Machine Learning (2024) 113:4407–4444
https://doi.org/10.1007/s10994-023-06403-z

Cost-sensitive sparse group online learning for imbalanced data streams

Zhong Chen[1] · Victor Sheng[2] · Andrea Edwards[3] · Kun Zhang[3]

Received: 6 April 2022 / Revised: 11 July 2023 / Accepted: 16 August 2023 /
Published online: 20 November 2023

Abstract

Effective streaming feature selection in dynamic online environments is essential in numerous applications. However, most existing methods evaluate high-dimensional features individually and ignore the potentially pertainable group structures of features. Moreover, the class imbalance underlying streaming data may further decrease the discriminative efficacy of the selected features, resulting in deteriorated classification performance. Motivated by this observation, we propose a cost-sensitive sparse group online learning (CSGOL) framework and its proximal version (PCSGOL) to handle imbalanced and high-dimensional streaming data. We formulate this issue as a new cost-sensitive online optimization problem by leveraging the ℓ_2-norm, ℓ_1-norm, and groupwise sparsity constraints in the dual averaging regularization. Inspired by the proximal optimization, we further introduce the average weighted distance in CSGOL and develop the PCSGOL method to achieve stable prediction results. We mathematically derive closed-form solutions to the optimization problems with four modified hinge loss functions, leading to four variants of CSGOL and PCSGOL. Extensive empirical studies on real-world streaming datasets and online anomaly detection tasks demonstrate the effectiveness of our proposed methods.

Keywords Imbalanced data streams · Cost-sensitive learning · Online learning · Group sparsity · Online anomaly detection

1 Introduction

With the rapid development of information technologies, a large amount of data are being generated and collected from open and dynamic environments across diverse fields such as financial analysis, surveillance systems, and sensor networks (Ho & Wechsler, 2010). Data generation over time is referred to as streaming data. They contain valuable information that needs to be processed and distilled for real-time data analytics. In traditional data mining tasks, data are presumably stationary. That is, the training and test data come from the same distribution, and their statistical properties do not change over time. However, in dynamic environments, changes in data distribution and changes in feature relevance are

Editors: Nuno Moniz, Nathalie Japkowicz, Michal Wozniak, Shuo Wang.

Extended author information available on the last page of the article common. Moreover, traditional batch offline processing requires the data to be completely available and stored in a database or file before processing. Such learning paradigms suffer from expensive retraining costs and poor scalability when facing new data with unknown patterns. Nevertheless, in streaming data, stationary characteristics no longer exist. The patterns are more likely to evolve unpredictably as time passes. For example, the network traffic flow fluctuates daily due to potential hacking attempts, malicious software outbreaks, unexpected mail server problems, etc. Hence, as a powerful learning paradigm, online learning (Hoi et al., 2021) has emerged to incrementally update the model and make real-time predictions on a stream of examples before obtaining feedback about the true label, thereby making the learning process efficient, scalable, and adaptable and continuously processing incoming instances.

The unbounded sequences of real-time data are often characterized by fast velocity, high-throughput, and large volume attributes, and with new properties such as high dimensionality and skewed distribution. These new properties introduce unprecedented challenges to traditional online learning techniques. To address these challenges, sparse online learning that combines the merits of online learning and sparse learning was proposed. Sparse online learning can incrementally update a trained model and learn a sparse solution with a limited number of active features. There are two major groups of sparse online learning methods. The first group seeks sparse solutions through subgradient descent with truncation (Langford et al., 2009; Duchi & Singer, 2009; Ma & Zheng, 2017). The second group focuses on the dual averaging methods that can explicitly exploit the regularization structure in an online manner. These include regularized dual averaging (RDA) (Xiao, 2010) and RDA+ (Lee and Wright, 2012). Despite their success, sparse online learning models ignore the underlying group structures of features (Hu et al., 2017) when minimizing the empirical loss. For example, most of the user data gathered or collected in network security management (NSM) are in open environments, and the features of the data, such as user profiles, user communities, visits/attacks, and access controls, are collected from different groups (e.g., user profiles and communities). For instance, the attacks may fall into some main categories: DOS (denial-of-service, e.g., syn flood), R2L (unauthorized access from a remote machine, e.g., guessing password), U2R (unauthorized access to local superuser privileges, e.g., various "buffer overflow" attacks), and PROBING (surveillance and other probing, e.g., port scanning) (Wang et al., 2011). These group features are collectively concatenated or merged in the final feature space for further analysis. As an effective method, the original group lasso can yield solutions with sparsity at the group level (Ni et al., 2019). However, this model is built for batch-mode offline learning and usually lacks the ability to further investigate the key factors in an important group. In contrast, in many real-world applications, to select important groups and key features within a group, online models need to promote sparsity at both the group and individual feature levels. Therefore, only seeking sparsity at the individual or group level (Yang et al., 2010; Simon et al., 2013; Wang et al., 2015; Zhou et al., 2021) may lose some useful information that is important to accurately interpret the continuously evolving streaming data.

Moreover, in real-world applications, such as fraud detection, streaming data are both high-dimensional and highly class imbalanced. There have been many attempts such as Multiset feature learning (MFL) (Wu et al., 2017), Confusion Matrix-based Kernel Logistic Regression (CM-KLOGR) (Ohsaki et al., 2017), DDAE (Yin et al., 2020), and Gaussian Distribution based Oversampling (GDO) (Xie et al., 2022) to handle imbalanced data. This occurs because the imbalance classification is crucial in a large number of applications related to the detection of outliers, anomalies, failures, and risks. In such cases, existing sparsity-aware online algorithms usually ignore the minority classes, which can be important in these applications. The class-imbalance issue seriously affects the performance of sparse (group) online learning methods since they treat the misclassification cost of different classes equally and choose the sparse features with the highest classification accuracy, which may deteriorate the performance for imbalanced streaming data. To solve this issue, cost-sensitive online learning methods that use more meaningful cost-sensitive metrics such as the F-measure (Wong, 2020) have been studied extensively. Representative methods include CPA (Crammer et al., 2006), CPA-PB (Crammer et al., 2006), PAUM (Li et al., 2002), CSOGD (Wang et al., 2013), CSTG (Chen et al., 2017), and CSRDA (Chen et al., 2021). For example, Wang et al. (2013) proposed CSOGD to directly optimize the weighted sum of sensitivity and specificity and classification cost of false positives and false negatives by minimizing the weighted indicator function. Chen et al. (2017, 2021) extended the TG and RDA techniques to cost-sensitive online learning scenarios by proposing CSTG and CSRDA, respectively. Although these methods are effective in combating skewed distributions in online settings, they often lack sparse group solutions for online interpretation. Hence, it is challenging to learn from high-dimensional and class imbalanced data streams in an online manner.

Although both learning paradigms have achieved promising performance, most methods are inappropriate to jointly solve the cost-sensitive and group sparsity problems because they often seek cost-insensitive measurements or unexplainable decision models. In light of these observations, we introduce a cost-sensitive sparse group online learning (CSGOL) method to classify imbalanced data streams with high dimensionality. We directly make CSGOL cost-sensitive by integrating the misclassification cost into the formulated objective function. Then, discriminative learning of the sparse group classifier is achieved through the groupwise ℓ_2-norm and ℓ_1-norm combination. To stabilize the prediction model with both imbalanced and high-dimensional issues, the next-round update involves the average of all past subgradients of the loss functions. As a result, the obtained model can automatically seek a favorable trade-off between the low misclassification cost and high group/feature sparsity. We also derive closed-form solutions of CSGOL and its proximal version PCSGOL. Four types of modified hinge loss functions are adopted by CSGOL and PCSGOL, leading to four versions of algorithm implementations. We further analyze the time and space complexity and derive the analytical regret bounds of CSGOL and PCSGOL. Empirical studies demonstrate that the proposed CSGOL and PCSGOL algorithms are more effective than the advanced sparse group and cost-sensitive (sparse) online learning algorithms for high-dimensional streaming data with varied imbalance ratios.

The main contributions of our work are summarized as follows.

(1) We formulate a new cost-sensitive sparse group online learning optimization problem and its proximal version by jointly optimizing the misclassification cost and sparsity of the weight vectors at the group level.
(2) We derive closed-form solutions to effectively solve both CSGOL and PCSGOL optimization problems for imbalanced data streams.
(3) We consider four types of modified hinge loss functions in CSGOL and PCSGOL optimization through a solid theoretical regret bound guarantee.

(4) We verify the effectiveness and interpretation ability of CSGOL and PCSGOL through a series of experiments on high-dimensional streaming datasets with various imbalance ratios.

The rest of the paper is organized as follows. The related work is discussed in Sect. 2. Our proposed methods and algorithms are elaborated in Sect. 3. The theoretical regret bound analysis is presented in Sect. 4. The experimental results are reported in Sect. 5. Our conclusion and future work are summarized in Sect. 6.

2 Related work

We summarize the state-of-the-art methods of sparse online learning, sparse group online learning, and cost-sensitive online learning, which are highly related to our work.

2.1 Sparse online learning

The goal of sparse online learning is to induce sparsity in the weights of online learning algorithms, ensuring that the prediction model only contains a limited size of active features. These algorithms thus have the potential to achieve better performance and interpretability in practice. Existing solutions for sparse online learning can be categorized into two main groups: truncation gradient-based methods and regularized dual averaging-based methods. The former group follows the general idea of subgradient descent with truncation. For example, Langford et al. (2009) proposed a simple yet efficient modification of the standard stochastic gradient via truncated gradient (TG) to achieve sparsity in online learning. Duchi and Singer (2009) further proposed a forward-backward splitting (FOBOS) algorithm to solve the sparse online learning problems by performing an unconstrained subgradient descent step and casting an instantaneous optimization problem with a trade-off between minimizing the regularization term and keeping close to the result obtained in the previous phase. However, with high-dimensional streaming data, the TG and FOBOS methods suffer from slow convergence and high variance due to heterogeneity in feature sparsity. To this end, Ma and Zheng (2017) introduced a stabilized truncated stochastic gradient descent (STSGD) algorithm where a soft-thresholding scheme on the weight vector is employed by imposing an adaptive shrinkage to the amount of information available in each feature. The latter group focuses on the dual averaging methods that can explicitly exploit the regularization structure in an online manner. One representative method is the regularized dual averaging (RDA) proposed in Xiao (2010), which learns the variables by solving a regularized optimization problem that involves the average of all past subgradients of the loss functions instead of the subgradient in the current iteration. Lee and Wright (2012) extended the RDA algorithm to RDA+ by using a more aggressive truncation threshold. Ushio and Yukawa (2019) proposed the projection-based regularized dual averaging (PDA) method to simultaneously exploit a sparsity-promoting metric and a sparsity-promoting regularizer. Zhou et al. (2019) proposed an online algorithm GraphDA for graph-structured sparsity constraint problems using the dual averaging method.

2.2 Sparse group online learning

To efficiently investigate the important explanatory factors in a grouped manner, Yang et al. (2010) developed an online learning algorithm DAGL for group lasso that updates the learning weight vector at each iteration by a closed-form solution based on the average of the previous subgradients. To address the group structures in the feature stream, Wang et al. (2015) developed an online group feature selection method OGFS including an online intragroup selection stage and online intergroup selection stage. A criterion based on spectral analysis is designed to select discriminative features in each group for intragroup selection, and a linear regression model is utilized to select an optimal subset for intergroup selection. However, OGFS needs to choose a small number of positive parameters in advance, which is relatively difficult without prior information in practical applications. To deal with new features that arrive by groups, Group-SAOLA (Yu et al., 2016) extended the SAOLA algorithm, which can online yield a set of feature groups that is sparse between groups and within each group. Ni et al. (2019) proposed a new algorithm called Group Follow The Regularized Leader (GFTRL) for neural feature selection models that directly adds a sparse group lasso regularizer into the FTRL optimizer.

2.3 Cost-sensitive online learning

Cost-sensitive classification has been extensively studied in the area of data mining (Leevy et al., 2018), where the weighted sum of sensitivity and specificity and the weighted misclassification cost of false positives and false negatives (Elkan, 2001) are widely used to quantitatively measure the asymmetric classification outcomes in imbalanced learning. However, there are only a few works specifically for cost-sensitive online learning for imbalanced data streams, including PAUM (Li et al., 2002), CPA (Crammer et al., 2006), CSOGD (Wang et al., 2013), CSDUOL (Zhao and Hoi, 2013), ARCSOGD (Zhao et al., 2015), CSTG (Chen et al., 2017), ACOG (Zhao et al., 2018), and CSRDA (Chen et al., 2021). Specifically, Wang et al. (2013) proposed a cost-sensitive online classification framework that directly optimizes two well-known cost-sensitive measures: weighted cost and weighted sum. Zhao and Hoi (2013) tackled the same problem by adopting the double updating technique and proposed a cost-sensitive double updating online learning (CSDUOL) algorithm. Zhao et al. (2015) proposed an adaptively regularized cost-sensitive online gradient descent (ARCSOGD) method based on the confidence-weighted strategy, which combines the first-order and second-order information for online model updates. Zhao et al. (2018) adopted adaptive regularization for cost-sensitive online classification problems by proposing ACOG, which can significantly reduce the regret bound by incorporating second-order information to enhance the prediction performance. Chen et al. (2017, 2021) proposed a cost-sensitive sparse online learning framework including CSTG and CSRDA by making TG and RDA cost-sensitive, which improves the trade-off between low cost and high sparsity for imbalanced high-dimensional streaming data.

For addressing the imbalanced data streams with concept drifts, Mirza et al. (2015) propose a computationally efficient framework, named ensemble of subset online sequential extreme learning machine (ESOS-ELM), which comprises an ensemble representing short-term memory, an information storage module representing long-term memory and a change detection mechanism to promptly detect concept drifts. Wang et al. (2016) introduce two resampling-based ensemble methods, named MOOB and MUOB, which can process multi-class data directly and strictly online with an adaptive sampling rate for multiclass imbalance and online learning. Cano and Krawczyk (2020) propose a new ensemble method named Kappa Updated Ensemble (KUE), which is a combination of online and block-based ensemble approaches that uses Kappa statistic for dynamic weighting and selection of base classifiers. Bernardo and Della Valle (2021) propose the very fast continuous synthetic minority oversampling technique (VFC-SMOTE). It is a novel meta-strategy to be prepended to any streaming machine learning classification algorithm aiming at oversampling the minority class using a new version of SMOTE and BORDERLINE-SMOTE inspired by Data Sketching. Liu et al. (2021) propose a comprehensive active learning method for multiclass imbalanced streaming data with concept drift (CALMID), which designs a novel sample weight formula that comprehensively considers the class imbalance ratio of the sample's category and the prediction difficulty. Recently, Cano and Krawczyk (2022) introduce a novel online ensemble classifier named Robust Online Self-Adjusting Ensemble (ROSE) for online training of base classifiers and online detection of concept drift and creation of a background ensemble for faster adaptation to changes.

Although the above learning paradigms have achieved promising performance in a large number of applications, they are somewhat inappropriate to jointly solve the cost-sensitive, individual-level sparsity and groupwise sparsity problems in online imbalanced classification, as they often seek cost-insensitive measurements or unexplainable individual- and group-level decision models. These observations motivate us to introduce a new cost-sensitive sparse group online learning method that incorporates the regularized dual averaging technique for stabilizing the prediction performance in handling imbalanced data streams with high dimensionality.

3 Proposed method

3.1 Problem statement

In this study, we concentrate on cost-sensitive sparse group online learning for imbalanced and high-dimensional binary classification, which can be easily extended to multi-class scenarios through one-vs-one or one-vs-all strategies. We make explicit assumptions on the data streams with stationary settings in our study. Let $\mathcal{X} \subseteq \mathbb{R}^d$ be the feature space and $\mathcal{Y} = \{-1, +1\}$ be the label space. We consider a data stream that sequentially comes $\mathcal{D} = \{(\boldsymbol{x}_t, y_t) \mid t \in [T]\}$, where $\boldsymbol{x}_t \in \mathcal{X}$ is the instance received at timestamp t, $y_t \in \mathcal{Y}$ is the true class label of $\boldsymbol{x}_t$, and $[T] = \{1, 2, \dots, T\}$. We assume that (1) d is a large number, (2) The high-dimensional features are sparse and have group effects, and (3) Among the two classes present in $\mathcal{D}$, the size of the positive class is much smaller than that of the negative class, that is, $T_p \ll T_n$. At each timestamp t, a linear model $\boldsymbol{w}_t \in \mathcal{X}$ will be learned and assigns a predicted label $\hat{y}_t$ to $\boldsymbol{x}_t$ by $\hat{y}_t = \text{sign}(\boldsymbol{w}_t^T \boldsymbol{x}_t)$. The model is then updated for the next-round prediction according to the true label y_t and predefined loss functions, i.e., $\ell(\boldsymbol{w}_t;(\boldsymbol{x}_t, y_t))$. To achieve high scalability and interpretability, most of the elements in $\boldsymbol{w}_t$ are required to be zero, making the obtained model have a limited number of active groups and features. Our learning task is thus to seek such a sparse group model that can simultaneously reduce the overall misclassification cost incurred by the skewed class distribution. Table 1 summarizes the major notations used in this paper.

