Robust Sparse Online Learning for Data Streams with Streaming Features

Zhong Chen* Yi He† Di Wu‡ Huixin Zhan§ Victor Sheng¶ Kun Zhang‖

Abstract

Sparse online learning has received extensive attention during the past few years. Most of existing algorithms that utilize ℓ_1-norm regularization or ℓ_1-ball projection assume that the feature space is fixed or changes by following explicit constraints. However, this assumption does not always hold in many real applications. Motivated by this observation, we propose a new online learning algorithm tailored for data streams described by open feature spaces, where new features can be occurred, and old features may be vanished over various time spans. Our algorithm named RSOL provides a strategy to adapt quickly to such feature dynamics by encouraging sparse model representation with an ℓ_1- and ℓ_2-mixed regularizer. We leverage the proximal operator of the $\ell_{1,2}$-mixed norm and show that our RSOL algorithm enjoys a closed-form solution at each iteration. A sub-linear regret bound of our proposed algorithm is guaranteed with a solid theoretical analysis. Empirical results benchmarked on nine streaming datasets validate the effectiveness of the proposed RSOL method over three state-of-the-art algorithms.
Keywords: online learning, sparse learning, streaming feature selection, open feature spaces, $\ell_{1,2}$ mixed norm

1 Introduction

Data streams can nowadays be generated from real applications in high velocity, thanks to advances in and ubiquity of sensing techniques [4–10]. These data streams provide a real-time description of our communities, cities, and natural and societal environments that constantly evolve. Online Learning (OL) that enables to train decision-making models on-the-fly in accordance with the evolving patterns of data thus leads to many powerful algorithms for streaming data analytics [6, 8–10, 37]. The initial focus of OL algorithms was to deal with an incremental sample space, where the instances of training data emerge one after the other and are processed in a single pass. All data instances are posited to reside in a *fixed* feature space. However, this assumption can be too restrictive in practice. To wit, consider an urban disaster monitoring system aided by OL, where streaming data are sent from crowd-sensing devices such as smart phones and sensor kits/sites that scatter across a geographically wide region in real time [2]. Fixing the set of features to be used in a prior is next to impossible for two reasons. First, new users join the sensing effort would commit data collected by their own devices (e.g., a new-brand cellphone), thus introducing new sensory features. Second, as users may stop sending data for reasons like battery exhaustion or network malfunction, any pre-existing features can become unobserved in later time snapshots. Another example is spam filtering, where new features (e.g., new words, hashtags, and URLs) relevant to classification tend to appear over time [32]. In extreme cases, the original set of features can become totally irrelevant for subsequent use in the learning task. We coin such data inputs as *streaming data in open feature spaces* (SDOFS).

To enable learning in an SDOFS setting, recent studies have proposed new OL models [1, 11, 16, 17, 19, 22, 28, 39, 44], with their shared idea being to establish a mapping relationship between old and new features. As such, for any new feature just emerging, its learning coefficient can be better initialized to expedite convergence; for old features turned unobservable, their learned coefficients can still be leveraged via reconstruction, thereby saving learning cost.

However, the previous SDOFS studies all suffer one limitation – to maximize their learning performance, all emerged features must be remembered in their models. This is impractical in data applications where new features are generated in wild velocities. For example, social opinion mining, where hundreds of millions of new features on social media, such as hashtags, buzzwords, or topics, are emerging, trending, and then vanish within short time spans [27, 29]. Keeping all features would result in an OL model with prohibitively huge size and dimension, eating up memory resources, slowing down computing speed, and usually ending up with inferior classification performance. Moreover, the possibly very large feature space, even of unknown or infinite sizes, can make the resultant OL model non-interpretable, where the features being relevant to learn-

*Southern Illinois University Carbondale. Email: zhong.chen@cs.siu.edu
†Old Dominion University. Email: yihe@cs.odu.edu
‡Southwest University. Email: wudi.cigit@gmail.com
§Cedars-Sinai Medical Center. Email: huixin.zhan@cshs.org
¶Texas Tech University. Email: victor.sheng@ttu.edu
‖Xavier University of Louisiana. Email: kzhang@xula.edu

ing tasks are overwhelmed by other irrelevant features.

Specifically, we propose a novel OL approach that encourages a sparse model solution in an SDOFS setup, termed Robust Sparse Online Learning in open feature spaces with $\ell_{1,2}$-Norm Constraint (RSOL, for short). Our key idea is two-fold. First, to tame the feature space dynamics, we tailor an online passive-aggressive (PA) program based on the margin-maximum principle. The proposed PA program reweighs the learning weights at each iteration only if the increment or decrement of feature space would incur prediction loss. Second, to encourage model sparsity, we impose an $\ell_{1,2}$-norm constraint on the PA program and solve it with a closed-form solution. We leverage an incremental matrix with memory of the learned feature weights and apply the sparsity-preserved operator on it to stabilize the resultant sparse solution. In the matrix, the learning weights of a feature should be set to zero consistently over the memory length, if the feature is irrelevant. We show that our RSOL enjoys a closed form solution, which lends itself amenable for implementation. Theoretical and empirical studies are carried out to substantiate the effectiveness of our proposed methods.

2 Related Work

We relate our proposed RSOL approach to two research threads: *online learning in open feature spaces*, which performs OL in the same data environment as we do, and *sparse online learning*, which aims to reduce the OL model dimension but mostly in fixed feature spaces, while ours are open.

Note, we are aware of another thread of studies termed *online streaming feature selection* (OSFS) [26, 33, 35, 36], which are seemingly similar to our study but essentially different in two aspects. First, OSFS allows incremental feature input but posits a *fixed instance space*, i.e., all instances are given in advance before feature selection starts. In our setting, however, the data instances are presented one after the other like normal streams. Second, OSFS decouples feature selection and learning, i.e., it selects a set of highly relevant features at first and then trains and evaluates a predictive model on it in an offline fashion. Our setting is more challenging as we need to perform learning and feature selection jointly in an online fashion.

2.1 Online Learning in Open Feature Spaces

In OL, a given learning algorithm tries to infer a prediction model from sequentially appearing instances. OL algorithms can be distinguished into first-order and second-order, where first-order algorithms use first-order information for the update, e.g., the Perceptron algorithm [14] or Online Gradient Descent [42]. Second-order algorithms aim to make use of the underlying structure between features [7, 38].

However, traditional OL methods cannot learn from open feature spaces since they assume the feature space remains constant. To alleviate this restriction, pioneer studies considered an increasing feature space [15, 32, 39], with the crux lying in initializing the learning weights of new features with an educated guess, such that the online learner can enjoy a jump-start with faster convergence over random initialization on new features. Later studies further relaxed this setting into an arbitrarily open feature space [1,16,18–22,40], where new features can emerge and any pre-existing features may stop to be observed at any time. Linear ensemble models became a core technique of these approaches. Its idea is to establish feature relationship, such that the learner can reconstruct the old features to leverage their learned weights for better prediction performance in cases where they become unobserved.

Unfortunately, such ensembled linear models would soon grow to unmanageably large size as new features keep emerging. To reduce the dimension, recent studies [17, 28] suggested to extract complex feature interplays in latent space. Alas, this latent model representation would inevitably sacrifice model interpretability, which is critical in various domains such as finance, medicine, and security. Our proposed RSOL approach excels in the sense that our sparse model solutions are directly associated with the original features. The leveraged $\ell_{1,2}$-norm would consistently encourage the model weights converging to a limited number of entries while setting others to zero. The higher its weight, the more relevant this feature is to the prediction. The model interpretability is thus preserved in an online process.

