Online Feature Selection by Adaptive Sub-gradient Methods

Tingting Zhai$^{1[0000-0002-4660-2125]}$, Hao Wang$^{1[0000-0003-2129-2148]}$,
Frédéric Koriche$^{2[0000-0002-6952-5775]}$, and Yang Gao$^{1[0000-0002-2488-1813]}$

[1] State Key Laboratory for Novel Software Technology,
Nanjing University, Nanjing, China, 210023
zhtt.go@gmail.com; wanghao@nju.edu.cn; gaoy@nju.edu.cn,
[2] Center of Research in Information in Lens,
Université d'Artois, Lens, France, 62307
koriche@cril.fr

Abstract. The overall goal of *online feature selection* is to iteratively select, from high-dimensional streaming data, a small, "budgeted" number of features for constructing accurate predictors. In this paper, we address the online feature selection problem using novel truncation techniques for two online sub-gradient methods: Adaptive Regularized Dual Averaging (ARDA) and Adaptive Mirror Descent (AMD). The corresponding truncation-based algorithms are called B-ARDA and B-AMD, respectively. The key aspect of our truncation techniques is to take into account the magnitude of feature values in the current predictor, together with their frequency in the history of predictions. A detailed regret analysis for both algorithms is provided. Experiments on six high-dimensional datasets indicate that both B-ARDA and B-AMD outperform two advanced online feature selection algorithms, OFS and SOFS, especially when the number of selected features is small. Compared to sparse online learning algorithms that use ℓ_1 regularization, B-ARDA is superior to ℓ_1-ARDA, and B-AMD is superior to Ada-Fobos.

Keywords: online feature selection, adaptive sub-gradient methods, high-dimensional streaming data

1 Introduction

Feature selection is an important topic of machine learning and data mining, for constructing sparse, accurate and interpretable models [7, 9, 13]. Given a batch of high-dimensional data instances, the overall goal is to find a small subset of relevant features, which are used to construct a low-dimensional predictive model. In modern applications involving streaming data, feature selection is not a "single-shot" offline operation, but an online process that iteratively updates the pool of relevant features, so as to track a sparse predictive model [16, 20]. A prototypical example of online feature selection is the anti-spam filtering task, in which the learner is required to classify each incoming message, using a small subset of features that is susceptible to evolve over time.

Conceptually, the online feature selection problem can be cast as a repeated prediction game between the learner and its environment. During each round t of the game, the learner starts by selecting a subset of at most B features over $\{1, \cdots, d\}$, where B is a predefined budget. Upon those selected features is built a predictive model $\boldsymbol{w}_t$ which, in the present paper, is assumed to be a linear function over $\mathbb{R}^d$. Then, a labelled example $(\boldsymbol{x}_t, y_t) \in \mathbb{R}^d \times \mathbb{R}$ is supplied by the environment, and the learner incurs a loss $f(\boldsymbol{w}_t; \boldsymbol{x}_t, y_t)$. The overall goal for the learner is to minimize its cumulative loss over T rounds of the game.

From a computational viewpoint, online feature selection is far from easy since, at each round t, the learner is required to solve a constrained optimization task, characterized by a budget (or ℓ_0 pseudo-norm) constraint on the model $\boldsymbol{w}_t$. Actually, this problem is known to be NP-hard for common loss functions advocated in classification and regression settings [10]. In order to alleviate this difficulty, two main approaches have been proposed in the literature. The first approach is to replace the nonconvex ℓ_0 constraint by a convex ℓ_1 constraint, or an ℓ_1 regularizer [3,4,6,8,11,15]. Though this approach is promoting the sparsity of solutions, it cannot guarantee that, at each iteration, the number of selected features is bounded by the predefined budget B. The second approach is divided in two main steps: first, solve a convex, unconstrained optimization problem, and next, seek a new solution that approximates the unconstrained solution while satisfying the ℓ_0 constraint. Based on this second approach, the OFS [16] and SOFS [20] strategies exploit truncation techniques for maintaining a budgeted number of features. However, OFS is oblivious to the history of predictions made so far, which might prove useful for assessing the frequencies of features. SOFS uses a suboptimal truncation rule that only considers the confidence of feature values in the current model, but ignores the magnitude of feature values which, again, could prove useful for estimating their relevance. Moreover, Wu et al. [20] did not provide any theoretical analysis for SOFS.

In this paper, we investigate the online feature selection problem using novel truncation techniques. Our contributions are threefold:

1. Two online feature selection algorithms, called Budgeted ARDA (B-ARDA) and Budgeted AMD (B-AMD), are proposed. B-ARDA and B-AMD perform truncation to eliminate irrelevant features. In our paper, the relevance of features is assessed by their frequency in the sequence of predictions, and their magnitude in the current predictor.
2. A detailed regret analysis for both algorithms is provided, which captures the intuition and rationale behind our truncation techniques.
3. Experiments on six high-dimensional datasets reveal the superiority of the proposed algorithms compared with both advanced feature selection algorithms and ℓ_1-based online learning algorithms.

The paper is organized as follows. Section 2 provides some related work in feature selection and online learning. Section 3 presents the notation used throughout the paper and elaborates on the problem setting. Our learning algorithms and their regret analysis are detailed in Section 4. Comparative experiments are given in Section 5. Finally, Section 6 concludes the paper.

2 Related work

Feature selection is a well-studied topic in machine learning and data mining [1,7,23]. Existing feature selection approaches include batch (or offline) methods and online methods. Batch methods, examined for instance in [12–14, 18], typically require an access to all available data, which makes them difficult to operate on sequential data. On the other hand, online methods are more suited to handle large-scale, and potentially streaming, information. Currently, there are two different “online modes” for selecting features. The first mode assumes that the number of examples is fixed but features arrive sequentially over time, such as in [17, 19, 22]. Contrastingly, the second mode assumes that the number of features is known in advance, but examples are supplied one by one, as studied for example in [16, 20]. We focus here on the second online mode, which is more natural for real-world streaming data. According to this mode, online feature selection methods can be grouped into three categories, summarized in Table 1.

Table 1. A list of recent works in online feature selection

Sparsity strategy	references / methods
ℓ_1 constraint	[2] [5]
ℓ_1 regularization	Fobos [3], TrunGrad [8], ℓ_1-RDA [21], CMD [6], ℓ_1-ARDA [4], Ada-Fobos [4], SOL [15]
ℓ_0 truncation	OFS [16], SOFS [20]

ℓ_1 *constraint/regularization.* Methods enforcing ℓ_1 constraints project the solution $\boldsymbol{w}$ after gradient descent update onto an ℓ_1 ball with radius r. Recent works, such as [2, 5], focus on designing efficient projection algorithms. There are also many researches which aim at solving an ℓ_1-regularized convex optimization problem. Notably, in [3], Duchi et al. propose the Fobos algorithm, which first performs a sub-gradient descent in order to get an intermediate solution, and then seeks a new solution that stays close to the intermediate solution and has a low ℓ_1 norm complexity. The second stage can be solved efficiently by truncating coefficients below a threshold in the intermediate solution. In [8], Langford et al. claim that such truncation operation is too aggressive and propose an alternative truncated gradient technique (TrunGrad), which gradually shrinks the coefficients to zero by a small amount. In [6], Duchi et al. generalize the Online Mirror Descent (OMD) to regularized losses, and propose the Composite Mirror Descent (CMD) algorithm, which exploits the composite structure of the objective to get desirable effects. Their derived algorithms include Fobos as an special case. In [21], Xiao presents an ℓ_1-Regularized Dual Averaging algorithm (ℓ_1-RDA) which, at each iteration, minimizes the sum of three terms: a linear function obtained by averaging all previous sub-gradients, an ℓ_1 regularization term and an additional strongly convex regularization term. In [4], Duchi et al. propose ARDA and ACMD, which adaptively modify the proximal function in order to incorporate the information related to the geometry of data observed in earlier iterations. The derived algorithms, ℓ_1-ARDA and Ada-Fobos, achieve better performance than their non-adaptive versions, namely, ℓ_1-RDA and Fobos. In [15], Wang et al. present a framework for sparse online classification. Their methods perform feature selection by carefully tuning the ℓ_1 regularization parameter.