Table 1 Major notations

Notation	Description
$\mathcal{X}$	Domain of an input feature space with d dimensions ($\mathcal{X} \subseteq \mathbf{R}^d$)
$\mathcal{Y}$	Domain of class labels (binary classes, $\mathcal{Y} = \{-1, +1\}$)
$\boldsymbol{x}$	An instance in the feature space $\mathcal{X}$
y	True class label of $\boldsymbol{x}$
$\mathcal{D}$	A data stream comes sequentially
$\hat{y}$	Predicted class label of $\boldsymbol{x}$
$\boldsymbol{w}$	Weight vector of the classifier
$\|\boldsymbol{w}\|_1$	ℓ_1-norm of the weight vector $\boldsymbol{w}$
$\|\boldsymbol{w}\|_2$	ℓ_2-norm of the weight vector $\boldsymbol{w}$, or $\|\boldsymbol{w}\|$ for short
T_p	Number of positive instances
T_n	Number of negative instances
I_p	Number of false negatives
I_n	Number of false positives
μ_+, μ_-	Cost parameters for a false negative and a false positive
κ	Ratio between μ_+ and μ_-
t	Index of the t-th streaming data
T	Number of instances
$[T]$	Index set: $\{1, 2, \cdots, T\}$
$sign(\cdot)$	Sign function
$\ell(\cdot)$	Loss function
$\mathbb{I}_\pi$	Indicator function that outputs 1 if π holds and 0 otherwise
$\Phi(\cdot)$	Smooth regularization term
$\Psi(\cdot)$	Sparsity regularization term
$\nabla\ell^\star(\cdot)$	Subgradient of loss function $\ell^\star(\cdot)$, where $\star \in \{I, II, III, IV\}$
$\mathcal{F}_T(\cdot)$	Objective function over T instances
$\boldsymbol{g}_t$	A single stochastic subgradient of the loss function at the current timestamp t
$\bar{\boldsymbol{g}}_t$	Average subgradients from the start timestamp to the current timestamp t
$\lambda_1, \lambda_2, \lambda_3, \gamma, \sigma > 0$	Regularization parameters
K	The number of groups over all the features
k	The group index of the features ($k = 1, 2, \cdots, K$)
d_k	The feature dimension of the k-th group ($k = 1, 2, \cdots, K$)
L	The length of the sliding window
$R_T^\star(\boldsymbol{w})$	Regret of CSGOL/PCSGOL-I/II/III/IV
$S(\boldsymbol{w})$	The Hoyer sparsity measurement of the weight vector $\boldsymbol{w} \in \mathcal{X}$

3.2 Proposed cost-sensitive online learning framework

Previous studies cast the online learning problem as learning a set of decision models to minimize regret. However, a critique of the standard surrogate loss functions is that they ignore the misclassification cost asymmetry between the majority and minority classes. To resolve this imbalance issue, cost-sensitive objectives such as weighted cost have been proposed for different weight costs of different classes. The accumulated weighted cost is defined by $cost = \mu_+ I_p + \mu_- I_n = \mu_+ \sum_{y_t=+1} \mathbb{I}_{(y_t \hat{y}_t < 0)} + \mu_- \sum_{y_t=-1} \mathbb{I}_{(y_t \hat{y}_t < 0)}$, where

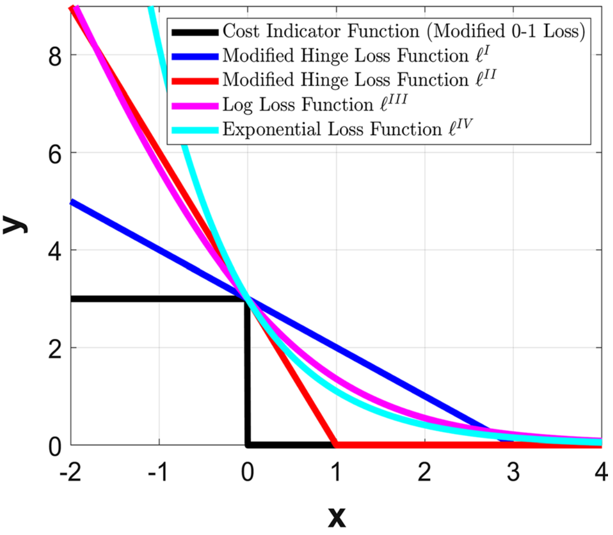


Fig. 1 Four modified hinge loss functions are utilized as tight convex surrogate of cost function in CSGOL and PCSGOL

$I_p = \sum_{y_t=+1} \mathbb{I}_{(y_t\hat{y}_t<0)}$ and $I_n = \sum_{y_t=-1} \mathbb{I}_{(y_t\hat{y}_t<0)}$ are the numbers of false positives and false negatives, respectively; $0 \le \mu_+ \le 1$ and $0 \le \mu_- \le 1$ $(\mu_+ + \mu_- = 1)$ are the misclassification cost parameters for false positives and false negatives, respectively; and $\mathbb{I}_{(\cdot)}$ denotes the indicator function. In the study, since we assume that the positives are the minority class, that is, $T_p \ll T_n$, we take $\mu_+ > \mu_-$ to weight more cost in misclassifying the true positives of the objective function. Note that minimizing the cost is equivalent to minimizing the objective function: $\kappa \sum_{y_t=+1} \mathbb{I}_{(y_t\hat{y}_t<0)} + \sum_{y_t=-1} \mathbb{I}_{(y_t\hat{y}_t<0)}$, where $\kappa = \frac{\mu_+}{\mu_-} > 1$.

However, it is NP-hard to minimize the above cost indicator function (Wang et al., 2013; Chen et al., 2017). Thus, we replace the objective function with its convex surrogates. The tightest surrogate of the indicator function is the hinge loss. Hence, we adopt four types of modified hinge loss as the tight and convex surrogates of the weighted cost, which are summarized as follows:

$$\ell^I(\boldsymbol{w};(\boldsymbol{x}, y)) = \max(0, \bar{\kappa} - y\boldsymbol{w}^T\boldsymbol{x}), \tag{1}$$

$$\ell^{II}(\boldsymbol{w};(\boldsymbol{x}, y)) = \bar{\kappa} * \max(0, 1 - y\boldsymbol{w}^T\boldsymbol{x}), \tag{2}$$

$$\ell^{III}(\boldsymbol{w};(\boldsymbol{x}, y)) = \bar{\kappa} * \log(1 + \exp(-y\boldsymbol{w}^T\boldsymbol{x})), \tag{3}$$

$$\ell^{IV}(\boldsymbol{w};(\boldsymbol{x}, y)) = \bar{\kappa} * \exp(-y\boldsymbol{w}^T\boldsymbol{x}), \tag{4}$$

where $\bar{\kappa} = \kappa\mathbb{I}_{(y=+1)} + \mathbb{I}_{(y=-1)}$. These four loss functions are illustrated in Fig. 1. Compared with the cost indicator function, ℓ^{II} is much tighter than ℓ^I and ℓ^{III} when $x > 0$. However, ℓ^I is much tighter than ℓ^{II} and ℓ^{IV}, and ℓ^{III} is much tighter than ℓ^{II} when $x < 0$. All of the loss functions enjoy the same value when $x = 0$.

Using these loss functions, we have four types of cost-sensitive algorithms to minimize the objective function in Eq. (9). The corresponding update rule in the average gradient step (i.e., $\bar{\boldsymbol{g}}_t = \frac{t-1}{t}\bar{\boldsymbol{g}}_{t-1} + \frac{1}{t}\boldsymbol{g}_t$, where $\boldsymbol{g}_t = \nabla\ell(\boldsymbol{w}_t;(\boldsymbol{x}_t, y_t))$) can be expressed as follows:

$$\bar{\boldsymbol{g}}_t = \frac{t-1}{t}\bar{\boldsymbol{g}}_{t-1} - \frac{y_t}{t}\mathbb{I}_{(\ell^{I}>0)}\boldsymbol{x}_t, \tag{5}$$

$$\bar{\boldsymbol{g}}_t = \frac{t-1}{t}\bar{\boldsymbol{g}}_{t-1} - \frac{\bar{\kappa}_t y_t}{t}\mathbb{I}_{(\ell^{II}>0)}\boldsymbol{x}_t, \tag{6}$$

$$\bar{\boldsymbol{g}}_t = \frac{t-1}{t}\bar{\boldsymbol{g}}_{t-1} - \frac{\bar{\kappa}_t y_t \exp(-y_t \boldsymbol{w}_t^T \boldsymbol{x}_t)}{t(1+\exp(-y_t \boldsymbol{w}_t^T \boldsymbol{x}_t))}\mathbb{I}_{(\ell^{III}>0)}\boldsymbol{x}_t, \tag{7}$$

$$\bar{\boldsymbol{g}}_t = \frac{t-1}{t}\bar{\boldsymbol{g}}_{t-1} - \frac{\bar{\kappa}_t y_t \exp(-y_t \boldsymbol{w}_t^T \boldsymbol{x}_t)}{t}\mathbb{I}_{(\ell^{IV}>0)}\boldsymbol{x}_t, \tag{8}$$

where $\bar{\kappa}_t = \kappa_t \mathbb{I}_{(y_t=+1)} + \mathbb{I}_{(y_t=-1)}$. Note that $\kappa_t = \frac{\mu_+(\sum_{r=max\{1,t-L+1\}}^{t} \mathbb{I}_{(y_r=-1)}+1)}{\mu_-(\sum_{r=max\{1,t-L+1\}}^{t} \mathbb{I}_{(y_r=+1)}+1)} = \kappa \frac{\sum_{r=max\{1,t-L+1\}}^{t} \mathbb{I}_{(y_r=-1)}+1}{\sum_{r=max\{1,t-L+1\}}^{t} \mathbb{I}_{(y_r=+1)}+1}$ is dynamically determined by the most recent L timestamps in the range of $[t-L+1, t]$, where L is the length of the sliding window. Therefore, $\bar{\kappa}_t$ and κ_t are both bounded by $\bar{\kappa}_t \leq \kappa_t \leq max\{1, \kappa\}(L+1)$ for any $t \in [T]$. In this way, we can dynamically update κ_t regardless of the initiation of the imbalance parameter κ. In addition, the term $\frac{\sum_{r=max\{1,t-L+1\}}^{t} \mathbb{I}_{(y_r=-1)}+1}{\sum_{r=max\{1,t-L+1\}}^{t} \mathbb{I}_{(y_r=+1)}+1}$ can still be bounded by $(L+1)$ even though a window includes all majorities only. Note that our proposed methods are single-pass online learning; the dynamical sliding windows are only used to estimate the local imbalance ratio.

To balance the low misclassification cost and high sparsity, we formulate the objective function as follows:

$$\mathcal{F}_T(\boldsymbol{w}; \{(\boldsymbol{x}_t, y_t)\}_{t=1}^{T}) = \sum_{t=1}^{T} \ell(\boldsymbol{w}; (\boldsymbol{x}_t, y_t)) + \Phi(\boldsymbol{w}) + \Psi(\boldsymbol{w}), \tag{9}$$

where $\ell(\boldsymbol{w}; (\boldsymbol{x}_t, y_t))$ is the loss function, which has the above four specific forms in cost-sensitive online learning. The smooth term $\Phi(\boldsymbol{w})$ regularizes the complexity of the classifier to avoid overfitting, and $\Psi(\boldsymbol{w})$ regularizes the (group) sparsity of the weight vector. Our goal is to find an online learning solution to tackle the convex optimization problem, which can be approximately solved by the RDA technique (Xiao, 2010). Instead of utilizing only a single stochastic subgradient $\boldsymbol{g}_t = \nabla \ell(\boldsymbol{w}_t; (\boldsymbol{x}_t, y_t))$ of the loss function at the current timestamp t, RDA updates the next-round weight vector $\boldsymbol{w}_{t+1}$ using the average of all past stochastic subgradients $\{\boldsymbol{g}_s\}_{s=1}^{t}$ (i.e., $\bar{\boldsymbol{g}}_t = \frac{1}{t}\sum_{s=1}^{t} \boldsymbol{g}_s$) and hence leads to improved empirical performance. Next, we will elaborate on the key techniques of CSGOL and PCSGOL and derive the update rule for iteratively minimizing the objective function at the group level.

3.3 Cost-sensitive sparse group online learning

Suppose that the d features are divided into K nonoverlapping groups with size d_k in the k-th group, i.e., $\sum_{k=1}^{K} d_k = d$. Hence, we can rewrite $\boldsymbol{x}_t = [(\boldsymbol{x}_t^{(1)})^T, (\boldsymbol{x}_t^{(2)})^T, \dots, (\boldsymbol{x}_t^{(K)})^T]^T$ with the group of variables $\boldsymbol{x}_t^{(k)} \in \mathbb{R}^{d_k}$, where $k = 1, 2, \dots, K$. The data do not form a group in the feature space when $d_k = 1$ for k. To handle imbalanced and high-dimensional streaming data, CSGOL is introduced to achieve the desired low cost and high sparsity at the group and within-group levels for real-time classification and online interpretation. To address the imbalance issue, we directly minimize the misclassification cost through the modified hinge loss functions. To yield both sparse group selection and sparse solutions in the selected group, we impose the ℓ_1/ℓ_2 mixed regularizations at the group level. The ℓ_1 -norm-based RDA technique is adopted in CSGOL to promote overall sparsity. For timestamp t, based on the current model $\boldsymbol{w}_t \in \mathbb{R}^d$, we use the following optimization solution to update the next-round $\boldsymbol{w}_{t+1} \in \mathbb{R}^d$:

$$\boldsymbol{w}_{t+1} = \underset{\boldsymbol{w}\in\mathbb{R}^d}{\operatorname{argmin}} \{< \bar{\boldsymbol{g}}_t, \boldsymbol{w} > + \frac{\gamma}{2\sqrt{t}} \|\boldsymbol{w}\|_2^2 + \tag{10}$$

$$\sum_{k=1}^{K} (\lambda_1 \left\| \boldsymbol{w}^{(k)} \right\|_2 + \lambda_2 \left\| \boldsymbol{w}^{(k)} \right\|_1) + \lambda_3 \|\boldsymbol{w}\|_1 \}, \tag{11}$$

where $\boldsymbol{w} = [(\boldsymbol{w}^{(1)})^T, (\boldsymbol{w}^{(2)})^T, \dots, (\boldsymbol{w}^{(K)})^T]^T$, $\boldsymbol{w}^{(k)} \in \mathbb{R}^{d_k}$ is the k-th group vector of $\boldsymbol{w}$; $\gamma > 0$, $\lambda_1 > 0$, $\lambda_2 > 0$, and $\lambda_3 > 0$ are the regularization parameters. The ℓ_1/ℓ_2 mixed regularizers at the group level can not only promote sparsity for selecting individual features but also endow the feature space with an additional structure so that features are not penalized individually but collectively, encouraging entire groups of features to be sparse by promoting intergroup and intragroup sparsity. The proposed cost-sensitive sparse group constraints for online learning have a mixing parameter representing the ratio of lasso to group lasso, thus providing a compromise between selecting a subset of sparse feature groups and introducing sparsity within each group.

We can derive the closed-form solution to solve the above optimization problem. That is, given $\bar{\boldsymbol{g}}_t$ in the t-th iteration, for the k-th group, the optimal solution is updated correspondingly as follows:

$$\boldsymbol{w}_{t+1}^{(k)} = -\frac{\sqrt{t}}{\gamma} \max\left(1 - \frac{\lambda_1}{\left\|\boldsymbol{p}_t^{(k)}\right\|_2}, 0\right) \boldsymbol{p}_t^{(k)}, \tag{12}$$

where $\boldsymbol{p}_t^{(k)} = \operatorname{sign}(\bar{\boldsymbol{g}}_t^{(k)}) \bigodot \max(|\ \bar{\boldsymbol{g}}_t^{(k)}\ | - (\lambda_2 + \lambda_3)\mathbf{1}, \mathbf{0})$, $\bar{\boldsymbol{g}}_t^{(k)} \in \mathbb{R}^{d_k}$ is the k-th group vector of $\bar{\boldsymbol{g}}_t$, $\bigodot$ is the elementwise multiplication, $\mathbf{1} = [1, \dots, 1]^T \in \mathbb{R}^{d_k}$, and $\mathbf{0} = [0, \dots, 0]^T \in \mathbb{R}^{d_k}$ $(k = 1, 2, \dots, K)$. Hence, the larger the $(\lambda_2 + \lambda_3)$, the higher the probability of the components of $\boldsymbol{p}_t^{(k)}$ being zero, leading to a higher sparsity of $\boldsymbol{w}_{t+1}^{(k)}$ in the CSGOL model overall. However, a decreased performance may occur at the same time.

3.4 Proximal cost-sensitive sparse group online learning

Inspired by the proximal optimization, we introduce a proximal constraint in CSGOL and develop the PCSGOL method. Specifically, in addition to the above minimization constraints, to achieve stable prediction results, we try to minimize the distance between the next-round weight vector $\boldsymbol{w}_{t+1}$ and all previous weight vectors $\{\boldsymbol{w}_s\}_{s=1}^t$. Hence, we introduce the average weighted distance $\frac{\sigma}{2t} \sum_{s=1}^t \left\|\boldsymbol{w} - \boldsymbol{w}_s\right\|_2^2$ in PCSGOL to replace $\frac{\gamma}{2\sqrt{t}} \|\boldsymbol{w}\|_2^2$ in

CSGOL, which tries to ensure that $\boldsymbol{w}$ is close to all previous weight vectors $\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_t$. Therefore, for timestamp t, based on the current model $\boldsymbol{w}_t \in \mathbb{R}^d$, we use the following optimization solution to update the next-round $\boldsymbol{w}_{t+1} \in \mathbb{R}^d$:

$$\boldsymbol{w}_{t+1} = \underset{\boldsymbol{w} \in \mathbb{R}^d}{\operatorname{argmin}} \left\{ < \bar{\boldsymbol{g}}_t, \boldsymbol{w} > + \frac{\sigma}{2t} \sum_{s=1}^{t} \|\boldsymbol{w} - \boldsymbol{w}_s\|_2^2 + \right. \tag{13}$$

$$\left. \sum_{k=1}^{K} (\lambda_1 \left\|\boldsymbol{w}^{(k)}\right\|_2 + \lambda_2 \left\|\boldsymbol{w}^{(k)}\right\|_1) + \lambda_3 \|\boldsymbol{w}\|_1 \right\}, \tag{14}$$

where $\boldsymbol{w}^{(k)} \in \mathbb{R}^{d_k}$ is the k-th group vector of $\boldsymbol{w}$; $\sigma > 0$, $\lambda_1 > 0$, $\lambda_2 > 0$, and $\lambda_3 > 0$ are the regularization parameters.

We can derive the closed-form solution to solve the above optimization problem. That is, given $\bar{\boldsymbol{g}}_t$ in the t-th iteration, for the k-th group, the optimal solution is updated correspondingly as follows:

$$\boldsymbol{w}_{t+1}^{(k)} = -\frac{1}{\sigma} \max \left(1 - \frac{\lambda_1}{\left\|\boldsymbol{q}_t^{(k)}\right\|_2}, 0 \right) \boldsymbol{q}_t^{(k)}, \tag{15}$$

where $\boldsymbol{q}_t^{(k)} = \operatorname{sign}(\bar{\boldsymbol{g}}_t^{(k)} - \sigma \bar{\boldsymbol{w}}_t^{(k)}) \bigodot \max(|\, \bar{\boldsymbol{g}}_t^{(k)} - \sigma \bar{\boldsymbol{w}}_t^{(k)} \,| - (\lambda_2 + \lambda_3)\mathbf{1}, \mathbf{0})$, $\bar{\boldsymbol{w}}_t^{(k)} \in \mathbb{R}^{d_k}$ is the k-th group vector of $\bar{\boldsymbol{w}}_t$, and $\bar{\boldsymbol{w}}_t = \frac{1}{t} \sum_{s=1}^{t} \boldsymbol{w}_s$ is the average weight. Similarly, the larger the $(\lambda_2 + \lambda_3)$, the higher the probability of the components of $\boldsymbol{q}_t^{(k)}$ being zero, leading to a higher sparsity of $\boldsymbol{w}_{t+1}^{(k)}$ in the PCSGOL model overall. However, a decreased performance may occur at the same time.

All proofs on the closed-form solutions of CSGOL and PCSGOL are provided in the Appendices. Note that these closed-form results of CSGOL and PCSGOL are based on the assumption that the groups are divided in nonoverlapping ways. However, if the data contain overlapped groups, then we can simply replicate the overlapped features as in Yang et al. (2010) to obtain similar solutions of CSGOL and PCSGOL.

3.5 Algorithms

We summarize the key steps of the CSGOL and PCSGOL algorithms in Algorithm 1. It is obvious that the overall time complexity of the algorithm is $\mathcal{O}(Td)$, which is linear with respect to the total number of instances T when the dimensionality d is not too high and can be treated as a constant. The space complexity of each learning step is $\mathcal{O}(d)$, linear with respect to d. In practice, when the dataset is high-dimensional, the sparse group mechanism introduced in CSGOL and PCSGOL can further reduce the computational cost.