2.2 Sparse Online Learning

The goal of sparse online learning is to induce sparsity in the weights of online learning algorithms [3, 43], ensuring the prediction model only contains a limited size of active features. The following algorithms, thus, have the potential to achieve better performance and interpretability as well. The 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. [24] propose a simple yet efficient modification of the standard stochastic gradient via truncated gradient (TG) to achieve sparsity in online learning. Duchi and Singer [12] further propose a forward-backward splitting (FOBOS) algorithm to solve the sparse online learning problems. 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 the end, Ma and Zhang [30] introduce a stabilized truncated stochastic gradient descent (STSGD) algorithm. Chen et al. [3] extend TG to cost-sensitive OL via truncated gradient (CSTG) and make it cost-sensitive and scalable for imbalanced and high-dimensional data streams.

The latter group focuses on the dual averaging methods that can explicitly exploit the regularization structure. One representative method is the regularized dual averaging (RDA) proposed in [34], which learns the variables by solving a regularized optimization problem that involves the average of all past subgradients of the loss functions. Lee and Wright [25] further extend the RDA algorithm to RDA+ by using a more aggressive truncation threshold. Tang et al. [45] propose a dual perspective of online learning algorithm, which concerns using a window method to achieve sparsity and robustness. The Fenchel conjugates and gradient ascent are used to perform online learning optimization process. Ushio and Yukawa [31] propose the projection based regularized dual averaging (PDA) method to exploit a sparsity-promoting metric and a sparsity-promoting regularizer simultaneously. Zhou et al. [41] propose an online algorithm GraphDA for graph-structured sparsity constraint problems using the dual averaging method. Chen et al. [7] develop a second-order projection dual averaging based online learning (SPDA) method to effectively handle high-throughput streaming data. By fully exploiting the regularized dual averaging optimization, the second-order information, and an optimal projection operator, SPDA converges fast with fairly optimal solutions.

Unfortunately, none of these methods can be generalized to an open features. Specifically, for a new feature, its weight either is initialized as zero, which can be interpreted as irrelevant, or is randomly initialized, which would require a sufficiently large number of instances to converge. Likewise, for an old feature becoming unobserved, no gradient information is available on its entry, thus its weight is not updated. Both cases lead to statistical bias. Our RSOL approach outperforms the prior studies by leveraging a passive-aggressive (PA) learner that 1) apportions the weights from other existing features to a new feature for its better initialization and 2) redistributes the weight of an unobserved old feature to other features. Closed-form solutions[1] are available for both cases, which lends our RSOL an advantage of fast convergence and be integrative to the tailored sparsity constraints.

[1]All detailed proofs are provided at the link: https://www.dropbox.com/scl/fi/v5o5hrlw0wuwzamm4fhn6/SDM24_Appendix.pdf?rlkey=797oeccx0ivzlvnwuekb1tn6u&dl=0

3 Proposed Method

In this section, we first formulate the learning problem (in Section 3.1) and then present our proposed RSOL algorithm. In particular, in Section 3.2, we elaborate the PA learner that deals with the feature space increment and decrement; in Section 3.3, we introduce the $\ell_{1,2}$-norm regularization and how it encourages model sparsity.

3.1 Problem Statement We start with a typical SDOFS modeling. Write an input sequence $\{(\boldsymbol{x}_t, y_t) \mid t \in [T]\}$. Each data instance $\boldsymbol{x}_t \in \mathbb{R}^{d_t}$ received at the t-th round is a vector of d_t-dimension, associated with a true class label $y_t \in \{-1, +1\}$. We hereby follow prior art [16, 20] to restrict our interest in a binary classification problem, as multi-class setups can be trivially reduced to binary cases with One-vs-One or One-vs-Rest strategies [1].

At each round, the learner observes $\boldsymbol{x}_t$ and returns a prediction $\hat{y}_t = \text{sign}(\boldsymbol{w}_t^\top \boldsymbol{x}_t)$. The true label y_t is then revealed, and the learner suffers a risk if the prediction was incorrect, e.g., gauged by hinge loss $\ell_t(\boldsymbol{w}_t, (\boldsymbol{x}_t, y_t)) = \max(0, 1 - y_t(\boldsymbol{w}_t^\top \boldsymbol{x}_t))$. The learner then is updated to $\boldsymbol{w}_{t+1}$ based on the loss information and gets ready to the next round. Our goal is to find an updating strategy $\mathcal{A}$ that minimizes empirical risk and, more importantly, yields a sparse model solution over T rounds, namely:

$$\min_{\boldsymbol{w}_t \in \mathbb{R}^{d_t}} \mathbb{E}_{t\in[T]}\left[\ell_t(\boldsymbol{w}_t, (\boldsymbol{x}_t, y_t))\right] + \|\boldsymbol{w}_t\|_0, \tag{3.1}$$

where the ℓ_0-norm counts the number of nonzero entries in weight vector $\boldsymbol{w}_t$.

The main challenge is imposed by the fact that the feature space is open and can be either decremental ($d_{t+1} \leq d_t$) or incremental ($d_{t+1} \geq d_t$), due to newly emerging features or unobserved old features, respectively. The survived features are represented by $\boldsymbol{x}_{t+1}^s = \boldsymbol{x}_t \cap \boldsymbol{x}_{t+1}$ or $\boldsymbol{w}_{t+1}^s = \boldsymbol{w}_t \cap \boldsymbol{w}_{t+1}$, the vanished features are represented by $\boldsymbol{x}_{t+1}^v = \boldsymbol{x}_t \setminus \boldsymbol{x}_{t+1}$ or $\boldsymbol{w}_{t+1}^v = \boldsymbol{w}_t \setminus \boldsymbol{w}_{t+1}$, and the new features are represented by $\boldsymbol{x}_{t+1}^n = \boldsymbol{x}_{t+1} \setminus \boldsymbol{x}_t$ or $\boldsymbol{w}_{t+1}^n = \boldsymbol{w}_{t+1} \setminus \boldsymbol{w}_t$.

3.2 Online Passive-Aggressive Feature Reweighing If the feature dimension is decreased from the t-th round to the $(t+1)$-th round (i.e., $d_t \geq d_{t+1}$), then we decompose the instance $\boldsymbol{x}_t = [\boldsymbol{x}_t^s; \boldsymbol{x}_t^d]$ and the corresponding weight vector $\boldsymbol{w}_t = [\boldsymbol{w}_t^s; \boldsymbol{w}_t^d]$, where $\boldsymbol{x}_t^s \in \mathbb{R}^{d_{t+1}}$ is the vector with survival features and $\boldsymbol{x}_t^d \in \mathbb{R}^{d_t - d_{t+1}}$ is the vector with vanished features. That is, $\boldsymbol{x}_t^s$ and $\boldsymbol{w}_t^s$ have the same dimension as $\boldsymbol{x}_{t+1}$ and $\boldsymbol{w}_{t+1}$. Moreover, to make the model robust to the noise, we use the soft-margin technique by introducing

a slack variable ξ into the optimization problem. In this case, we extend the passive-aggressive (PA) algorithm to update $\boldsymbol{w}_{t+1}$ by solving the following optimization task:

$$\boldsymbol{w}_{t+1} = \underset{\boldsymbol{w}\in\mathbb{R}^{d_{t+1}},\ell_{t+1}\leq\xi,\xi\in\mathbb{R}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{w}-\boldsymbol{w}_t^s\|_2^2 + \mu\xi^2 \tag{3.2}$$

where $\mu > 0$ is a penalty parameter that can tradeoff the rigidness and slackness of the online model. A larger value of μ implies a more rigid update step, and $\ell_{t+1} = \ell_{t+1}(\boldsymbol{w}, (\boldsymbol{x}_{t+1}, y_{t+1})) = \max(0, 1 - y_{t+1}(\boldsymbol{w}^T\boldsymbol{x}_{t+1}))$ is the loss at round $t+1$. Then, we derive the closed-form solution for the above equation in Theorem 3.1.

