Healing Truncation Bias : Self-weighted Truncation framework for Dual Averaging

Hidekazu Oiwa
Graduate School of Information Science and Technology, The University of Tokyo
Tokyo, Japan
hidekazu.oiwa@gmail.com

Shin Matsushima
Graduate School of Information Science and Technology, The University of Tokyo
Tokyo, Japan
masin@r.dl.itc.u-tokyo.ac.jp

Hiroshi Nakagawa
Information Technology Center
The University of Tokyo
Tokyo, Japan
nakagawa@dl.itc.u-tokyo.ac.jp

***Abstract*—We propose a new truncation framework for online supervised learning. Learning a compact predictive model in an online setting has recently attracted a great deal of attention. The combination of online learning with sparsity-inducing regularization enables faster learning with a smaller memory space than a conventional learning framework. However, a simple combination of these triggers the truncation of weights whose corresponding features rarely appear, even if these features are crucial for prediction. Furthermore, it is difficult to emphasize these features in advance while preserving the advantages of online learning. We develop an extensional truncation framework to Dual Averaging, which retains rarely occurring but informative features. Our proposed framework integrates information on all previous subgradients of the loss functions into a regularization term. Our enhancement of a conventional L_1-regularization accomplishes the automatic adjustment of each feature's truncations. This extension enables us to identify and retain rare but informative features without preprocessing. In addition, our framework achieves the same computational complexity and regret bound as standard Dual Averaging. Experiments demonstrated that our framework outperforms other sparse online learning algorithms.**

***Keywords*-Online Learning, Supervised Learning, Sparsity-inducing Regularization, Feature Selection, Sentiment Analysis**

I. Introduction

Online learning is a training method where a prediction and an update take place in a sequential setting each time a learner receives one datum. Online learning is beneficial for learning from large-scale data in terms of used memory space and computational complexity. If training data is very large, many batch algorithms cannot derive a global optimal solution within a reasonable amount of time because the computational cost is very high. When all instances cannot be simultaneously loaded into the main memory, optimization in batch learning requires some reformulation for exact solving [1]. On the other hand, many online learning algorithms update predictors with simple update formula based on only one new instance. Thus, online learning runs faster with a smaller memory space than batch learning in the result. Online learning algorithms are efficient especially in learning from a dataset where the dimension of instances or the number of instances is very large. Many algorithms of batch learning have been transformed into online ones for the ease of handling large-scale data.

L_1-regularization is a well-known regularization method of generating a compact predictive model. L_1-regularization eliminates parameters that are insignificant for prediction on the fly. As a result, L_1-regularization promotes the derivation of a compact predictive model. A compact model is able to reduce the computational time and required memory space for prediction because it deals with a smaller number of parameters. In addition, L_1-regularization has a generalization effect for preventing over-fitting to training data.

To combine online learning with L_1-regularization, some frameworks had recently been developed for efficiently solving large-scale learning problems without preprocessing, like TF-IDF [2]. One state-of-the-art framework, which is a combination of online learning with L_1-regularization, is Regularized Dual Averaging (RDA) [3]. RDA integrates a regularization term into the optimization problem of Dual Averaging (DA) [4] and enables to obtain a sparse solution by applying L_1-regularization. RDA solves a simple minimization problem that consists of the summation of all past subgradients of loss functions and the whole regularization term to search for the optimal parameters at each update. RDA outperforms other sparse online learning frameworks in terms of model compactness and precision [3].

Although online learning combined with L_1-regularization is indispensable for large-scale learning, conventional frameworks do not take into consideration the information on feature occurrence frequency. As a result, rare features are easily dropped from a predictive model, even though these features are crucial for prediction. The frequency of feature occurrences is not usually uniform in many tasks, such as natural language processing and pattern recognition. Furthermore, if there is a heterogeneity of the value range among features, these algorithms also tend to truncate parameters the features of which take values around 0.

We propose a new truncation framework to retain rare but informative features in an online setting. The key idea behind our proposed framework is to integrate the absolute values for the subgradients of loss functions into the regularization term. By applying this extension into RDA, our framework can dynamically weaken truncation effects in rare features.

We also analyze the theoretical aspects of the new truncation framework. As a result, the same computational complexity and the regret upper bound are derived as those for RDA. Finally, we evaluated how effective our framework was through several experiments. Experimental results revealed that our framework healed the truncation bias occurred in previous work and outperformed state-of-the-art methods while maintaining a similar compactness to other algorithms.

The composition of this paper is as follows.

- Section II: we introduce a conventional RDA framework, which introduces a regularization term into DA.
- Section III: we present a new truncation framework, which can be applied to RDA. We show some analyses for our framework and a closed-form solution.
- Section IV: we prove the regret bound for our algorithms to ensure the upper bound of objective functions.
- Section V: we compare our framework with other sparse online learning algorithms through several tasks.
- Section VI: we conclude the paper and refer to future work.

A. Related Work

First note that some extentional work for DA have been done. Shalev-Shwartz et al. [5] proposed a primal-dual framework. This primal-dual framework develops a new universal bound in optimization problems to search for the optimal hypothesis. Moreover, some algorithms for online learning have been proposed to obtain tighter upper bounds for convergence than other algorithms. DA is a special case of this primal-dual framework. Dekel et al. [6] proposed a mini-batch version of DA so that it could process instances in a distributed environment. They proved an asymptotically optimal regret bound for smooth convex loss functions and stochastic examples. Lee et al. [7] focused on characteristics where RDA could identify a low-dimensional manifold induced by a regularized term in a weight space. This observation promoted the development of a new DA-based algorithm for faster search for the optimal weight. As a state-of-the-art optimization framework for sparse online learning, many researchers had focused on and utilized DA.

Other than DA, a splitting method framework has also been proposed for sparse online learning. This framework consists of two steps at each round. In the first step, a predictor is updated to improve the accuracy of prediction by using the received instance. Optimization methods, such as subgradient method [8] or Stochastic Gradient Descent (SGD), have often been used in the first step. Then, regularization has been applied in the second step. Splitting method is a well known approach because it can integrate online learning with L_1-regularization while preserving the advantages of the two techniques. Carpenter [9] proposed a splitting approach that combined SGD with L_1-regularization. Then, Duchi and Singer [10] and Langford et al. [11] generalized splitting frameworks. Duchi and Singer's framework was called FOBOS [10]. These algorithms were guaranteed to asymptotically offer regret $O(\sqrt{T})$ under certain conditions.

Oiwa et al. [12] proposed an extention to FOBOS that integrated information on feature occurrence into a regularization term to solve the truncation problems. Specifically, subgradient method with frequency-aware truncation in FOBOS was called SGFT. SGFT achieved the same computational cost and the same convergence rate as FOBOS. The idea behind our framework, healing truncation bias, was inspired by Oiwa et al., however, the definition of introduced parameters is different. The treatment of step width is the most important different part between these two frameworks. This difference achieves our intention more precisely as described below.

Moreover, Duchi et al. [13] proposed a new update scheme called AdaGrad. AdaGrad incorporated the knowledge of instances observed in earlier iterations into a proximal function to emphasize rare features without preprocessing. However, AdaGrad could neither normalize each feature's value range nor reflect information on a just received datum because it controlled the feature importance only in a proximal function. Our proposed framework solved these problems as will be explained later.