Algorithm 1 Cost-sensitive sparse group online learning (CSGOL) and proximal CSGOL (PCSGOL) algorithms

```
Online input: $\{(\boldsymbol{x}_t, y_t)\}_{t=1}^T$, $\gamma$, $\sigma$, $\lambda_1$, $\lambda_2$, $\lambda_3$, $\ell^\star(\cdot;\ (\cdot,\cdot))$ (where $\star \in \{I, II, III, IV\}$), $\kappa = \frac{\mu_+}{\mu_-}$, and $L$.
Online output: $\boldsymbol{w}_{T+1}$.
Initialization: $\boldsymbol{w}_1 = \mathbf{0} \in \mathbb{R}^d$, $\boldsymbol{g}_0 = \mathbf{0} \in \mathbb{R}^d$;
for $t = 1, 2, \ldots, T$ do
    Receive $\boldsymbol{x}_t \in \mathbb{R}^d$;
    Predict $\hat{y}_t = \mathrm{sign}(\boldsymbol{w}_t^T \boldsymbol{x}_t)$ and receive $y_t \in \{-1, +1\}$;
    Compute the loss function $\ell^\star(\boldsymbol{w}_t;\ (\boldsymbol{x}_t, y_t))$;
    Update $\bar{\kappa}_t = \kappa_t \mathbb{I}_{(y_t=+1)} + \mathbb{I}_{(y_t=-1)}$;
    Compute the subgradient $\boldsymbol{g}_t = \nabla \ell^\star(\boldsymbol{w}_t;\ (\boldsymbol{x}_t, y_t))$;
    Update average subgradient $\bar{\boldsymbol{g}}_t = \frac{t-1}{t} \bar{\boldsymbol{g}}_{t-1} + \frac{1}{t} \boldsymbol{g}_t$;
    if choose the CSGOL method then
        for $k = 1, 2, \ldots, K$ do
            Compute $\boldsymbol{p}_t^{(k)} = \mathrm{sign}(\bar{\boldsymbol{g}}_t^{(k)}) \bigodot \max(|\ \bar{\boldsymbol{g}}_t^{(k)}\ | - (\lambda_2 + \lambda_3)\mathbf{1}, \mathbf{0})$;
            Update $\boldsymbol{w}_{t+1}^{(k)}$ through Eq. (12);
        end for
        $\boldsymbol{w}_{t+1} = [(\boldsymbol{w}_{t+1}^{(1)})^T, (\boldsymbol{w}_{t+1}^{(2)})^T, \ldots, (\boldsymbol{w}_{t+1}^{(K)})^T]^T$;
    elsechoose the PCSGOL method
        Update average weight $\bar{\boldsymbol{w}}_t = \frac{t-1}{t} \bar{\boldsymbol{w}}_{t-1} + \frac{1}{t} \boldsymbol{w}_t$;
        for $k = 1, 2, \ldots, K$ do
            Compute $\boldsymbol{q}_t^{(k)} = \mathrm{sign}(\bar{\boldsymbol{g}}_t^{(k)} - \sigma \bar{\boldsymbol{w}}_t^{(k)}) \bigodot \max(|\ \bar{\boldsymbol{g}}_t^{(k)} - \sigma \bar{\boldsymbol{w}}_t^{(k)}\ | - (\lambda_2 + \lambda_3)\mathbf{1}, \mathbf{0})$;
            Update $\boldsymbol{w}_{t+1}^{(k)}$ through Eq. (15);
        end for
        $\boldsymbol{w}_{t+1} = [(\boldsymbol{w}_{t+1}^{(1)})^T, (\boldsymbol{w}_{t+1}^{(2)})^T, \ldots, (\boldsymbol{w}_{t+1}^{(K)})^T]^T$;
    end if
end for
```

4 Theoretical analysis

Based on the regret bound of RDA (Xiao, 2010), we can derive the regret bounds of CSGOL and PCSGOL in Theorems 1 and 2, respectively. We also compare the regret bounds of some existing cost-sensitive and sparse group online learning methods.

4.1 Regret bound of CSGOL

Theorem 1 (Regret Bound of CSGOL) *Let the sequences of* $\{\mathbf{w}_t\}_{t=1}^T$ *and* $\{\mathbf{g}_t = \nabla \ell(\mathbf{w}_t;(\mathbf{x}_t, y_t))\}_{t=1}^T$ *be generated by the CSGOL algorithm, where* $\mathbf{w}_t, \mathbf{x}_t \in \mathbb{R}^d$, $y_t \in \{-1, +1\}$, *and* $\ell(\mathbf{w}_t;(\mathbf{x}_t, y_t))$ *is a convex loss function. We assume that there is a constant* $A > 0$ *such that* $\|\mathbf{g}_t\|_2^2 \leq A^2$ *and* $\|\mathbf{x}_t\|_2^2 \leq 1$ *for all* $t \geq 1$. *Then, for any* $\mathbf{w} \in \mathbb{R}^d$ *with* $\frac{1}{2}\|\mathbf{w}\|_2^2 \leq B^2$ $(B > 0)$, *we have the regret* $\mathcal{R}_T^*(\mathbf{w}) = \sum_{t=1}^T [\ell^*(\mathbf{w}_t;(\mathbf{x}_t, y_t)) + \Phi(\mathbf{w}_t) + \Psi(\mathbf{w}_t)] - \sum_{t=1}^T [\ell^*(\mathbf{w};(\mathbf{x}_t, y_t)) + \Phi(\mathbf{w}) + \Psi(\mathbf{w})]$ $(* \in \{I, II, III, IV\})$ *of CSGOL is bounded by* $\mathcal{R}_T^*(\mathbf{w}) \leq \gamma B^2 \sqrt{T} + \frac{A^2}{2\gamma} \sum_{t=1}^T \frac{1}{\sqrt{t}} \leq \gamma B^2 \sqrt{T} + \frac{A^2}{2\gamma} 2\sqrt{T} = (\gamma B^2 + \frac{A^2}{\gamma})\sqrt{T}$.

Remark 1 (I) If $\ell^*(\cdot) = \ell^I(\cdot)$, then we can derive the following bound of $\|\boldsymbol{g}_t\|_2^2$ for all $t \geq 1$: $\|\boldsymbol{g}_t\|_2^2 = \|\nabla \ell^I(\boldsymbol{w}_t;(\boldsymbol{x}_t, y_t))\|_2^2 = \| - y_t\boldsymbol{x}_t\|_2^2 \leq 1$. Thus, replacing $A^2 = 1$ in Theorem 1 will obtain the regret bound of CSGOL-I: $\mathcal{R}_T^I(\boldsymbol{w}) \leq (\gamma B^2 + \frac{1}{\gamma})\sqrt{T}$; (II) If $\ell^*(\cdot) = \ell^{II}(\cdot)$, then we can derive the following bound of $\|\boldsymbol{g}_t\|_2^2$ for all $t \geq 1$: $\|\boldsymbol{g}_t\|_2^2 = \|\nabla \ell^{II}(\boldsymbol{w}_t;(\boldsymbol{x}_t, y_t))\|_2^2 = \| - \bar{\kappa}_t y_t \boldsymbol{x}_t\|_2^2 \leq \bar{\kappa}_t^2 \leq \kappa_t^2 \leq \max^2(1,\kappa)(L+1)^2$. Thus, replacing $A^2 = \max^2(1,\kappa)(L+1)^2$ in Theorem 1 will obtain the regret bound of CSGOL-II: $\mathcal{R}_T^{II}(\boldsymbol{w}) \leq (\gamma B^2 + \frac{\max^2(1,\kappa)(L+1)^2}{\gamma})\sqrt{T}$; (III) If $\ell^*(\cdot) = \ell^{III}(\cdot)$, then we can derive the following bound of $\|\boldsymbol{g}_t\|_2^2$ for all $t \geq 1$: $\|\boldsymbol{g}_t\|_2^2 = \|\nabla \ell^{III}(\boldsymbol{w}_t;(\boldsymbol{x}_t, y_t))\|_2^2 = \| - \bar{\kappa}_t y_t \boldsymbol{x}_t \frac{\exp(-y_t \boldsymbol{w}_t^T \boldsymbol{x}_t)}{1+\exp(-y_t \boldsymbol{w}_t^T \boldsymbol{x}_t)}\|_2^2 \leq \bar{\kappa}_t^2$ $\|\boldsymbol{g}_t\|_2^2 = \|\nabla \ell^{III}(\boldsymbol{w}_t;(\boldsymbol{x}_t, y_t))\|_2^2 = \| - \bar{\kappa}_t y_t \boldsymbol{x}_t \frac{\exp(-y_t \boldsymbol{w}_t^T \boldsymbol{x}_t)}{1+\exp(-y_t \boldsymbol{w}_t^T \boldsymbol{x}_t)}\|_2^2 \leq \bar{\kappa}_t^2 \leq \kappa_t^2 \leq \max^2(1,\kappa)(L+1)^2$, where $\frac{\exp(-y_t \boldsymbol{w}_t^T \boldsymbol{x}_t)}{1+\exp(-y_t \boldsymbol{w}_t^T \boldsymbol{x}_t)} \leq 1$. Thus, replacing $A^2 = \max^2(1,\kappa)(L+1)^2$ in Theorem 1 will obtain the regret bound of CSGOL-III: $\mathcal{R}_T^{III}(\boldsymbol{w}) \leq (\gamma B^2 + \frac{\max^2(1,\kappa)(L+1)^2}{\gamma})\sqrt{T}$; and (IV) If $\ell^*(\cdot) = \ell^{IV}(\cdot)$, then we can derive the following bound of $\|\boldsymbol{g}_t\|_2^2$ for all $t \geq 1$: $\|\boldsymbol{g}_t\|_2^2 = \|\nabla \ell^{IV}(\boldsymbol{w}_t;(\boldsymbol{x}_t, y_t))\|_2^2 = \| - \bar{\kappa}_t y_t \boldsymbol{x}_t \exp(-y_t \boldsymbol{w}_t^T \boldsymbol{x}_t)\|_2^2 \leq \exp(-\sqrt{2}B)\bar{\kappa}_t^2 \leq \exp(-\sqrt{2}B)$ $\kappa_t^2 \leq \exp(-\sqrt{2}B)\max^2(1,\kappa)(L+1)^2$, where $\exp(-y_t \boldsymbol{w}_t^T \boldsymbol{x}_t) \leq \exp(-\sqrt{2}B)$. Thus, replacing $A^2 = \exp(-\sqrt{2}B)\max^2(1,\kappa)(L+1)^2$ in Theorem 1 will obtain the regret bound of CSGOL-IV: $\mathcal{R}_T^{IV}(\boldsymbol{w}) \leq (\gamma B^2 + \frac{\exp(-\sqrt{2}B)\max^2(1,\kappa)(L+1)^2}{\gamma})\sqrt{T}$.

4.2 Regret bound of PCSGOL

Theorem 2 (Regret Bound of PCSGOL) *Let the sequences of* $\{\mathbf{w}_t\}_{t=1}^T$ *and* $\{\mathbf{g}_t = \nabla \ell(\mathbf{w}_t;(\mathbf{x}_t, y_t))\}_{t=1}^T$ *be generated by the PCSGOL algorithm, where* $\mathbf{w}_t, \mathbf{x}_t \in \mathbb{R}^d$, $y_t \in \{-1, +1\}$, *and* $\ell(\mathbf{w}_t;(\mathbf{x}_t, y_t))$ *is a convex loss function. We assume that there is a constant* $A > 0$ *such that* $\|\mathbf{g}_t\|_2^2 \leq A^2$ *and* $\|\mathbf{x}_t\|_2^2 \leq 1$ *for all* $t \geq 1$. *Then, for any* $\mathbf{w} \in \mathbb{R}^d$ *with* $\frac{1}{2}\|\mathbf{w}\|_2^2 \leq B^2$ $(B > 0)$, *we have the regret* $\mathcal{R}_T^*(\mathbf{w}) = \sum_{t=1}^T [\ell^*(\mathbf{w}_t;(\mathbf{x}_t, y_t)) + \Phi(\mathbf{w}_t) + \Psi(\mathbf{w}_t)] - \sum_{t=1}^T [\ell^*(\mathbf{w};(\mathbf{x}_t, y_t)) + \Phi(\mathbf{w}) + \Psi(\mathbf{w})]$ $(* \in \{I, II, III, IV\})$ *of PCSGOL is bounded by* $\mathcal{R}_T^*(\mathbf{w}) \leq \sigma B^2 \sqrt{T} + \frac{A^2}{2\sigma}\sum_{t=1}^T \frac{1}{t} \leq \sigma B^2\sqrt{T} + \frac{A^2}{2\sigma}\log T$.

Remark 2 (I) If $\ell^*(\cdot) = \ell^I(\cdot)$, then we can derive the following bound of $\|\boldsymbol{g}_t\|_2^2$ for all $t \geq 1$: $\|\boldsymbol{g}_t\|_2^2 = \|\nabla \ell^I(\boldsymbol{w}_t;(\boldsymbol{x}_t, y_t))\|_2^2 = \| - y_t\boldsymbol{x}_t\|_2^2 \leq 1$. Thus, replacing $A^2 = 1$ in Theorem 2 will obtain the regret bound of PCSGOL-I: $\mathcal{R}_T^I(\boldsymbol{w}) \leq \sigma B^2\sqrt{T} + \frac{1}{2\sigma}\log T$; (II) If $\ell^*(\cdot) = \ell^{II}(\cdot)$, then we can derive the following bound of $\|\boldsymbol{g}_t\|_2^2$ for all $t \geq 1$: $\|\boldsymbol{g}_t\|_2^2 = \|\nabla \ell^{II}(\boldsymbol{w}_t;(\boldsymbol{x}_t, y_t))\|_2^2 = \| - \bar{\kappa}_t y_t \boldsymbol{x}_t\|_2^2 \leq \bar{\kappa}_t^2 \leq \kappa_t^2 \leq \max^2(1,\kappa)(L+1)^2$. Thus, replacing $A^2 = \max^2(1,\kappa)(L+1)^2$ in Theorem 2 will obtain the regret bound of PCSGOL-II: $\mathcal{R}_T^{II}(\boldsymbol{w}) \leq \sigma B^2\sqrt{T} + \frac{\max^2(1,\kappa)(L+1)^2}{2\sigma}\log T$; (III) If $\ell^*(\cdot) = \ell^{III}(\cdot)$, then we can derive the following bound of $\|\boldsymbol{g}_t\|_2^2$ for all $t \geq 1$: $\|\boldsymbol{g}_t\|_2^2 = \|\nabla \ell^{III}(\boldsymbol{w}_t;(\boldsymbol{x}_t, y_t))\|_2^2 = \| - \bar{\kappa}_t y_t \boldsymbol{x}_t \frac{\exp(-y_t \boldsymbol{w}_t^T \boldsymbol{x}_t)}{1+\exp(-y_t \boldsymbol{w}_t^T \boldsymbol{x}_t)}\|_2^2 \leq \bar{\kappa}_t^2 \leq \kappa_t^2 \leq \max^2(1,\kappa)(L+1)^2$, where $\frac{\exp(-y_t \boldsymbol{w}_t^T \boldsymbol{x}_t)}{1+\exp(-y_t \boldsymbol{w}_t^T \boldsymbol{x}_t)} \leq 1$. Thus, replacing $A^2 = \max^2(1,\kappa)(L+1)^2$ in Theorem 2

Table 2 Regret bound comparison: TG, RDA, CSOGD, CSTG, CSRDA, DAGL/DASGL/DAESGL, CSGOL, and PCSGOL

Method	Regret bound	Complexity
TG (Langford et al., 2009)	$\frac{1+2B^2}{2}\sqrt{T}$	$\mathcal{O}(\sqrt{T})$
RDA (Xiao, 2010)	$(\gamma B^2 + \frac{A^2}{\gamma})\sqrt{T}$	$\mathcal{O}(\sqrt{T})$
CSOGD-I (Wang et al., 2013)	$\sqrt{2}B^2\sqrt{T}$	$\mathcal{O}(\sqrt{T})$
CSOGD-II (Wang et al., 2013)	$\max(1,\kappa)\sqrt{2}B^2\sqrt{T}$	$\mathcal{O}(\sqrt{T})$
CSTG-I (Chen et al., 2017)	$\frac{C(1+2B^2)}{2}\sqrt{T}$	$\mathcal{O}(\sqrt{T})$
CSTG-II (Chen et al., 2017)	$\frac{C(\max^2(1,\kappa)+2B^2)}{2}\sqrt{T}$	$\mathcal{O}(\sqrt{T})$
CSRDA-I (Chen et al., 2021)	$(\gamma B^2 + \frac{1}{\gamma})\sqrt{T}$	$\mathcal{O}(\sqrt{T})$
CSRDA-II (Chen et al., 2021)	$(\gamma B^2 + \frac{\max^2(1,\kappa)(L+1)^2}{\gamma})\sqrt{T}$	$\mathcal{O}(\sqrt{T})$
DAGL/DASGL/DAESGL (Yang et al., 2010)	$(\gamma B^2 + \frac{A^2}{\gamma})\sqrt{T}$	$\mathcal{O}(\sqrt{T})$
CSGOL-I	$(\gamma B^2 + \frac{1}{\gamma})\sqrt{T}$	$\mathcal{O}(\sqrt{T})$
CSGOL-II	$(\gamma B^2 + \frac{\max^2(1,\kappa)(L+1)^2}{\gamma})\sqrt{T}$	$\mathcal{O}(\sqrt{T})$
CSGOL-III	$(\gamma B^2 + \frac{\max^2(1,\kappa)(L+1)^2}{\gamma})\sqrt{T}$	$\mathcal{O}(\sqrt{T})$
CSGOL-IV	$(\gamma B^2 + \frac{\exp(-\sqrt{2}B)\max^2(1,\kappa)(L+1)^2}{\gamma})\sqrt{T}$	$\mathcal{O}(\sqrt{T})$
PCSGOL-I	$\sigma B^2\sqrt{T} + \frac{1}{2\sigma}\log T$	$\mathcal{O}(\sqrt{T} + \log T)$
PCSGOL-II	$\sigma B^2\sqrt{T} + \frac{\max^2(1,\kappa)(L+1)^2}{2\sigma}\log T$	$\mathcal{O}(\sqrt{T} + \log T)$
PCSGOL-III	$\sigma B^2\sqrt{T} + \frac{\max^2(1,\kappa)(L+1)^2}{2\sigma}\log T$	$\mathcal{O}(\sqrt{T} + \log T)$
PCSGOL-IV	$\sigma B^2\sqrt{T} + \frac{\exp(-\sqrt{2}B)\max^2(1,\kappa)(L+1)^2}{2\sigma}\log T$	$\mathcal{O}(\sqrt{T} + \log T)$

will obtain the regret bound of PCSGOL-III: $\mathcal{R}_T^{III}(\boldsymbol{w}) \leq \sigma B^2\sqrt{T} + \frac{\max^2(1,\kappa)(L+1)^2}{2\sigma}\log T$; and (IV) If $\ell^*(\cdot) = \ell^{IV}(\cdot)$, then we can derive the following bound of $\|\boldsymbol{g}_t\|_2^2$ for all $t \geq 1$: $\|\boldsymbol{g}_t\|_2^2 = \|\nabla\ell^{IV}(\boldsymbol{w}_t;(\boldsymbol{x}_t, y_t))\|_2^2 = \| - \bar{\kappa}_t y_t \boldsymbol{x}_t \exp(-y_t \boldsymbol{w}_t^T \boldsymbol{x}_t)\|_2^2 \leq \exp(-\sqrt{2}B)\bar{\kappa}_t^2 \leq \exp(-\sqrt{2}B)\kappa_t^2 \leq \exp(-\sqrt{2}B)\max^2(1,\kappa)(L+1)^2$, where $\exp(-y_t \boldsymbol{w}_t^T \boldsymbol{x}_t) \leq \exp(-\sqrt{2}B)$. Thus, replacing $A^2 = \exp(-\sqrt{2}B)\max^2(1,\kappa)(L+1)^2$ in Theorem 2 will obtain the regret bound of PCSGOL-IV: $\mathcal{R}_T^{IV}(\boldsymbol{w}) \leq \sigma B^2\sqrt{T} + \frac{\exp(-\sqrt{2}B)\max^2(1,\kappa)(L+1)^2}{2\sigma}\log T$.