THEOREM 3.1. (**Closed-form Solution of Eq. (3.2)**) *The closed-form solution for minimizing Eq. (3.2) is* $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t^s + \gamma_t y_{t+1}\boldsymbol{x}_{t+1}$, *where* $\gamma_t = \frac{\ell_{t+1}(\boldsymbol{w}_t^s, (\boldsymbol{x}_{t+1}, y_{t+1}))}{\|\boldsymbol{x}_{t+1}\|_2^2 + \frac{1}{2\mu}}$.

If the feature dimension is increased from the t-th round to the $(t+1)$-th round (i.e., $d_t \leq d_{t+1}$), then we decompose the instance $\boldsymbol{x}_{t+1} = [\boldsymbol{x}_{t+1}^s; \boldsymbol{x}_{t+1}^n]$ and the corresponding weight vector $\boldsymbol{w}_{t+1} = [\boldsymbol{w}_{t+1}^s; \boldsymbol{w}_{t+1}^n]$, where $\boldsymbol{x}_{t+1}^s \in \mathbb{R}^{d_t}$ is the vector with survival features and $\boldsymbol{x}_{t+1}^n \in \mathbb{R}^{d_{t+1}-d_t}$ is the vector with newly-observed features. That is, $\boldsymbol{x}_{t+1}^s$ and $\boldsymbol{w}_{t+1}^s$ have the same dimension as $\boldsymbol{x}_t$ and $\boldsymbol{w}_t$. In this case, similarly, we extend the PA algorithm to update $\boldsymbol{w}_{t+1}$ by solving the following optimization task:

$$\boldsymbol{w}_{t+1} = \underset{\boldsymbol{w}=[\boldsymbol{w}^s;\boldsymbol{w}^n]\in\mathbb{R}^{d_{t+1}},\ell_{t+1}\leq\xi,\xi\in\mathbb{R}}{\operatorname{argmin}} \tfrac{1}{2}\|\boldsymbol{w}^s-\boldsymbol{w}_t\|_2^2 + \tfrac{1}{2}\|\boldsymbol{w}^n\|_2^2 + \mu\xi^2 \tag{3.3}$$

where $\mu > 0$ is a penalty parameter, and $\ell_{t+1} = \ell_{t+1}(\boldsymbol{w}, (\boldsymbol{x}_{t+1}, y_{t+1})) = \max(0, 1 - y_{t+1}(\boldsymbol{w}^T\boldsymbol{x}_{t+1})) = \max(0, 1 - y_{t+1}((\boldsymbol{w}^s)^T\boldsymbol{x}_{t+1}^s) - y_{t+1}((\boldsymbol{w}^n)^T\boldsymbol{x}_{t+1}^n))$ is the loss at round $t+1$. Then, we derive the closed-form solution for the above equation in Theorem 3.2.

THEOREM 3.2. (**Closed-form Solution of Eq. (3.3)**) *The general update strategy is the closed-form solution of Eq. (3.3),* $\boldsymbol{w}_{t+1} = [\boldsymbol{w}_{t+1}^s; \boldsymbol{w}_{t+1}^n] = [\boldsymbol{w}_t + \gamma_t y_{t+1}\boldsymbol{x}_{t+1}^s; \gamma_t y_{t+1}\boldsymbol{x}_{t+1}^n]$, *where* $\gamma_t = \frac{\max(0, 1-y_{t+1}\boldsymbol{w}_t^T\boldsymbol{x}_{t+1}^s)}{\|\boldsymbol{x}_{t+1}^s\|_2^2 + \|\boldsymbol{x}_{t+1}^n\|_2^2 + \frac{1}{2\mu}} = \frac{\ell_{t+1}(\boldsymbol{w}_t, (\boldsymbol{x}_{t+1}^s, y_{t+1}))}{\|\boldsymbol{x}_{t+1}^s\|_2^2 + \|\boldsymbol{x}_{t+1}^n\|_2^2 + \frac{1}{2\mu}} = \frac{\ell_{t+1}([\boldsymbol{w}_t;\boldsymbol{0}], (\boldsymbol{x}_{t+1}, y_{t+1}))}{\|\boldsymbol{x}_{t+1}\|_2^2 + \frac{1}{2\mu}}$, *and* $[\boldsymbol{w}_t; \boldsymbol{0}] \in \mathbb{R}^{d_{t+1}}$.

Hence, using the above two strategies with closed-form solutions, we can alternately update the online model in an SDOFS setup with widely evolving vanish, survival, and new features.

3.3 Memory-aware $\ell_{1,2}$-Norm Model Sparsifying In this section, we consider the problem setting of the online binary classification task for SDOFS and present the RSOL method to achieve sparse solution by leveraging the ℓ_1- and ℓ_2-mixed regularizer. We observe that when using the ℓ_2 norm as the regularization function, we obtain an all zeros vector if $\|\boldsymbol{w}\|_2 \leq \lambda$ (Theorem 3.3). The zero vector does not carry any generalization properties, which surfaces a concern regarding the usability of these norms as a form of regularization. This seemingly problematic phenomenon can, however, be useful in the incremental online setting. In many applications, the set of weights can be grouped into subsets where each subset of weights should be dealt with uniformly. For example, in the sparse online learning problem for SDOFS, each sliding window is associated with a different weight vector $\boldsymbol{w}^l \in \mathbb{R}^{d_l}$ $(l = 1, 2, \cdots, L)$. The prediction for a new instance $\boldsymbol{x}$ is a vector $\langle \boldsymbol{w}^1, \boldsymbol{x}\rangle$, $\langle \boldsymbol{w}^2, \boldsymbol{x}\rangle$, $\cdots$, $\langle \boldsymbol{w}^L, \boldsymbol{x}\rangle$, where L is the length of a specific sliding window. The predicted class is the index of the inner-product attaining the largest of the L values, $\operatorname{argmax}_{l\in\{1,2,\cdots,L\}}\langle \boldsymbol{w}^l, \boldsymbol{x}\rangle$. Since all the weight vectors operate over the same instance space, in order to achieve a sparse solution, it may be beneficial to tie the weights corresponding to the input features. That is, we would like to employ a regularization function that tends to zero the row of weights $w_1^l, w_2^l, \cdots, w_{d_l}^l$ $(l = 1, 2, \cdots, L)$ simultaneously. In these circumstances, the nullification of the entire weight vector by the ℓ_2 regularization becomes a powerful tool.