Other than sparse online learning, some significant algorithms have been proposed in the field of online learning for classification. Perceptron [14], Passive-Aggressive [15], and Confidence-Weighted [16], [17] algorithms (CW) are well-known methods that are alternatives to subgradient-based approaches. CW algorithms are state-of-the-art in this field. CW introduced a Gaussian distribution into a weight vector to capture functionalities of rare features. When CW updated a learner, CW emphasized informative rare features according to confidence parameters updated on the fly. However, these algorithms did not generate sparse solutions. In addition, they were specialized to classification.

II. Regularized Dual Averaging (RDA)

First, let us introduce the notations we have used in this paper. Scalars are in lower-case italics, e.g., λ, and the absolute value of each scalar is $|\lambda|$. Vectors are in lower-case bold, such as $\mathbf{x}$, and the i-th entry of vector $\mathbf{x}$ is represented as $x^{(i)}$. Matrices are in upper-case bold, e.g., $\mathbf{X}$. $\|\mathbf{x}\|_p$ represents an L_p norm of vector $\mathbf{x}$. $\langle \mathbf{x}, \mathbf{y} \rangle$ denotes an inner product of two vectors $\mathbf{x}, \mathbf{y}$. Let $\mathbf{dom} f$ be the domain of function f. Function sign is defined as $\mathrm{sign}(\lambda) = \lambda/|\lambda|$ where λ is a scalar (If $\lambda = 0$, $\mathrm{sign}(\lambda) = 0$). Let $\mathrm{argmin}\, h$ be a unique point for minimizing function h. Let $\mathrm{Argmin}\, \Psi$ be the set of minimizing points of function Ψ. Table I summarizes the notations used here.

We focus on sparse online learning with a linear predictor in this work. At round t, the algorithm receives instance $\mathbf{x}_t \in X \subset \Re^d$ where X is an input space. Then, it applies current weight vector $\mathbf{w}_t \in W \subset \Re^d$ to make a prediction. Let W be a closed convex set. It then receives true value

Table I
NOTATION

a	scalar
$\mathbf{a}$	vector
$\mathbf{A}$	matrix
$a^{(i)}$	i-th entry of vector $\mathbf{a}$
$\|\lambda\|$	absolute value
$\|\mathbf{a}\|_p$	L_p norm
$\langle \mathbf{a}, \mathbf{b} \rangle$	inner product
$\mathbf{dom} f$	domain of function f
$\operatorname{argmin} h$	unique point for minimizing function h
$\operatorname{Argmin} \Psi$	set of minimizing points of function Ψ

$y_t \in Y \subset \Re$ and generates subgradient $\mathbf{g}_t \in \partial \ell_t(\mathbf{w}_t)$ using loss function $\ell_t(\cdot) : W \to \Re$.

Here, we deal with a linear predictive model. In a linear setting, we make a prediction through the value of the inner product of a current weight vector and an input vector. To evaluate the loss through a linear setting, we assume a loss function ℓ_t that exists in function $\hat{\ell}_t(\cdot) : \Re \to \Re$ where $\ell_t(\mathbf{w}_t) = \hat{\ell}_t(\langle \mathbf{w}_t, \mathbf{x}_t \rangle)$. Generally, a loss function is defined as non-decreasing for the distance between y_t and $\langle \mathbf{w}_t, \mathbf{x}_t \rangle$. Many well-known loss functions satisfy this condition, such as the squared loss function,

$$\ell_t(\mathbf{w}_t) = (y_t - \langle \mathbf{w}_t, \mathbf{x}_t \rangle)^2 \ , \tag{1}$$

and the hinge loss function,

$$\ell_t(\mathbf{w}_t) = [1 - y_t \langle \mathbf{w}_t, \mathbf{x}_t \rangle]_+ \ . \tag{2}$$

Then, we update a predictor for minimizing the value of loss functions. At the same time, we truncate parameters to make a predictive model compact. The objective of sparse online learning is to derive an optimal weight vector $\mathbf{w}$, which is a sparse vector and effective for prediction.

We introduce Regularized Dual Averaging (RDA) [3]. RDA updates a learner subject to (3) at each round.

$$\mathbf{w}_{t+1} = \operatorname*{argmin}_{\mathbf{w}} \sum_{\tau=1}^{t} \langle \mathbf{g}_\tau, \mathbf{w} \rangle + t\Psi(\mathbf{w}) + \beta_t h(\mathbf{w}) \tag{3}$$

The first term describes the inner product of the weight vector and the summation of all previous subgradients. This term works for the minimization of loss functions through the first-order approximation of these functions. The second term indicates regularized term $\Psi(\mathbf{w})$, which is a closed convex function. To obtain a sparse weight vector, it is possible that the L_1 norm of the weight vector is inserted into $\Psi(\mathbf{w})$, i.e., L_1-regularization. The third term describes a proximal function term. $\{\beta_t\}_{t \geq 1}$ is a positive and non-decreasing input sequence, which determines the convergence properties of RDA and $h(\mathbf{w})$ is a strongly convex auxiliary function that satisfies

$$\operatorname*{argmin}_{\mathbf{w}} h(\mathbf{w}) \in \operatorname*{Argmin}_{\mathbf{w}} \Psi(\mathbf{w}) \ . \tag{4}$$

A function $h : W \to \Re$ is called strongly convex if there is a constant $\sigma > 0$ where it satisfies inequality (5) with respect to a norm $\| \cdot \|$ for any $\mathbf{a}, \mathbf{b} \in W$ and for any $\alpha \in [0, 1]$.

$$h(\alpha \mathbf{a} + (1-\alpha)\mathbf{b}) \leq \ \alpha h(\mathbf{a}) + (1-\alpha) h(\mathbf{b}) - \frac{\sigma}{2} \alpha (1-\alpha) \|\mathbf{a} - \mathbf{b}\|^2 \ . \tag{5}$$

σ is called the convexity parameter. RDA is guaranteed to offer regret bound $O(\sqrt{T})$.

III. Self-weighted Truncation framework for Dual Averaging (STDA)

As noted in Section I, online learning that applies conventional L_1-regularization does not take into account the occurrence counts of features. When learning with a linear model in an online setting, conventional L_1-regularization tends to truncate some characteristic features because simple L_1-regularization applies the same penalty to all features independent of their value range or occurrence frequency. This property produces biases toward the truncation. We show two examples where conventional L_1-regularization causes problems that are common in sparse online learning.

A. Feature Occurrence Frequency Problem

Let us assume RDA with conventional L_1-regularization. Let $\Psi(\mathbf{w})$ be $\lambda \|\mathbf{w}\|_1$ where λ is a scalar, $h(\mathbf{w})$ be $\|\mathbf{w}\|_2^2$, and β_t be $1/\sqrt{t}$. We apply this algorithm to a dataset in which the occurrence rate of feature A is 1/100 and that of feature B is 1/2. Feature A inevitably becomes 0 unless 100λ is exceeded by the average of the A-th absolute values of subgradients whose A-th index is non-zero, i.e.,

$$\hat{g}^{(A)} \geq 100\lambda \ , \tag{6}$$

is satisfied where $\hat{g}^{(i)}$ is the average value of the i-th absolute values of subgradients only taking the i-th index to be non-zero. On the other hand, the weight of feature B does not always drop to 0 where

$$\hat{g}^{(B)} \geq 2\lambda \ . \tag{7}$$

If there is heterogeneity of occurrence frequency among features, the algorithm may fail to retain features that are rare but significant for prediction. Occurrence frequency is skewed in many tasks such as NLP and pattern recognition.