ℓ_0 *Truncation.* In contrast with the above approaches, Jin et al. [16] propose a truncation method that satisfies the budget (or ℓ_0) constraint at each iteration. Their OFS algorithm first projects the predictor $\boldsymbol{w}$ (obtained from gradient descent) onto an ℓ_2 ball, so that most of the numerical values of $\boldsymbol{w}$ are concentrated to their largest elements, and then keeps only the B largest weights in $\boldsymbol{w}$. Wu et al. [20] further explore the truncation method for a confidence-weighted learning algorithm AROW, and proposed SOFS, which simply truncates the elements with least confidence after the update step in the diagonal version of AROW.

Our proposed online feature selection algorithms are also based on truncation techniques. Yet, our approaches differ from OFS and SOFS in the sense that truncation strategies are tailored to advanced adaptive sub-gradient methods, namely ARDA and AMD, which can perform more informative gradient descent, and which can find highly discriminative but rarely seen features. Moreover, we provide a detailed regret analysis for truncated versions of ARDA and AMD.

3 Notation and Problem Setting

In what follows, lowercase letters denote scalars or vectors, and uppercase letters represent matrices. An exception is the parameter B that captures our budget on the number of selected features. Let $[d]$ denote the set $\{1, \cdots, d\}$. We use $\boldsymbol{I}$ to denote the identity matrix, and $\text{diag}(\boldsymbol{v})$ to denote the diagonal matrix with vector $\boldsymbol{v}$ on the diagonal. For a linear predictor $\boldsymbol{w}_t$ chosen at iteration t, we use $w_{t,i}$ to denote its ith entry. As usual, we use $\langle \boldsymbol{v}, \boldsymbol{w} \rangle$ to denote the inner product between $\boldsymbol{v}$ and $\boldsymbol{w}$, and for any $p \in [1, \infty]$, we use $||\boldsymbol{w}||_p$ to denote the ℓ_p norm of $\boldsymbol{w}$. We also use $||\boldsymbol{w}||_0$ to denote the ℓ_0 pseudo-norm of $\boldsymbol{w}$, that is, $||\boldsymbol{w}||_0 = |\{i \in [d] : w_i \neq 0\}|$. For a convex loss function f_t, the sub-differential set of f_t at $\boldsymbol{w}$ is denoted by $\partial f_t(\boldsymbol{w})$, and $\boldsymbol{g}_t$ is used to denote a sub-gradient of f_t at $\boldsymbol{w}_t$, i.e. $\boldsymbol{g}_t \in \partial f_t(\boldsymbol{w}_t)$. When f_t is differentiable at $\boldsymbol{w}$, we use $\nabla f_t(\boldsymbol{w})$ to denote its unique sub-gradient (called gradient). Let $\boldsymbol{g}_{1:t} = [\boldsymbol{g}_1\ \boldsymbol{g}_2\ \cdots \boldsymbol{g}_t]$ be a $d \times t$ matrix obtained by concatenating the sub-gradients $\boldsymbol{g}_j$ from $j = 1$ to t. The ith row vector of $\boldsymbol{g}_{1:t}$ is denoted by $\boldsymbol{g}_{1:t,i}$. Let ψ_t be a strictly convex and continuously differentiable function defined, at each iteration t, on a closed convex set $\mathcal{C} \subseteq \mathbb{R}^d$ and let $\mathcal{D}_{\psi_t}(\boldsymbol{x}, \boldsymbol{y})$ denote the corresponding Bregman divergence, given by:

$$\mathcal{D}_{\psi_t}(\boldsymbol{x}, \boldsymbol{y}) = \psi_t(\boldsymbol{x}) - \psi_t(\boldsymbol{y}) - \langle \nabla \psi_t(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle, \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{C} \ .$$

By construction, we have $\mathcal{D}_{\psi_t}(\boldsymbol{x}, \boldsymbol{y}) \geq 0$ and $\mathcal{D}_{\psi_t}(\boldsymbol{x}, \boldsymbol{x}) = 0$ for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{C}$.

As mentioned above, the online feature selection problem can be formulated as a repeated prediction game between the learner and its environment. At iteration t, a new data point $\boldsymbol{x}_t \in \mathbb{R}^d$ is supplied to the learner, which is required to predict a label for $\boldsymbol{x}_t$ according to its current model $\boldsymbol{w}_t$. We assume that $\boldsymbol{w}_t$ is a sparse linear function in $\mathbb{R}^d$ such that $||\boldsymbol{w}_t||_0 \leq B$, where B is a predefined budget. Once the learner has committed to its prediction, the true label $y_t \in \mathbb{R}$ of $\boldsymbol{x}_t$ is revealed, and the learner suffers a loss $l(\boldsymbol{w}_t; (\boldsymbol{x}_t, y_t))$. We use here $l_t(\boldsymbol{w}_t) = l(\boldsymbol{w}_t; (\boldsymbol{x}_t, y_t))$, and we assume that $l_t(\boldsymbol{w}_t) = f_t(\boldsymbol{w}_t) + \varphi(\boldsymbol{w}_t)$, where $f_t(\boldsymbol{w}_t)$ is a convex loss function and $\varphi(\boldsymbol{w}_t)$ is a regularization function. The performance of the learner is measured according to its *regret*:

$$\mathcal{R}^T = \sum_{t=1}^{T} l_t(\boldsymbol{w}_t) - \min_{\boldsymbol{w} \in \mathbb{R}^d : ||\boldsymbol{w}||_0 \leq B} \sum_{t=1}^{T} l_t(\boldsymbol{w}) \ ,$$

where $||\boldsymbol{w}_t||_0 \leq B$ for all t. Our goal is to devise online feature selection strategies for which, regrets are sublinear in T. The nonconvex ℓ_0 constraint makes our problem more challenging than standard online convex optimization tasks.

4 B-ARDA and B-AMD

Advanced ARDA and AMD algorithms can take full advantage of the sub-gradient information observed in earlier iterations to perform more informative learning. Since ARDA and AMD are different methods, we need to develop specific truncation strategies for each of them.

4.1 B-ARDA and its regret analysis

A straightforward approach for performing ℓ_0 truncations is to keep the B elements with largest magnitude (in absolute value) in the current predictor $\boldsymbol{w}_t$. Such a naive approach suffers from an important shortcoming: frequently occurring discriminative features tend to be removed. This flaw results from the updating rule of adaptive sub-gradient methods: frequent attributes are given *low* learning rates, while infrequent attributes are given *high* learning rates.

Thus, we need to consider a more sophisticated truncation approach which takes into account the frequencies of features, together with their magnitude. To this end, we present the pseudocode of B-ARDA described in Algorithm 1. Basically, B-ARDA starts with a standard ARDA iteration from Step 1 to Step 9, and provides an intermediate solution $\boldsymbol{z}_{t+1}$, for which $||\boldsymbol{z}_{t+1}||_0 \leq B$ may not hold; then at Step 10, the algorithm truncates $\boldsymbol{z}_{t+1}$ in order to find a new solution $\boldsymbol{w}_{t+1}$ so that $||\boldsymbol{w}_{t+1}||_0 \leq B$ is satisfied. In our truncation operation, we consider both the magnitude of elements in $\boldsymbol{z}_{t+1}$, and the frequency of features conveyed by the diagonal matrix $\boldsymbol{H}_t$.

