Frequency-Aware Truncated Methods for Sparse Online Learning

Hidekazu Oiwa, Shin Matsushima, and Hiroshi Nakagawa

University of Tokyo,
7-3-1, Hongo, Bunkyo-ku, Tokyo, 113-0033, Japan
{oiwa,masin}@r.dl.itc.u-tokyo.ac.jp,
n3@dl.itc.u-tokyo.ac.jp

Abstract. Online supervised learning with L_1-regularization has gained attention recently because it generally requires less computational time and a smaller space of complexity than batch-type learning methods. However, a simple L_1-regularization method used in an online setting has the side effect that rare features tend to be truncated more than necessary. In fact, feature frequency is highly skewed in many applications. We developed a new family of L_1-regularization methods based on the previous updates for loss minimization in linear online learning settings. Our methods can identify and retain low-frequency occurrence but informative features at the same computational cost and convergence rate as previous works. Moreover, we combined our methods with a cumulative penalty model to derive more robust models over noisy data. We applied our methods to several datasets and empirically evaluated the performance of our algorithms. Experimental results showed that our frequency-aware truncated models improved the prediction accuracy.

Keywords: Online Learning, L_1-regularization, Sparse Learning, Convex Programming, Low-frequency Occurrence Features, Natural Language Processing.

1 Introduction

Online learning is a training method using a sequence of instances, and it executes a learning process on one piece of data at each round. When learning from a large quantity of data, many well-known batch-type algorithms cannot solve an optimization problem within a reasonable time because the computational cost is very high. In addition, all instances may not be loaded into the main memory simultaneously. An online learning framework calculates what components of the weight vector are to be updated and by how much, based on only one instance, resulting in use of much less memory space. In this aspect, online learning is competitive for training from large-scale datasets in which the instances are high dimensional or the number of instances is very large. Online learning has recently attracted much attention owing to these properties, and many algorithms have been transformed into online ones.

D. Gunopulos et al. (Eds.): ECML PKDD 2011, Part II, LNAI 6912, pp. 533–548, 2011.

L_1-regularization, Lasso, is regarded as a useful technique for large-scale data analysis. Normal L_1-regularization introduces the L_1 norm into optimization problems to penalize the weight vector. By applying L_1-regularization in algorithms, we can generate compact models to eliminate the features that do not contribute to the prediction. Compact models are also able to reduce the computational time and memory space used.

Carpenter[3] proposed an approach that combines online learning with L_1-regularization while maintaining the advantages of both techniques. Duchi et al.[7] and Langford et al.[9] generalized online learning with regularization and proved the regret bound. These methods consist of two steps. In the first step, the weight vector is updated to improve precision by reducing the value of the loss function using the received instance. Then, in the second step, regularization is applied to the weight vector. This learning scheme is the most famous in the field of sparse online learning. Therefore, many algorithms associated with the two-step scheme have been developed and analyzed, e.g., the lazy-update and cumulative-update forms. Thus, we focus on two-step scheme in this paper.

The widely known form of two-step algorithms is a subgradient method with L_1-regularization. This framework updates the weight vector in a loss minimization step according to the subgradient method, and then it truncates parameters using a normal L_1-regularized term. In this paper, we call this method SubGradient method with L_1-regularization (SG-L_1). Although SG-L_1 is an effective learning framework, this algorithm does not take into account feature frequency information. As a result, a set of rarely occurring features tends to be truncated to zero even if they are important or critical features. In many applications, such as natural language processing and pattern recognition tasks, the frequency of feature occurrence is not usually uniform. If there are value range differences among features, the truncated problem also occurs. However, these properties were not studied in detail in previous works.

Parts of infrequently occurring features are often informative for prediction. To capture these parts, pre-emphasizing methods have been developed, such as TF-IDF[14]. Another pre-processing method is to normalize the value range of each feature to standardize each feature. However, in an online learning setting, it is difficult to use these pre-processing methods while preserving the essence of online learning, i.e., to process samples sequentially.