Formally, let $\boldsymbol{W} \in \mathbb{R}^{d\times L}$ represent a $d \times L$ matrix where the l-th $(l = 1, 2, \cdots, L)$ column of the matrix is the weight vector $\boldsymbol{w}^l$, where d is the total number of all evolvable features. Thus, the i-th $(i = 1, 2, \cdots, d)$ row corresponds to the weight of the i-th feature with respect to all instances. The mixed $\ell_{1,2}$-norm of $\boldsymbol{W}$, denoted $\|\boldsymbol{W}\|_{\ell_{1,2}}$, is obtained by computing the ℓ_2-norm of each row of $\boldsymbol{W}$ and then applying the ℓ_1-norm to the resulting d dimensional vector, i.e., $\|\boldsymbol{W}\|_{\ell_{1,2}} = \sum_{i=1}^d \|\boldsymbol{w}_i\|_2$. Thus, in a mixed-norm regularized optimization problem, we seek the minimizer of the objective function,

$$f(\boldsymbol{W}) + \lambda\|\boldsymbol{W}\|_{\ell_{1,2}}. \tag{3.4}$$

Given the specific variants of various norms, the model update for the $\ell_{1,2}$ mixed-norm is readily available. Let $\boldsymbol{w}^l \in \mathbb{R}^d$ denote the l-th $(l = 1, 2, \cdots, L)$ column of the matrix $\boldsymbol{W} \in \mathbb{R}^{d\times L}$, i.e., $\boldsymbol{W} = [\boldsymbol{w}^1, \boldsymbol{w}^2, \cdots, \boldsymbol{w}^L]$, and $\bar{\boldsymbol{w}}^i \in \mathbb{R}^L$ denote the i-th $(i = 1, 2, \cdots, d)$ row of the matrix $\boldsymbol{W} \in \mathbb{R}^{d\times L}$, i.e., $\boldsymbol{W} = [\bar{\boldsymbol{w}}^1; \bar{\boldsymbol{w}}^2; \cdots; \bar{\boldsymbol{w}}^d]$. Analogously to the standard norm-based regularization, we let $\boldsymbol{W}_t = [[\boldsymbol{w}_{t-L+1}; \boldsymbol{0}], [\boldsymbol{w}_{t-L+2}; \boldsymbol{0}], \cdots, [\boldsymbol{w}_t; \boldsymbol{0}]] \in \mathbb{R}^{d\times L}$ be the incremental matrix with all good feature alignment, where $[\boldsymbol{w}_{t-l+1}; \boldsymbol{0}] \in \mathbb{R}^d$ and $\boldsymbol{w}_{t-l+1} \in \mathbb{R}^{d_{t-l+1}}$ $(l = 1, 2, \cdots, L)$,

which can be obtained by online learning with decremental or incremental features or mixed features (Section 3.2). For the $\ell_{1,2}$ mixed-norm, we need to solve the problem,

$$\min_{\boldsymbol{W}\in\mathbb{R}^{d\times L}} \{\frac{1}{2}\|\boldsymbol{W}-\boldsymbol{W}_t\|_F^2 + \lambda\|\boldsymbol{W}\|_{\ell_{1,2}}\} \tag{3.5}$$

where $\|\cdot\|_F^2$ is the Frobenius norm of a matrix and $\lambda > 0$ is the regularization parameter.

This problem is equivalent to

$$\min_{\boldsymbol{W}=[\bar{\boldsymbol{w}}^1;\bar{\boldsymbol{w}}^2;\cdots;\bar{\boldsymbol{w}}^d]\in\mathbb{R}^{d\times L}} \sum_{i=1}^{d} \{\frac{1}{2}\|\bar{\boldsymbol{w}}^i - \bar{\boldsymbol{w}}_t^i\|_2^2 + \lambda\|\bar{\boldsymbol{w}}^i\|_2\} \tag{3.6}$$

where $\bar{\boldsymbol{w}}_t^i$ is the i-th row of $\boldsymbol{W}_t$. It is immediate to see that the problem given in Eq. (3.5) is decomposable into d separate problems of dimension L in Eq. (3.6), each of which can be solved by the procedures described in the following Theorem 3.3. The end result of solving these types of mixed-norm problems is a sparse matrix with numerous zero rows. In this way, RSOL can not only alleviate the curse of dimensionality by the incremental learning strategy, but also promote the sparsity of decremental and incremental features by considering feature correlations over time. Hence, RSOL has a big potential to improve the prediction performance compared with most existing methods.

THEOREM 3.3. (CLOSED-FORM SOLUTION OF RSOL) *The closed-form solution of the following ℓ_2-norm minimization: $\bar{\boldsymbol{w}}_\star^i = \mathrm{argmin}_{\bar{\boldsymbol{w}}^i\in\mathbb{R}^L}\{\frac{1}{2}\|\bar{\boldsymbol{w}}^i - \bar{\boldsymbol{w}}_t^i\|_2^2 + \lambda\|\bar{\boldsymbol{w}}^i\|_2\}$, where $i = 1, 2, \cdots, d$, is:*

$$\bar{\boldsymbol{w}}_\star^i = \begin{cases} \boldsymbol{0} & \text{if} \quad \|\bar{\boldsymbol{w}}_t^i\|_2 \le \lambda \\ (1 - \frac{\lambda}{\|\bar{\boldsymbol{w}}_t^i\|_2})\bar{\boldsymbol{w}}_t^i & \text{if} \quad \|\bar{\boldsymbol{w}}_t^i\|_2 > \lambda \end{cases} \tag{3.7}$$

Remark 1: It is worth noting that the ℓ_2 regularization results in a zero weight vector under the condition that $\|\bar{\boldsymbol{w}}_t^i\|_2 \le \lambda$. This condition is rather more stringent for sparsity than the condition for ℓ_1 (where a weight is sparse based only on its value, while here, sparsity happens only if the entire weight vector has ℓ_2-norm less than λ), so it is unlikely to hold in high dimensions. However, it does constitute a very important building block when using a mixed ℓ_1/ℓ_2-norm as the regularization function.

In summary, the pseudo codes of the proposed RSOL method are present in Algorithm 1.

4 Theoretical Analysis

Clearly, for the online update of decremental features, the regret of RSOL can be bounded by $\mathcal{O}(\sqrt{T})$ as the conventional online gradient descent with fixed feature space. Here, we introduce Lemma 4.1 and derive the regret bound of RSOL with incremental features in Theorem 4.1.