Note that the update at Step 9 often takes a closed-form. For example, if we use the standard Euclidean regularizer $\varphi(\boldsymbol{w}) = \frac{\lambda}{2}||\boldsymbol{w}||_2^2$, we get that

$$\boldsymbol{z}_{t+1} = -\eta(\lambda\eta t \boldsymbol{I} + \boldsymbol{H}_t)^{-1} \sum_{i=1}^{t} \boldsymbol{g}_i \ .$$

Algorithm 1: B-ARDA

```
   Input: Data stream $\{(\boldsymbol{x}_t, y_t)\}_{t=1}^{\infty}$, constant $\delta > 0$, step-size $\eta > 0$, budget $B$
   Output: $\boldsymbol{w}_t$
1  $\boldsymbol{w}_1 = \boldsymbol{0}$, $\boldsymbol{g}_{1:0} = []$;
2  for $t = 1, 2, \cdots$ do
3      Receive $\boldsymbol{x}_t$;
4      Predict the label of $\boldsymbol{x}_t$ with $\boldsymbol{w}_t$;
5      Receive $y_t$ and suffer loss $f_t(\boldsymbol{w}_t)$;
6      Receive sub-gradient $\boldsymbol{g}_t \in \partial f_t(\boldsymbol{w}_t)$;
7      Update $\boldsymbol{g}_{1:t} = [\boldsymbol{g}_{1:t-1}\ \boldsymbol{g}_t]$, $s_{t,i} = ||\boldsymbol{g}_{1:t,i}||_2$;
8      Set $\boldsymbol{H}_t = \delta \boldsymbol{I} + \mathrm{diag}(\boldsymbol{s}_t)$, $\bar{\boldsymbol{g}}_t = \frac{1}{t}\sum_{i=1}^{t} \boldsymbol{g}_i$, $\psi_t(\boldsymbol{w}) = \frac{1}{2}\langle \boldsymbol{w}, \boldsymbol{H}_t \boldsymbol{w} \rangle$;
9      ARDA update:
           $\boldsymbol{z}_{t+1} = \arg\min_{\boldsymbol{w} \in \mathbb{R}^d} \left\{ \eta \langle \bar{\boldsymbol{g}}_t, \boldsymbol{w} \rangle + \eta \varphi(\boldsymbol{w}) + \frac{1}{t} \psi_t(\boldsymbol{w}) \right\}$     (1)
10     Truncation operation:
           $\boldsymbol{w}_{t+1} = \arg\min_{\boldsymbol{w} \in \mathbb{R}^d} \langle \boldsymbol{w} - \boldsymbol{z}_{t+1}, \boldsymbol{H}_t(\boldsymbol{w} - \boldsymbol{z}_{t+1}) \rangle$, subject to $||\boldsymbol{w}||_0 \leq B$     (2)
```

The truncation operation at Step 10 can be efficiently solved by a simple greedy procedure. Let $\boldsymbol{v}_{t+1} \in \mathbb{R}^d$ be the vector with entries $v_{t+1,j} = H_{t,jj} z_{t+1,j}^2$. Based on this notation, if $||\boldsymbol{z}_{t+1}||_0 \leq B$, $\boldsymbol{w}_{t+1} = \boldsymbol{z}_{t+1}$; otherwise, $\boldsymbol{w}_{t+1} = \boldsymbol{z}_{t+1}^B$, where

$$z_{t+1,i}^B = \begin{cases} z_{t+1,i} & \text{if } H_{t,ii} z_{t+1,i}^2 \text{ occurs in the } B \text{ largest values of } \boldsymbol{v}_{t+1}, \\ 0 & \text{otherwise.} \end{cases}$$

The following result demonstrates that our truncation strategy for ARDA can lead to a sublinear regret. The proof, built essentially on the work of [4], is included in Appendix 1 for completeness.

Theorem 1. *Let $\xi_t^2 = \langle \boldsymbol{w}_t - \boldsymbol{z}_t, \boldsymbol{H}_{t-1}(\boldsymbol{w}_t - \boldsymbol{z}_t) \rangle$, which is the factual truncation error at iteration $t-1$. Set $\max_t ||\boldsymbol{g}_t||_\infty \leq \delta$ and $\max_t \xi_t \leq \xi$. For any $\boldsymbol{w}^* \in \mathbb{R}^d$, B-ARDA achieves the following regret bound:*

$$\mathcal{R}_{B\text{-}ARDA}^T \leq \frac{\delta}{2\eta} ||\boldsymbol{w}^*||_2^2 + \left(\frac{1}{2\eta} ||\boldsymbol{w}^*||_\infty + \eta \right) \sum_{i=1}^{d} ||\boldsymbol{g}_{1:T,i}||_2 + \xi \sqrt{2T \sum_{i=1}^{d} ||\boldsymbol{g}_{1:T,i}||_2} \ .$$

To see why the bound is sublinear, we notice from [4] that

$$\sum_{i=1}^{d} ||\boldsymbol{g}_{1:T,i}||_2 = \sqrt{d} \sqrt{\inf_{\boldsymbol{s}: \boldsymbol{s} \succeq 0, \langle 1, \boldsymbol{s} \rangle \leq d} \left\{ \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \mathrm{diag}(\boldsymbol{s})^{-1} \boldsymbol{g}_t \rangle \right\}} \leq \sqrt{d} \sqrt{\sum_{t=1}^{T} ||\boldsymbol{g}_t||_2^2} \ .$$

For the maximum truncation error $\xi = 0$, we directly recover the regret bound of ARDA. If $\xi \neq 0$, we get bounds of the form:

1. if ξ is $O(||\boldsymbol{w}^*||_\infty \sqrt{\sum_{i=1}^d ||\boldsymbol{g}_{1:T,i}||_2/T})$, $\mathcal{R}^T_{\text{B-ARDA}} = O(||\boldsymbol{w}^*||_\infty \sum_{i=1}^d ||\boldsymbol{g}_{1:T,i}||_2)$.
2. if ξ is $\Omega(||\boldsymbol{w}^*||_\infty \sqrt{\sum_{i=1}^d ||\boldsymbol{g}_{1:T,i}||_2/T})$, $\mathcal{R}^T_{\text{B-ARDA}} = O(\xi \sqrt{2T \sum_{i=1}^d ||\boldsymbol{g}_{1:T,i}||_2})$.

In other words, the cumulative loss of B-ARDA using only B features converges to that of an optimal solution in hindsight as T approaches infinity. The value of ξ is determined by the budget parameter B; larger values of B produce a smaller ξ, while smaller values of B yield a larger ξ.

We mention in passing that the naive truncation method, described in the beginning of this section, may be implemented by replacing the Step 10 in Algorithm 1 with

$$\boldsymbol{w}_{t+1} = \arg\min_{\boldsymbol{w} \in \mathbb{R}^d} \langle \boldsymbol{w} - \boldsymbol{z}_{t+1}, \boldsymbol{w} - \boldsymbol{z}_{t+1} \rangle, \text{ subject to } ||\boldsymbol{w}||_0 \leq B.$$

The regret produced by such truncation is, however, *not* sublinear since:

$$\sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{w}_t - \boldsymbol{z}_t \rangle \leq \sum_{t=1}^T ||\boldsymbol{g}_t||_2 ||\boldsymbol{w}_t - \boldsymbol{z}_t||_2 \leq \xi \sum_{t=1}^T ||\boldsymbol{g}_t||_2 \quad (\xi_t = ||\boldsymbol{w}_t - \boldsymbol{z}_t||_2) \ .$$

4.2 B-AMD and its regret analysis

We now focus on a truncation technique for the sub-gradient method AMD. Our approach is also considering both the magnitude of elements and the frequency of features. The pseudocode of B-AMD is presented in Algorithm 2, where $\mathcal{D}_{\psi_t}(\boldsymbol{w}, \boldsymbol{w}_t)$ is the Bregman divergence between $\boldsymbol{w}$ and $\boldsymbol{w}_t$. Note that we use AMD rather than ACMD since we do not use the composite structure of the objective function, but the truncation operation, to produce sparse solutions.

In essence, B-AMD performs an AMD iteration and then truncates the returned solution. Importantly, the AMD update at Step 9 admits a closed-form solution: $\boldsymbol{z}_{t+1} = \boldsymbol{w}_t - \eta \boldsymbol{H}_t^{-1} \mathbf{g}_t$. Similarly to B-ARDA, the truncation operation at Step 10 can be solved efficiently: if $||\boldsymbol{z}_{t+1}||_0 \leq B$, $\boldsymbol{w}_{t+1} = \boldsymbol{z}_{t+1}$; otherwise, $\boldsymbol{w}_{t+1} = \boldsymbol{z}^B_{t+1}$ where $\boldsymbol{z}^B_{t+1,i} = \boldsymbol{z}_{t+1,i}$ if $H_{t,ii}|\boldsymbol{z}_{t+1,i}|$ occurs in the B largest values of $\{H_{t,jj}|\boldsymbol{z}_{t+1,j}|, j \in [d]\}$, and $\boldsymbol{z}^B_{t+1,i} = 0$, otherwise.