B. Value Range Problem

The disparity of value range of features also affects the truncation. Let us assume that there are two features: feature C is an arbitrary feature and feature D is one whose value is 1000 times larger than that of feature C. While learning from this dataset,

$$1000 |\bar{g}^{(C)}| = |\bar{g}^{(D)}| \ , \tag{8}$$

is always satisfied where $\bar{g}^{(i)}$ is the average value of the i-th index of all previous subgradients. Thus, the weight of

feature C is truncated faster than that of feature D by RDA, although they both have the same effect in prediction. That is,

$$|\bar{g}^{(C)}| \leq \lambda < |\bar{g}^{(D)}| \ , \tag{9}$$

might be satisfied. If $|\bar{g}^{(i)}| \leq \lambda$, the weight of feature i becomes 0. Thus, there is a possibility that feature C is truncated even though feature D is not. The converse phenomenon doesn't occur.

C. Self-weighted Truncation Modeling for DA

To overcome the above problems, we propose a self-weighted truncation framework — a framework that automatically tunes the intensities of truncation. Our proposed framework enables to retain rare but informative features in an online setting. Self-weighted truncation framework is inspired by frequency-aware approaches in splitting methods [12]. Self-weighted truncation modeling integrates the information from all previous subgradients into a regularization term. This extension enables to adjust for the effect of truncation taking each subgradient's occurrence frequency into consideration. Self-weighted Truncated Dual Averaging (STDA) defines the regularization term as:

$$\Psi_t(\mathbf{w}) = \lambda \|\mathbf{R}_{t,p}\mathbf{w}\|_1 \ , \tag{10}$$

where

$$\mathbf{R}_{t,p} = \begin{pmatrix} r_{t,p}^{(1)} & 0 & \dots & 0 \\ 0 & r_{t,p}^{(2)} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & r_{t,p}^{(d)} \end{pmatrix} \ s.t. \ r_{t,p}^{(i)} = \sqrt[p]{\sum_{\tau=1}^{t} \left| g_\tau^{(i)} \right|^p} \ .$$

$r_{t,p}^{(i)}$ is the value of L_p norm of a vector that consists of feature i's all previous subgradients. $\mathbf{R}_{t,p}$ is a matrix consisting of $r_{t,p}^{(i)}$ of all features in a diagonal component. λ is a regularization parameter to control the trade-off between loss-minimization and regularization.

We can place a wide variety of values into p to adjust the difference in truncation intensity. We give a simple example to demonstrate how parameter p influences self-weighted parameters $r_{t,p}^{(i)}$. Let us assume that the components of all subgradients are limited to either 0 or 1. To represent the relationship between the value of $r_{t,p}^{(i)}$ and the occurrence count of i-th index of subgradients in this setting, we show an illustrative example in Figure 1. The horizontal plot shows the number of occurrences in ascending order. The vertical plot indicates the value of $r_{t,p}^{(i)}$. Figure 1 indicates that the smaller the value of p is, the slower a rare feature is truncated. Note that conventional L_1-regularization can be regarded as the algorithm of $r_{t,p}^{(i)} = 1$ for all t, i.

In the remainder of this section, we fix parameter p and omit it, such as that from $\mathbf{R}_{t,p}$ to $\mathbf{R}_t$ and that from $r_{t,p}^{(i)}$ to $r_t^{(i)}$ to simplify our explanation.

By using definition $\Psi_t(\mathbf{w})$ above, we modify the optimization problem as:

$$\mathbf{w}_{t+1} = \operatorname*{argmin}_{\mathbf{w}} \sum_{\tau=1}^{t} \left(\langle \mathbf{g}_\tau, \mathbf{w} \rangle + \Psi_\tau(\mathbf{w}) \right) + \beta_t h(\mathbf{w}) \ . \tag{11}$$

From equation (10), when the value of $r_t^{(i)}$ is large, the algorithm tends to truncate the weight $w_t^{(i)}$ whose corresponding features frequently appear. If a feature is rare where the number of non-zero subgradient values at this feature is small, $r_t^{(i)}$ has a high probability of being a small value. Thus, we can adjust the value of truncation according to the subgradient information.

D. Feature Occurrence vs. Subgradient Occurrence

Truncation tuning based on the feature occurrence frequency could solve the problems described above. However, we adjust truncation intensity in accordance not with the feature occurrence frequency but with the occurrence frequency of ***subgradients***. This is because a weight vector is updated as proportional to subgradient counts rather than feature occurrence counts. Thus, regularization should be performed along with subgradient occurrence counts.

We give one example that explains why a subgradient is better than a feature. Let us assume that the dataset contains a feature that frequently occurs. In this dataset, RDA can obtain an appropriate weight value for this feature just after we receive the first few data. Even after obtaining a good weight for that feature, RDA receives many data containing the feature. Thus, the feature occurrence count continues to increase. In this case, a self-weighted parameter works to drop the weight even if the optimal weight has already been obtained. On the other hand, in the case of truncation based on a subgradient, a self-weighted parameter moves slowly if the weight is already sufficiently learned. This is because the values of the corresponding subgradients become small. For these reasons, a subgradient works better as the information for a self-weighted truncation framework than a feature. We verify the desirability of our modeling through the experiments.

E. Closed-form Solution to STDA

We derive the closed-form update formula for weight vector $\mathbf{w}_t$ in the case of $h(\mathbf{w}) = \|\mathbf{w}\|_2^2/2$. Our derivation is similar to that by Xiao [3], except for the definition of $\Psi_t(\mathbf{w})$.

Let $\mathbf{r}_t$ be a vector that consists of self-weighted parameter $r_t^{(i)}$ at each component and let $\bar{\mathbf{r}}_t$ be the average of all previous self-weighted parameters $\mathbf{r}_\tau$ until t. Let $\bar{\mathbf{g}}_t$ be the average of all previous subgradients $\mathbf{g}_\tau$ until round t. Moreover, we define $\mathbf{u}_t$, each element of which follows equation (12), to simplify our explanation.

$$u_t^{(i)} = \sum_{\tau=1}^{t} |g_\tau^{(i)}|^p \tag{12}$$

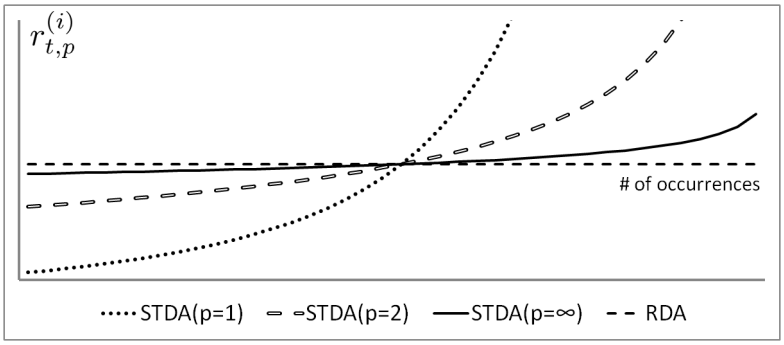

Figure 1. Comparison of self-weighted parameter $r_{t,p}^{(i)}$ against parameter p