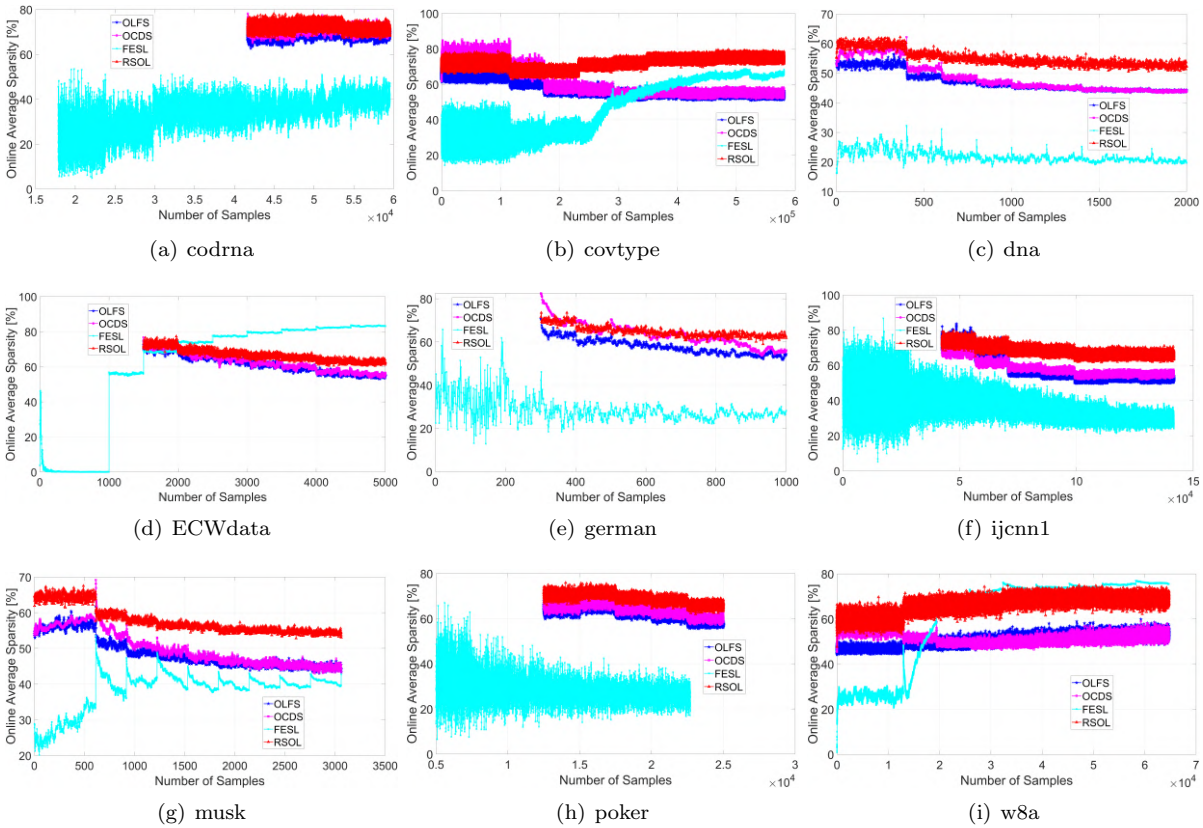
LEMMA 4.1. *Let $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_T, y_T)$ be a sequence of training instances, where $\boldsymbol{x}_t \in \mathbb{R}^{d_t}$, $d_t \le d_{t+1}$, and $y_t \in \{-1, +1\}$ for all $t \in [T]$. Let the learning rate γ_t for the online learning with incremental features. Then, the following bound holds for any $\boldsymbol{w} \in \mathbb{R}^{d_T}$ ($d_1 \le d_2 \le \cdots \le d_t \le d_T \le d_T$), $\sum_{t=0}^{T-1} \gamma_t(2\ell_{t+1}([\boldsymbol{w}_t; \boldsymbol{0}], (\boldsymbol{x}_{t+1}, y_{t+1})) - \gamma_t\|\boldsymbol{x}_{t+1}\|_2^2 - 2\ell_{t+1}(\Pi_{\boldsymbol{w}_{t+1}}\boldsymbol{w}, (\boldsymbol{x}_{t+1}, y_{t+1}))) \le \|\boldsymbol{w}\|_2^2$, where $\Pi_{\boldsymbol{w}_{t+1}}\boldsymbol{w} = \Pi_{\boldsymbol{x}_{t+1}}\boldsymbol{w} \in \mathbb{R}^{d_{t+1}}$ is the sub-vector of $\boldsymbol{w}$ and has the same dimension as $\boldsymbol{w}_{t+1}$ and $\boldsymbol{x}_{t+1}$.*

Algorithm 1 The RSOL Algorithm

Online input: streaming instance $\boldsymbol{x}_{t+1}$; true label y_{t+1}; regularization parameter λ, penalty parameter μ, and sliding window size L.
Online output: sparse solution, $\boldsymbol{w}_{t+1}$.

Initialization: $\boldsymbol{w}_0 = \boldsymbol{0} \in \mathbb{R}^{d_0}$;
for $t = 0, 1, \ldots, T-1$ **do**
  receive $\boldsymbol{x}_{t+1} \in \mathbb{R}^{d_{t+1}}$;
  if $(d_t \ge d_{t+1})$ **then**
    predict $\hat{y}_{t+1} = \mathrm{sign}((\boldsymbol{w}_t^s)^T\boldsymbol{x}_{t+1})$ and receive $y_{t+1} \in \{-1, +1\}$;
    suffer loss $\ell_t(\boldsymbol{w}_t) = \ell_{t+1}(\boldsymbol{w}_t^s, (\boldsymbol{x}_{t+1}, y_{t+1}))$;
    update $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t^s + \gamma_t y_{t+1}\boldsymbol{x}_{t+1}$, where $\gamma_t = \frac{\ell_{t+1}(\boldsymbol{w}_t^s, (\boldsymbol{x}_{t+1}, y_{t+1}))}{\|\boldsymbol{x}_{t+1}\|_2^2 + \frac{1}{2\mu}}$;
    sparse update $\boldsymbol{w}_{t+1} = \mathrm{argmin}_{\bar{\boldsymbol{w}}^i\in\mathbb{R}^L}\{\frac{1}{2}\|\bar{\boldsymbol{w}}^i - \bar{\boldsymbol{w}}_{t+1}^i\|_2^2 + \lambda\|\bar{\boldsymbol{w}}^i\|_2\}(i = 1, 2, \cdots, d_{t+1})$ through Eq. (3.7);
  else if $(d_t \le d_{t+1})$ **then**
    predict $\hat{y}_{t+1} = \mathrm{sign}([\boldsymbol{w}_t; \boldsymbol{0}]^T\boldsymbol{x}_{t+1})$ and receive $y_{t+1} \in \{-1, +1\}$;
    suffer loss $\ell_t(\boldsymbol{w}_t) = \ell_{t+1}([\boldsymbol{w}_t; \boldsymbol{0}], (\boldsymbol{x}_{t+1}, y_{t+1}))$;
    update $\boldsymbol{w}_{t+1} = [\boldsymbol{w}_{t+1}^s; \boldsymbol{w}_{t+1}^n] = [\boldsymbol{w}_t + \gamma_t y_{t+1}\boldsymbol{x}_{t+1}^s; \gamma_t y_{t+1}\boldsymbol{x}_{t+1}^n]$, where $\gamma_t = \frac{\ell_{t+1}([\boldsymbol{w}_t; \boldsymbol{0}], (\boldsymbol{x}_{t+1}, y_{t+1}))}{\|\boldsymbol{x}_{t+1}\|_2^2 + \frac{1}{2\mu}}$;
    sparse update $\boldsymbol{w}_{t+1} = \mathrm{argmin}_{\bar{\boldsymbol{w}}^i\in\mathbb{R}^L}\{\frac{1}{2}\|\bar{\boldsymbol{w}}^i - \bar{\boldsymbol{w}}_{t+1}^i\|_2^2 + \lambda\|\bar{\boldsymbol{w}}^i\|_2\}(i = 1, 2, \cdots, d_{t+1})$ through Eq. (3.7);
  end if
end for

THEOREM 4.1. (REGRET BOUND OF RSOL) *Let $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_T, y_T)$ be a sequence of training instances, where $\boldsymbol{x}_t \in \mathbb{R}^{d_t}$, $d_t \le d_{t+1}$, $y_t \in \{-1, +1\}$, and $\|\boldsymbol{x}_t\|_2^2 \le R^2$ ($R > 0$) for all $t \in [T]$. Let*



(a) codrna (b) covtype (c) dna

(d) ECWdata (e) german (f) ijcnn1

(g) musk (h) poker (i) w8a

Figure 1: Dynamic learning curves in terms of average sparsity of all competing online algorithms.