Remark 3 Theorems 1 and 2 reveal the mathematical relationships of regret bounds of CSGOL and PCSGOL with the imbalanced parameter κ, size of sliding windows L, and regularization parameters γ and σ, respectively. Since the gradient of ℓ^I is independent of κ_t, $\mathcal{R}_T^I$ is also irrelevant to κ_t. Furthermore, it is easy to derive the cost bounds of CSGOL and PCSGOL by $cost^* \leq \mu_- \sum_{t=1}^T \ell^*(\boldsymbol{w}_t;(\boldsymbol{x}_t, y_t))$, where $* \in \{I, II, III, IV\}$. Hence, using Theorems 1 and 2, we can determine that the cost bounds of CSGOL and PCSGOL depend on the corresponding regret bounds and the trade-off between the misclassification cost, individual feature sparsity, and groupwise sparsity. The lower the regret bound and/or higher feature or groupwise sparsity of CSGOL (or PCSGOL), the lower the cost bound of CSGOL (or PCSGOL).

Table 3 Summarization of real-world binary datasets

Dataset	#Sample	#Feature	#Positive:#Negative	#Positive(Sub):#Negative
PCMAC[a]	1943	7, 510	$961 : 982 \approx 1 : 1.0$	$96 : 982 \approx 1 : 10.2$
spambase[b]	4, 601	56	$1813 : 2788 \approx 1 : 1.5$	$181 : 2788 \approx 1 : 15.4$
MITFace[b]	6, 977	361	$2429 : 4548 \approx 1 : 1.9$	$242 : 4548 \approx 1 : 18.8$
a9a[a]	48, 842	123	$11687 : 37155 \approx 1 : 3.2$	$1168 : 37155 \approx 1 : 31.8$
usps1all[b]	7291	256	$1194 : 6097 \approx 1 : 5.1$	$119 : 6097 \approx 1 : 51.2$
w8a[b]	64, 700	300	$1933 : 62767 \approx 1 : 32.5$	$1933 : 62767 \approx 1 : 32.5$

[a] http://archive.ics.uci.edu/ml/datasets.php

[b] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

4.3 Regret bound comparisons

Table 2 compares the regret bounds of CSGOL-I/II/III/IV and PCSGOL-I/II/III/IV with those of TG, RDA, CSOGD-I/II, CSTG-I/II, and CSRDA-I/II. The complexity of the regret bounds of CSGOL-I/II/III/IV is $\mathcal{O}(\sqrt{T})$, the same as those of TG, RDA, CSOGD-I/II, CSTG-I/II, and CSRDA-I/II. It is clear that the algorithms with the first type of loss have tighter regret bounds than their counterparts with the other three types of loss. The difference between these two groups can be large when the data are highly skewed. Among these methods, CSGOL, PCSGOL, CSRDA, and CSTG have the loosest regret bounds, as both sparsity and loss are considered in the bounds. TG, RDA, and CSOGD, on the other hand, only consider sparsity or loss in their bounds. Compared to CSOGD, TG and RDA have a looser regret bound. This is because TG and RDA may have higher misclassification costs when sparsity is introduced in the high-dimensional feature space. Note that since the complexity of the regret bounds of all of the methods is $\mathcal{O}(\sqrt{T})$ or $\mathcal{O}(\sqrt{T} + \log T)$, the average regret bounds $\bar{\mathcal{R}}_T^*(\boldsymbol{w}) = \frac{1}{T}\mathcal{R}_T^*(\boldsymbol{w})$ of the methods and their variants converges to $\mathcal{O}(\frac{1}{\sqrt{T}}) \to 0$ or $\mathcal{O}(\frac{1}{\sqrt{T}} + \frac{\log T}{T}) \to 0$ as the number of streaming samples $T \to \infty$.

5 Experiments

5.1 Datasets and evaluation metrics

Six real-world streaming datasets are utilized in the experiments. These datasets are also utilized as real-world streaming benchmarks in many state-of-the-art (online) studies (Wang et al., 2013; Zhang et al., 2016; Zhao et al., 2018; Chen et al. 2021) for mining data streams. Table 3 summarizes the attributes including the imbalance ratio of each binary dataset (last column). These original datasets except “w8a” are also preprocessed by the one-side subsampling method (Kubat & Matwin, 1997), which selectively subsamples the minority classes such that they are suitable for the severely imbalanced online classification tasks. The size of each subsample is approximately 10% of the positive samples. We use classification cost, sparsity, and running time as the comparison metrics. For the sparsity comparison, we utilize the Hoyer measurement (Hurley and Rickard, 2009) of the weight vector $\boldsymbol{w} \in \mathcal{X}$ as the sparsity measurement: $S(\boldsymbol{w}) = \frac{\sqrt{d} - (\frac{\|\boldsymbol{w}\|_1}{\|\boldsymbol{w}\|_2})}{\sqrt{d-1}}$, where d is the length of $\boldsymbol{w}$, $0 \le S(\boldsymbol{w}) \le 1$. The smaller the magnitude, the lower the sparsity of the measure.

5.2 Competing algorithms

In the experiments, we compare CSGOL/PCSGOL-I/II/III/IV with several state-of-the-art sparse group online learning algorithms and cost-sensitive (sparse) online learning algorithms: DAGL (Yang et al., 2010), DASGL (Yang et al., 2010), DAESGL (Yang et al., 2010), OGFS (Wang et al., 2015), CSOGD-I/II (Wang et al., 2013), CSTG-I/II (Chen et al., 2017), and CSRDA-I/II (Chen et al., 2021). More specifically, DAGL (Yang et al., 2010) is a group online learning method; DASGL (Yang et al., 2010), DAESGL (Yang et al., 2010), and OGFS (Yang et al., 2010) are sparse group online learning methods; CSOGD-I/II (Wang et al., 2013) are cost-sensitive online learning methods; CSTG-I/II (Chen et al., 2017) and CSRDA-I/II (Chen et al., 2021) are cost-sensitive sparse online learning methods; and our proposed CSGOL-I/II/III/IV and PCSGOL-I/II/III/IV are cost-sensitive sparse group online learning methods.

5.3 Experimental settings

We implement CSTG-I/II, CSRDA-I/II, CSGOL-I/II/III/IV, and PCSGOL-I/II/III/IV in MATLAB. The MATLAB implementations of DAGL (Yang et al., 2010), DASGL (Yang et al., 2010), DAESGL (Yang et al., 2010), OGFS (Wang et al., 2015), and CSOGD-I/II (Wang et al., 2013) are conducted from (Yang et al., 2010), (Wang et al., 2015), and (Wang et al., 2013), respectively. For a fair comparison, the same experimental setup is applied to all algorithms. We set $\mu_+ = 0.9$ and $\mu_- = 0.1$ for the misclassification cost measurement. After the preliminary studies, we set the group size by $d_k = 10$, length of the sliding window by $L = 100$, and regularized parameters are tested by $\gamma = 0.05$, $\lambda_1 = 0.1$, $\lambda_2 = 0.05$, and $\lambda_3 = 0.05$ for CSGOL and by $\sigma = 0.1$, $\lambda_1 = 0.01$, $\lambda_2 = 0.05$, and $\lambda_3 = 0.05$ for PCSGOL. All other parameter values are determined based upon the recommendations in Yang et al. (2010), Wang et al. (2013), and Wang et al. (2015). One hundred independent runs for each dataset are performed, and the average result of each method is reported. We perform all experiments on a Windows machine with a 3.7-GHz Intel Core processor and 64.0-GB main memory.

5.4 Overall comparisons

Table 4 presents the overall performance, including misclassification cost (± standard deviation), Hoyer sparsity measurement (± standard deviation), and average running time of all competing algorithms on these imbalanced streaming datasets. Some observations can be summarized as follows: (1) In terms of the average misclassification cost, CSGOL-IV achieved the lowest cost on the "PCMAC" dataset, followed by PCSGOL-IV; CSRDA-II achieved the lowest cost on the "MITFace" dataset, followed by CSGOL-IV; otherwise, PCSGOL-III or PCSGOL-IV exhibits the best performance on the other

Table 4 Performance comparison of different algorithms on all datasets. Best results are highlighted in bold

Dataset/Metrics	Method	Cost	Sparsity (%)	Running time (s)	Dataset/Metrics	Method	Cost	Sparsity (%)	Running time (s)
PCMAC	DAGL	781.11 ± 8.42	21.26 ± 0.22	**0.062**	a9a	DAGL	3067.18 ± 16.32	47.23 ± 5.32	**0.150**
	DASGL	735.58 ± 10.04	75.15 ± 0.56	0.066		DASGL	3373.95 ± 13.80	47.14 ± 5.21	0.191
	DAESGL	569.79 ± 6.42	**76.76** ± **0.79**	0.088		DAESGL	3547.53 ± 6.63	52.83 ± 1.53	0.223
	OGFS	230.51 ± 10.80	20.47 ± 0.35	0.067		OGFS	3606.65 ± 6.23	45.04 ± 5.62	0.161
	CSOGD-I	193.90 ± 20.53	23.38 ± 0.68	0.068		CSOGD-I	2565.04 ± 115.10	24.42 ± 0.07	0.162
	CSOGD-II	135.71 ± 7.91	21.23 ± 0.21	0.069		CSOGD-II	2520.00 ± 118.63	23.53 ± 0.29	0.163
	CSTG-I	137.02 ± 7.79	63.96 ± 0.86	0.065		CSTG-I	1829.10 ± 13.06	17.11 ± 0.10	0.151
	CSTG-II	136.12 ± 7.82	66.47 ± 1.09	0.314		CSTG-II	1813.99 ± 12.96	17.43 ± 0.04	0.409
	CSRDA-I	137.02 ± 7.79	70.23 ± 0.11	0.100		CSRDA-I	1821.51 ± 13.74	17.46 ± 0.01	0.176
	CSRDA-II	187.17 ± 5.17	21.33 ± 0.24	0.071		CSRDA-II	1389.64 ± 14.50	52.35 ± 0.54	0.184
	CSGOL-I	160.93 ± 9.16	21.26 ± 0.22	0.068		CSGOL-I	1828.10 ± 16.56	47.77 ± 5.07	0.152
	CSGOL-II	384.43 ± 28.79	21.25 ± 0.20	0.064		CSGOL-II	1579.32 ± 31.15	76.58 ± 14.92	0.155
	CSGOL-III	165.92 ± 4.38	48.29 ± 1.91	0.859		CSGOL-III	1337.21 ± 13.16	**86.89** ± **6.25**	0.610
	CSGOL-IV	**121.28** ± **16.32**	23.33 ± 0.79	0.864		CSGOL-IV	1246.85 ± 8.90	83.78 ± 12.60	0.614
	PCSGOL-I	200.71 ± 6.54	21.24 ± 0.07	0.137		PCSGOL-I	1752.63 ± 7.95	42.34 ± 0.24	0.200
	PCSGOL-II	196.64 ± 18.99	19.67 ± 0.16	0.137		PCSGOL-II	1675.00 ± 9.61	28.99 ± 1.57	0.201
	PCSGOL-III	230.97 ± 43.76	19.27 ± 0.16	0.463		PCSGOL-III	1258.78 ± 70.20	25.92 ± 1.21	0.363
	PCSGOL-IV	132.86 ± 10.86	19.77 ± 0.18	0.474		PCSGOL-IV	**1167.08** ± **10.70**	26.62 ± 0.18	0.364

Table 4 (continued)

Dataset/Metrics	Method	Cost	Sparsity (%)	Running time (s)	Dataset/Metrics	Method	Cost	Sparsity (%)	Running time (s)
Spambase	DAGL	182.64 ± 8.92	34.16 ± 4.98	**0.008**	usps1all	DAGL	598.18 ± 9.52	22.90 ± 1.26	0.031
	DASGL	227.67 ± 6.73	34.27 ± 4.93	0.009		DASGL	600.41 ± 9.26	22.92 ± 1.25	0.035
	DAESGL	253.47 ± 3.77	36.63 ± 4.87	0.011		DAESGL	607.83 ± 4.13	23.26 ± 1.68	0.040
	OGFS	267.02 ± 6.15	33.68 ± 4.84	**0.008**		OGFS	609.68 ± 3.66	22.67 ± 1.52	0.032
	CSOGD-I	135.96 ± 4.74	36.77 ± 0.42	**0.008**		CSOGD-I	304.34 ± 6.96	19.50 ± 0.15	0.032
	CSOGD-II	134.25 ± 5.09	41.52 ± 0.62	**0.008**		CSOGD-II	301.85 ± 7.42	22.70 ± 0.20	0.032
	CSTG-I	117.24 ± 3.23	21.06 ± 0.92	0.009		CSTG-I	277.75 ± 4.69	13.84 ± 0.38	0.031
	CSTG-II	115.57 ± 2.98	44.37 ± 1.95	0.012		CSTG-II	275.89 ± 4.68	13.83 ± 0.04	0.125
	CSRDA-I	116.25 ± 3.05	24.64 ± 0.09	0.009		CSRDA-I	277.75 ± 4.69	13.80 ± 0.59	0.039
	CSRDA-II	85.25 ± 2.56	50.33 ± 1.49	0.010		CSRDA-II	206.01 ± 3.39	48.17 ± 2.87	0.036
	CSGOL-I	113.21 ± 3.96	34.16 ± 4.98	**0.008**		CSGOL-I	271.97 ± 5.17	22.88 ± 1.26	0.030
	CSGOL-II	110.12 ± 2.83	62.48 ± 9.44	**0.008**		CSGOL-II	280.91 ± 4.74	21.45 ± 1.52	**0.029**
	CSGOL-III	82.11 ± 2.35	**73.37 ± 5.11**	0.024		CSGOL-III	187.26 ± 3.83	**92.01 ± 2.86**	0.185
	CSGOL-IV	75.58 ± 3.14	61.09 ± 10.59	0.024		CSGOL-IV	165.01 ± 3.68	21.76 ± 1.65	0.186
	PCSGOL-I	108.38 ± 2.30	50.58 ± 0.81	0.009		PCSGOL-I	278.70 ± 1.82	50.37 ± 1.33	0.045
	PCSGOL-II	114.96 ± 2.91	33.80 ± 2.42	0.010		PCSGOL-II	276.41 ± 5.63	30.21 ± 2.17	0.045
	PCSGOL-III	83.57 ± 5.97	30.37 ± 1.90	0.016		PCSGOL-III	**154.98 ± 13.43**	24.12 ± 1.53	0.101
	PCSGOL-IV	**70.64 ± 2.96**	27.60 ± 0.07	0.015		PCSGOL-IV	173.71 ± 3.28	13.37 ± 0.14	0.101

Table 4 (continued)

Dataset/Metrics	Method	Cost	Sparsity (%)	Running time (s)	Dataset/Metrics	Method	Cost	Sparsity (%)	Running time (s)
MITFace	DAGL	454.30 ± 6.64	22.23 ± 0.95	0.031	w8a	DAGL	4892.99 ± 24.92	67.91 ± 1.11	0.621
	DASGL	452.47 ± 12.87	22.31 ± 0.97	0.034		DASGL	5602.36 ± 18.74	67.73 ± 1.06	**0.575**
	DAESGL	454.80 ± 1.12	21.88 ± 0.88	0.039		DAESGL	5956.04 ± 11.10	57.46 ± 0.65	0.669
	OGFS	454.78 ± 2.45	22.74 ± 0.98	0.031		OGFS	6093.52 ± 7.61	62.26 ± 1.01	0.592
	CSOGD-I	247.94 ± 5.90	33.52 ± 0.18	0.031		CSOGD-I	978.65 ± 15.39	22.57 ± 0.13	0.631
	CSOGD-II	242.92 ± 5.48	33.88 ± 0.17	0.031		CSOGD-II	858.11 ± 10.85	37.94 ± 0.24	0.597
	CSTG-I	241.74 ± 4.10	22.47 ± 0.15	0.030		CSTG-I	901.63 ± 12.93	16.98 ± 0.07	0.613
	CSTG-II	240.45 ± 3.59	22.49 ± 0.02	0.157		CSTG-II	887.84 ± 12.44	17.26 ± 0.02	1.084
	CSRDA-I	241.72 ± 3.66	22.44 ± 0.02	0.042		CSRDA-I	892.32 ± 14.94	17.28 ± 0.05	0.642
	CSRDA-II	**186.09** ± **2.06**	24.15 ± 1.22	0.033		CSRDA-II	1059.93 ± 11.49	58.47 ± 0.44	0.646
	CSGOL-I	232.11 ± 4.02	22.20 ± 0.95	**0.029**		CSGOL-I	944.37 ± 13.58	65.44 ± 0.72	**0.575**
	CSGOL-II	283.83 ± 4.95	21.52 ± 1.09	**0.029**		CSGOL-II	2116.80 ± 20.64	71.35 ± 0.88	0.586
	CSGOL-III	191.88 ± 1.95	**91.05** ± **2.37**	0.197		CSGOL-III	873.70 ± 11.45	**73.20** ± **5.23**	2.463
	CSGOL-IV	189.55 ± 2.18	21.78 ± 1.08	0.197		CSGOL-IV	801.14 ± 12.16	70.16 ± 1.06	2.468
	PCSGOL-I	190.50 ± 2.08	19.61 ± 0.78	0.046		PCSGOL-I	2467.50 ± 13.04	47.75 ± 0.32	0.812
	PCSGOL-II	243.24 ± 3.83	19.26 ± 1.15	0.046		PCSGOL-II	1077.95 ± 8.56	38.51 ± 0.62	0.777
	PCSGOL-III	222.43 ± 5.15	19.59 ± 0.80	0.107		PCSGOL-III	930.32 ± 66.52	34.66 ± 0.48	1.457
	PCSGOL-IV	198.01 ± 2.77	20.51 ± 0.02	0.106		PCSGOL-IV	**711.22** ± **8.41**	24.77 ± 0.26	1.443

datasets; (2) In terms of the sparsity measurement, in most cases, CSGOL-III obtains the highest sparsity with potential the least activated features for model interpretation on all the datasets except "PCMAC", where DAESGL presents the highest sparsity, followed by DASGL; (3) In terms of average running time, PCSGOL-III and PCSGOL-IV achieve the highest computational cost almost in all of the datasets since they need more time to compute the adjusted weighting factor during the online gradient updating.

Quantitatively, over these six datasets, the average misclassification cost by CSGOL-I/II/III/IV is reduced by 1.89, 8.44, 21.15, 61.14, and 62.78% compared to the average misclassification cost by CSRDA-I/II, CSTG-I/II, CSOGD-I/II, OGFS, and DAGL/DASGL/DAESGL, respectively. Similarly, the average misclassification cost by PCSGOL-I/II/III/IV is reduced by 2.35, 8.87, 21.52, 61.33, and 62.95% compared to the average misclassification cost by CSRDA-I/II, CSTG-I/II, CSOGD-I/II, OGFS, and DAGL/DASGL/DAESGL, respectively. There is no significant difference in the average misclassification cost by CSGOL-I/II/III/IV and PCSGOL-I/II/III/IV.

At the same time, over these six datasets, the average sparsity by CSGOL-I/II/III/IV is improved by 79.28, 46.80, 83.10, 81.11, 49.26, and 22.84% compared to the average sparsity by PCSGOL-I/II/III/IV, CSRDA-I/II, CSTG-I/II, CSOGD-I/II, OGFS, and DAGL/DASGL/DAESGL, respectively. This indicates that CSGOL-I/II/III/IV achieved the largest sparsity on average, followed by DAGL/DASGL/DAESGL, and CSRDA-I/II. Overall, the proposed CSGOL and PCSGOL methods achieved state-of-the-art performance by optimizing the online misclassification cost by averaging gradients of loss functions at the group level. However, the proposed CSGOL methods have better sparsity or potentially better model interpretation than the PCSGOL methods. The sparsity measurement comparisons validate the interpretation ability of CSGOL for handling high-dimensional streaming data.