In this paper, we propose simple truncated methods for retaining rarely occurring but informative features in an online setting. The key idea is to integrate the updating values in the loss minimization step into the L_1-regularization step. We call these methods frequency-aware truncated methods. In this way, we can decrease the truncation effects of rare features in an online setting. We also analyzed theoretical guarantees of our methods and derived the same computational cost and regret bound as for the SG-L_1 method. Furthermore, we investigated frequency-aware truncated methods with a cumulative penalty[15] to achieve robust solutions for noisy instances. We evaluated the effectiveness of our methods in experiments comparing our approach to other sparse online algorithms.

Table 1. Notation

a	scalar	$\lvert\lambda\rvert$	absolute value
$\mathbf{a}$	vector	$a^{(i)}$	i-th entry of vector $\mathbf{a}$
$\mathbf{A}$	matrix	$A^{(i,j)}$	(i,j)-th entry of matrix $\mathbf{A}$
$\lVert\mathbf{a}\rVert_p$	L_p norm	$\langle\mathbf{a},\mathbf{b}\rangle$	inner product

The outline of this paper is as follows. First, we introduce the problem setting and related works of sparse online learning in section 2. Next, we point out the disadvantages of previous works, namely, that low-frequency features are readily truncated, and propose frequency-aware truncated methods for solving rare-frequency feature truncated problems in section 3. Moreover, we analyze some properties of our methods and give theoretical guarantees. In section 4, we derive additional algorithms for combining our proposed methods with cumulative penalty models. In section 5, we evaluate the performance of our methods using classification tasks. From the experimental results, we discuss the properties of frequency-aware truncation and our contribution. We conclude the paper in section 6.

2 Linear Sparse Online Supervised Learning

2.1 Problem Setting

First, we introduce our notation to formally describe the problem setting. In this paper, scalars are lower-case italic letters, e.g., λ, and an absolute value of each scalar is $|\lambda|$. Vectors are lower-case bold letters, such as $\mathbf{x}$. Matrices are upper-case bold letters, e.g., $\mathbf{X}$. $\|\mathbf{x}\|_p$ represents L_p norm of vector $\mathbf{x}$, and $\langle\mathbf{x},\mathbf{y}\rangle$ denotes an inner product of two vectors $\mathbf{x},\mathbf{y}$. Table 1 summarizes the notation in this paper.

In this work, we develop a new family of truncated strategies for linear online learning. In the setting of standard linear sparse online learning, algorithms perform a sequential prediction and updating scheme. The objective is to derive the optimal weight vector $\mathbf{w} \in W \subset \mathbf{R}^d$, where W is a closed convex set. The updating process is conducted as follows.

1. Receive input data $\mathbf{x}_t \in X \subset \mathbf{R}^d$. Input data $\mathbf{x}_t$ is a feature vector taken from a d-dimensional closed convex set X.
2. Make a prediction based on an inner product of feature vector $\mathbf{x}_t$ and a weight vector $\mathbf{w}_t$. The predicted value is $\hat{y}_t = \langle\mathbf{w}_t,\mathbf{x}_t\rangle$.
3. Observe a true output y_t.
4. Update weight vector $\mathbf{w}_t$ to $\mathbf{w}_{t+1/2}$ using a loss function $\ell_t(\cdot)$.
5. Update $\mathbf{w}_{t+1/2}$ to $\mathbf{w}_{t+1}$ using an L_1-regularized term $r_t(\cdot)$.
6. Iterate steps 1 through 5 until no input data remains.

We update a weight vector according to loss function ℓ_t at step 4 and regularization term r_t at step 5. ℓ_t is a loss function of the form[1]

$$\ell_t(\mathbf{w}_t) : W \to \mathbf{R}_+ \ ,$$

where ℓ_t is convex with respect to weight vector $\mathbf{w}_t$. In this paper, we deal with linear online learning framework. Thus, we consider a loss function that exists a function $\hat{\ell}_t$, where

$$\ell_t(\mathbf{w}) = \hat{\ell}_t(\langle \mathbf{w}, \mathbf{x}_t \rangle) = \hat{\ell}_t(\hat{y}_t) \ , \qquad (1)$$

and loss function $\hat{\ell}_t$ is generally non-decreasing for the difference between $\hat{y}_t$ and y_t. We call a loss function that satisfies the restriction above a linear online learning problem.