First, the algorithm updates parameter $\mathbf{u}_t$ as:

$$u_t^{(i)} = u_{t-1}^{(i)} + |g_t^{(i)}|^p \ . \tag{13}$$

Then, we calculate self-weighted parameters $\mathbf{r}_t$ and $\bar{\mathbf{r}}_t$ as:

$$r_t^{(i)} = \sqrt[p]{u_t^{(i)}} \ . \tag{14}$$

$$\bar{\mathbf{r}}_t = \frac{t-1}{t}\bar{\mathbf{r}}_{t-1} + \frac{1}{t}\mathbf{r}_t \ . \tag{15}$$

We calculate new weight vector $\mathbf{w}_{t+1}$ following equation (11). First note that the optimization problem (11) can be decomposed into d independent coordinates of $\mathbf{w}_t$ as

$$w_{t+1}^{(i)} = \operatorname*{argmin}_w \left(t\bar{g}_t^{(i)} w + t\lambda |\bar{r}_t^{(i)} w| + \frac{\beta_t}{2} w^2 \right) \ , \tag{16}$$

where coefficient $\lambda > 0$ can be arbitrary. From the definition of $r_t^{(i)}$, clearly $\bar{r}_t^{(i)} \geq 0$. Thus, the optimal solution to formula (16) is subject to

$$\bar{g}_t^{(i)} + \lambda \bar{r}_t^{(i)} \xi^{(i)} + \frac{\beta_t}{t} w_{t+1}^{(i)} = 0 \ , \tag{17}$$

where $\xi^{(i)}$ is a subgradient of $|w_{t+1}^{(i)}|$. The differential of $|w|$ are 1 if $w > 0$, -1 if $w < 0$, or $\{\xi \in R | -1 \leq \xi \leq 1\}$ if $w = 0$. Therefore, we can solve the optimization problem as:

- If $|\bar{g}_t^{(i)}| \leq \lambda \bar{r}_t^{(i)}$, we set $w_{t+1}^{(i)} = 0$ and $\xi^{(i)} = -\bar{g}_t^{(i)} / \lambda \bar{r}_t^{(i)}$. When $w_{t+1}^{(i)} \neq 0$, equation (17) cannot be satisfied.
- If $\bar{g}_t^{(i)} > \lambda \bar{r}_t^{(i)} > 0$, we must set $w_{t+1}^{(i)} < 0$ and $\xi^{(i)} = -1$.
- If $\bar{g}_t^{(i)} < -\lambda \bar{r}_t^{(i)} < 0$, we must set $w_{t+1}^{(i)} > 0$ and $\xi^{(i)} = 1$.

We finally obtain a closed-form solution as:

$$w_{t+1}^{(i)} = \begin{cases} 0 & |\bar{g}_t^{(i)}| \leq \lambda \bar{r}_t^{(i)} \\ -\dfrac{t}{\beta_t} \epsilon_t^{(i)} & \text{otherwise} \end{cases} \ , \tag{18}$$

where we define $\bar{g}_t^{(i)} - \mathrm{sign}(\bar{g}_t^{(i)}) \lambda \bar{r}_t^{(i)}$ as $\epsilon_t^{(i)}$. We summarize the algorithm for STDA in Algorithm 1.

Algorithm 1 Self-weighted Truncated Dual Averaging (STDA)

Require:

$\{\beta_t\}_{t \geq 1}$ is a positive and non-decreasing sequence.
$\mathbf{w}_1 = \mathbf{0}$, $\mathbf{u}_0 = \mathbf{0}$, $\bar{\mathbf{g}}_0 = \mathbf{0}$ and $\bar{\mathbf{r}}_0 = \mathbf{0}$.

Algorithm:

for $t = 1, 2, \ldots$ **do**
  Given loss function ℓ_t, and compute subgradient $\mathbf{g}_t \in \partial \ell_t(\mathbf{w}_t)$.
  Update the average of all previous subgradients $\bar{\mathbf{g}}_t$ as $\bar{\mathbf{g}}_t = (t-1)\bar{\mathbf{g}}_{t-1}/t + \mathbf{g}_t/t$.
  Calculate self-weighted parameters $\mathbf{r}_t$ by eq. (13) and eq. (14).
  Derive new parameters $\bar{\mathbf{r}}_t$ by eq. (15).
  Update weight vector by eq. (18).
end for

The existence of parameters $\bar{r}_t^{(i)}$ makes a difference between the proposed formula and the update formula in RDA. From the update formula in STDA, we can see that parameters $\bar{r}_t^{(i)}$, which is an average value of the sequence $\{r_t^{(i)}\}$, tune the intensity of truncation. The bigger the value of $\bar{r}_t^{(i)}$, the smaller the value of $\epsilon_t^{(i)}$, that is, the stronger the intensity of truncation. The value of $\epsilon_t^{(i)}$ becomes big when that weight is updated in a certain direction. If an update direction fluctuates with respect to a weight, $\bar{r}_t^{(i)}$ becomes big while the absolute value of $\bar{r}_t^{(i)}$ is small. Thus, STDA tends to retain rare features that have a strong polarity.

Note that STDA can compute updates in lazy form for faster calculation. We only use parameters whose features occur in a received datum when it comes to prediction. Thus, the evaluations of eqs. (14), (15), and (18) can be postponed until corresponding features occur. When we run this algorithm in lazy update form, the computational cost becomes O(the number of occurring features) at each round.

IV. Regret Analysis of STDA

Even if an algorithm receives any instance or convex loss function sequence, the algorithm's regret bound is guaranteed by $o(T)$; then, we can say that weight vector $\mathbf{w}_t$ converges to static optimal vector $\mathbf{w}^*$ in convex set W. In this paper, regret is measured as:

$$R_{\ell+\Psi}(T) = \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \inf_{\mathbf{w}} \sum_{t=1}^{T} f_t(\mathbf{w}) \ , \tag{19}$$

where $f_t(\mathbf{w}) = \ell_t(\mathbf{w}) + \Psi_t(\mathbf{w})$.

We assume the sequence of subgradients $\{\mathbf{g}_t\}$ is bounded by constant G, i.e.,

$$\|\mathbf{g}_t\|_* \leq G \ , \tag{20}$$

where $\|\mathbf{g}\|_* = \max_{\|\mathbf{w}\|\leq 1}\langle \mathbf{g}, \mathbf{w}\rangle$ is a dual norm in space W^*, which is the vector space of all linear functions on W endowed with norm $\|\mathbf{w}\|$.

First, we bound $r_t^{(i)}$ using a scalar V so that we can ensure $\mathbf{dom}\Psi_t$ does not go to infinity. We redefine $\mathbf{r}_t$ as:

$$r_t^{(i)} = \min\left(V, \sqrt[p]{\sum_{\tau=1}^{t}\left|g_\tau^{(i)}\right|^p}\right) \ , \tag{21}$$

i.e., we set the upper bound for $r_t^{(i)}$ to V. From this definition, if we assume formula (20) is satisfied, and let D be a positive scalar value, we can derive the following inequality for any $t \geq 1$ and any $\mathbf{w} \in \{\mathbf{v} \in \mathbf{dom}\Psi_t | h(\mathbf{v}) \leq D^2\}$,

$$\frac{1}{t}\sum_{\tau=1}^{t}\Psi_\tau(\mathbf{w}) \leq V\|\mathbf{w}\|_1 \ . \tag{22}$$

Next, we prove the regret bound for our algorithms. While our regret bound is similar to that in Xiao's analysis [3], it is not the same because we must consider the effect of $\mathbf{R}_t$.

