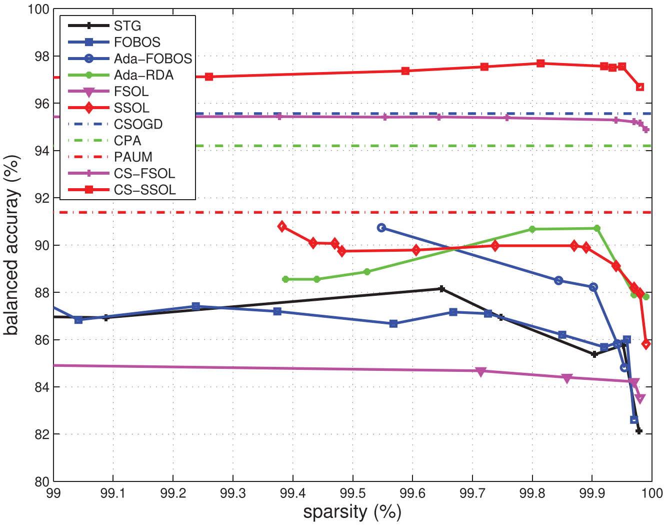


Fig. 4. Balanced accuracy of different algorithms for malicious URL detection.

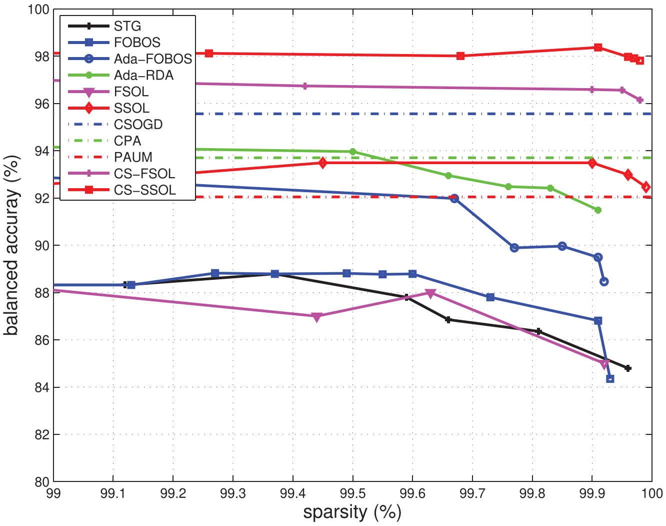


Fig. 5. Balanced accuracy of different algorithms for web spam detection.

set was 1 \ 99. For test dataset, we collected 10,000 samples from the original test set with the same positive-negative ratio.

The imbalance web spam dataset is denoted as "WEBSPAM2," as shown in Table 2. The feature dimension of WEBSPAM2 dataset (16,071,971) is much higher than the one of URL2 (3,231,961). Hence the feature representations of WEBSPAM2 dataset are extremely sparse (96.19% versus 44.96%). The anomaly detection task on WEBSPAM2 dataset is very challenging, given the high-dimensional sparse features and unbalanced data distributions. The experiment settings keep the same with the Section 5.5.1. The experiment results are shown in Figure 5.

First, the performances of non-sparse cost-sensitive algorithms are very poor. Second, similar to the previous experiment, the second order online learning algorithms are better than the first-order ones among all the cost-insensitive/cost-sensitive algorithms. Third, the proposed CS-SSOL algorithm consistently achieves the best performance, which demonstrates the efficacy of the proposed technique for real-world data stream classification problems.

6 CONCLUSIONS AND FUTURE WORK

In this article, we propose a framework of sparse online classification for large-scale high-dimensional data stream classification task. First, we present the proposed framework can easily derive an existing first-order sparse online classification algorithm as its special case. Using the proposed framework, we can further derive a new sparse online classification algorithms by exploiting second-order information. Second, we develop the proposed technique to solve the cost-sensitive data stream classification problem and explore the applications of online anomaly detection, including *malicious URL detection* and *web spam detection*. Finally, we exhaustively evaluate the performance of the proposed algorithms on both theoretical and empirical datasets, where the encouraging experimental results demonstrate that the proposed algorithms can achieve the state-of-the-art performance in comparison to a large family of existing online learning algorithms.