The next theorem provides a regret bound for B-AMD, and conveys the rationale for the designed truncation. The proof is given in Appendix 2.

Theorem 2. *Set* $\xi_t = \sum_{i=1}^d H_{t,ii} |z_{t+1,i} - w_{t+1,i}|$. *For any* $\boldsymbol{w}^* \in \mathbb{R}^d$, *B-AMD achieves the following regret bound:*

$$\mathcal{R}^T_{B\text{-}AMD} \leq \frac{1}{\eta} ||\boldsymbol{w}^*||_\infty \sum_{t=1}^T \xi_t + \left(\frac{1}{2\eta} \max_{t \leq T} ||\boldsymbol{w}^* - \boldsymbol{w}_t||^2_\infty + \eta \right) \sum_{i=1}^d ||\boldsymbol{g}_{1:T,i}||_2 \ ,$$

where the first term of right-hand side is obtained from truncation.

Informally, the regret bound in Theorem 2 indicates that the cumulative loss of B-AMD converges toward the cumulative loss of the optimal $\boldsymbol{w}^*$ as T tends toward infinity, and the gap between the two is mainly dominated by the sum of truncation errors, that is, $\sum_{t=1}^T \xi_t$. This observation implies that we should try to minimize ξ_t at each round in order to reduce the gap. If the truncation error is set to $\xi_t = 0$ for any t, the regret bound of AMD is immediately recovered.

```
Algorithm 2: B-AMD
Input: Data stream $\{(\boldsymbol{x}_t, y_t)\}_{t=1}^{\infty}$, constant $\delta$, step-size $\eta$, budget $B$
Output: $\boldsymbol{w}_t$
$\boldsymbol{w}_1 = \boldsymbol{0}$, $\boldsymbol{g}_{1:0} = []$ ;
for $t = 1, 2, \cdots$ do
    Receive $\boldsymbol{x}_t$;
    Predict the label of $\boldsymbol{x}_t$ with $\boldsymbol{w}_t$;
    Receive $y_t$ and suffer loss $f_t(\boldsymbol{w}_t)$;
    Receive sub-gradient $\boldsymbol{g}_t \in \partial f_t(\boldsymbol{w}_t) + \partial \varphi(\boldsymbol{w}_t)$;
    Update $\boldsymbol{g}_{1:t} = [\boldsymbol{g}_{1:t-1}\ \boldsymbol{g}_t]$, $s_{t,i} = ||\boldsymbol{g}_{1:t,i}||_2$;
    Set $\boldsymbol{H}_t = \delta \boldsymbol{I} + \mathrm{diag}(\boldsymbol{s}_t)$, $\psi_t(\boldsymbol{w}) = \frac{1}{2}\langle \boldsymbol{w}, \boldsymbol{H}_t \boldsymbol{w} \rangle$;
    AMD update:
        $\boldsymbol{z}_{t+1} = \arg\min_{\boldsymbol{w} \in \mathbb{R}^d} \{\eta \langle \boldsymbol{g}_t, \boldsymbol{w} \rangle + \mathcal{D}_{\psi_t}(\boldsymbol{w}, \boldsymbol{w}_t)\}$        (3)
    Truncation operation:
        $\boldsymbol{w}_{t+1} = \arg\min_{\boldsymbol{w} \in \mathbb{R}^d} \sum_{i=1}^{d} H_{t,ii} |w_i - z_{t+1,i}|$, subject to $||\boldsymbol{w}||_0 \le B$        (4)
```

5 Experiments

This section reports two experimental studies[3]. In the first experiment, we compare B-ARDA and B-AMD with OFS and SOFS; in the second one, we compare our algorithms with ℓ_1-ARDA and Ada-Fobos, which achieve feature selection by carefully tuning the ℓ_1 regularization parameter. Although the theoretical analysis of our algorithms holds for many convex losses and regularization functions, we use here the squared hinge loss and ℓ_2 regularizer, that is, $f_t(\boldsymbol{w}_t) = (\max\{0, 1 - y_t \langle \boldsymbol{w}_t, \boldsymbol{x}_t \rangle\})^2$ and $\varphi(\boldsymbol{w}_t) = \frac{\lambda}{2} ||\boldsymbol{w}_t||_2^2$.

5.1 Datasets

Our experiments were performed on six high-dimensional binary classification datasets, selected from different domains. Their statistics are presented in Table 2, where "data density" is the maximal number of non-zero features per instance divided by the total number of features. Arcene's task is to distinguish cancer versus normal patterns from mass-spectrometric data. Dexter and farm_ads are text classification problems in a bag-of-words representation. Gisette aims to separate the highly confusable digits '4' and '9'. The above four datasets are available in UCI repository. Pcmac and basehock are a subset extracted from 20newsGroup[4]. Pcmac is to separate documents from "ibm.pc.hardware" and "mac.hardware", and basehock is to distinguish "baseball" versus "hockey".

[3] Our codes are available at https://github.com/LUCKY-ting/online-feature-selection

[4] http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html

Table 2. A summary of datasets

Dataset	# features (d)	# train (n)	# test	density
arcene	10000	100	100	71.25%
dexter	20000	300	300	1.65%
gisette	5000	6000	1000	29.6%
basehock	26214	1197	796	6.48%
pcmac	26214	1168	777	4.5%
farm_ads	54877	3313	830	4.19%

5.2 Comparison with online feature selection algorithms

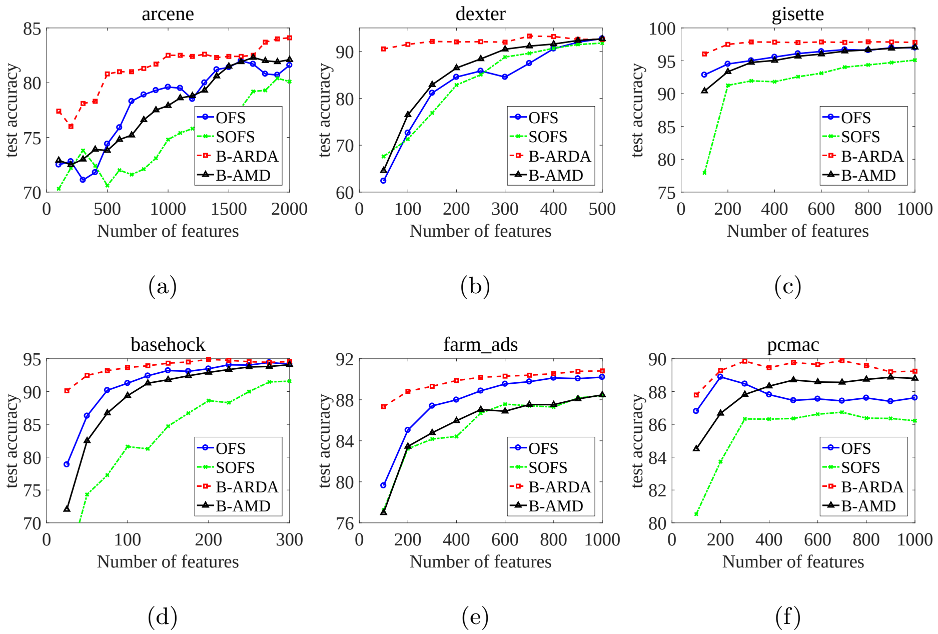
We first compared B-ARDA and B-AMD with OFS [16] and SOFS [20] on datasets in Table 2. For OFS, B-ARDA and B-AMD algorithms, the regularization parameter λ and the step-size η were obtained by choosing values in $\{10^{-1}, 10^{-1.5}, \cdots, 10^{-8}\}$, and taking the best performance in the training set. A similar interval was used for selecting the best parameter $1/\gamma$ for SOFS. We set $\delta = 10^{-2}$ for B-ARDA and B-AMD on all datasets. Based on these empirically optimal parameter values, we vary the budget B in order to plot the test accuracy versus the number of selected features.