We can prove Theorem 1 to bound the regret.

Theorem 1: Let sequences $\{\mathbf{w}_t\}_{t\geq 1}$ and $\{\mathbf{g}_t\}_{t\geq 1}$ be generated by Algorithm 1. Furthermore, let us assume that conditions (20) and (21) are satisfied. Then, we define $\beta_t = \gamma\sqrt{t}$ where γ is a constant and D is a positive constant. We also assume optimal $\mathbf{w}$ is restricted in the set of $\{\mathbf{v} \in \mathbf{dom}\Psi_T | h(\mathbf{v}) \leq D^2\}$. In this case, for any $T \geq 1$ we have

$$R_{\ell+\Psi}(T) \leq \left(\gamma D^2 + \frac{G^2}{\gamma}\right)\sqrt{T} \ . \tag{23}$$

The proof of Theorem 1 is in the Appendix.

V. Experiments

We evaluated STDA using two-type classification tasks.

First, we used sentiment classification tasks [18] for reviews of Amazon.com goods. In these tasks, algorithms try to classify whether a positive or negative opinion was noted from a review text. We selected four categories in these datasets: books, dvd, electronics, and kitchen. Feature vector consists of unigram and bigram features.

Second, we used the 20 Newsgroups dataset (news20) [19]. The news20 is a news categorization task where algorithms try to predict to what category each news article will be assigned. This dataset consists of about 20,000 news articles. Each article is assigned to one of 20 predetermined categories. We used two subsets of news20: ob-2-1 and sb-2-1 [1]. The number of categories and the closeness between categories differed in each subset. The 'o' indicates 'overlap' and 's' denotes 'separated' for the first letter of each subset name. Classifying categories correctly is more difficult with an 'overlap' dataset.

We have provided the specifications for each dataset in Table II, including the number of features, instances, categories, and the types of categories.

A. Experimental Settings

We examined STDA, RDA, FOBOS [10], and SGFT [12] to compare the error and sparseness rates. The hinge loss function (2) was used as loss functions in these experiments.

The parameter setting in STDA is as follows. We set $h(\mathbf{w}) = 1/2\|\mathbf{w}\|_2^2$ and $\beta_t = \sqrt{t}$. It should be noted that the difference of β_t had an insignificant effect on the results as long as it satisfied $\beta_t \propto \sqrt{t}$. As an upper bound of self-weighted parameter, we set $V = 10^8$ to satisfy the regret bound restriction. The value of $r_t^{(i)}$ had never attained 10^8; thus, the value of V did not influence the result.

In RDA, the settings of $h(\cdot)$ and β_t were the same as those in STDA. In FOBOS and SGFT, the sequence of step size was set to $\eta_t = \eta_{t+1/2} = 1/\sqrt{t}$ to satisfy the regret bound restriction.

We executed 10-fold cross-validation by adjusting parameter λ to derive a highly predictable sparse weight vector. First, we divide each dataset into 10 subsets to search the appropriate parameter for achieving a low error rate with a sparse weight vector by using 9 subsets, called a training set. After tuning λ, the algorithms learn parameters from the training set through 20 iterations, which is to say, we run through all training examples 20 times. Then, we evaluate the error and sparseness rate using the remaining subset. Sparseness rate is calculated by (the number of zero components in the final weight vector) / (the number of components in the weight). This process recurs until all combinations are tried.

B. Experimental Results in STDA

We show the functionality of parameter p and the validity of our framework based on a subgradient. We evaluated the performance of STDA when we changed the parameter p and the definition of self-weighted parameters $r_{t,p}^{(i)}$. In the

[1] http://mlg.ucd.ie/datasets/20ng.html

Table II
DATASET SPECIFICATIONS

	# of instances	# of features	# of categories	type of categories
books	4,465	332,440	2	positive / negative
dvd	3,586	282,900	2	positive / negative
electronics	5,681	235,796	2	positive / negative
kitchen	5,945	205,665	2	positive / negative
ob-2-1	1,000	5,942	2	graphics / space
sb-2-1	1,000	6,276	2	christian / windows

case of modeling based on a feature occurrence frequency, we define $r_{t,p}^{(i)}$ as follows:

$$r_{t,p}^{(i)} = \sqrt[p]{\sum_{\tau=1}^{t} \left| x_{\tau}^{(i)} \right|^p}. \quad (24)$$

From STDA, we evaluated the algorithms of $p = 1, 2, \infty$ for each self-weighted parameter setting. Table III shows the experimental results.

From the results, we can see that STDA $p = 2$ and $p = \infty$ based on a subgradient outperformed other algorithms in terms of both sparseness and precision in most tasks. These results support the discussion in Section III-D. There are no significant differences of results between $p = 2$ and $p = \infty$.

We note that the results are very unstable in STDA $p = 1$ because it is difficult to derive a well-predictable sparse weight vector. In most cases, they cannot obtain a sparse weight vector except for 100%, i.e., a zero vector. The reason for this is that the incremental value of $r_t^{(i)}$ added at round t is the same as $g_t^{(i)}$ or $x_t^{(i)}$. Once $\bar{r}_t^{(i)}$ is larger than $\bar{g}_t^{(i)}$, i-th weight is clipped to 0 for any round t. It is very hard to control hyper-parameter λ and obtain a sparse weight vector that is also useful for prediction. When $p > 1$, this phenomenon is unlikely to occur because the incremental value of $r_t^{(i)}$ added at round t decreases.

C. Experimental Results compared with those from Previous Work

Table IV shows the experimental results for STDA $p = \infty$, RDA, FOBOS, and SGFT. In SGFT, we set $p = 2$ when SGFT achieved the best performance as in [12].

When comparing STDA with RDA, STDA outperforms RDA in terms of both error and sparseness rates in four out of six datasets. In the other two datasets, the figures of STDA $p = \infty$ is not below those of RDA at all aspects. From these experimental results, we note that self-weighted truncation framework could improve prediction accuracy by retaining rare but informative features. At the same time, we can attain the same sparseness rate because STDA can truncate unimportant features for prediction.

For finding out the functionality of our extensions, we show that STDA and RDA obtain what discriminative features are important in Table V. In this experiment, we used books dataset. Features are listed where one algorithm has determined that a feature is important while the other algorithm has determined that it is unimportant for prediction. These results indicate that STDA could retain rare features, e.g., *smearing* and *some interesting*, while in RDA these features were overly truncated. On the other hand, we can see that frequently occurring but not good predictive terms, such as *his* and *more*, are well truncated in STDA.

Table V
SAMPLES OF IMPORTANT FEATURES IN EACH ALGORITHM. OCCURRENCE COUNTS IN TRAINING DATASET ARE DENOTED IN PARENTHESES.