In the setting of a standard linear online learning problem, a subgradient method(SG)[1][17] is often used for learning. In subgradient methods, the weight vector is updated as stated in formula (2):

$$\mathbf{w}_{t+1/2} = \mathbf{w}_t - \eta_t \mathbf{g}_t^f \quad s.t. \quad \mathbf{g}_t^f \in \partial f_t(\mathbf{w}_t) \ , \qquad (2)$$

where $\mathbf{g}_t^f$ is a subgradient[2] of f_t with respect to $\mathbf{w}_t$ and η_t is a learning rate. $\partial f(\mathbf{w}_t)$ is a set of all subgradients of f_t at $\mathbf{w}_t$. A subgradient method updates parameters sequentially to minimize f_t. It has been proved that the regret bound of the subgradient method is $O(\sqrt{T})$ when $\eta_t = 1/\sqrt{t}$, and consequently the regret bound per data vanishes as $T \to \infty$.

r_t is a regularized term of the form

$$r_t(\mathbf{w}_{t+1/2}) : W \to \mathbf{R}_+ \ ,$$

where r_t is convex in $\mathbf{w}_{t+1/2}$. Many algorithms use L_1 norm to penalize the weight vector in a sparsity-induced regularization.

$$r_t(\mathbf{w}) = r(\mathbf{w}) = \lambda \|\mathbf{w}\|_1 \ , \qquad (3)$$

where λ is a regularization parameter.

In SG-L_1, we penalize the weight vector according to formula (4) at step 5.

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}} \left\{ \|\mathbf{w} - \mathbf{w}_{t+1/2}\|_2^2/2 + \lambda \eta_{t+1/2} \|\mathbf{w}\|_1 \right\} \ , \qquad (4)$$

where $\eta_{t+1/2}$ is the second learning rate. In this step, we find a weight vector that is in between the previous weight $\mathbf{w}_{t+1/2}$ and a truncated one.

In this paper, we focus on step 5, the regularization step, and propose a new family of L_1-regularization methods.

[1] Squared loss function $\ell_t(\mathbf{w}_t) = (y_t - \langle \mathbf{w}_t, \mathbf{x}_t \rangle)^2$ and Hinge loss function $\ell_t(\mathbf{w}_t) = [1 - y_t \langle \mathbf{w}_t, \mathbf{x}_t \rangle]_+$ are usually used for ℓ_t.

[2] A subgradient of f at $\mathbf{x}$ is the vector $\mathbf{g} \in \mathbf{R}^n$ that satisfies

$$\forall \mathbf{y} \quad f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \mathbf{g}, \mathbf{y} - \mathbf{x} \rangle \ .$$

Even if f is non-differentiable, at least one subgradient exists when f is convex.

2.2 Related Works

As previously mentioned, SG-L_1 is the most common method in sparse online learning frameworks. Carpenter[3] split the update procedure into two steps and proposed a method to obtain a sparse solution in an online setting. In addition, FOBOS[7] and truncated gradient methods[9] generalized a splitting form method and analyzed the optimal step size and the regret bound of sparse online learning. These algorithms are guaranteed to asymptotically offer regret $O(\sqrt{T})$ in the restriction of the loss function and regularized term in section 2.1.

Furthermore, Nesterov[12] proposed the dual averaging method for online learning. This method updates the weight vector to solve the simple optimization problem that includes the average of all previous subgradients of the loss functions at each iteration. Xiao[16] developed the extension of the dual averaging method to include a regularization term, such as L_1 norm. The regularized dual averaging form (RDA) solves the minimization problem that takes into account both a regularized term and the average of all previous subgradients. A family of dual averaging methods ensures the $O(\sqrt{T})$ regret bound, but, this scheme also has the low-occurrence feature truncation problem because it applies the same penalty to all features. Duchi et al.[6] proposed a new family of subgradient methods as an alternative to previously used subgradient methods, named AdaGrad. AdaGrad incorporates the knowledge of the data observed in earlier iterations to emphasize the infrequently occurring instance in an online setting. However, when a feature occurs for the first time, AdaGrad cannot standardize it. This is because AdaGrad adjusts the update in a loss minimization step. In addition, AdaGrad also has the value range problem explained in section 3.