In the future, we would like also explore SOL for distributed settings, including centralized and decentralized distributed settings [22, 36, 37].

REFERENCES

[1] Rehan Akbani, Stephen Kwek, and Nathalie Japkowicz. 2004. Applying support vector machines to imbalanced datasets. In *Proceedings of the European Conference on Machine Learning*. 39–50.

[2] Suhrid Balakrishnan and David Madigan. 2008. Algorithms for sparse linear classifiers in the massive data setting. *Journal of Machine Learning Research* 9 (2008), 313–337.

[3] Antoine Bordes, Léon Bottou, and Patrick Gallinari. 2009. SGD-QN: Careful Quasi-Newton stochastic gradient descent. *Journal of Machine Learning Research* 10 (2009), 1737–1754.

[4] Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M. Buhmann. 2010. The balanced accuracy and its posterior distribution. In *Proceedings of the 20th International Conference on Pattern Recognition*. 3121–3124.

[5] Nicolò Cesa-Bianchi, Alex Conconi, and Claudio Gentile. 2005. A second-order perceptron algorithm. *SIAM Journal on Computing* 34, 3 (2005), 640–668.

[6] Nicolo Cesa-Bianchi and Gabor Lugosi. 2006. *Prediction, Learning, and Games*. Cambridge University Press.

[7] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research* 7 (2006), 551–585.

[8] Koby Crammer, Mark Dredze, and Fernando Pereira. 2008. Exact convex confidence-weighted learning. In *Proceedings of the Conference on Neural Information Processing Systems*. 345–352.

[9] Koby Crammer, Alex Kulesza, and Mark Dredze. 2009. Adaptive regularization of weight vectors. *Machine Learning* 91, 2 (2009), 1–33.

[10] Mark Dredze, Koby Crammer, and Fernando Pereira. 2008. Confidence-weighted linear classification. In *Proceedings of the International Conference on Machine Learning*. 264–271.

[11] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12 (2011), 2121–2159.

[12] John Duchi and Yoram Singer. 2009. Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research* 10 (2009), 2899–2934.

[13] Charles Elkan. 2001. The foundations of cost-sensitive learning. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*. 973–978.

[14] Yoav Freund and Robert E. Schapire. 1999. Large margin classification using the perceptron algorithm. *Machine Learning* 37, 3 (1999), 277–296.

[15] Claudio Gentile. 2001. A new approximate maximal margin classification algorithm. *Journal of Machine Learning Research* 2 (2001), 213–242.

[16] Steven C. H. Hoi, Jialei Wang, and Peilin Zhao. 2014. LIBOL: A library for online learning algorithms. *Journal of Machine Learning Research* 15 (2014), 495–499.

[17] Steven C. H. Hoi, Doyen Sahoo, Jing Lu, and Peilin Zhao. 2018. Online learning: A comprehensive survey. *CoRR* abs/1802.02871 (2018). arxiv:1802.02871 http://arxiv.org/abs/1802.02871

[18] Jyrki Kivinen, Alex J. Smola, and Robert C. Williamson. 2001. Online learning with kernels. In *Proceedings of the Conference on Neural Information Processing Systems*. 785–792.

[19] John Langford, Lihong Li, and Tong Zhang. 2009. Sparse online learning via truncated gradient. *Journal of Machine Learning Research* 10 (2009), 777–801.

[20] Sangkyun Lee and Stephen J. Wright. 2012. Manifold identification in dual averaging for regularized stochastic online learning. *Journal of Machine Learning Research* 13, 55 (2012), 1705–1744.

[21] Yaoyong Li, Hugo Zaragoza, Ralf Herbrich, John Shawe-Taylor, and Jaz Kandola. 2002. The perceptron algorithm with uneven margins. In *Proceedings of the International Conference on Machine Learning*. Vol. 2. 379–386.

[22] Tie-Yan Liu, Wei Chen, and Taifeng Wang. 2017. Distributed machine learning: Foundations, trends, and practices. In *Proceedings of the 26th International Conference on World Wide Web Companion*. Rick Barrett, Rick Cummings, Eugene Agichtein, and Evgeniy Gabrilovich (Eds.). ACM, 913–915. DOI: https://doi.org/10.1145/3041021.3051099

[23] Justin Ma, Alex Kulesza, Mark Dredze, Koby Crammer, Lawrence K. Saul, and Fernando Pereira. 2010. Exploiting feature covariance in high-dimensional online learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*. 493–500.