STDA ($p = \infty$)	RDA
some interesting (117)	his (1491)
a constructive (101)	more (877)
be successful (64)	time (1161)
was blatantly (29)	almost (376)
smearing (30)	say (2407)

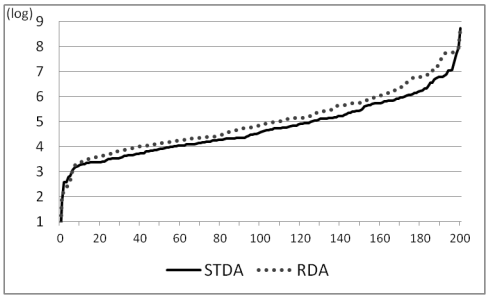

Figure 2. Comparison between feature occurrence frequencies in the predictive models of STDA and RDA

Additionally, in Figure 2, we show frequency statistics of features that each algorithm retains. Features are sorted by the occurrence frequency in ascending order from left to right for each result. The vertical plot indicates the occurrence frequency of each feature. The scale is logarithmic. Figure 2 indicates that STDA obtains rarer features than RDA. This is verified by the fact that the line of STDA is below that of RDA in most parts of the figure.

When comparing STDA with FOBOS, STDA largely outperformed FOBOS in terms of precision in all datasets. This indicates that the RDA learning framework had a significant advantage over the FOBOS learning framework. In addition, we can see that SGFT cannot obtain the highly predictable solution when the sparse rate is above a threshold. Since the step width in SGFT enlarges self-weighted parameters

Table III
EXPERIMENTAL RESULTS AMONG STDA FAMILY : THE RATES OF ERROR AND SPARSENESS (ITERATIONS : 20). THE SPARSENESS RATE ARE IN PARENTHESES. THE BEST FIGURES AMONG ALL ALGORITHMS EXCEPT STDA ($p = 1$) FOR EACH DATASET ARE WRITTEN IN **BOLD**.

	STDA ($p = 1$) Subgradient	STDA ($p = 2$) Subgradient	STDA ($p = \infty$) Subgradient	STDA ($p = 1$) Feature Frequency	STDA ($p = 2$) Feature Frequency	STDA ($p = \infty$) Feature Frequency
books	13.65	**13.12**	13.50	12.86	13.44	13.64
	(30.34)	(87.73)	(**91.13**)	(32.28)	(86.77)	(91.09)
dvd	12.66	**13.25**	13.69	12.88	14.64	13.86
	(31.31)	(85.16)	(**94.74**)	(32.36)	(89.23)	(89.12)
electronics	9.86	**10.51**	10.70	10.08	10.93	10.51
	(31.98)	(**95.89**)	(88.93)	(39.66)	(87.44)	(88.56)
kitchen	7.89	**8.98**	9.05	7.96	9.54	9.12
	(35.08)	(95.90)	(**97.36**)	(40.51)	(89.40)	(88.34)
ob-2-1	1.50	**2.80**	3.80	4.90	5.40	6.20
	(54.39)	(80.74)	(**82.30**)	(78.28)	(78.44)	(75.01)
sb-2-1	0.80	1.40	**1.20**	9.20	2.60	2.70
	(60.25)	(86.99)	(**93.69**)	(78.92)	(87.80)	(87.87)

Table IV
EXPERIMENTAL RESULTS COMPARED TO PREVIOUS WORK : THE RATE OF ERROR AND SPARSENESS (ITERATIONS : 20). THE SPARSENESS RATE ARE IN PARENTHESES. THE LOWEST ERROR RATES AND THE HIGHEST SPARSENESS RATES FOR EACH DATASET ARE WRITTEN IN **BOLD**.

	STDA ($p = \infty$)	RDA	FOBOS	SGFT(p=2)
books	**13.50**	14.00	15.72	45.52
	(**91.13**)	(88.69)	(80.39)	(84.19)
dvd	**13.69**	14.42	18.13	37.56
	(**94.74**)	(93.23)	(91.60)	(82.91)
electronics	**10.70**	11.06	12.59	30.67
	(88.93)	(**91.93**)	(90.54)	(84.77)
kitchen	**9.05**	9.77	10.73	27.21
	(**97.36**)	(91.23)	(90.36)	(86.15)
ob-2-1	3.80	**3.10**	6.00	15.00
	(82.30)	(76.96)	(**83.05**)	(79.89)
sb-2-1	**1.20**	2.30	4.00	19.30
	(**93.69**)	(89.74)	(70.91)	(82.37)

included in the first few features more than expected, it over-truncates features even if they are very useful. These results backed up the experimental results obtained by Oiwa et al. [12] where RDA is more precise than frequency-aware truncated FOBOS in most tasks.

VI. CONCLUSION

We proposed self-weighted truncation framework for Dual Averaging to retain rare but informative features in an online setting. Our proposed truncation framework integrated information on all previous subgradients into a regularized term to adjust the truncation effect on the fly. We could solve the problem of conventional L_1-regularization in this way, where rare features were truncated on a priority basis even if these features were important for prediction. Furthermore, we proved the theoretical guarantees of STDA by deriving the computational cost and finding the upper regret bound. Finally, we evaluated the performance of our methods in experiments. The results revealed that the methods we proposed outperformed previous sparse online learning algorithms while preserving the sparseness. Moreover, we showed that STDA could retain rare but informative features.

One remaining issue is whether we can modify the algorithm to choose the optimal figure of parameter p in an online setting. We aim to investigate these questions and further extend our proposed methods.

ACKNOWLEDGEMENT

This work was supported by JSPS KAKENHI, Grant-in-Aid for JSPS Fellows for Hidekazu Oiwa and Shin Matsushima.

REFERENCES

[1] H.-F. Yu, C.-J. Hsieh, K.-W. Chang, and C.-J. Lin, "Large linear classification when data cannot fit in memory," in *KDD*. ACM, 2010, pp. 833–842.

[2] G. Salton and C. Buckley, *Term-weighting approaches in automatic text retrieval*. Morgan Kaufmann Publishers Inc., 1997, pp. 323–328.

[3] L. Xiao, "Dual averaging methods for regularized stochastic learning and online optimization," *Journal of Machine Learning Research*, vol. 11, pp. 2543–2596, 2010.

[4] Y. Nesterov, "Primal-dual subgradient methods for convex problems," *Mathematical Programming*, vol. 120, no. 1, pp. 221–259, 2009.

[5] S. Shalev-Shwartz and Y. Singer, "Convex repeated games and fenchel duality," in *Advances in NIPS*. MIT Press, 2007, pp. 1265–1272.

[6] O. Dekel, R. Gilad-Bachrach, O. Shamir, and L. Xiao, "Optimal distributed online prediction using mini-batches," *Journal of Machine Learning Research*, vol. 13, pp. 165–202, 2012.

[7] S. Lee and S. J. Wright, "Manifold identification of dual averaging methods for regularized stochastic online learning," in *Proc. of ICML*. Omnipress, 2011, pp. 1121–1128.

[8] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Athena Scientific, 1999.

[9] B. Carpenter, "Lazy sparse stochastic gradient descent for regularized multinomial logistic regression," 2008.

[10] J. Duchi and Y. Singer, "Efficient Online and Batch Learning Using Forward Backward Splitting," *Journal of Machine Learning Research*, vol. 10, pp. 2899–2934, 2009.

[11] J. Langford, L. Li, and T. Zhang, "Sparse online learning via truncated gradient," *Journal of Machine Learning Research*, vol. 10, pp. 777–801, 2009.

[12] H. Oiwa, S. Matsushima, and H. Nakagawa, "Frequency-aware truncated methods for sparse online learning," in *ECML/PKDD*, vol. 2. Springer-Verlag, 2011, pp. 533–548.