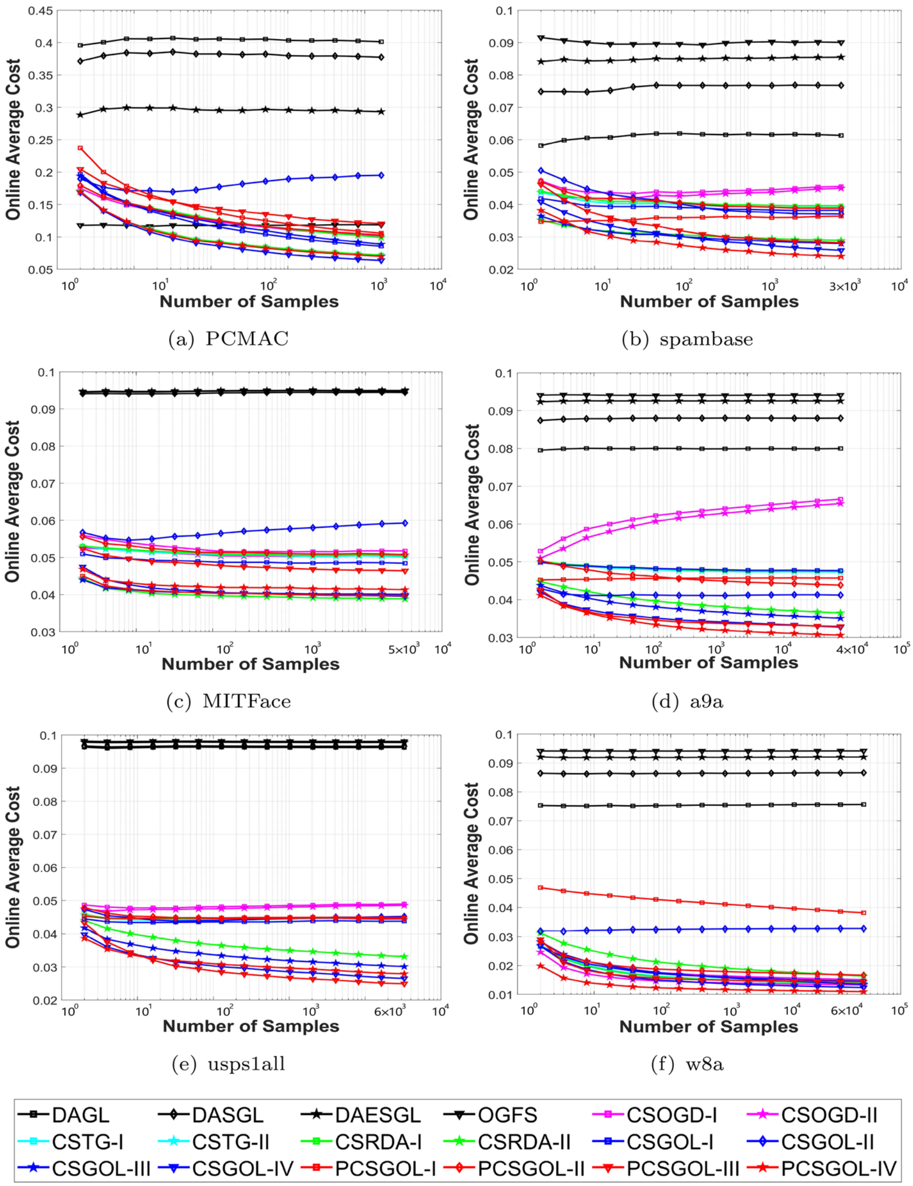
Finally, over these six datasets, the average running time of CSGOL-I/II/III/IV is almost 2.62, 1.74, 2.85, 2.92, and 2.71 times that of CSRDA-I/II, CSTG-I/II, CSOGD-I/II, OGFS, and DAGL/DASGL/DAESGL, respectively. Similarly, the average running time by PCSGOL-I/II/III/IV is almost 1.88, 1.25, 2.04, 2.10, and 1.94 times that by CSRDA-I/II, CSTG-I/II, CSOGD-I/II, OGFS, and DAGL/DASGL/DAESGL, respectively. This indicates that OGFS is the most efficient online algorithm. However, the average time consumptions of CSGOL and PCSGOL are very competitive with OGFS, making the proposed methods relatively fast to process high-throughput data streams.

5.5 Dynamic performance comparisons

In this section, we dynamically show the real-time classification performance of all competing algorithms as data streams sequentially come.

5.5.1 Dynamic cost comparisons

As shown in Fig. 2, we investigate the dynamic misclassification cost of all algorithms with the progression of a data stream. For these six data streams, the online average cost curves of CSGOL-III and PCSGOL-III/IV consistently dominate the corresponding curves of other algorithms without much variation. The superiority of CSGOL-III and PCSGOL-IV over others is evident on the "PCMAC" and "a9a" data streams, where the severe imbalance ratios of positives and negatives are present. This indicates that CSGOL-III and



(a) PCMAC

(b) spambase

(c) MITFace

(d) a9a

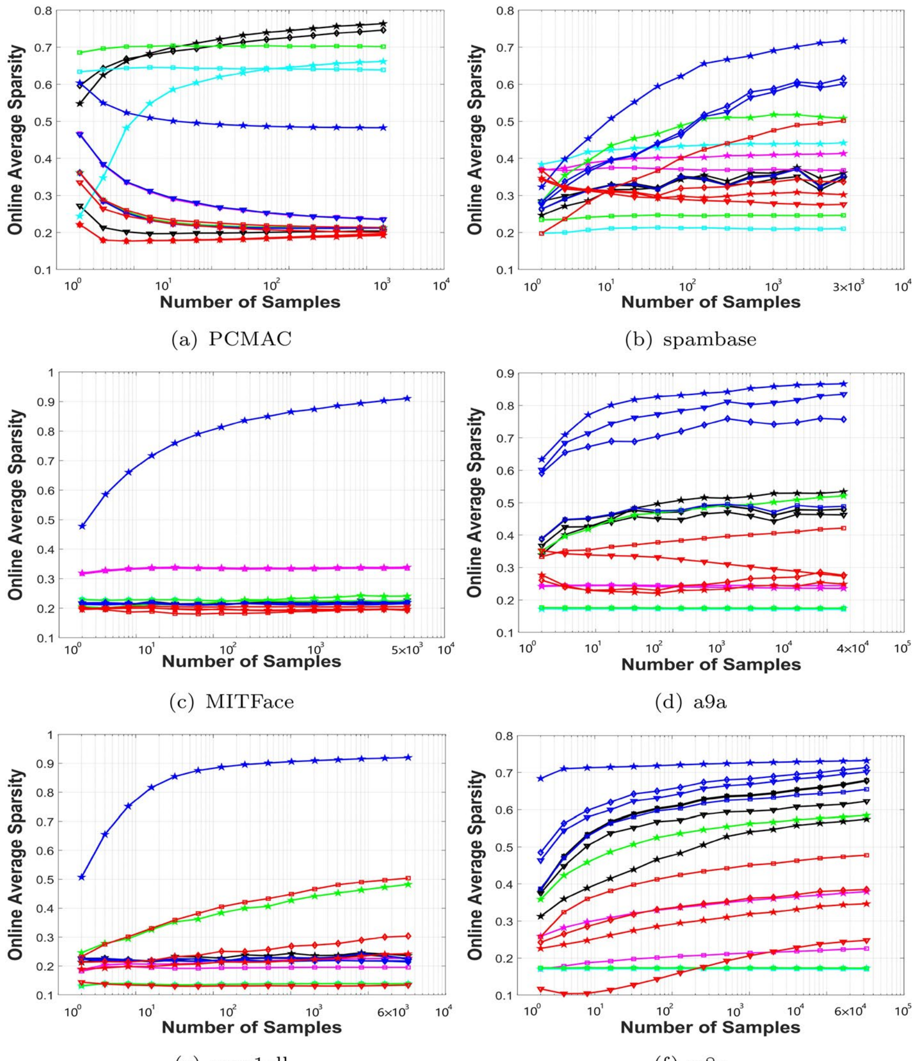
(e) usps1all

(f) w8a

(g) Legends of All Methods

Fig. 2 Dynamic learning curves in terms of online average misclassification cost of all competing algorithms as data streams progress. Legends for all methods also apply to Figs. 3, 5, 8, 9, and 10

PCSGOL-IV are able to capture the underlying structure of the minority classes associated with the ever-evolving distributions of imbalanced streaming data.



(a) PCMAC

(b) spambase

(c) MITFace

(d) a9a

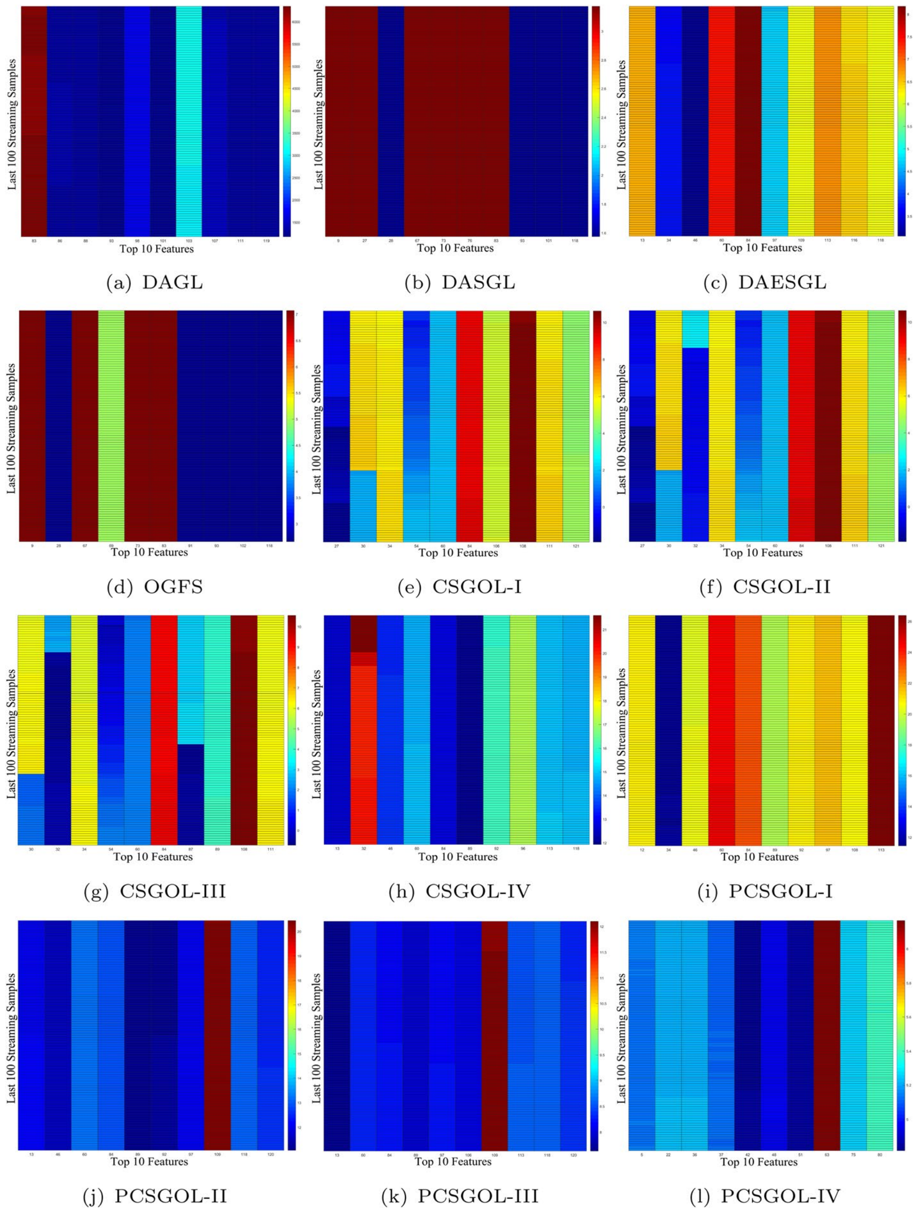
(e) usps1all

(f) w8a

Fig. 3 Dynamic learning curves in terms of online average sparsity of all competing algorithms as data streams progress

5.5.2 Dynamic sparsity comparisons and interpretation

Figure 3 presents the online average sparsity measurement of all algorithms with the progression of a data stream. CSGOL-III significantly achieves much higher sparsity than other methods on all datasets except the "PCMAC" dataset with extremely high feature dimensions, where DAESGL and DASGL dominate the curves of other methods. The Hoyer sparsity measurement achieved by CSGOL-III is approximately 0.90 and 0.70 on the "MITFace" and "w8a" datasets with over 300 features, respectively. This indicates that our proposed methods not only achieve lower misclassification cost for imbalanced data



(a) DAGL (b) DASGL (c) DAESGL

(d) OGFS (e) CSGOL-I (f) CSGOL-II

(g) CSGOL-III (h) CSGOL-IV (i) PCSGOL-I

(j) PCSGOL-II (k) PCSGOL-III (l) PCSGOL-IV

Fig. 4 Dynamic heatmaps on weight matrix are compared for model interpretation on the “a9a” dataset

streams but also obtain a better sparsity level for model interpretation in handling high-dimensional streaming data.

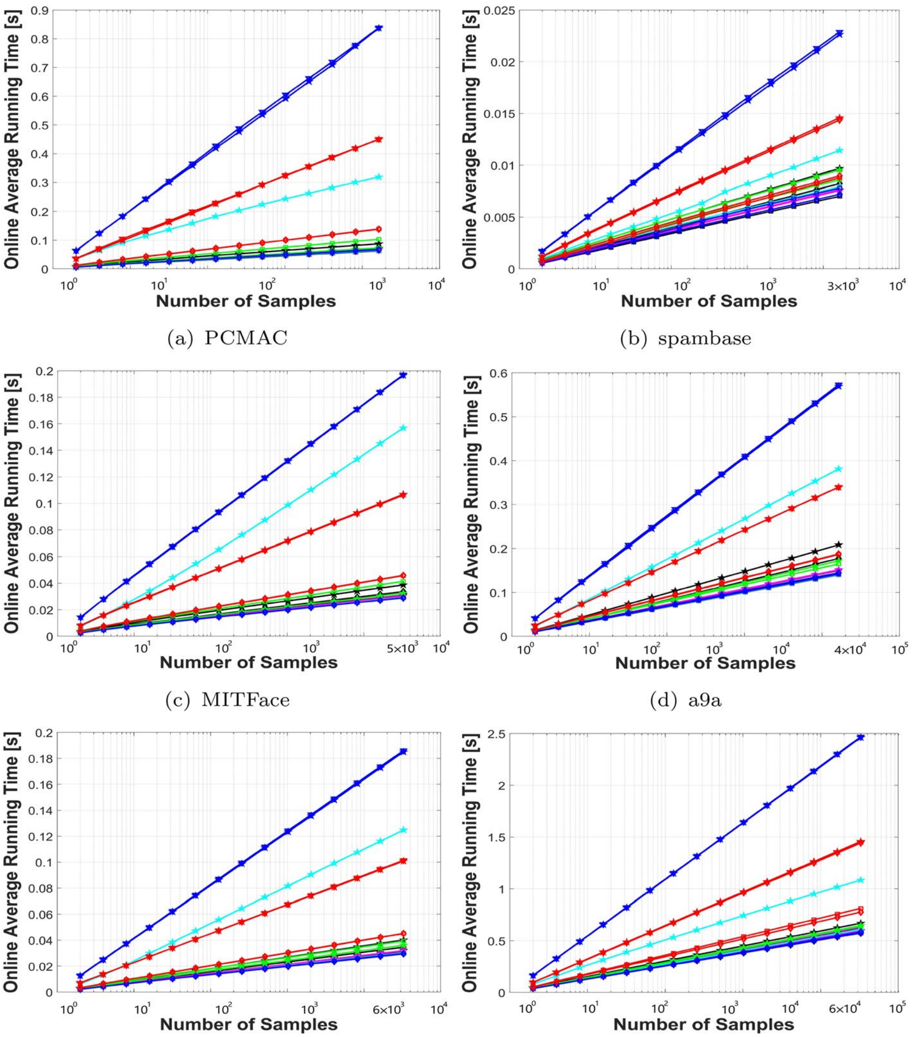
For the model interpretation comparisons, Fig. 4 presents the heatmaps on the weight matrix of the compared methods on the “a9a” dataset. The online prediction task is to determine whether a person makes over $50K$ in salary over a year. The x-axis represents the top 10 selected features determined by absolute values of the weight vectors, and the y-axis represents the last 100 streaming samples via convergence. We found that the 9-, 28-, 67-, 73-, 83-, 93-, and 118-th features [i.e., age (42-44), work class (never-worked), occupation (armed-forces), relationship (unmarried), hours-per-week, and native country (South Korean, Yugoslavia)] are selected multiple times among the four cost-insensitive sparse group online learning methods, i.e., DAGL, DASGL, DAESGL, and OGFS. The 27-, 30-, 32-, 34-, 54-, 60-, 84-, 108-, 111-, and 121-th [i.e., work class (without-pay), education (bachelors, 11-th grade, prof-school), occupation (tech-support, handlers-cleaners), and native country (United States, Laos, Haiti, Peru)] features are selected multiple times among the CSGOL-I/II/III/IV methods. The 48-, 60-, 84-, 89-, 92-, 97-, 113-, and 118-th features [i.e., marital status (divorced), occupation (handlers-cleaners), and native country (United States, Outlying US, Greece, Honduras, Hungary, Yugoslavia)] are selected multiple times among the PCSGOL-I/II/III/IV methods. This indicates that CSGOL achieves better interpretable features at the group level (i.e., work class, education, occupation, and native country) and features within groups than the DAGL, DASGL, DAESGL, OGFS, and PCSGOL methods.

5.5.3 Dynamic running time comparisons

Figure 5 presents the online average running time of one hundred independent runs for all methods on those six datasets. The average time consumptions of CSGOL and PCSGOL are approximately three and two times that of the most efficient online algorithm OGFS, making the proposed methods relatively fast to process high-throughput data streams. In addition, the average time consumed by CSGOL and PCSGOL is still competitive compared with the cost-sensitive sparse online learning methods such as CSRDA-I/II and CSTG-I/II. These results validate the efficiency of CSGOL and PCSGOL compared with state-of-the-art methods.