In order to make our results reliable under the optimal parameter setting, each algorithm was run 10 times, each time with τ passes on the training examples. Namely, each pass is done with a random permutation of the training set, and the classifier output at the end of τ passes is evaluated on a separated test set. The number of passes τ was set as $\lceil \frac{2d}{n} \rceil$ for each dataset. Fig. 1 and Fig. 2 display the average test accuracy of all algorithms for varying feature budgets.

Based on Fig. 1, we can observe that B-ARDA achieves the highest test accuracy for every budget parameter B. By contrast, B-AMD is outperformed by B-ARDA, but remains better than SOFS. By coupling Fig. 1 and Fig. 2, we observe that the performance gap between B-ARDA and the other algorithms decreases as the budget B increases. The results for B-AMD are mixed: for small values of B, this strategy is outperformed by OFS, due to a large truncation error; but when the budget is gradually increasing, B-AMD outperforms OFS at some value of B. For example, on the gisette and farm_ads datasets, B-AMD outperforms OFS at $B \geq 1000$ and $B \geq 2000$, respectively. SOFS achieves poor accuracy for small budgets, but its performance is approaching B-ARDA and B-AMD by increasing B. This steams from the fact SOFS tends to keep more features to achieve an accuracy that is competitive with that of B-ARDA and B-AMD. We can clearly see that B-ARDA, B-AMD and SOFS are all outperforming OFS for large values of B. To sum up, when a small number of features is desired, B-ARDA is the best choice, and when more features are allowed, both B-ARDA and B-AMD are better than OFS and SOFS.

5.3 Comparison with sparse online learning algorithms

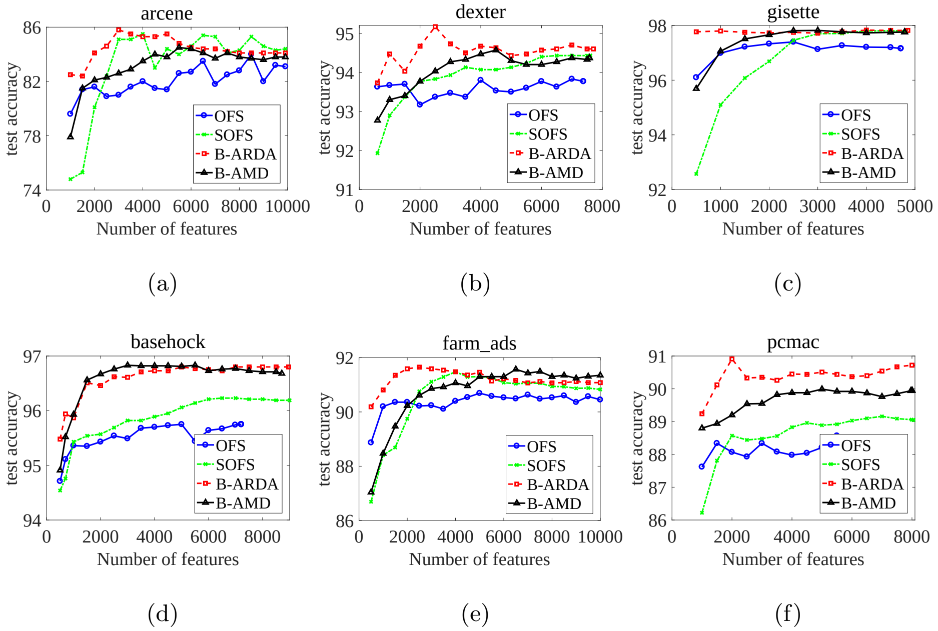
We have also compared our proposed algorithms with ℓ_1-ARDA [4] and Ada-Fobos [4], which achieve feature selection by carefully tuning the ℓ_1 regularization



(a) (b) (c)

(d) (e) (f)

Fig. 1. Test performance w.r.t. OFS and SOFS (small feature budgets)

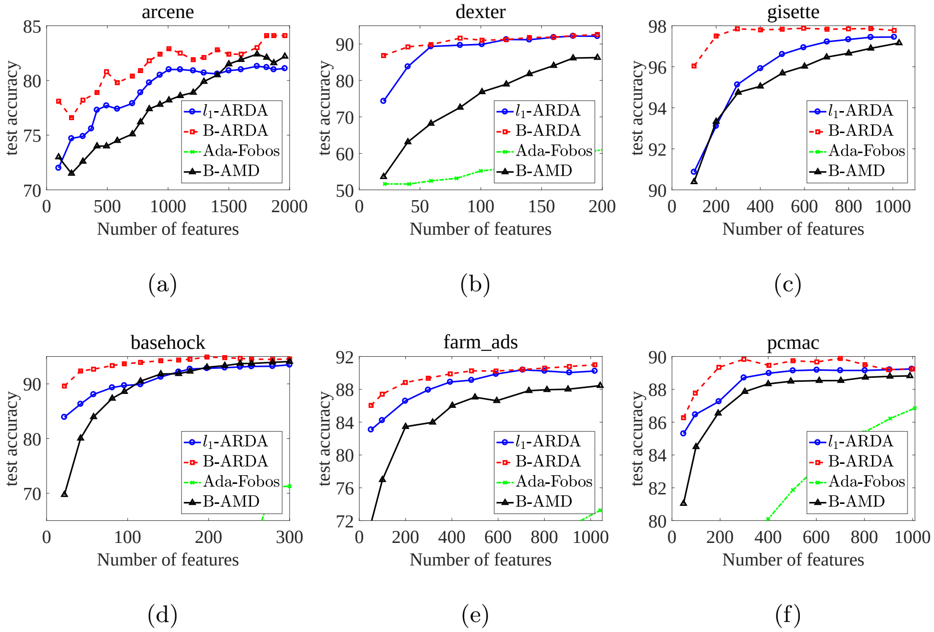


(a) (b) (c)

(d) (e) (f)

Fig. 2. Test performance w.r.t. OFS and SOFS (large feature budgets)

parameter. For fair comparisons, the choice of step-sizes follows the experimental setup in Section 5.2. Once the step-size value is determined, the ℓ_1 regularization parameter is gradually modified for deriving different numbers B of features for ℓ_1-ARDA and Ada-Fobos. For B-ARDA and B-AMD, the input budget values B are those obtained by ℓ_1-ARDA and Ada-Fobos, respectively. Fig. 3 presents the test accuracy of these algorithms when a small number of features is selected. The plot for Ada-Fobos does not appear in some subfigures since its accuracy falls outside the specified range.



(a) (b) (c)

(d) (e) (f)

Fig. 3. Test performance w.r.t. ℓ_1-ARDA and Ada-Fobos (small feature budgets)

Based on Fig. 3, we observe that both B-ARDA and ℓ_1-ARDA outperform B-AMD and Ada-Fobos. This indicates that the regularized dual averaging method is more competitive than the mirror descent method especially when very sparse solutions are desired. Remarkably, B-ARDA is better than ℓ_1-ARDA when a small number of features is required, which means that our truncation strategy for ARDA is successful. We also notice that Ada-Fobos has a poor performance for small budgets; by contrast, B-AMD is much better. We do not present the plots for large number of features due to space constraints, but we report the observed results: as the number of features increases, the performance gaps among these algorithms are gradually shrinking, and finally, these algorithms empirically attain a similar test accuracy. Yet, from a practical viewpoint, it is much simpler to select a desired number of features for B-ARDA and B-AMD. For ℓ_1-

ARDA and Ada-Fobos, the number of features cannot be determined in advance: it is empirically conditioned by the choice of the regularization parameter.

6 Conclusion

In this paper, two novel online feature selection algorithms, called B-ARDA and B-AMD, have been proposed and analyzed. Both algorithms perform feature selection via truncation techniques, which take into account the magnitude of feature values in the current predictor, together with the frequency of features in the observed data stream. By taking as input a desired budget, both algorithms are easy to control, especially in comparison with ℓ_1-based feature selection techniques. We have shown on six high-dimensional datasets that B-ARDA outperforms advanced OFS and SOFS especially when a small number of features is required; when more features are allowed, both B-ARDA and B-AMD are better than OFS and SOFS. Compared with ℓ_1-ARDA and Ada-Fobos that achieve feature selection by carefully tuning the ℓ_1 regularization parameter, B-ARDA is shown to superior to ℓ_1-ARDA and B-AMD superior to Ada-Fobos, which corroborates the interest of our truncation strategies. A natural perspective of research is to investigate whether our approach may be extended to "structured" feature selection tasks, such as group structures.