[13] J. C. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.

[14] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958.

[15] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online Passive-Aggressive Algorithms," *Journal of Machine Learning Research*, vol. 7, pp. 551–585, 2006.

[16] M. Dredze and K. Crammer, "Confidence-weighted linear classification," in *Proc. of ICML*. ACM, 2008, pp. 264–271.

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

[18] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification," in *Proc. of ACL*. Association for Computational Linguistics, 2007, pp. 440–447.

[19] K. Lang, "Newsweeder: Learning to filter netnews," in *Proc. of ICML*. Morgan Kaufmann, 1995, pp. 331–339.

APPENDIX : REGRET ANALYSIS OF STDA

We prove Theorem 1 in this Appendix. Our proof is written in line with Xiao's [3] proof because this proof is similar to that of Xiao [3]. The difference between these two proof procedures is the definition of $\Psi_t(\mathbf{w})$ and the effect of this change.

First, we define the summation of all previous subgradients as $\mathbf{s}_t$, i.e.,

$$\mathbf{s}_t = \sum_{\tau=1}^{t} \mathbf{g}_\tau = t\bar{\mathbf{g}}_t \ . \tag{25}$$

Then, let $\mathbf{w}_0$ be the minimizer of $h(\mathbf{w})$. From the assumption (4), we obtain

$$\mathbf{w}_0 = \operatorname*{argmin}_{\mathbf{w}} h(\mathbf{w}) \in \operatorname*{Argmin}_{\mathbf{w}} \Psi_0(\mathbf{w}) \ . \tag{26}$$

In addition, let $\{\beta_t\}_{t\geq 1}$ be a positive and non-decreasing sequence where $\beta_0 = \beta_1 > 0$.

Then, we define two conjugate-type functions for each $t \geq 0$:

$$U_t(\mathbf{s}) = \max_{\mathbf{w}\in F_D} \left\{ \langle \mathbf{s}, \mathbf{w} - \mathbf{w}_0 \rangle - \sum_{\tau=1}^{t} \Psi_\tau(\mathbf{w}) \right\} , \tag{27}$$

$$V_t(\mathbf{s}) = \max_{\mathbf{w}} \{ \langle \mathbf{s}, \mathbf{w} - \mathbf{w}_0 \rangle - \sum_{\tau=1}^{t} \Psi_\tau(\mathbf{w}) - \beta_t h(\mathbf{w}) \} \ , \tag{28}$$

where $F_D = \{\mathbf{w} \in \mathbf{dom}\Psi_t | h(\mathbf{w}) \leq D^2\}$ and D is a scalar with a positive value. Because $\beta_t > 0$, function $\sum_{\tau=1}^{t} \Psi_\tau(\mathbf{w}) + \beta_t h(\mathbf{w})$ is always strongly convex; therefore, the maximizer of equation (28) is unique and we have $\mathbf{dom}U_t = \mathbf{dom}V_t = W^*$ for all $t \geq 0$.

We can derive Lemma 1 in this setting.

Lemma 1: For any $\mathbf{s} \in W^*$ and $t \geq 0$, we have

$$U_t(\mathbf{s}) \leq V_t(\mathbf{s}) + \beta_t D^2 \ . \tag{29}$$

Proof: See the proof of Lemma 9 in Xiao [3]. The procedure of proof is exactly the same while the definitions of U_t and V_t are different from those in Xiao.

□

Let $\pi_t(\mathbf{s})$ denote the unique maximizer of $V_t(\mathbf{s})$, and then we have the following equation for each $t \geq 0$.

$$\mathbf{w}_{t+1} = \pi_t(-\mathbf{s}_t) \ . \tag{30}$$

Lemma 2: Function V_t is convex and differentiable, and its gradient is given as:

$$\nabla V_t(\mathbf{s}) = \pi_t(\mathbf{s}) - \mathbf{w}_0 \ . \tag{31}$$

Moreover, the gradient is Lipschitz continuous with constant $1/\beta_t$.

Proof: This lemma follows from the characteristics of convexity because function $\sum_{\tau=1}^{t} \Psi_\tau(\mathbf{w}) + \beta_t h(\mathbf{w})$ is strongly convex with convexity parameter β_t.

□

From Lemma 2, we can obtain the following inequality for any vector $\mathbf{a}, \mathbf{b}$ in W^*.

$$V_t(\mathbf{a}+\mathbf{b}) \leq V_t(\mathbf{a}) + \langle \mathbf{b}, \nabla V_t(\mathbf{a}) \rangle + \frac{1}{2\beta_t} \|\mathbf{b}\|_*^2 \ . \tag{32}$$

Lemma 3: For each $t \geq 1$, we have

$$\begin{aligned} &V_t(-\mathbf{s}_t) + \Psi_{t+1}(\mathbf{w}_{t+1}) \\ &\qquad \leq V_{t-1}(-\mathbf{s}_t) + (\beta_{t-1} - \beta_t) h(\mathbf{w}_{t+1}) \ . \end{aligned} \tag{33}$$

Proof:

$$
\begin{aligned}
&V_{t-1}(-\mathbf{s}_t) \\
&= \max_{\mathbf{w}} \left\{ \langle -\mathbf{s}_t, \mathbf{w} - \mathbf{w}_0 \rangle - \sum_{\tau=1}^{t-1} \Psi_\tau(\mathbf{w}) - \beta_{t-1} h(\mathbf{w}) \right\} \\
&\leq \langle -\mathbf{s}_t, \mathbf{w}_{t+1} - \mathbf{w}_0 \rangle - \sum_{\tau=1}^{t-1} \Psi_\tau(\mathbf{w}_{t+1}) - \beta_{t-1} h(\mathbf{w}_{t+1}) \\
&= \langle -\mathbf{s}_t, \mathbf{w}_{t+1} - \mathbf{w}_0 \rangle - \sum_{\tau=1}^{t} \Psi_\tau(\mathbf{w}_{t+1}) - \beta_t h(\mathbf{w}_{t+1}) \\
&\qquad + \Psi_t(\mathbf{w}_{t+1}) + (\beta_t - \beta_{t-1}) h(\mathbf{w}_{t+1}) \\
&\leq V_t(-\mathbf{s}_t) + \Psi_{t+1}(\mathbf{w}_{t+1}) + (\beta_t - \beta_{t-1}) h(\mathbf{w}_{t+1}) \ . \quad (34)
\end{aligned}
$$

From equation (30), we can use $\mathbf{w}_{t+1} = \pi_t(-\mathbf{s}_t)$. Also, from the definition of $\Psi_{t+1}(\mathbf{w})$, we can derive $\Psi_t(\mathbf{w}) \leq \Psi_{t+1}(\mathbf{w})$ for any $\mathbf{w}$. By using these properties, we can show that the last inequality is satisfied.