5.6 Parameter sensitivity analysis

To run CSGOL and PCSGOL, one needs to specify several parameters, especially the regularization parameters λ_1, λ_2, and λ_3. Since the sum of λ_2 and λ_3 determines the sparsity level of both CSGOL and PCSGOL, we investigate how the alterations of λ_1 and $(\lambda_2 + \lambda_3)$ affect the performance of by grid search when γ and σ are fixed, respectively. Taking the severely imbalanced dataset "a9a" as an example, we summarize the performance of CSGOL and PCSGOL when λ_1 and $(\lambda_2 + \lambda_3)$ are both selected from $[5 \times 10^{-3}, 1 \times 10^{-2}, 5 \times 10^{-2}, 1 \times 10^{-1}]$. In Fig. 6, we compare the dynamic misclassification cost when varying these parameters. It is evident that CSGOL-III/IV achieves a much lower cost than CSGOL-I/II in Fig. 6a. The performances of CSGOL-II/III/IV are relatively stable without much variation when $\lambda_1 \in [5 \times 10^{-3}, 1 \times 10^{-2}, 5 \times 10^{-2}, 1 \times 10^{-1}]$ and $\lambda_2 + \lambda_3 \in [5 \times 10^{-3}, 1 \times 10^{-2}, 5 \times 10^{-2}, 1 \times 10^{-1}]$; however, the misclassification cost of CSGOL-I fluctuates as proper λ_1 and $(\lambda_2 + \lambda_3)$ are vital to smooth the model. When λ_1 is fixed, the misclassification cost of CSGOL-I is significantly decreased when the values of $(\lambda_2 + \lambda_3)$ are decreased from 1×10^{-1} to 5×10^{-3}. The reason is that a relatively large $(\lambda_2 + \lambda_3)$ will promote the sparsity level; however, it deteriorates the misclassification cost of CSGOL. Overall, CSGOL-II/III/IV are relatively robust to these parameters. Similar results are observed for PCSGOL in Fig. 6b.



(a) PCMAC

(b) spambase

(c) MITFace

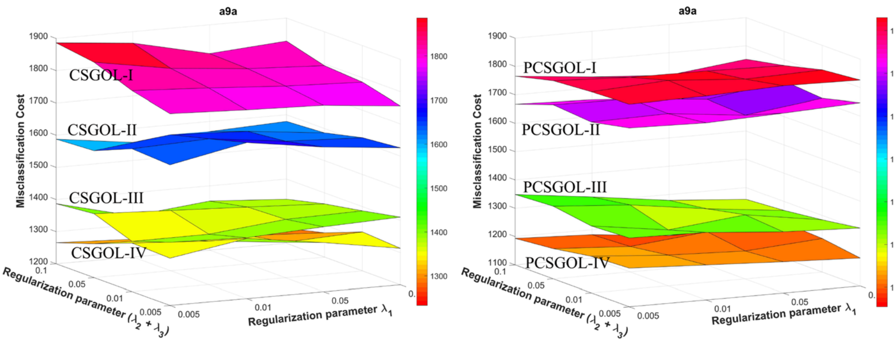
(d) a9a

(e) usps1all

(f) w8a

Fig. 5 Dynamic learning curves in terms of online average running time (seconds) of all competing algorithms as data streams progress

We also investigate how the size of sliding windows L affects the performance of CSGOL and PCSGOL. Figure 7 presents the online average accumulated cost of CSGOL and PCSGOL when L is selected from [10, 20, 50, 100, 200, 500, 1000] on the "w8a" dataset. Overall, CSGOL-I/IV and PCSGOL-III/IV are relatively robust to the size of sliding windows L since their online average accumulated cost is stable without much variation. The reason could be the dynamical design of $\kappa_t = \kappa \frac{\sum_{r=max\{1,t-L+1\}}^{t} \mathbb{I}_{(y_r=-1)}+1}{\sum_{r=max\{1,t-L+1\}}^{t} \mathbb{I}_{(y_r=+1)}+1}$, which is relatively robust to L since it is embedded with the initiation of the imbalance parameter κ for imbalance online learning. Moreover, the misclassification cost of both CSGOL and PCSGOL is almost minimized when $L = 100$. The reason is probably that a small L leads to an



(a) CSGOL (b) PCSGOL

Fig. 6 Sensitivity analysis of regularization parameters of CSGOL (left panel) and PCSGOL (right panel) on the "a9a" dataset

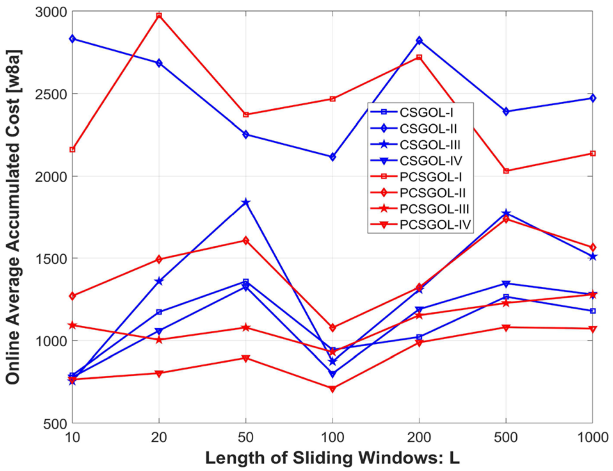


Fig. 7 Sensitivity analysis of length of sliding windows L on the "w8a" dataset

Table 5 Summarization of real world (binary) datasets for online anomaly detection in experiments

Dataset	#Sample	#Feature	#Positive:#Negative
KDDCUP'08[c]	102, 294	117	$623 : 101,671 \approx 1 : 163.2$
Speech[d]	3686	400	$61 : 3,625 \approx 1 : 59.4$
Musk[e]	3062	166	$97 : 2,965 \approx 1 : 30.6$
Covtype (class 4 vs. class 2)[f]	286, 048	54	$2,747 : 283,301 \approx 1 : 103.1$
Credit Card[g]	284, 807	29	$492 : 284,315 \approx 1 : 578$

[c] https://www.kdd.org/kdd-cup/view/kdd-cup-2008/Data

[d] http://odds.cs.stonybrook.edu/speech-dataset/

[e] http://odds.cs.stonybrook.edu/musk-dataset/

[f] https://archive.ics.uci.edu/ml/datasets/covertype

[g] https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

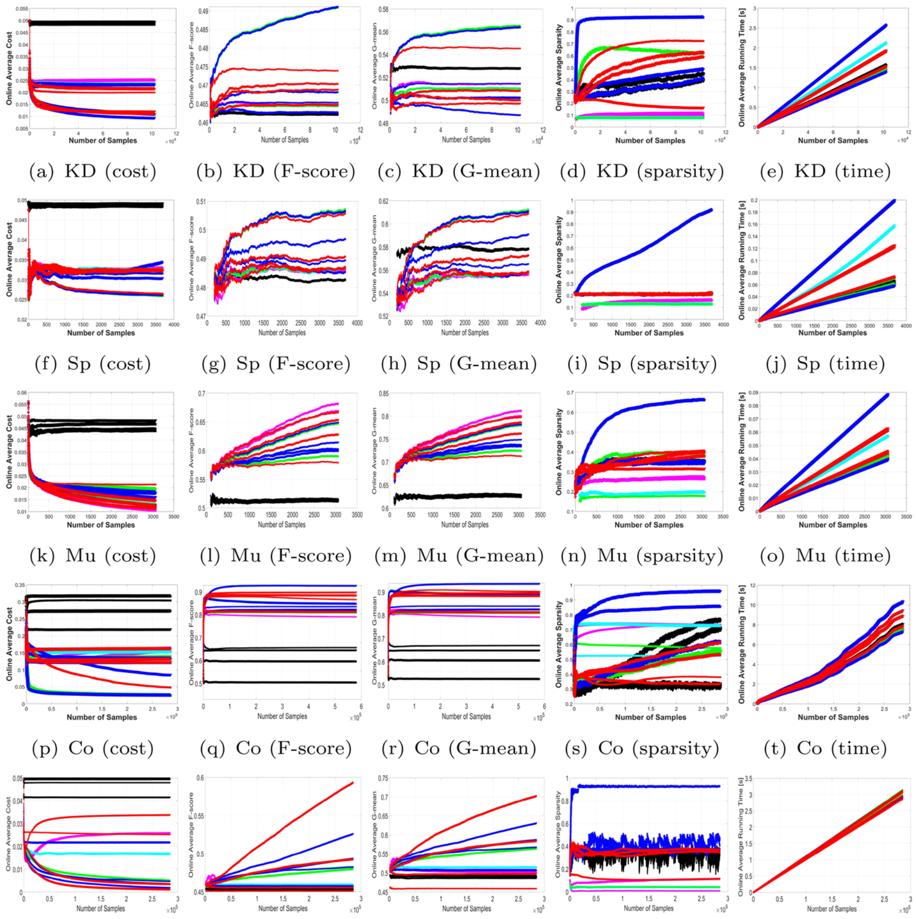
underestimation of the true imbalance ratio; however, a large L leads to an overestimation of the true imbalance ratio, which can increase the misclassification cost. Therefore, we set $L = 100$ in the experiments to balance these two unexpected cases.

5.7 Online anomaly detection

The proposed cost-sensitive sparse group online classification algorithms (i.e., CSGOL-I/II/III/IV and PCSGOL-I/II/III/IV) can potentially be applied to solve a wide range of real-world applications in streaming data mining. To verify their practical application value, we apply them to tackle the five online anomaly detection tasks (Table 5) below:

- *Medical imaging (KDDCUP'08)*: The KDDCUP'08 breast cancer dataset belongs to the medical image anomaly detection problem. The main goal is to detect breast cancer from X-ray images, where "benign" is assigned as normal and "malignant" is abnormal.
- *Speech accents recognition (Speech)*: The real-world speech dataset consists of 3,686 segments of English speech spoken with different accents. This dataset is provided by the Speech processing group at Brno University of Technology, Czech Republic. The majority data corresponds to the American accent, and only 1.65% corresponds to one of seven other accents (these are referred to as anomalies).
- *Musk classification (Musk)*: This dataset describes a set of 102 molecules, of which 39 are judged by human experts to be musks and the remaining 63 molecules are judged to be nonmusks. The goal is to learn to predict whether new molecules will be musks or nonmusks.
- *Forest covertype (Covtype)*: This dataset contains information about the forest cover type of 30 × 30-meter cells obtained from the US Forest Service Region 2 Resource Information System. It contains 581,012 instances, 54 attributes, and 7 class labels (i.e., class 1: Spruce/Fir, class 2: Lodgepole Pine, class 3: Ponderosa Pine, class 4: Cottonwood/Willow, class 5: Aspen, class 6: Douglas-fir, and class 7: Krummholz) related to different forest cover types. However, the original dataset is unsuitable for the imbalance classification task. Thus, we conduct the experiments by considering two extreme classes, where class 4 (Cottonwood/Willow) is the most minority class and class 2 (Lodgepole Pine) is the most majority class, i.e., their imbalance ratio is $2,747 : 283,301 \approx 1 : 103.1$ with a total 286,048 instances.
- *Credit card fraud detection (creditcard)*: The dataset contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

Figure 8 exhibits the experimental results on these four datasets. We make several observations: (1) CSGOL-III/IV achieve the lowest misclassification cost on the "KDDCUP'08," "Speech," and "covtype" datasets, while CSOGD-II and PCSGOL-III achieve the lowest misclassification cost on the "Musk" dataset. This confirms the superiority of cost-sensitive online learning for imbalanced data streams. (2) CSGOL-III/IV achieves the highest sparsity measurement among all four tasks in different domains. Overall,



(a) KD (cost) (b) KD (F-score) (c) KD (G-mean) (d) KD (sparsity) (e) KD (time)

(f) Sp (cost) (g) Sp (F-score) (h) Sp (G-mean) (i) Sp (sparsity) (j) Sp (time)

(k) Mu (cost) (l) Mu (F-score) (m) Mu (G-mean) (n) Mu (sparsity) (o) Mu (time)

(p) Co (cost) (q) Co (F-score) (r) Co (G-mean) (s) Co (sparsity) (t) Co (time)

(u) Cr (cost) (v) Cr (F-score) (w) Cr (G-mean) (x) Cr (sparsity) (y) Cr (time)

Fig. 8 Dynamic learning curves in terms of online metrics (i.e., cost, sparsity, F-score, G-mean, and running time) of all competing algorithms for online anomaly detection

CSGOL-III/IV achieves a better balance between low cost and high sparsity than the other methods on these four datasets. This indicates that the online cost-sensitive and group sparsity optimizations for CSGOL are effective in handling high-dimensional and imbalanced data streams using regularized dual averaging over time. (3) Although the running speed of CSGOL is the slowest among all of the algorithms, its cost and sparsity performance is relatively competitive with those of the other methods among these datasets. (4) For the credit card dataset with the severe imbalanced ratio, we observe that PCSGOL-III achieved the lowest misclassification cost and highest classification F1-score and G-mean of all the competing algorithms, with a competitive sparsity metric as well. CSGOL-IV achieved highest sparsity with relative low misclassification cost and high classification F1-score and G-mean. In conclusion, all promising results confirm the superiority of our proposed CSGOL algorithms for real-world online anomaly detection problems, where the datasets are embedded with high-dimensional and highly class-imbalanced properties.

Table 6 Summarization of real-world gas sensor data streams

Dataset/File	Ethylene_CO	Ethylene_Methane
# Samples	4, 208, 261	4, 178, 504
# Features	16	16
#Abnormal samples: #Normal samples	$966,564 : 3,241,697 \approx 1 : 3.4$	$848,598 : 3,329,906 \approx 1 : 3.9$

5.8 Results on real-world data streams

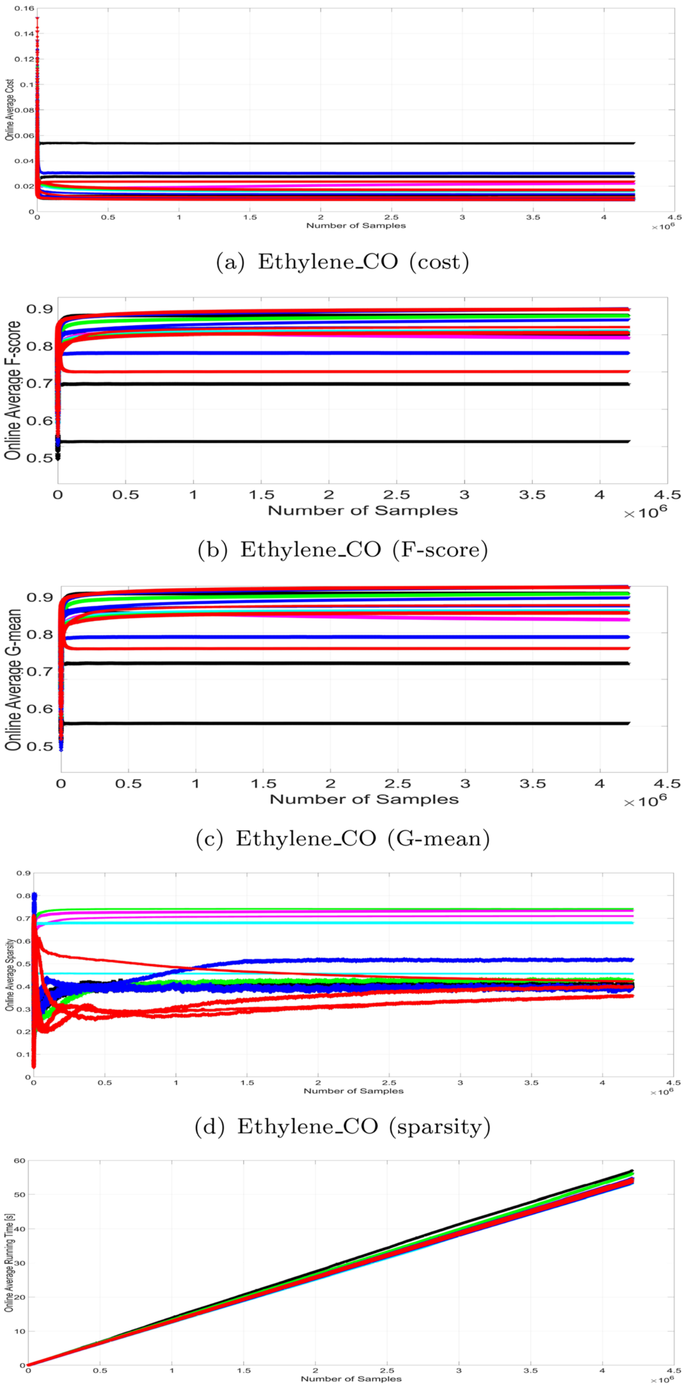
To test our methods on real-world data streams, we tried our best to located a public dataset - Gas sensor array under dynamic gas mixtures at[1] (Fonollosa et al., 2015). Using this real-world sensory data, we added additional experiments to compare our proposed methods with all baselines in terms of the misclassification cost, sparsity, and running time.

This labeled real-world sensory data contains the acquired time series from 16 chemical sensors exposed to gas mixtures at varying concentration levels. In particular, Fonollosa et al. (2015) generated two gas mixtures: Ethylene and Methane in air, and Ethylene and CO in air. Each measurement was constructed by the continuous acquisition of the 16-sensor array signals for a duration of about 12 h without interruption. The data is presented in two different files and each file contains the data from one mixture. The file "ethylene_CO.txt" contains the recordings from the sensors when exposed to mixtures of Ethylene and CO in air. The file "ethylene_methane.txt" contains the acquired time series induced by the mixture of Methane and Ethylene in air.

Using those two files, we generated two datasets of real-world gas sensor data streams: "Ethylene_CO" and "Ethylene_Methane". For each dataset, we considered the air quality was abnormal if and only if Methane (or CO) concentration > 0 ppm and Ethylene concentration > 0 ppm. Otherwise, the air quality was normal. Table 6 summarizes the characteristics of "Ethylene_CO" and "Ethylene_Methane" datasets. It is apparent that both datasets have imbalanced class distributions, which are suitable cases for the anomaly detection task.

As shown in Figs. 9 and 10, we found that the overtime cost curves of PCSGOL-III/IV or CSGOL-IV consistently dominate the corresponding curves of other algorithms without much variation. Moreover, the average running time of CSGOL and PCSGOL is much less than the baseline algorithms. Since the dimension of each dataset is relatively low, in most cases, the sparsity performances of CSGOL and PCSGOL are lower than CSRDA, CSOGD, and CSTG, but are very competitive with them in terms of interpretations. In summary, the proposed CSGOL/PCSGOL methods achieved better performances than the existing methods on real-world sensory data streams.

[1] https://goo.gl/zcAijP.