Appendix 1: Proof of Theorem 1

Proof. Denote the Mahalanobis norm with respect to a symmetric matrix $\boldsymbol{A}$ by $||\cdot||_{\boldsymbol{A}} = \sqrt{\langle \cdot, \boldsymbol{A}\cdot\rangle}$. If $\boldsymbol{A}$ is positive definite, $||\cdot||_{\boldsymbol{A}}$ is a norm.

Recall that a function ψ is *μ-strongly convex* with respect to a norm $||\cdot||$ if the following inequality holds for any $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$:

$$\psi(\boldsymbol{w}_1) - \psi(\boldsymbol{w}_2) - \langle \nabla\psi(\boldsymbol{w}_2), \boldsymbol{w}_1 - \boldsymbol{w}_2\rangle \geq \frac{\mu}{2}||\boldsymbol{w}_1 - \boldsymbol{w}_2||^2 \ .$$

Note that since $\psi_t(\boldsymbol{w}) = \frac{1}{2}\langle \boldsymbol{w}, \boldsymbol{H}_t\boldsymbol{w}\rangle$, and $\boldsymbol{H}_t$ is positive definite, $t\varphi(\boldsymbol{w}) + \frac{\psi_t(\boldsymbol{w})}{\eta}$ is $\frac{1}{\eta}$-strongly convex with respect to the norm $||\cdot||_{\boldsymbol{H}_t}$.

Let ψ_t^* be the conjugate dual of $t\varphi(\boldsymbol{w}) + \frac{\psi_t(\boldsymbol{w})}{\eta}$, that is,

$$\psi_t^*(\boldsymbol{v}) = \sup_{\boldsymbol{w}\in\mathbb{R}^d}\left\{\langle \boldsymbol{v}, \boldsymbol{w}\rangle - t\varphi(\boldsymbol{w}) - \frac{\psi_t(\boldsymbol{w})}{\eta}\right\} \ .$$

Since $\psi_t^*(\boldsymbol{v})$ is a linear function of $\boldsymbol{v}$, we have

$$\nabla\psi_t^*(\boldsymbol{v}) = \arg\min_{\boldsymbol{w}\in\mathbb{R}^d}\left\{-\langle \boldsymbol{v}, \boldsymbol{w}\rangle + t\varphi(\boldsymbol{w}) + \frac{\psi_t(\boldsymbol{w})}{\eta}\right\} \ .$$

Owing to the duality of strong convexity and strong smoothness, ψ_t^* is η-smooth with respect to $||\cdot||_{\boldsymbol{H}_t^{-1}}$. From the definition of smoothness, we have for any $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$,

$$\psi_t^*(\boldsymbol{v}_1) - \psi_t^*(\boldsymbol{v}_2) - \langle \nabla\psi_t^*(\boldsymbol{v}_2), \boldsymbol{v}_1 - \boldsymbol{v}_2\rangle \leq \frac{\eta}{2}||\boldsymbol{v}_1 - \boldsymbol{v}_2||^2_{\boldsymbol{H}_t^{-1}} \ . \tag{5}$$

Let $\boldsymbol{v}_t = \sum_{i=1}^{t} \boldsymbol{g}_i$. For any $\boldsymbol{w}^* \in \mathbb{R}^d$, we have

$$\begin{aligned}
&\sum_{t=1}^{T}(l_t(\boldsymbol{w}_t) - l_t(\boldsymbol{w}^*)) = \sum_{t=1}^{T}\left(f_t(\boldsymbol{w}_t) + \varphi(\boldsymbol{w}_t) - f_t(\boldsymbol{w}^*) - \varphi(\boldsymbol{w}^*)\right)\\
&= \varphi(\boldsymbol{w}_1) - \varphi(\boldsymbol{w}_{T+1}) + \sum_{t=1}^{T}\left(f_t(\boldsymbol{w}_t) + \varphi(\boldsymbol{w}_{t+1}) - f_t(\boldsymbol{w}^*) - \varphi(\boldsymbol{w}^*)\right)\\
&\le \sum_{t=1}^{T}\left(\langle \boldsymbol{g}_t, \boldsymbol{w}_t - \boldsymbol{w}^*\rangle + \varphi(\boldsymbol{w}_{t+1}) - \varphi(\boldsymbol{w}^*)\right)\\
&\le \sum_{t=1}^{T}(\langle \boldsymbol{g}_t, \boldsymbol{w}_t\rangle + \varphi(\boldsymbol{w}_{t+1})) + \frac{1}{\eta}\psi_T(\boldsymbol{w}^*) + \sup_{\boldsymbol{w}\in\mathbb{R}^d}\left\{-\sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{w}\rangle - T\varphi(\boldsymbol{w}) - \frac{1}{\eta}\psi_T(\boldsymbol{w})\right\}\\
&\le \sum_{t=1}^{T}(\langle \boldsymbol{g}_t, \boldsymbol{w}_t\rangle + \varphi(\boldsymbol{w}_{t+1})) + \frac{1}{\eta}\psi_T(\boldsymbol{w}^*) + \psi_T^*(-\boldsymbol{v}_T) \ .
\end{aligned}$$

According to the update equation (1) of B-ARDA, we have

$$\begin{aligned}
&\psi_T^*(-\boldsymbol{v}_T) = -\langle \boldsymbol{v}_T, \boldsymbol{z}_{T+1}\rangle - T\varphi(\boldsymbol{z}_{T+1}) - \frac{1}{\eta}\psi_T(\boldsymbol{z}_{T+1})\\
&\le_1 - \langle \boldsymbol{v}_T, \boldsymbol{z}_{T+1}\rangle - (T-1)\varphi(\boldsymbol{z}_{T+1}) - \frac{1}{\eta}\psi_{T-1}(\boldsymbol{z}_{T+1}) - \varphi(\boldsymbol{z}_{T+1})\\
&\le \sup_{\boldsymbol{w}\in\mathbb{R}^d}\left\{-\langle \boldsymbol{v}_T, \boldsymbol{w}\rangle - (T-1)\varphi(\boldsymbol{w}) - \frac{1}{\eta}\psi_{T-1}(\boldsymbol{w})\right\} - \varphi(\boldsymbol{z}_{T+1})\\
&= \psi_{T-1}^*(-\boldsymbol{v}_T) - \varphi(\boldsymbol{z}_{T+1})\\
&\le_2 \psi_{T-1}^*(-\boldsymbol{v}_{T-1}) - \langle \nabla\psi_{T-1}^*(-\boldsymbol{v}_{T-1}), \boldsymbol{g}_T\rangle + \frac{\eta}{2}||\boldsymbol{g}_T||^2_{\boldsymbol{H}_{T-1}^{-1}} - \varphi(\boldsymbol{z}_{T+1})\\
&= \psi_{T-1}^*(-\boldsymbol{v}_{T-1}) - \langle \boldsymbol{z}_T, \boldsymbol{g}_T\rangle + \frac{\eta}{2}||\boldsymbol{g}_T||^2_{\boldsymbol{H}_{T-1}^{-1}} - \varphi(\boldsymbol{z}_{T+1}) \ ,
\end{aligned}$$

where $\le_1$ follows from the monotonicity of $\psi_t(\boldsymbol{w})$, that is, $\psi_{t+1}(\boldsymbol{w}) \ge \psi_t(\boldsymbol{w})$ and $\le_2$ follows from (5).

Combining the above inequality and using the fact that $\varphi(\boldsymbol{w}_t) \le \varphi(\boldsymbol{z}_t)$ for any t, we obtain that

$$\begin{aligned}
\sum_{t=1}^{T}(l_t(\boldsymbol{w}_t) - l_t(\boldsymbol{w}^*)) \le{}& \frac{1}{\eta}\psi_T(\boldsymbol{w}^*) + \psi_{T-1}^*(-\boldsymbol{v}_{T-1}) + \frac{\eta}{2}||\boldsymbol{g}_T||^2_{\boldsymbol{H}_{T-1}^{-1}}\\
&+ \langle \boldsymbol{g}_T, \boldsymbol{w}_T - \boldsymbol{z}_T\rangle + \sum_{t=1}^{T-1}(\langle \boldsymbol{g}_t, \boldsymbol{w}_t\rangle + \varphi(\boldsymbol{w}_{t+1})) \ .
\end{aligned}$$

Since $\{\boldsymbol{w} \in \mathbb{R}^d : ||\boldsymbol{w}||_0 \le B\}$ is a subset of $\mathbb{R}^d$, the above upper bound holds for $\mathcal{R}^T_{\text{B-ARDA}}$.