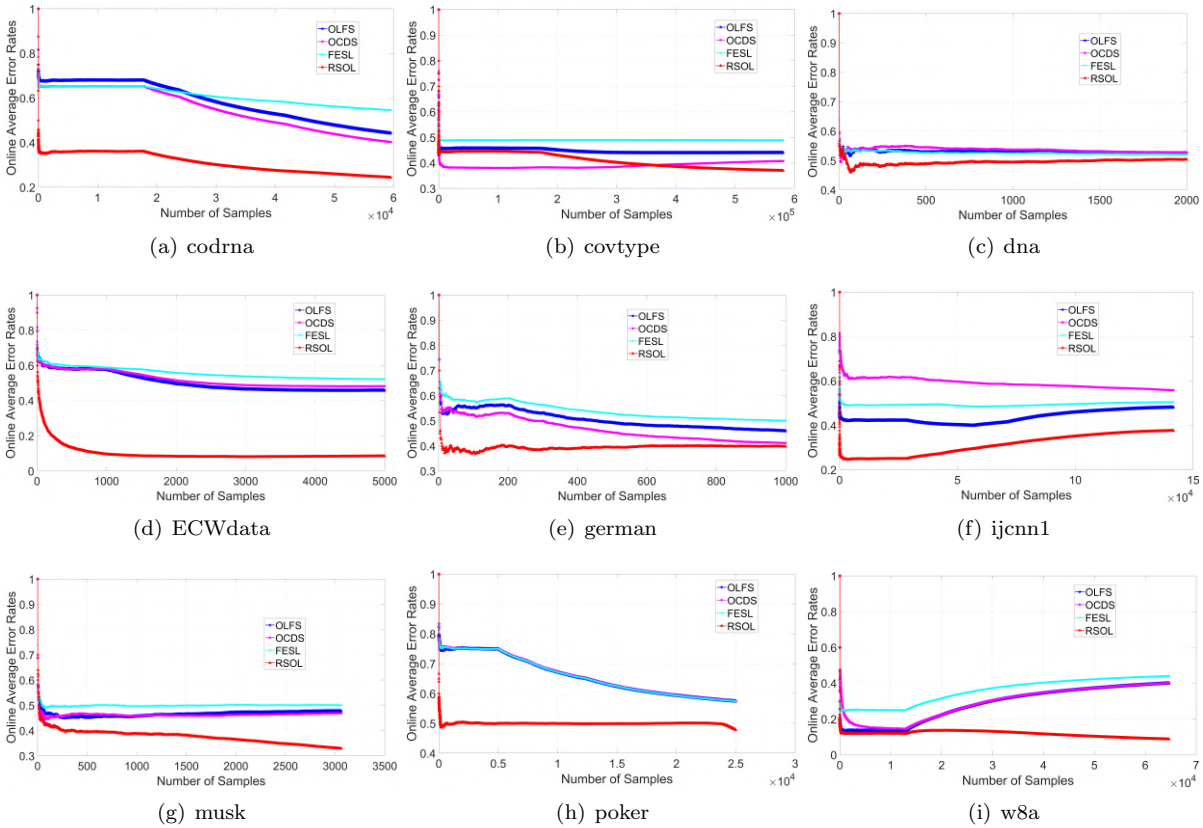
the learning rate $\gamma_t = \frac{\ell_{t+1}([\boldsymbol{w}_t;\mathbf{0}],(\boldsymbol{x}_{t+1},y_{t+1}))}{\|\boldsymbol{x}_{t+1}\|_2^2+\frac{1}{2\mu}}$ *for the RSOL with online learning with incremental features. Then, the following regret bound* $\mathcal{R}_T(\boldsymbol{w})$ *holds for any* $\boldsymbol{w} \in \mathbb{R}^{d_T}$ $(d_1 \leq d_2 \leq \cdots \leq d_t \leq d_T \leq d_T)$, $\mathcal{R}_T(\boldsymbol{w}) = \sum_{t=0}^{T-1} \ell_{t+1}([\boldsymbol{w}_t;\mathbf{0}],(\boldsymbol{x}_{t+1},y_{t+1})) - \sum_{t=0}^{T-1} \ell_{t+1}(\Pi_{\boldsymbol{w}_{t+1}}\boldsymbol{w},(\boldsymbol{x}_{t+1},y_{t+1})) \leq \sqrt{T}(\frac{\|\boldsymbol{w}\|_2}{2} + U_T) + (\frac{1}{2\mu} + R^2)\|\boldsymbol{w}\|_2^2$, *where* $U_T = \sqrt{\sum_{t=0}^{T-1} \ell_{t+1}^2(\Pi_{\boldsymbol{w}_{t+1}}\boldsymbol{w},(\boldsymbol{x}_{t+1},y_{t+1}))}$.

Remark 2: Theorem 4.1 indicates that the regret bound of RSOL is upper bounded by a sub-linear bound plus $(\frac{1}{2\mu}+R^2)\|\boldsymbol{w}\|_2^2$. If we assume that for any $\boldsymbol{w} \in \mathbb{R}^{d_T}$, we have $\|\boldsymbol{w}\|_2^2 \leq C^2$ $(C > 0)$, we can obtain that $\mathcal{R}_T(\boldsymbol{w}) \leq \sqrt{T}(\frac{C}{2} + U_T) + (\frac{1}{2\mu} + R^2)C^2$, which implies that the regret bound of RSOL enjoys $\mathcal{O}(\sqrt{T})$. Hence, the average regret bound of RSOL is $\mathcal{O}(\frac{1}{\sqrt{T}})$, which will converge to zero as the number of streaming samples $T \to \infty$.

5 Experiments

5.1 Datasets and Evaluation Protocol Our evaluations are conducted on nine benchmark datasets since these datasets originally contain large-scale instances and relatively high-dimensional features. The evaluated datasets span diverse application domains. Table 1 summarizes their statistics. We follow the same protocol of prior studies [16, 39] to simulate the streaming feature dynamics, where the later inputs tend to carry incrementally more features and decrementally less features. We split the original datasets into twenty chunks, where in the i-th $(i = 1, 2, \cdots, 10)$ chunk only the first $i \times 10\%$ features would be retained, i.e., the first data batch will retain the first 10% features and so forth. In the i-th $(i = 11, 12, \cdots, 20)$ chunk only $(21 - i) \times 10\%$ features would be retained. All the datasets are implemented with $5\% \sim 25\%$ outliers for the experiments. The Hoyer sparsity measurement [23] and dynamic error rate are employed to measure the algorithm performance.