(a) Ethylene_CO (cost)

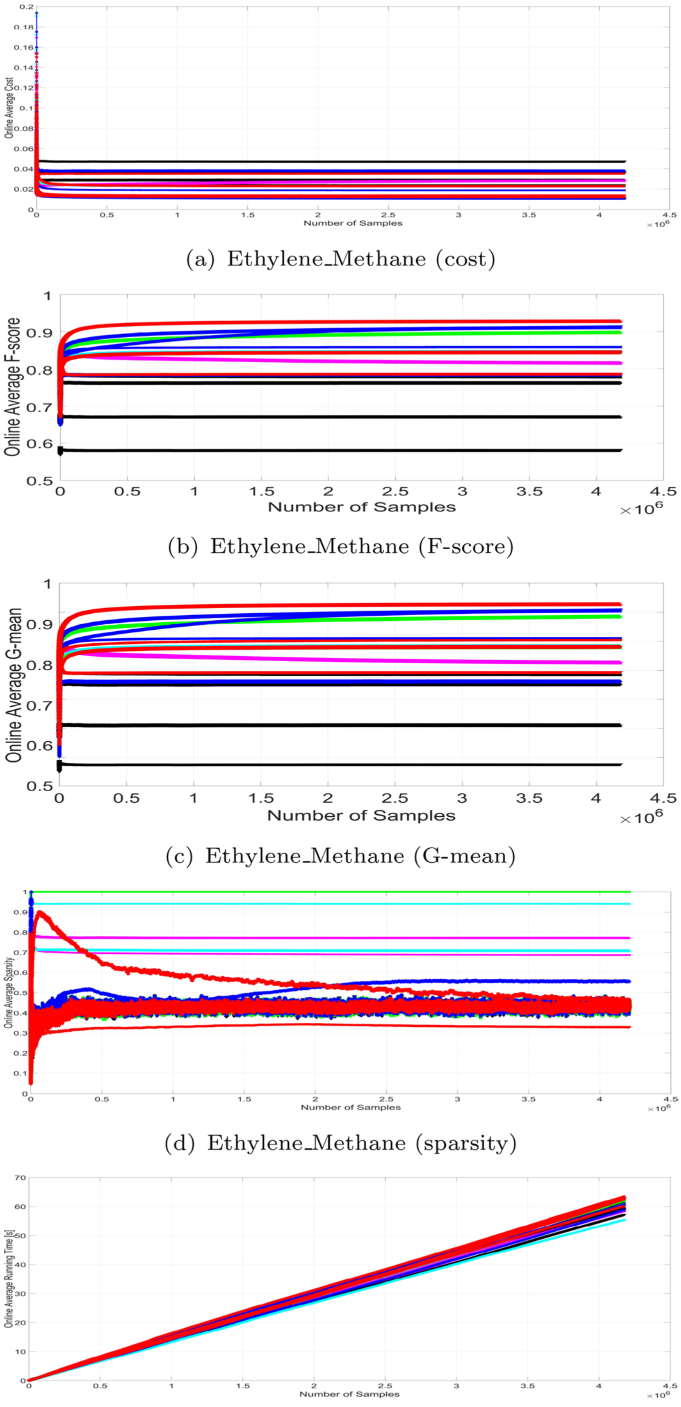
(b) Ethylene_CO (F-score)

(c) Ethylene_CO (G-mean)

(d) Ethylene_CO (sparsity)

(e) Ethylene_CO (time)

Fig. 9 Dynamic learning curves in terms of online metrics (i.e., cost, F-score, G-mean, sparsity, and running time) of all competing algorithms



(a) Ethylene_Methane (cost)

(b) Ethylene_Methane (F-score)

(c) Ethylene_Methane (G-mean)

(d) Ethylene_Methane (sparsity)

(e) Ethylene_Methane (time)

Fig. 10 Dynamic learning curves in terms of online metrics (i.e., cost, F-score, G-mean, sparsity, and running time) of all competing algorithms

6 Conclusion

In this paper, we propose a novel framework CSGOL and its proximal version PCSGOL for cost-sensitive sparse group online learning. CSGOL and PCSGOL meet the key expectations set by ever-evolving skewed data streams with high dimensionality: (1) They have at most $\mathcal{O}(d)$ space complexity and $\mathcal{O}(\sqrt{T})$ or $\mathcal{O}(\sqrt{T} + \log T)$ complexity for regret bounds, (2) CSGOL can automatically seek a favorable trade-off between low classification cost and high sparsity among and within groups, and (3) They have a relatively fast response time and are robust to different parameter settings. Experimental evaluations on multiple benchmark datasets verify the effectiveness of the proposed methods in the cost-sparsity tradeoff for the minority class of the data streams.

As part of future work, we plan to improve the stability of CSGOL and PCSGOL by incorporating ensemble learning strategies. Additionally, we may introduce a general and adaptive robust loss function (Barron, 2019) in the proposed methods to address the challenges in noisy data streams and improve their noise tolerance properties. Another potential direction is to design effective local adaptive strategies in both CSGOL and PCSGOL for imbalanced streaming data with abrupt and gradual concept drifts (Brzezinski et al., 2021), which may achieve good performance but are least affected by the changing statistical properties of data streams. Last, by following the structural risk minimization principle, we may adopt the passive-aggressive update rule in Zhang et al. (2016) to extend and apply CSGOL and PCSGOL for feature-evolving data streams with old features that may vanish and new features that may appear over time.

Appendix A Proof of closed-form solution of CSGOL

Proof To prove the closed-form solution of CSGOL, we introduce two lemmas below. □

Lemma 1 (Solution of ℓ_1 Proximity Operator) *The soft-thresholding function,* $\mathbf{w}^* = \operatorname{argmin}_{\mathbf{w}\in\mathbb{R}^d}\{\frac{1}{2}\|\mathbf{w}-\mathbf{x}\|_2^2 + \lambda\|\mathbf{w}\|_1\}$, *has the closed-form solution*: $\mathbf{w}^* = \operatorname{sign}(\mathbf{x}) \bigodot \max(|\,\mathbf{x}\,| - \lambda\mathbf{1}, \mathbf{0})$, *where* $\bigodot$ *is the elementwise multiplication,* $\mathbf{1} = [1, 1, \dots, 1]^T \in \mathbb{R}^d$, *and* $\mathbf{0} = [0, 0, \dots, 0]^T \in \mathbb{R}^d$.

Lemma 2 (Closed-Form Solution of DAGL Yang et al., 2010) *For the timestamp t, based on the current model* $\mathbf{w}_t \in \mathbb{R}^d$, *we use the following optimization solution to update the next-round model* $\mathbf{w}_{t+1} \in \mathbb{R}^d$:

$$\mathbf{w}_{t+1} = \operatorname{argmin}_{\mathbf{w}\in\mathbb{R}^d}\{\bar{\mathbf{g}}_t^T\mathbf{w} + \frac{\gamma}{2\sqrt{t}}\|\mathbf{w}\|_2^2 + \lambda\sum_{k=1}^{K}\sqrt{d_k}\|\mathbf{w}^{(k)}\|_2\} \tag{A1}$$

where $\gamma > 0$ *and* $\lambda > 0$ *are the regularization parameters,* d_k *is the size of group* k $(k = 1, 2, \dots, K)$, $\sum_{k=1}^{K} d_k = d$, $\mathbf{w} = [(\mathbf{w}^{(1)})^T, (\mathbf{w}^{(2)})^T, \dots, (\mathbf{w}^{(K)})^T]^T \in \mathbb{R}^d$, *and* $\mathbf{w}^{(k)} \in \mathbb{R}^{d_k}$ *is the k-th group vector of* $\mathbf{w}$. *Then, given* $\bar{\mathbf{g}}_t$ *in each iteration, for the k-th group* $(k = 1, 2, \dots, K)$, *the optimal closed-form solution is updated correspondingly as follows*: $\mathbf{w}_{t+1}^{(k)} = -\frac{\sqrt{t}}{\gamma}\max(1 - \frac{\lambda\sqrt{d_k}}{\|\bar{\mathbf{g}}_t^{(k)}\|_2}, 0)\bar{\mathbf{g}}_t^{(k)}$.

Proof of Lemma 2 Since the objective of the target function is componentwise, we can focus on the solution in a specific group such as the k-th group. Thus, we have $\boldsymbol{w}_{t+1}^{(k)} = \operatorname{argmin}_{\boldsymbol{w}^{(k)}\in\mathbb{R}^{d_k}}\{(\bar{\boldsymbol{g}}_t^{(k)})^T\boldsymbol{w}^{(k)} + \frac{\gamma}{2\sqrt{t}}\|\boldsymbol{w}^{(k)}\|_2^2 + \lambda\sqrt{d_k}\|\boldsymbol{w}^{(k)}\|_2\} = \operatorname{argmin}_{\boldsymbol{w}\in\mathbb{R}^{d_k}}\{(\bar{\boldsymbol{g}}_t^{(k)})^T\boldsymbol{w}$ $+\frac{\gamma}{2\sqrt{t}}\|\boldsymbol{w}\|_2^2 + \lambda\sqrt{d_k}\|\boldsymbol{w}\|_2\}$.

The optimal $\boldsymbol{w}_{t+1}^{(k)}$ should be $\boldsymbol{w}_{t+1}^{(k)} = \tau_k\bar{\boldsymbol{g}}_t^{(k)}$ with $\tau_k \le 0$. Otherwise, we can assume for the sake of contradiction that $\boldsymbol{w}_{t+1}^{(k)} = \tau_k\bar{\boldsymbol{g}}_t^{(k)} + \boldsymbol{\delta}^{(k)}$, where $\tau_k \in \mathbb{R}$ and $\boldsymbol{\delta}^{(k)} \in \mathbb{R}^{d_k}$ is the null space of $\bar{\boldsymbol{g}}_t^{(k)}$. It is easy to verify that $\boldsymbol{\delta}^{(k)} = \mathbf{0}$. Otherwise, the objection function is not minimized.

On the other hand, if $\tau_k > 0$, then $-\tau_k\bar{\boldsymbol{g}}_t^{(k)}$ will continue to decrease the objective function. Thus, $\tau_k \le 0$. Since we have $\boldsymbol{w}_{t+1}^{(k)} = \tau_k\bar{\boldsymbol{g}}_t^{(k)}$ with $\tau_k \le 0$, the objective function becomes $\boldsymbol{w}_{t+1}^{(k)} = \operatorname{argmin}_{\boldsymbol{w}\in\mathbb{R}^{d_k}}\{(\bar{\boldsymbol{g}}_t^{(k)})^T\boldsymbol{w} + \frac{\gamma}{2\sqrt{t}}\|\boldsymbol{w}\|_2^2 + \lambda\sqrt{d_k}\|\boldsymbol{w}\|_2\} = \operatorname{argmin}_{\tau_k\le 0}\{\tau_k\|\bar{\boldsymbol{g}}_t^{(k)}\|_2^2 + \frac{\gamma\tau_k^2}{2\sqrt{t}}\|\bar{\boldsymbol{g}}_t^{(k)}\|_2^2$ $-\tau_k\lambda\sqrt{d_k}\|\bar{\boldsymbol{g}}_t^{(k)}\|_2\}$.

By constructing the Lagrangian function, $\mathcal{L}(\tau_k,\epsilon) = \tau_k\|\bar{\boldsymbol{g}}_t^{(k)}\|_2^2 + \frac{\gamma\tau_k^2}{2\sqrt{t}}\|\bar{\boldsymbol{g}}_t^{(k)}\|_2^2 - \tau_k\lambda\sqrt{d_k}\|\bar{\boldsymbol{g}}_t^{(k)}\|_2 + \tau_k\epsilon$, we have $\epsilon \ge 0$.

The Karush-Kuhn-Tucker (KKT) condition indicates that the optimal solution must satisfy

$$\begin{cases} \frac{\partial\mathcal{L}(\tau_k,\epsilon)}{\partial\tau_k} = \|\bar{\boldsymbol{g}}_t^{(k)}\|_2^2 + \frac{\gamma\tau_k}{\sqrt{t}}\|\bar{\boldsymbol{g}}_t^{(k)}\|_2^2 - \lambda\sqrt{d_k}\|\bar{\boldsymbol{g}}_t^{(k)}\|_2 + \epsilon = 0 \\ \tau_k\epsilon = 0 \\ \epsilon \ge 0 \end{cases} \tag{A2}$$

That is,

$$\begin{cases} (\|\bar{\boldsymbol{g}}_t^{(k)}\|_2 - \lambda\sqrt{d_k})\|\bar{\boldsymbol{g}}_t^{(k)}\|_2 + \epsilon = -(\frac{\gamma}{\sqrt{t}}\|\bar{\boldsymbol{g}}_t^{(k)}\|_2^2)\tau_k \\ \tau_k\epsilon = 0 \\ \epsilon \ge 0 \end{cases} \tag{A3}$$

(1) If $\|\bar{\boldsymbol{g}}_t^{(k)}\|_2 - \lambda\sqrt{d_k} < 0$, then $\epsilon > 0$; hence, $\tau_k = 0$; (2) If $\|\bar{\boldsymbol{g}}_t^{(k)}\|_2 - \lambda\sqrt{d_k} \ge 0$, then $\epsilon \le 0$. Since $\epsilon \ge 0$, $\epsilon = 0$. In this case, we have $\tau_k = \frac{(\|\bar{\boldsymbol{g}}_t^{(k)}\|_2 - \lambda\sqrt{d_k})\|\bar{\boldsymbol{g}}_t^{(k)}\|_2}{-\frac{\gamma}{\sqrt{t}}\|\bar{\boldsymbol{g}}_t^{(k)}\|_2^2} = -\frac{\sqrt{t}}{\gamma}(1 - \frac{\lambda\sqrt{d_k}}{\|\bar{\boldsymbol{g}}_t^{(k)}\|_2})$.

In summary, we have

$$\tau_k = \begin{cases} 0 & \text{if } \|\bar{\boldsymbol{g}}_t^{(k)}\|_2 - \lambda\sqrt{d_k} < 0 \\ -\frac{\sqrt{t}}{\gamma}(1 - \frac{\lambda\sqrt{d_k}}{\|\bar{\boldsymbol{g}}_t^{(k)}\|_2}) & \text{if } \|\bar{\boldsymbol{g}}_t^{(k)}\|_2 - \lambda\sqrt{d_k} \ge 0 \end{cases} \tag{A4}$$

Therefore, $\tau_k = -\frac{\sqrt{t}}{\gamma}\max(1 - \frac{\lambda\sqrt{d_k}}{\|\bar{\boldsymbol{g}}_t^{(k)}\|_2}, 0)$ and $\boldsymbol{w}_{t+1}^{(k)} = -\frac{\sqrt{t}}{\gamma}\max(1 - \frac{\lambda\sqrt{d_k}}{\|\bar{\boldsymbol{g}}_t^{(k)}\|_2}, 0)\bar{\boldsymbol{g}}_t^{(k)}$. This proves Lemma 2.

Using these two lemmas, we provide the following proof details for solving CSGOL.

Since the objective of the target function of CSGOL is componentwise, we can focus on the solution in a specific group such as the k-th group. Thus, we have $\boldsymbol{w}_{t+1}^{(k)} = \operatorname{argmin}_{\boldsymbol{w}^{(k)}\in\mathbb{R}^{d_k}}\{(\bar{\boldsymbol{g}}_t^{(k)})^T\boldsymbol{w}^{(k)} + \frac{\gamma}{2\sqrt{t}}\|\boldsymbol{w}^{(k)}\|_2^2 + \lambda_1\|\boldsymbol{w}^{(k)}\|_2 + \lambda_2\|\boldsymbol{w}^{(k)}\|_1 + \lambda_3\|\boldsymbol{w}^{(k)}\|_1\}$ $= \operatorname{argmin}_{\boldsymbol{w}\in\mathbb{R}^{d_k}}\{(\bar{\boldsymbol{g}}_t^{(k)})^T\boldsymbol{w} + \frac{\gamma}{2\sqrt{t}}\|\boldsymbol{w}\|_2^2 + \lambda_1\|\boldsymbol{w}\|_2 + \lambda_2\|\boldsymbol{w}\|_1 + \lambda_3\|\boldsymbol{w}\|_1\}$.

Define the function $L(\boldsymbol{w}) = \frac{\gamma}{2\sqrt{t}}\|\boldsymbol{w}\|_2^2 + \lambda_1\|\boldsymbol{w}\|_2$, which is a nonnegative function. Since the objective function is elementwise, we can consider one entry i, that is $w_{t+1,i}^{(k)} = \operatorname{argmin}_{w\in\mathbb{R}}\{\bar{g}_{t,i}^{(k)}w + L(w^2) + (\lambda_2+\lambda_3) \mid w \mid\}$, where $L(w^2)$ is also a nonnegative function on w^2, and $L(w^2) = 0$ only if $w = 0$ for all i in the k-th group.

(1) If $\bar{g}_{t,i}^{(k)} = 0$, then $w_{t+1,i}^{(k)} = 0$;

(2) If $0 < \bar{g}_{t,i}^{(k)} \le (\lambda_2 + \lambda_3)$, then $w_{t+1,i}^{(k)} \le 0$. Thus, $[\bar{g}_{t,i}^{(k)} - (\lambda_2+\lambda_3)]w + L(w^2) \le \bar{g}_{t,i}^{(k)}w + L(w^2) + (\lambda_2+\lambda_3) \mid w \mid$ to minimize the value $w_{t+1,i}^{(k)} = 0$;

(3) If $(\lambda_2+\lambda_3) < \bar{g}_{t,i}^{(k)}$, then $w_{t+1,i}^{(k)} \le 0$. Thus, the objective function becomes $w_{t+1,i}^{(k)} = \operatorname{argmin}_{w\in\mathbb{R}}\{[\bar{g}_{t,i}^{(k)} - (\lambda_2+\lambda_3)]w + L(w^2)\} = \operatorname{argmin}_{w\in\mathbb{R}}\{[\bar{g}_{t,i}^{(k)} - (\lambda_2+\lambda_3)]w + \frac{\gamma}{2\sqrt{t}}\|w\|_2^2 + \lambda_1\|w\|_2\}$, which has the same form as DAGL. Thus, we have $w_{t+1,i}^{(k)} = -\frac{\sqrt{t}}{\gamma}\max(1 - \frac{\lambda_1}{\|\bar{g}_{t,i}^{(k)} - (\lambda_2+\lambda_3)\|_2}, 0)[\bar{g}_{t,i}^{(k)} - (\lambda_2+\lambda_3)]$;

(4) If $-(\lambda_2+\lambda_3) \le \bar{g}_{t,i}^{(k)} < 0$, then, $w_{t+1,i}^{(k)} \ge 0$. Thus, $[\bar{g}_{t,i}^{(k)} + (\lambda_2+\lambda_3)]w + L(w^2) \le \bar{g}_{t,i}^{(k)}w + L(w^2) + (\lambda_2+\lambda_3) \mid w \mid$, to minimize the value, $w_{t+1,i}^{(k)} = 0$;

(5) If $\bar{g}_{t,i}^{(k)} < -(\lambda_2+\lambda_3)$, then, $w_{t+1,i}^{(k)} \ge 0$. Thus, the objective function becomes $w_{t+1,i}^{(k)} = \operatorname{argmin}_{w\in\mathbb{R}}\{[\bar{g}_{t,i}^{(k)} + (\lambda_2+\lambda_3)]w + L(w^2)\} = \operatorname{argmin}_{w\in\mathbb{R}}\{[\bar{g}_{t,i}^{(k)} + (\lambda_2+\lambda_3)]w + \frac{\gamma}{2\sqrt{t}}\|w\|_2^2 + \lambda_1\|w\|_2\}$, which has the same form as DAGL. Thus, we have $w_{t+1,i}^{(k)} = -\frac{\sqrt{t}}{\gamma}\max(1 - \frac{\lambda_1}{\|\bar{g}_{t,i}^{(k)} + (\lambda_2+\lambda_3)\|_2}, 0)[\bar{g}_{t,i}^{(k)} + (\lambda_2+\lambda_3)]$.