By repeating the above process, we get that

$$\begin{aligned}
\mathcal{R}^T_{\text{B-ARDA}} &\leq \frac{1}{\eta}\psi_T(\boldsymbol{w}^*) + \psi^*_0(-\boldsymbol{v}_0) + \frac{\eta}{2}\sum_{t=1}^{T}||\boldsymbol{g}_t||^2_{\boldsymbol{H}^{-1}_{t-1}} + \sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{w}_t - \boldsymbol{z}_t\rangle \\
&\leq_1 \frac{1}{\eta}\psi_T(\boldsymbol{w}^*) + \frac{\eta}{2}\sum_{t=1}^{T}||\boldsymbol{g}_t||^2_{\boldsymbol{H}^{-1}_{t-1}} + \sum_{t=1}^{T}||\boldsymbol{w}_t - \boldsymbol{z}_t||_{\boldsymbol{H}_{t-1}}||\boldsymbol{g}_t||_{\boldsymbol{H}^{-1}_{t-1}} \\
&\leq_2 \frac{1}{\eta}\psi_T(\boldsymbol{w}^*) + \frac{\eta}{2}\sum_{t=1}^{T}||\boldsymbol{g}_t||^2_{\boldsymbol{H}^{-1}_{t-1}} + \xi\sum_{t=1}^{T}||\boldsymbol{g}_t||_{H^{-1}_{t-1}} \\
&\leq \frac{1}{\eta}\psi_T(\boldsymbol{w}^*) + \frac{\eta}{2}\sum_{t=1}^{T}||\boldsymbol{g}_t||^2_{\boldsymbol{H}^{-1}_{t-1}} + \xi\sqrt{T\sum_{t=1}^{T}||\boldsymbol{g}_t||^2_{\boldsymbol{H}^{-1}_{t-1}}}, \qquad (6)
\end{aligned}$$

where we used $\psi^*_0(-\boldsymbol{v}_0) = 0$ and the Hölder's inequality (for dual norms) for $\leq_1$. For $\leq_2$, we used $\xi_t = ||\boldsymbol{w}_t - \boldsymbol{z}_t||_{\boldsymbol{H}_{t-1}}$ and the assumption that $\xi_t \leq \xi$ for $t = 1, 2, \cdots T$. We now give a bound for each term.

$$\psi_T(\boldsymbol{w}^*) = \frac{\delta}{2}||\boldsymbol{w}^*||^2_2 + \frac{1}{2}\langle \boldsymbol{w}^*, \operatorname{diag}(\mathbf{s}_T)\boldsymbol{w}^*\rangle \leq \frac{\delta}{2}||\boldsymbol{w}^*||^2_2 + \frac{1}{2}||\boldsymbol{w}^*||_\infty\sum_{i=1}^{d}||\boldsymbol{g}_{1:T,i}||_2$$

With the assumption $\max_t ||\boldsymbol{g}_t||_\infty \leq \delta$, we can use Lemma 4 in [4] and get

$$\sum_{t=1}^{T}||\boldsymbol{g}_t||^2_{\boldsymbol{H}^{-1}_{t-1}} \leq \sum_{t=1}^{T}\langle \boldsymbol{g}_t, \operatorname{diag}(\boldsymbol{s}_t)^{-1}\boldsymbol{g}_t\rangle \leq 2\sum_{i=1}^{d}||\boldsymbol{g}_{1:T,i}||_2$$

The main result follows by plugging these local bounds into (6).

Appendix 2: Proof of Theorem 2

Proof. For any $\boldsymbol{w}^* \in \mathbb{R}^d$, we have

$$\begin{aligned}
&\eta(f_t(\boldsymbol{w}_t) + \varphi(\boldsymbol{w}_t) - f_t(\boldsymbol{w}^*) - \varphi(\boldsymbol{w}^*)) \\
&\leq \langle \eta\boldsymbol{g}_t, \boldsymbol{w}_t - \boldsymbol{w}^*\rangle = \langle \eta\boldsymbol{g}_t, \boldsymbol{w}_t - \boldsymbol{z}_{t+1} + \boldsymbol{z}_{t+1} - \boldsymbol{w}^*\rangle \\
&= \langle \eta\boldsymbol{g}_t + \boldsymbol{H}_t(\boldsymbol{z}_{t+1} - \boldsymbol{w}_t), \boldsymbol{z}_{t+1} - \boldsymbol{w}^*\rangle + \langle \eta\boldsymbol{g}_t, \boldsymbol{w}_t - \boldsymbol{z}_{t+1}\rangle \\
&\quad + \langle \boldsymbol{H}_t(\boldsymbol{w}_t - \boldsymbol{z}_{t+1}), \boldsymbol{z}_{t+1} - \boldsymbol{w}^*\rangle \\
&\leq_1 \langle \eta\boldsymbol{g}_t, \boldsymbol{w}_t - \boldsymbol{z}_{t+1}\rangle + \langle \boldsymbol{H}_t(\boldsymbol{w}_t - \boldsymbol{z}_{t+1}), \boldsymbol{z}_{t+1} - \boldsymbol{w}^*\rangle \\
&= \eta\langle \sqrt{\eta}\boldsymbol{g}_t, \frac{1}{\sqrt{\eta}}(\boldsymbol{w}_t - \boldsymbol{z}_{t+1})\rangle + \mathcal{D}_{\psi_t}(\boldsymbol{w}^*, \boldsymbol{w}_t) - \mathcal{D}_{\psi_t}(\boldsymbol{w}^*, \boldsymbol{z}_{t+1}) - \mathcal{D}_{\psi_t}(\boldsymbol{z}_{t+1}, \boldsymbol{w}_t) \\
&\leq_2 \frac{\eta^2}{2}||\boldsymbol{g}_t||^2_{\boldsymbol{H}^{-1}_t} + \frac{1}{2}||\boldsymbol{w}_t - \boldsymbol{z}_{t+1}||^2_{\boldsymbol{H}_t} - \mathcal{D}_{\psi_t}(\boldsymbol{z}_{t+1}, \boldsymbol{w}_t) \\
&\quad + \mathcal{D}_{\psi_t}(\boldsymbol{w}^*, \boldsymbol{w}_t) - \mathcal{D}_{\psi_t}(\boldsymbol{w}^*, \boldsymbol{z}_{t+1}) \\
&= \frac{\eta^2}{2}||\boldsymbol{g}_t||^2_{\boldsymbol{H}^{-1}_t} + \mathcal{D}_{\psi_t}(\boldsymbol{w}^*, \boldsymbol{w}_t) - \mathcal{D}_{\psi_t}(\boldsymbol{w}^*, \boldsymbol{z}_{t+1})
\end{aligned}$$

$$
\begin{aligned}
&= \frac{\eta^2}{2}||\boldsymbol{g}_t||^2_{\boldsymbol{H}_t^{-1}} + \mathcal{D}_{\psi_t}(\boldsymbol{w}^*, \boldsymbol{w}_t) - \mathcal{D}_{\psi_t}(\boldsymbol{w}^*, \boldsymbol{w}_{t+1}) + (\mathcal{D}_{\psi_t}(\boldsymbol{w}^*, \boldsymbol{w}_{t+1}) - \mathcal{D}_{\psi_t}(\boldsymbol{w}^*, \boldsymbol{z}_{t+1})) \\
&\leq_3 \frac{\eta^2}{2}||\boldsymbol{g}_t||^2_{\boldsymbol{H}_t^{-1}} + \mathcal{D}_{\psi_t}(\boldsymbol{w}^*, \boldsymbol{w}_t) - \mathcal{D}_{\psi_t}(\boldsymbol{w}^*, \boldsymbol{w}_{t+1}) + \xi_t||\boldsymbol{w}^*||_\infty \ ,
\end{aligned}
$$

where $\leq_1$ follows from the KKT optimality condition for (3), i.e. for any $\boldsymbol{w} \in \mathbb{R}^d$,

$$
\langle \eta \boldsymbol{g}_t + \boldsymbol{H}_t(\boldsymbol{z}_{t+1} - \boldsymbol{w}_t), \boldsymbol{w} - \boldsymbol{z}_{t+1} \rangle \geq 0 \ .
$$