□

We assume that $h(\mathbf{w}_{t+1}) \geq 0$ and sequence $\{\beta_t\}_{t \geq 0}$ is non-decreasing. Thus, we can derive the following function.

$$
\forall t \geq 1 \quad V_t(-\mathbf{s}_t) + \Psi_{t+1}(\mathbf{w}_{t+1}) \leq V_{t-1}(-\mathbf{s}_t) \ . \quad (35)
$$

From these lemmas, we will prove Theorem 1. To measure the quality of the solutions, $\{\mathbf{w}_t\}_{t \geq 1}$, we define gap sequences $\{\delta_t\}_{t \geq 1}$ and use it for bounding regret $R_{\ell+r}(t)$. For each $t \geq 1$, the following inequality is satisfied as:

$$
\begin{aligned}
\delta_t &= \max_{\mathbf{w} \in F_D} \left\{ \sum_{\tau=1}^{t} (\langle \mathbf{g}_\tau, \mathbf{w}_\tau - \mathbf{w} \rangle + \Psi_\tau(\mathbf{w}_\tau)) - \sum_{\tau=1}^{t} \Psi_\tau(\mathbf{w}) \right\} \\
&\geq \max_{\mathbf{w} \in F_D} \{ \sum_{\tau=1}^{t} (f_\tau(\mathbf{w}_\tau) - f_\tau(\mathbf{w}) + \Psi_\tau(\mathbf{w}_\tau)) \\
&\qquad - \sum_{\tau=1}^{t} \Psi_\tau(\mathbf{w}) \} \\
&= \sum_{\tau=1}^{t} (f_\tau(\mathbf{w}_\tau) + \Psi_\tau(\mathbf{w}_\tau)) \\
&\qquad - \min_{\mathbf{w} \in F_D} \left\{ \sum_{\tau=1}^{t} (f_\tau(\mathbf{w}) + \Psi_\tau(\mathbf{w})) \right\} \\
&= R_{\ell+r}(t) \ . \quad (36)
\end{aligned}
$$

In addition, we can derive the upper bound for δ_t. To achieve this, we reformalize gap sequences as:

$$
\begin{aligned}
\delta_t &= \sum_{\tau=1}^{t} (\langle \mathbf{g}_\tau, \mathbf{w}_\tau - \mathbf{w}_0 \rangle + \Psi_\tau(\mathbf{w}_\tau)) \\
&\qquad + \max_{\mathbf{w} \in F_D} \{ \langle \mathbf{s}_t, \mathbf{w}_0 - \mathbf{w} \rangle - \sum_{\tau=1}^{t} \Psi_\tau(\mathbf{w}) \} \\
&= \sum_{\tau=1}^{t} (\langle \mathbf{g}_\tau, \mathbf{w}_\tau - \mathbf{w}_0 \rangle + \Psi_\tau(\mathbf{w}_\tau)) + U_t(-\mathbf{s}_t) \\
&\leq \sum_{\tau=1}^{t} (\langle \mathbf{g}_\tau, \mathbf{w}_\tau - \mathbf{w}_0 \rangle + \Psi_\tau(\mathbf{w}_\tau)) \\
&\qquad + V_t(-\mathbf{s}_t) + \beta_t D^2 \ . \quad (37)
\end{aligned}
$$

Next, we will discuss the upper bound for the right-hand side of inequality (37). For each $\tau \geq 2$,

$$
\begin{aligned}
&V_\tau(-\mathbf{s}_\tau) + \Psi_{\tau+1}(\mathbf{w}_{\tau+1}) \\
&\qquad \leq V_{\tau-1}(-\mathbf{s}_\tau) \\
&\qquad = V_{\tau-1}(-\mathbf{s}_{\tau-1} - \mathbf{g}_\tau) \\
&\qquad \leq V_{\tau-1}(-\mathbf{s}_{\tau-1}) \\
&\qquad\quad + \langle -\mathbf{g}_\tau, \nabla V_{\tau-1}(-\mathbf{s}_{\tau-1}) \rangle + \frac{\|\mathbf{g}_\tau\|_*^2}{2\beta_{\tau-1}} \\
&\qquad = V_{\tau-1}(-\mathbf{s}_{\tau-1}) \\
&\qquad\quad + \langle -\mathbf{g}_\tau, \mathbf{w}_\tau - \mathbf{w}_0 \rangle + \frac{\|\mathbf{g}_\tau\|_*^2}{2\beta_{\tau-1}} \ , \quad (38)
\end{aligned}
$$

where these four steps above used (35), (25), (32), and (31), respectively. In summary, we can derive

$$
\begin{aligned}
&\langle \mathbf{g}_\tau, \mathbf{w}_\tau - \mathbf{w}_0 \rangle + \Psi_{\tau+1}(\mathbf{w}_{\tau+1}) \\
&\qquad \leq V_{\tau-1}(-\mathbf{s}_{\tau-1}) - V_\tau(-\mathbf{s}_\tau) + \frac{\|\mathbf{g}_\tau\|_*^2}{2\beta_{\tau-1}} \ , \quad (39)
\end{aligned}
$$

where $\tau \geq 2$.

Summing up the above inequalities from $\tau = 2$ to t, and noting that $V_0(-\mathbf{s}_0) = V_0(\mathbf{0}) = 0$, we can derive

$$
\begin{aligned}
&\sum_{\tau=1}^{t} (\langle \mathbf{g}_\tau, \mathbf{w}_\tau - \mathbf{w}_0 \rangle + \Psi_{\tau+1}(\mathbf{w}_{\tau+1})) + V_t(-\mathbf{s}_t) \\
&\qquad \leq \frac{1}{2} \sum_{\tau=1}^{t} \frac{\|\mathbf{g}_\tau\|_*^2}{\beta_{\tau-1}} \ . \quad (40)
\end{aligned}
$$

From the definitions of $\mathbf{w}_0 = \mathbf{w}_1$ and $\Psi_t(\mathbf{w})$, note that $\Psi_{t+1}(\mathbf{w}_{t+1}) \geq \Psi_1(\mathbf{w}_1)$ for all $t \geq 1$. Therefore, we add non-positive term $\Psi_1(\mathbf{w}_1) - \Psi_{t+1}(\mathbf{w}_{t+1})$ to the left-hand side of inequality (40) and obtain the following inequality.

$$
\begin{aligned}
&\sum_{\tau=1}^{t} (\langle \mathbf{g}_\tau, \mathbf{w}_\tau - \mathbf{w}_0 \rangle + \Psi_\tau(\mathbf{w}_\tau)) + V_t(-\mathbf{s}_t) \\
&\qquad \leq \frac{1}{2} \sum_{\tau=1}^{t} \frac{\|\mathbf{g}_\tau\|_*^2}{\beta_{\tau-1}} \ . \quad (41)
\end{aligned}
$$

Combining (37) with (41), we can derive

$$
R_{\ell+r}(t) \leq \delta_t \leq \beta_t D^2 + \frac{1}{2} \sum_{\tau=1}^{t} \frac{\|\mathbf{g}_\tau\|_*^2}{\beta_{\tau-1}} \ . \quad (42)
$$

From the condition of $\|\mathbf{g}_\tau\|_* \leq G$ for all τ and $\beta_t = \gamma\sqrt{t}$, we can re-formalize (42) as:

$$
R_{\ell+r}(t) \leq \delta_t \leq \left(\gamma D^2 + \frac{G^2}{\gamma} \right) \sqrt{t} \ , \quad (43)
$$

where we use inequality

$$
\sum_{\tau=1}^{t-1} \frac{1}{\sqrt{\tau}} \leq 1 + \int_1^t \frac{1}{\sqrt{\tau}} d\tau = 2\sqrt{t} - 1 \ . \quad (44)
$$

□