[24] Yurii Nesterov. 2009. Primal-dual subgradient methods for convex problems. *Mathematical Programming* 120, 1 (2009), 221–259.

[25] Sara Radicati. 2013. *Email Statistics Report, 2013-2017*. Technical Report. The Radicati Group,Inc.

[26] Frank Rosenblatt. 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* 65, 6 (1958), 386.

[27] Nicol N. Schraudolph, Jin Yu, and Simon Günter. 2007. A stochastic Quasi-Newton method for online convex optimization. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*. 436–443.

[28] Shai Shalev-Shwartz. 2012. Online learning and online convex optimization. *Foundations and Trends in Machine Learning* 4, 2 (2012), 107–194.

[29] Shai Shalev-Shwartz and Ambuj Tewari. 2011. Stochastic methods for l1-regularized loss minimization. *Journal of Machine Learning Research* 12 (2011), 1865–1892.

[30] D. Wang, P. Wu, P. Zhao, Y. Wu, C. Miao, and S. C. H. Hoi. 2014. High-dimensional data stream classification via sparse online learning. In *Proceedings of the IEEE International Conference on Data Mining*. 1007–1012.

[31] Jialei Wang, Peilin Zhao, and Steven C. H. Hoi. 2012. Exact soft confidence-weighted learning. In *Proceedings of the International Conference on Machine Learning*.

[32] Jialei Wang, Peilin Zhao, Steven C. H. Hoi, and Rong Jin. 2013. Online feature selection and its applications. *IEEE Transactions on Knowledge and Data Engineering* 26, 3 (2013), 1–14.

[33] Jialei Wang, Peilin Zhao, and Steven C. H. Hoi. 2012. Cost-sensitive online classification. In *Proceedings of the 2012 IEEE 12th International Conference on Data Mining*. 1140–1145.

[34] Lin Xiao. 2010. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research* 11 (2010), 2543–2596.

[35] Seongwook Youn and Dennis McLeod. 2007. Spam email classification using an adaptive ontology. *Journal of Software* 2 (2007), 43–55.

[36] Chi Zhang, Qianxiao Li, and Peilin Zhao. 2019. Decentralized optimization with edge sampling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. Sarit Kraus (Ed.). 658–664. DOI: https://doi.org/10.24963/ijcai.2019/93

[37] Chi Zhang, Peilin Zhao, Shuji Hao, Yeng Chai Soh, Bu-Sung Lee, Chunyan Miao, and Steven C. H. Hoi. 2018. Distributed multi-task classification: A decentralized online learning approach. *Machine Language* 107, 4 (2018), 727–747. DOI: https://doi.org/10.1007/s10994-017-5676-y

[38] Peilin Zhao and Steven C. H. Hoi. 2013. Cost-sensitive online active learning with application to malicious URL detection. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 919–927.

[39] Peilin Zhao, Steven C. H. Hoi, and Rong Jin. 2011. Double updating online learning. *Journal of Machine Learning Research* 12 (2011), 1587–1615.

[40] Peilin Zhao, Yifan Zhang, Min Wu, Steven C. H. Hoi, Mingkui Tan, and Junzhou Huang. 2019. Adaptive cost-sensitive online classification. *IEEE Transactions on Knowledge and Data Engineering* 31, 2 (2019), 214–228. DOI: https://doi.org/10.1109/TKDE.2018.2826011

[41] Peilin Zhao, Furen Zhuang, Min Wu, Xiaoli Li, and Steven C. H. Hoi. 2015. Cost-sensitive online classification with adaptive regularization and its applications. In *Proceedings of the 2015 IEEE International Conference on Data Mining*, Charu C. Aggarwal, Zhi-Hua Zhou, Alexander Tuzhilin, Hui Xiong, and Xindong Wu (Eds.). IEEE Computer Society, 649–658. DOI: https://doi.org/10.1109/ICDM.2015.51