5.2 Experimental Settings We implement RSOL in Matlab. The implementations of OLSF, OCDS, and FESL are conducted from [39], [16], and [22], respectively. For a fair comparison, the same experimental setup is applied to all algorithms. We set λ to be 50, L to be 10^3, and μ to be 1. The parameters are chosen



(a) codrna (b) covtype (c) dna

(d) ECWdata (e) german (f) ijcnn1

(g) musk (h) poker (i) w8a

Figure 2: Dynamic learning curves in terms of online average classification error rate of all competing algorithms.

with cross validation. All the other parameter values are determined based on [39], [16], and [22]. Twenty independent runs for each dataset are performed and the average results of each method are reported.

Table 1: Summary of the datasets.

Dataset	#Samples	#Features	#Classes
codrna[2]	59,535	8	2
covtype[2]	581,012	54	7
dna[3]	2,000	180	2
ECWdata[3]	5,000	20	2
german[3]	1,000	24	2
ijcnn1[3]	141,691	22	2
musk[3]	3,062	166	2
poker[3]	25,010	10	2
w8a[3]	64,700	300	2

5.3 Dynamic sparsity comparisons As shown in Fig. 1, we investigate the dynamic sparsity of all algorithms with the progression of incoming instances. RSOL significantly achieve much higher sparsity measurement than other methods on the "covtype", "dna", "musk", and "w8a" datasets. The Hoyer sparsity measurement achieved by RSOL is approximate 0.80, 0.50, 0.55, and 0.75 on these four datasets with relatively high feature dimension, which is superior to other three methods. This indicates that our proposed method not only achieves lower classification error rates for the online updates (Section 5.4), but also obtains a better sparsity level of the online model for handling streaming data with varying feature spaces.

5.4 Impacts of RSOL In this section, we dynamically show the real time classification performance of all competing algorithms when the streaming data comes sequentially in Fig. 2. The online average error rate curves of RSOL dominate the corresponding curves of all other algorithms (without much variation). The superiority of RSOL over others is evident on "codrna", "dna", "ECWdata", "ijcnn1", "musk", "poker", and "w8a" datasets. It is worth noting that most of these datasets have relatively high feature spaces. These results validate the efficiency of RSOL in handling high-dimensional issues compared with competing online learning algorithms tailored for data streams de-

[2]http://archive.ics.uci.edu/ml/datasets.php
[3]https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

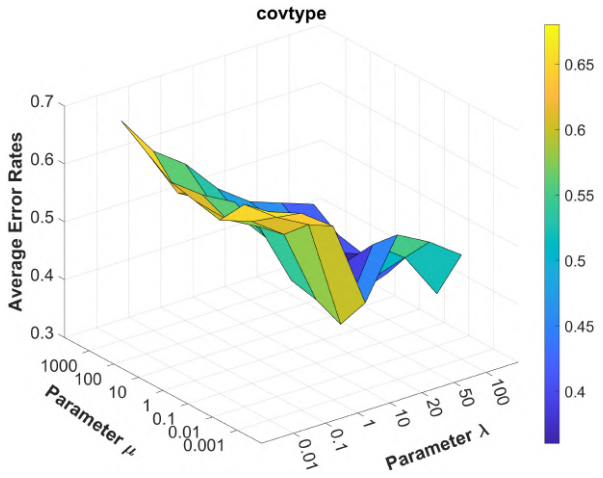


Figure 3: The sensitivity analysis of parameters λ and μ of RSOL on the "covtype" dataset.

scribed by open feature spaces. Moreover, the dynamical average error rate of RSOL is much lower than all other methods on the "w8a" dataset, where the severe imbalance ratios of positives and negatives are present. This indicates that RSOL has the potentials to effectively capture the underlying structure of the minority classes associated with the ever-evolving distributions of streaming data.

5.5 Sensitivity Analysis of RSOL To run RSOL, one needs to specify two regularizaton parameters λ and μ. Since λ determines the sparsity level of RSOL and μ is to balance rigidness and slackness, we investigate how the alterations of λ and μ affect the performance of RSOL by grid search. Taking the large-scale dataset "covtype" as the example, we summarize the performance of RSOL when λ and μ are selected from $[10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 2 \times 10^{1}, 5 \times 10^{1}, 10^{2}]$ and $[10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}]$, respectively. In Fig. 3, we compare the dynamic average error rates when varying these parameters. It is evident that the performances of RSOL are relatively stable without much variation when μ is in the range of $[10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}]$. However, the average error rates of RSOL are fluctuated as proper λ is vital to determine the sparsity and performance of the model. When μ is fixed, the average error rates of RSOL significantly is increased when the values of λ are decreased from 10^{2} to 10^{-2}. The reason is that a relatively small λ will promote the sparsity level, however, deteriorate the classification error rates of RSOL as well. Overall, RSOL is relatively robust to the parameter μ but is somewhat sensitive to the parameter λ.

6 Conclusion

In this paper, we focus on a general and challenging setting - online learning from SDOFS with dynamically vanished, survived and new features over time by proposing RSOL. By leveraging the power of the $\ell_{1,2}$-norm constraint, we exploit sparse 'non-zero' weights of the memory-aware matrix, resulting in truly sparse solutions in this complex prediction problem. We theoretically prove the regret bound of the proposed RSOL method with a sub-linear setup. A wide experiments on multiple benchmark datasets demonstrate the effectiveness of the proposed RSOL method over three advanced state-of-the-art online methods.

Acknowledgements

This work has been supported in part by the NSF Grants IIS-2245946 and IIS-2236578.

References

[1] E. Beyazit, J. Alagurajah, and X. Wu, *Online learning from data streams with varying feature spaces*, AAAI, pp. 3232–3239, 2019.

[2] A. Capponi, C. Fiandrino, B. Kantarci, L. Foschini, D. Kliazovich, and P. Bouvry, *A survey on mobile crowdsensing systems: Challenges, solutions, and opportunities*, IEEE Commun. Surv. Tutor., 21(3), pp. 2419–2465, 2019.

[3] Z. Chen, Z. Fang, W. Fan, A. Edwards, and K. Zhang, "CSTG: An effective framework for cost-sensitive sparse online learning," SDM, pp. 759–767, 2017.

[4] Z. Chen, Z. Fang, J. Zhao, W. Fan, A. Edwards, and K. Zhang, "Online density estimation over streaming data: A local adaptive solution," IEEE BigData, pp. 201–210, 2018.

[5] Z. Chen, Z. Fang, V. Sheng, A. Edwards, and K. Zhang, "CSRDA: Cost-sensitive regularized dual averaging for handling imbalanced and high-dimensional streaming data," ICBK, pp. 164–173, 2021.

[6] Z. Chen, Z. Fang, V. Sheng, J. Zhao, W. Fan, A. Edwards, and K. Zhang, *Adaptive robust local online density estimation for streaming data*, Int. J. Mach. Learn. Cybern., 12, pp. 1803–1824, 2021.

[7] Z. Chen, H. Zhan, V. Sheng, A. Edwards, and K. Zhang, *Projection dual averaging based second-order online learning*, ICDM, pp. 51–60, 2022.

[8] Z. Chen, H. Zhan, V. Sheng, A. Edwards, and K. Zhang, "Proximal cost-sensitive sparse group online learning," IEEE BigData, pp. 495–504, 2022.