In $\leq_2$, Fenchel-Yong inequality is used, and $\leq_3$ follows from

$$
\begin{aligned}
&\mathcal{D}_{\psi_t}(\boldsymbol{w}^*, \boldsymbol{w}_{t+1}) - \mathcal{D}_{\psi_t}(\boldsymbol{w}^*, \boldsymbol{z}_{t+1}) \\
&= \frac{1}{2}\left(||\boldsymbol{w}_{t+1}||^2_{\boldsymbol{H}_t} - ||\boldsymbol{z}_{t+1}||^2_{\boldsymbol{H}_t}\right) + \langle \boldsymbol{w}^*, \boldsymbol{H}_t(\boldsymbol{z}_{t+1} - \boldsymbol{w}_{t+1}) \rangle \\
&\leq \langle \boldsymbol{w}^*, \boldsymbol{H}_t(\boldsymbol{z}_{t+1} - \boldsymbol{w}_{t+1}) \rangle \leq \ ||\boldsymbol{w}^*||_\infty \sum_{i=1}^{d} H_{t,ii}|z_{t+1,i} - w_{t+1,i}| = \xi_t||\boldsymbol{w}^*||_\infty \ .
\end{aligned}
$$

Summing over $t = 1, 2, \cdots T$, we have that

$$
\begin{aligned}
\mathcal{R}^T_{\text{B-AMD}} &\leq \frac{\eta}{2}\sum_{t=1}^{T}||\boldsymbol{g}_t||^2_{\boldsymbol{H}_t^{-1}} + \frac{1}{\eta}||\boldsymbol{w}^*||_\infty \sum_{t=1}^{T}\xi_t + \frac{1}{\eta}\mathcal{D}_{\psi_1}(\boldsymbol{w}^*, \boldsymbol{w}_1) \\
&\quad + \frac{1}{\eta}\sum_{t=1}^{T-1}(\mathcal{D}_{\psi_{t+1}}(\boldsymbol{w}^*, \boldsymbol{w}_{t+1}) - \mathcal{D}_{\psi_t}(\boldsymbol{w}^*, \boldsymbol{w}_{t+1})) \\
&\leq \frac{\eta}{2}\sum_{t=1}^{T}||\boldsymbol{g}_t||^2_{\boldsymbol{H}_t^{-1}} + \frac{1}{\eta}||\boldsymbol{w}^*||_\infty \sum_{t=1}^{T}\xi_t + \frac{1}{2\eta}\max_{t\leq T}||\boldsymbol{w}^* - \boldsymbol{w}_t||^2_\infty \sum_{i=1}^{d}||\boldsymbol{g}_{1:T,i}||_2 \\
&\leq_1 \eta\sum_{i=1}^{d}||\boldsymbol{g}_{1:T,i}||_2 + \frac{1}{\eta}||\boldsymbol{w}^*||_\infty \sum_{t=1}^{T}\xi_t + \frac{1}{2\eta}\max_{t\leq T}||\boldsymbol{w}^* - \boldsymbol{w}_t||^2_\infty \sum_{i=1}^{d}||\boldsymbol{g}_{1:T,i}||_2 \ ,
\end{aligned}
$$

where the last inequality follows from Lemma 4 in [4].

Acknowledgment

The authors would like to acknowledge support for this project from the National Key R&D Program of China (2017YFB0702600, 2017YFB0702601), the National Natural Science Foundation of China (Nos. 61432008, 61503178) and the Natural Science Foundation of Jiangsu Province of China (BK20150587).

References

1. Brown, G., Pocock, A.C., Zhao, M., Luján, M.: Conditional likelihood maximisation: A unifying framework for information theoretic feature selection. J. Mach. Learn. Res. **13**, 27–66 (2012)

2. Condat, L.: Fast projection onto the simplex and the ℓ_1 ball. Math. Program. **158**(1-2), 575–585 (2016)
3. Duchi, J., Singer, Y.: Efficient online and batch learning using forward backward splitting. J. Mach. Learn. Res. **10**, 2899–2934 (2009)
4. Duchi, J.C., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. **12**, 2121–2159 (2011)
5. Duchi, J.C., Shalev-Shwartz, S., Singer, Y., Chandra, T.: Efficient projections onto the ℓ_1-ball for learning in high dimensions. In: Proc. ICML. pp. 272–279 (2008)
6. Duchi, J.C., Shalev-Shwartz, S., Singer, Y., Tewari, A.: Composite objective mirror descent. In: Proc. COLT. pp. 14–26 (2010)
7. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. **3**, 1157–1182 (2003)
8. Langford, J., Li, L., Zhang, T.: Sparse online learning via truncated gradient. J. Mach. Learn. Res. **10**, 777–801 (2009)
9. Rao, N.S., Nowak, R.D., Cox, C.R., Rogers, T.T.: Classification with the sparse group lasso. IEEE Trans. Signal Process. **64**(2), 448–463 (2016)
10. Shalev-Shwartz, S., Srebro, N., Zhang, T.: Trading accuracy for sparsity in optimization problems with sparsity constraints. SIAM J. Optim. **20**(6), 2807–2832 (2010)
11. Shalev-Shwartz, S., Tewari, A.: Stochastic methods for ℓ_1-regularized loss minimization. J. Mach. Learn. Res. **12**, 1865–1892 (2011)
12. Song, L., Smola, A.J., Gretton, A., Bedo, J., Borgwardt, K.M.: Feature selection via dependence maximization. J. Mach. Learn. Res. **13**, 1393–1434 (2012)
13. Tan, M., Tsang, I.W., Wang, L.: Towards ultrahigh dimensional feature selection for big data. J. Mach. Learn. Res. **15**(1), 1371–1429 (2014)
14. Tan, M., Wang, L., Tsang, I.W.: Learning sparse SVM for feature selection on very high dimensional datasets. In: Proc. ICML. pp. 1047–1054 (2010)
15. Wang, D., Wu, P., Zhao, P., Wu, Y., Miao, C., Hoi, S.C.H.: High-dimensional data stream classification via sparse online learning. In: Proc. ICDM. pp. 1007–1012 (2014)
16. Wang, J., Zhao, P., Hoi, S.C., Jin, R.: Online feature selection and its applications. IEEE Trans. Knowl. Data Eng. **26**(3), 698–710 (2014)
17. Wang, J., Wang, M., Li, P., Liu, L., Zhao, Z., Hu, X., Wu, X.: Online feature selection with group structure analysis. IEEE Trans. Knowl. Data Eng. **27**(11), 3029–3041 (2015)
18. Woznica, A., Nguyen, P., Kalousis, A.: Model mining for robust feature selection. In: Proc. SIGKDD. pp. 913–921 (2012)
19. Wu, X., Yu, K., Ding, W., Wang, H., Zhu, X.: Online feature selection with streaming features. IEEE Trans. Pattern Anal. Mach. Intell. **35**(5), 1178–1192 (2013)
20. Wu, Y., Hoi, S.C.H., Mei, T., Yu, N.: Large-scale online feature selection for ultra-high dimensional sparse data. ACM Trans. Knowl. Discov. Data **11**(4), 48:1–48:22 (2017)
21. Xiao, L.: Dual averaging methods for regularized stochastic learning and online optimization. J. Mach. Learn. Res. **11**, 2543–2596 (2010)
22. Yu, K., Wu, X., Ding, W., Pei, J.: Scalable and accurate online feature selection for big data. ACM Trans. Knowl. Discov. Data **11**(2), 16:1–16:39 (2016)
23. Yu, L., Liu, H.: Efficient feature selection via analysis of relevance and redundancy. J. Mach. Learn. Res. **5**, 1205–1224 (2004)