In summary, we have

$$w_{t+1,i}^{(k)} = \begin{cases} 0 & \text{if } \mid \bar{g}_{t,i}^{(k)} \mid \le (\lambda_2+\lambda_3) \\ -\frac{\sqrt{t}}{\gamma}\max(1 - \frac{\lambda_1}{\| \mid \bar{g}_{t,i}^{(k)} \mid - (\lambda_2+\lambda_3)\|_2}, 0)\operatorname{sign}(\bar{g}_{t,i}^{(k)}) \cdot [\mid \bar{g}_{t,i}^{(k)} \mid - (\lambda_2+\lambda_3)] & \text{if } \mid \bar{g}_{t,i}^{(k)} \mid > (\lambda_2+\lambda_3) \end{cases} \tag{A5}$$

That is

$$\boldsymbol{w}_{t+1}^{(k)} = \begin{cases} \mathbf{0} & \text{if } \mid \bar{\boldsymbol{g}}_t^{(k)} \mid \le (\lambda_2+\lambda_3) \\ -\frac{\sqrt{t}}{\gamma}\max(1 - \frac{\lambda_1}{\| \mid \bar{\boldsymbol{g}}_t^{(k)} \mid - (\lambda_2+\lambda_3)\mathbf{1}\|_2}, 0)\operatorname{sign}(\bar{\boldsymbol{g}}_t^{(k)}) \odot [\mid \bar{\boldsymbol{g}}_t^{(k)} \mid - (\lambda_2+\lambda_3)\mathbf{1}] & \text{if } \mid \bar{\boldsymbol{g}}_t^{(k)} \mid > (\lambda_2+\lambda_3) \end{cases} \tag{A6}$$

Hence, $\boldsymbol{w}_{t+1}^{(k)} = -\frac{\sqrt{t}}{\gamma}\max(1 - \frac{\lambda_1}{\| \mid \bar{\boldsymbol{g}}_t^{(k)} \mid - (\lambda_2+\lambda_3)\mathbf{1}\|_2}, 0)\operatorname{sign}(\bar{\boldsymbol{g}}_t^{(k)}) \odot \max(\mid \bar{\boldsymbol{g}}_t^{(k)} \mid - (\lambda_2+\lambda_3)\mathbf{1}, \mathbf{0})$ $= -\frac{\sqrt{t}}{\gamma}\max(1 - \frac{\lambda_1}{\| \mid \bar{\boldsymbol{g}}_t^{(k)} \mid - (\lambda_2+\lambda_3)\mathbf{1}\|_2}, 0)\boldsymbol{p}_t^{(k)} = -\frac{\sqrt{t}}{\gamma}\max(1 - \frac{\lambda_1}{\|\boldsymbol{p}_t^{(k)}\|_2}, 0)\boldsymbol{p}_t^{(k)}$, where $\boldsymbol{p}_t^{(k)} = \operatorname{sign}(\bar{\boldsymbol{g}}_t^{(k)}) \odot \max(\mid \bar{\boldsymbol{g}}_t^{(k)} \mid - (\lambda_2+\lambda_3)\mathbf{1}, \mathbf{0})$. This concludes the proof. □

Appendix B Proof of closed-form solution of PCSGOL

Proof The objective optimization function of PCSGOL is equivalent to

$$\boldsymbol{w}_{t+1} = \text{argmin}_{\boldsymbol{w}\in\mathbb{R}^d}\{\bar{\boldsymbol{g}}_t^T\boldsymbol{w} + \tfrac{1}{2t}\textstyle\sum_{s=1}^t \sigma\|\boldsymbol{w}-\boldsymbol{w}_s\|_2^2 + \sum_{k=1}^K(\lambda_1\|\boldsymbol{w}^{(k)}\|_2 + \lambda_2\|\boldsymbol{w}^{(k)}\|_1) + \lambda_3\|\boldsymbol{w}\|_1\}$$

$$= \text{argmin}_{\boldsymbol{w}\in\mathbb{R}^d}\{\bar{\boldsymbol{g}}_t^T\boldsymbol{w} + \tfrac{\sigma}{2t}\textstyle\sum_{s=1}^t(\|\boldsymbol{w}\|_2^2 - 2\boldsymbol{w}_s^T\boldsymbol{w} + \|\boldsymbol{w}_s\|_2^2) + \sum_{k=1}^K(\lambda_1\|\boldsymbol{w}^{(k)}\|_2 + \lambda_2\|\boldsymbol{w}^{(k)}\|_1) + \lambda_3\|\boldsymbol{w}\|_1\}$$

$$= \text{argmin}_{\boldsymbol{w}\in\mathbb{R}^d}\{(\bar{\boldsymbol{g}}_t - \tfrac{\sigma}{t}\textstyle\sum_{s=1}^t \boldsymbol{w}_s)^T\boldsymbol{w} + \tfrac{\sigma}{2}\|\boldsymbol{w}\|_2^2 + \sum_{k=1}^K(\lambda_1\|\boldsymbol{w}^{(k)}\|_2 + \lambda_2\|\boldsymbol{w}^{(k)}\|_1) + \lambda_3\|\boldsymbol{w}\|_1\}$$

By defining $\bar{\boldsymbol{w}}_t = \frac{1}{t}\sum_{s=1}^t \boldsymbol{w}_s$, we have

$\boldsymbol{w}_{t+1} = \text{argmin}_{\boldsymbol{w}\in\mathbb{R}^d}\{(\bar{\boldsymbol{g}}_t - \sigma\bar{\boldsymbol{w}}_t)^T\boldsymbol{w} + \frac{\sigma}{2}\|\boldsymbol{w}\|_2^2 + \sum_{k=1}^K(\lambda_1\|\boldsymbol{w}^{(k)}\|_2 + \lambda_2\|\boldsymbol{w}^{(k)}\|_1) + \lambda_3\|\boldsymbol{w}\|_1\}$. Since the objective of the target function is componentwise, we can focus on the solution in a specific group such as the k-th group. Thus, we have

$$\boldsymbol{w}_{t+1}^{(k)} = \text{argmin}_{\boldsymbol{w}^{(k)}\in\mathbb{R}^{d_k}}\{(\bar{\boldsymbol{g}}_t^{(k)} - \sigma\bar{\boldsymbol{w}}_t^{(k)})^T\boldsymbol{w}^{(k)} + \tfrac{\sigma}{2}\|\boldsymbol{w}^{(k)}\|_2^2 + \lambda_1\|\boldsymbol{w}^{(k)}\|_2 + \lambda_2\|\boldsymbol{w}^{(k)}\|_1 + \lambda_3\|\boldsymbol{w}^{(k)}\|_1\}$$

$$= \text{argmin}_{\boldsymbol{w}\in\mathbb{R}^{d_k}}\{(\bar{\boldsymbol{g}}_t^{(k)} - \sigma\bar{\boldsymbol{w}}_t^{(k)})^T\boldsymbol{w} + \tfrac{\sigma}{2}\|\boldsymbol{w}\|_2^2 + \lambda_1\|\boldsymbol{w}\|_2 + (\lambda_2+\lambda_3)\|\boldsymbol{w}\|_1\}$$

$= \text{argmin}_{\boldsymbol{w}\in\mathbb{R}^{d_k}}\{(\frac{\bar{\boldsymbol{g}}_t^{(k)}}{\sigma} - \bar{\boldsymbol{w}}_t^{(k)})^T\boldsymbol{w} + \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \frac{\lambda_1}{\sigma}\|\boldsymbol{w}\|_2 + \frac{\lambda_2+\lambda_3}{\sigma}\|\boldsymbol{w}\|_1\}$, which is similar to CSGOL; thus, we have the following closed-form solution:

$$\boldsymbol{w}_{t+1}^{(k)} = -\max\left(1 - \frac{\frac{\lambda_1}{\sigma}}{\|\,|\frac{\bar{\boldsymbol{g}}_t^{(k)}}{\sigma} - \bar{\boldsymbol{w}}_t^{(k)}| - \frac{\lambda_2+\lambda_3}{\sigma}\mathbf{1}\|_2}, 0\right)\text{sign}\left(\frac{\bar{\boldsymbol{g}}_t^{(k)}}{\sigma} - \bar{\boldsymbol{w}}_t^{(k)}\right)$$

$$\odot \max\left(\left|\frac{\bar{\boldsymbol{g}}_t^{(k)}}{\sigma} - \bar{\boldsymbol{w}}_t^{(k)}\right| - \frac{\lambda_2+\lambda_3}{\sigma}\mathbf{1}, \mathbf{0}\right)$$

$$= -\frac{1}{\sigma}\max\left(1 - \frac{\lambda_1}{\|\,|\bar{\boldsymbol{g}}_t^{(k)} - \sigma\bar{\boldsymbol{w}}_t^{(k)}| - (\lambda_2+\lambda_3)\mathbf{1}\|_2}, 0\right)\text{sign}(\bar{\boldsymbol{g}}_t^{(k)} - \sigma\bar{\boldsymbol{w}}_t^{(k)})$$

$\odot \max(|\bar{\boldsymbol{g}}_t^{(k)} - \sigma\bar{\boldsymbol{w}}_t^{(k)}| - (\lambda_2+\lambda_3)\mathbf{1}, \mathbf{0}) = -\frac{1}{\sigma}\max(1 - \frac{\lambda_1}{\|\boldsymbol{q}_t^{(k)}\|_2}, 0)\boldsymbol{q}_t^{(k)}$, where $\boldsymbol{q}_t^{(k)} = \text{sign}(\bar{\boldsymbol{g}}_t^{(k)} - \sigma\bar{\boldsymbol{w}}_t^{(k)}) \odot \max(|\bar{\boldsymbol{g}}_t^{(k)} - \sigma\bar{\boldsymbol{w}}_t^{(k)}| - (\lambda_2+\lambda_3)\mathbf{1}, \mathbf{0})$. This concludes the proof. □

Acknowledgements This publication was made possible by funding from the NIH grant 2U54MD007595 and the DOD ARO Grant #W911NF-20-1-0249. The contents are solely the responsibility of the authors and do not necessarily represent the official views of the NIH and the DOD.

Author Contributions ZC performed the experiments and wrote the manuscript. VS and KZ contributed to the algorithmic analysis, provided feedback, and edited the manuscript. VS and AE provided feedback on the results presentation and the manuscript. ZC and KZ contributed to the literature review. All authors read and approved the final manuscript.

Data Availability The data used to conduct the computational experiments are available in public repositories. The specific sources are outlined in the Sects. 5.1, 5.7, and 5.8.

Code availability The code for this work will be available upon request.

Declarations

Conflict of interest The authors declare that they have no conflicts of interest.

Ethical approval Not Applicable.

Consent to participate Not Applicable.

Consent for publication Not Applicable.

References

Barron, J. T. (2019). A general and adaptive Rubost loss function. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* IEEE, (pp. 4331–4339).

Bernardo, A., & Della Valle, E. (2021). Vfc-smote: very fast continuous synthetic minority oversampling for evolving data streams. *Data Mining and Knowledge Discovery, 35*(6), 2679–2713. https://doi.org/10.1007/s10618-021-00786-0

Brzezinski, D., Minku, L. L., Pewinski, T., et al. (2021). The impact of data difficulty factors on classification of imbalanced and concept drifting data streams. *Knowledge and Information Systems, 63*(6), 1429–1469. https://doi.org/10.1007/s10115-021-01560-w

Cano, A., & Krawczyk, B. (2020). Kappa updated ensemble for drifting data stream mining. *Machine Learning, 109*(10), 175–218. https://doi.org/10.1007/s10994-019-05840-z

Cano, A., & Krawczyk, B. (2022). Rose: Robust online self-adjusting ensemble for continual learning on imbalanced drifting data streams. *Machine Learning, 111*(7), 2561–2599. https://doi.org/10.1007/s10994-022-06168-x

Chen, Z., Fang, Z., Fan, W., et al. (2017). Cstg: An effective framework for cost-sensitive sparse online learning. In: *Proceedings of the 2017 SIAM International Conference on Data Mining, Society for Industrial and Applied Mathematics,* (pp. 759–767).

Chen, Z., Fang, Z., Sheng, V., et al. (2021). Csrda: Cost-sensitive regularized dual averaging for handling imbalanced and high-dimensional streaming data. In: *The 12th IEEE International Conference on Big Knowledge, IEEE,* (pp. 164–173).

Crammer, K., Dekel, O., Keshet, J., et al. (2006). Online passive-aggressive algorithms. *Journal of Machine Learning Research, 7*(1), 551–585.

Duchi, J., & Singer, Y. (2009). Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research, 10*(1), 2899–2934.

Elkan, C. (2001). The foundations of cost-sensitive learning. In: *The 17th International Joint Conference on Artificial Intelligence, American Association for Artificial Intelligence*, (pp. 973–978).

Fonollosa, J., Sheik, S., Huerta, R., et al. (2015). Reservoir computing compensates slow response of chemosensor arrays exposed to fast varying gas concentrations in continuous monitoring. *Sensors and Actuators, B: Chemical, 215*(1), 618–629. https://doi.org/10.1016/j.snb.2015.03.028

Ho, S. S., & Wechsler, H. (2010). A martingale framework for detecting changes in data streams by testing exchangeability. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 32*(12), 2113–2127. https://doi.org/10.1109/TPAMI.2010.48

Hoi, S. C., Sahoo, D., Lu, J., et al. (2021). Online learning: A comprehensive survey. *Neurocomputing, 459*, 249–289. https://doi.org/10.1016/j.neucom.2021.04.112

Hu, Y., Li, C., Meng, K., et al. (2017). Group sparse optimization via lp, q regularization. *Journal of Machine Learning Research, 18*(1), 960–1011.

Hurley, N., & Rickard, S. (2009). Comparing measures of sparsity. *IEEE Transactions on Information Theory, 55*(10), 4723–4741. https://doi.org/10.1109/TIT.2009.2027527

Kubat, M., & Matwin, S. (1997). Addressing the curse of imbalanced training sets: one-sided selection. In: *Proceedings of the 14th International Conference on International Conference on Machine Learning*, Morgan Kaufmann Publishers Inc., San Francisco, (p. 179).

Langford, J., Li, L., & Zhang, T. (2009). Sparse online learning via truncated gradient. *Journal of Machine Learning Research, 10*(3), 777–801.

Lee, S., & Wright, S. J. (2012). Manifold identification in dual averaging for regularized stochastic online learning. *Journal of Machine Learning Research, 13*(6), 1665–1705.

Leevy, J. L., Khoshgoftaar, T. M., Bauder, R. A., et al. (2018). A survey on addressing high-class imbalance in big data. *Journal of Big Data, 5*(1), 1–30. https://doi.org/10.1186/s40537-018-0151-6

Li, Y., Zaragoza, H., Herbrich, R., et al. (2002). The perceptron algorithm with uneven margins. In: *Proceedings of the 19th International Conference on International Conference on Machine Learning*, International Machine Learning Society, (pp. 379–386).

Liu, W., Zhang, H., Ding, Z., et al. (2021). A comprehensive active learning method for multiclass imbalanced data streams with concept drift. *Knowledge and Information Systems, 215*(3), 106778. https://doi.org/10.1016/j.knosys.2021.106778

Ma, Y., & Zheng, T. (2017). Stabilized sparse online learning for sparse data. *Journal of Machine Learning Research, 18*(1), 4773–4808.

Mirza, B., Lin, Z., & Liu, N. (2015). Ensemble of subset online sequential extreme learning machine for class imbalance and concept drift. *Neurocomputing, 149*(2), 316–329. https://doi.org/10.1016/j.neucom.2014.03.075

Ni, X., Yu, Y., Wu, P., et al. (2019). Feature selection for facebook feed ranking system via a group-sparsity-regularized training algorithm. In: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, Association for Computing Machinery, (pp. 2085–2088).

Ohsaki, M., Wang, P., Matsuda, K., et al. (2017). Confusion-matrix-based kernel logistic regression for imbalanced data classification. *IEEE Transactions on Knowledge and Data Engineering, 29*(9), 1806–1819. https://doi.org/10.1109/TKDE.2017.2682249

Simon, N., Friedman, J., Hastie, T., et al. (2013). A sparse-group lasso. *Journal of Computational and Graphical Statistics, 22*(2), 231–245. https://doi.org/10.1080/10618600.2012.681250

Ushio, A., & Yukawa, M. (2019). Projection-based regularized dual averaging for stochastic optimization. *IEEE Transactions on Signal Processing, 67*(10), 2720–2733. https://doi.org/10.1109/TSP.2019.2908901

Wang, C., Lai, J., Huang, D., et al. (2011). Vstream: A support vector-based algorithm for clustering data streams. *IEEE Transactions on Knowledge and Data Engineering, 25*(6), 1410–1424. https://doi.org/10.1109/TKDE.2011.263

Wang, J., Zhao, P., & Hoi, S. C. (2013). Cost-sensitive online classification. *IEEE Transactions on Knowledge and Data Engineering, 26*(10), 2425–2438. https://doi.org/10.1109/TKDE.2013.157

Wang, J., Wang, M., Li, P., et al. (2015). Online feature selection with group structure analysis. *IEEE Transactions on Knowledge and Data Engineering, 27*(11), 3029–3041. https://doi.org/10.1109/TKDE.2015.2441716

Wang, S., Minku, L. L., & Yao, X. (2016). Dealing with multiple classes in online class imbalance learning. In: *Proceedings of 25th International Joint Conference on Artificial Intelligence*, (pp. 2118–2124).

Wong, T. T. (2020). Linear approximation of f-measure for the performance evaluation of classification algorithms on imbalanced data sets. *IEEE Transactions on Knowledge and Data Engineering, 34*(2), 753–763. https://doi.org/10.1109/TKDE.2020.2986749

Wu, F., Jing, X. Y., Shan, S., et al. (2017). Multiset feature learning for highly imbalanced data classification. In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, Association for the Advancement of Artificial Intelligence*, (pp. 139–156).

Xiao, L. (2010). Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research, 11*(88), 2543–2596.

Xie, Y., Qiu, M., Zhang, H., et al. (2022). Gaussian distribution based oversampling for imbalanced data classification. *IEEE Transactions on Knowledge and Data Engineering, 34*(2), 667–679. https://doi.org/10.1109/TKDE.2020.2985965

Yang, H., Xu, Z., King, I., et al. (2010). Online learning for group lasso. In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*, International Machine Learning Society, Haifa, Israel, (pp. 1191–1198).

Yin, J., Gan, C., Zhao, K., et al. (2020). A novel model for imbalanced data classification. In: *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, Association for the Advancement of Artificial Intelligence*, (pp. 6680–6687).

Yu, K., Wu, X., Ding, W., et al. (2016). Scalable and accurate online feature selection for big data. *ACM Transactions on Knowledge Discovery from Data, 11*(2), 1–39. https://doi.org/10.1145/2976744

Zhang, Q., Zhang, P., Long, G., et al. (2016). Online learning from trapezoidal data streams. *IEEE Transactions on Knowledge and Data Engineering, 28*(10), 2709–2723. https://doi.org/10.1109/TKDE.2016.2563424

Zhao, P., & Hoi, S. C. (2013). Cost-sensitive double updating online learning and its application to online anomaly detection. In: *Proceedings of the 2013 SIAM International Conference on Data Mining* Society for Industrial and Applied Mathematics, Austin, (pp. 207–215).

Zhao, P., Zhuang, F., Wu, M., et al. (2015). Cost-sensitive online classification with adaptive regularization and its applications. In: *The 2015 IEEE International Conference on Data Mining*, IEEE, Atlantic, (pp. 649–658).

Zhao, P., Zhang, Y., Wu, M., et al. (2018). Adaptive cost-sensitive online classification. *IEEE Transactions on Knowledge and Data Engineering, 31*(2), 214–228. https://doi.org/10.1109/TKDE.2018.2826011

Zhou, B., Chen, F., & Ying, Y. (2019). Dual averaging method for online graph-structured sparsity. In: *The 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, Anchorage, (pp. 436–446).

Zhou, P., Wang, N., & Zhao, S. (2021). Online group streaming feature selection considering feature interaction. *Knowledge-Based Systems, 226*(17), 107–157. https://doi.org/10.1016/j.knosys.2021.107157

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Zhong Chen[1] · Victor Sheng[2] · Andrea Edwards[3] · Kun Zhang[3]

✉ Kun Zhang
kzhang@xula.edu

Zhong Chen
zhong.chen@cs.siu.edu

Victor Sheng
victor.sheng@ttu.edu

Andrea Edwards
aedwards@xula.edu

[1] School of Computing, Southern Illinois University, 1263 Lincoln Dr, Carbondale, IL 62901, USA

[2] Department of Computer Science, Texas Tech University, 2500 Broadway, Lubbock, TX 79409, USA

[3] Department of Computer Science, Xavier University of Louisiana, 1 Drexel Dr, New Orleans, LA 70125, USA