[9] Z. Chen, V. Sheng, A. Edwards, and K. Zhang, *An effective cost-sensitive sparse online learning framework for imbalanced streaming data classification and its application to online anomaly detection*, Knowl. Inf. Syst., 65(1), pp. 59–87, 2023.

[10] Z. Chen, V. Sheng, A. Edwards, and K. Zhang, *Cost-sensitive sparse group online learning for imbalanced data streams*, Mach. Learn., 11, pp. 1–38, 2023.
[11] D. Wu, S. Zhuo, Y. Wang, Z. Chen, and Y. He, "Online semi-supervised learning with mix-typed streaming features," AAAI, pp. 4720–4728, 2023.
[12] J. Duchi and Y. Singer, *Efficient online and batch learning using forward backward splitting*, J. Mach. Learn. Res., 10, pp. 2899-2934, 2009.
[13] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, *Efficient projections onto the l1-ball for learning in high dimensions*, ICML, pp. 272–279, 2008.
[14] Y. Freund and R. E. Schapire, *Large margin classification using the perceptron algorithm*, Mach. Learn., 37(3), pp. 277–296, 1999.
[15] J. B. Gomes, M. M. Gaber, P. A. Sousa, and E. Menasalvas, *Mining recurring concepts in a dynamic feature space*, IEEE Trans. Neural Netw. Learn. Syst., 25(1), pp. 95–110, 2013.
[16] Y. He, B. Wu, D. Wu, E. Beyazit, S. Chen, and X. Wu, *Online learning from capricious data streams: a generative approach*, IJCAI, pp. 2491–2497, 2019.
[17] Y. He, J. Dong, B. J. Hou, Y. Wang, and F. Wang, *Online learning in variable feature spaces with mixed data*, ICDM, pp. 181–190, 2021.
[18] Y. He, X. Yuan, S. Chen, and X. Wu, *Online learning in variable feature spaces under incomplete supervision*, AAAI, pp. 4106–4114, 2021.
[19] C. Hou and Z. H. Zhou, *One-pass learning with incremental and decremental features*, IEEE Trans. Pattern Anal. Mach. Intell., 40(11), pp. 2776–2792, 2017.
[20] B. J. Hou, L. Zhang, and Z. H. Zhou, *Learning with feature evolvable streams*, NeurIPS, pp. 1416–1426, 2017.
[21] B. J. Hou, Y. H. Yan, P. Zhao, and Z. H. Zhou, *Storage fit learning with feature evolvable streams*, AAAI, pp. 7729–7736, 2021.
[22] C. Hou, L. Zhang, and Z. H. Zhou, *Learning with feature evolvable streams*, IEEE Trans. Knowl. Data Eng., 33(06), pp. 2602–2615, 2021.
[23] N. Hurley and S. Rickard, *Comparing measures of sparsity*, IEEE Trans. Inf. Theory, 55(10), pp. 4723–4741, 2009.
[24] J. Langford, L. Li, and T. Zhang, *Sparse online learning via truncated gradient*, J. Mach. Learn. Res., 10(3), pp. 777–801, 2009.
[25] S. Lee and S. Wright, *Manifold identification in dual averaging for regularized stochastic online learning*, J. Mach. Learn. Res., 13(1), pp. 1705–1744, 2012.
[26] H. Li, X. Wu, Z. Li, and W. Ding, *Group feature selection with streaming features*, ICDM, pp. 1109–1114, 2013.
[27] Q. Li, S. Shah, A. Nourbakhsh, X. Liu, and R. Fang, *Hashtag recommendation based on topic enhanced embedding, tweet entity data and learning to rank*, ACM CIKM, pp. 2085–2088, 2016.
[28] H. Lian, J. S. Atwood, B. J. Hou, J. Wu, and Y. He, *Online deep learning from doubly-streaming data*, ACM-MM, pp. 3185–3194, 2022.
[29] X. Liu, Q. Li, A. Nourbakhsh, R. Fang, M. Thomas, K. Anderson, R. Kociuba, M. Vedder, S. Pomerville, R. Wudali, and R. Martin, *Reuters tracer: A large scale system of detecting & verifying real-time news events from twitter*, ACM CIKM, pp. 207–216, 2016.
[30] Y. Ma and T. Zheng, *Stabilized sparse online learning for sparse data*, J. Mach. Learn. Res., 18(1), pp. 4773–4808, 2017.
[31] A. Ushio and M. Yukawa, *Projection-based regularized dual averaging for stochastic optimization*, IEEE Trans. Signal Process., 67(10), pp. 2720–2733, 2019.
[32] B. Wenerstrom and C. Giraud-Carrier, *Temporal data mining in dynamic feature spaces*, ICDM, pp. 1141–1145, 2006.
[33] X. Wu, K. Yu, W. Ding, H. Wang, and X. Zhu, *Online feature selection with streaming features*, IEEE Trans. Pattern Anal. Mach. Intell., 35(5), pp. 1178–1192, 2012.
[34] L. Xiao, *Dual averaging methods for regularized stochastic learning and online optimization*, J. Mach. Learn. Res., 11(88), pp. 2543–2596, 2010.
[35] D. You, R. Li, S. Liang, M. Sun, X. Ou, F. Yuan, L. Shen, and X. Wu, *Online causal feature selection for streaming features*, IEEE Trans. Neural Netw. Learn. Syst., 34(3), pp. 1563–1577, 2021.
[36] K. Yu, W. Ding, and X. Wu, *LOFS: A library of online streaming feature selection*, Knowl. Based Syst., 113, pp 1–3, 2016.
[37] H. Yu, M. Neely, and X. Wei, *Online convex optimization with stochastic constraints*, NeurIPS, pp. 1428–1438, 2017.
[38] P. Zhang, C. Zhou, P. Wang, B. J. Gao, X. Zhu, and L. Guo, *E-tree: An efficient indexing structure for ensemble models on data streams*, IEEE Trans. Knowl. Data Eng., 27(2), pp. 461–474, 2014.
[39] Q. Zhang, P. Zhang, G. Long, W. Ding, C. Zhang, and X. Wu, *Online learning from trapezoidal data streams*, IEEE Trans. Knowl. Data Eng., 28(10), pp. 2709–2723, 2016.
[40] Z. Zhang, P. Zhao, Y. Jiang, and Z. H. Zhou, *Learning with feature and distribution evolvable streams*, ICML, pp. 11317–11327, 2020.
[41] B. Zhou, F. Chen, and Y. Ying, *Dual averaging method for online graph-structured sparsity*, ACM SIGKDD, pp. 436–446, 2019.
[42] M. Zinkevich, *Online convex programming and generalized infinitesimal gradient ascent*, ICML, pp. 928–936, 2003.
[43] D. Wang, P. Wu, P. Zhao, Y. Wu, C. Miao, and S. C. Hoi, *High-dimensional data stream classification via sparse online learning*, ICDM, pp. 1007–1012, 2014.
[44] A Ben-Tal, E. Hazan, T. Koren, and S. Mannor, *Oracle-based robust optimization via online learning*, Oper. Res., 63(3), pp. 628–638, 2015.
[45] M. Tang, B. Sun, and G. Li, *A dual perspective of sparse and robust online learning algorithm*, ICIMCS, pp. 217–221, 2014.