Useful methods have been proposed in the field of online learning for classification. For example, Perceptron[13], Passive-Aggressive[4], and Confidence-Weighted[5] algorithms are often used as alternatives to subgradient methods. In particular, Confidence-Weighted algorithms introduce a Gaussian distribution into a weight vector and update parameters using the covariance parameters of the weight vector in order to emphasize informative low-frequency features. However, Confidence-Weighted algorithms do not generate a sparse solution.

3 Frequency-Aware Truncated Methods

As noted in section 1, SG-L_1 and other sparse online algorithms apply the same penalty to all features independent of the previous update of each feature. In the linear online learning framework, algorithms update the weight of the feature that occurs in a piece of input data. As a result, the set of rarely occurring features tends to be sparse because the value range of these parameters must be larger than that of other features that are not truncated.

For example, we apply SG-L_1 to the dataset in which feature A's occurrence rate is 1/2 and feature B's rate is 1/100. In this case, feature B inevitably becomes 0 unless feature B's update satisfies

$$\eta_t |g_t^{\ell,(B)}| \geq \lambda \sum_{s=t}^{t+100} \eta_{s+1/2} \; .$$

In this paper, $g^{(i)}$ represents the i-th entry of the vector $\mathbf{g}$. On another front, the weight of feature A does not always drop to 0, where

$$\eta_t |g_t^{\ell,(A)}| \geq \lambda \sum_{s=t}^{t+1} \eta_{s+1/2} \ .$$

Therefore, in a normal sparse online learning framework, if the feature occurrence rate is non-uniform, we may fail to retain rarely occurring but important features. In many tasks, such as NLP and pattern recognition, a feature's occurrence rate is usually non-uniform.

In addition, each feature's value range affects the truncation of parameters. Assume that there are two features: one is an arbitrary feature and the other is one whose value is 1000 times larger than the first feature. If we learn from this dataset using normal sparse online learning, which applies the same penalty to all features, the weight of the first feature is truncated faster than that of the second feature, despite them both having almost the same effect for prediction.

We designed a family of frequency-aware truncated methods to capture low-frequency features and solve the value range problem in an online setting. A frequency-aware truncated method redefines step 5 in a sparse online learning alternative to normal L_1 norm by using each feature's previous update.

Let $\mathbf{u}_t$ be the t-th update at step 4[3]. In this case, we can write step 4 as

$$\mathbf{w}_{t+1/2} = \mathbf{w}_t + \mathbf{u}_t \ .$$

Then, the frequency-aware truncated method defines step 5 as follows:

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}} \left\{ \|\mathbf{w} - \mathbf{w}_{t+1/2}\|_2^2/2 + \lambda \eta_{t+1/2} \|\mathbf{H}_{t,p}\mathbf{w}\|_1 \right\} \ , \tag{5}$$

where

$$\mathbf{H}_{t,p} = \begin{pmatrix} h_{t,p}^{(1)} & 0 & \dots & 0 \\ 0 & h_{t,p}^{(2)} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & h_{t,p}^{(d)} \end{pmatrix} \quad s.t. \ \ h_{t,p}^{(j)} = \sqrt[p]{\sum_{s=1}^{t} \left| u_s^{(j)} \right|^p} \ .$$

$h_{t,p}^{(j)}$, which represents frequency-awareness, is the L_p norm of a vector that consists of feature j's update at step 4 in each iteration. $\mathbf{H}_{t,p}$ is a matrix consisting of $h_{t,p}^{(j)}$ of all features in a diagonal component. In this definition, we can derive the vector in which each component is $h_{t,p}^{(j)} w_t^{(j)}$, or $\mathbf{H}_{t,p}\mathbf{w}_t$. Thus, from equation (5), vector component $w_t^{(j)}$ tends to be truncated when the value of $h_{t,p}^{(j)}$ is large. If a feature is rarely occurring, the number of updates is also small; thus, $h_{t,p}^{(j)}$ also tends to have a small value. In addition, if the value of a feature C is 1000

[3] The update value of this form can be obtained in the setting of linear online learning (e.g., $-\eta_t \mathbf{g}_t^{\ell}$ in a subgradient method).

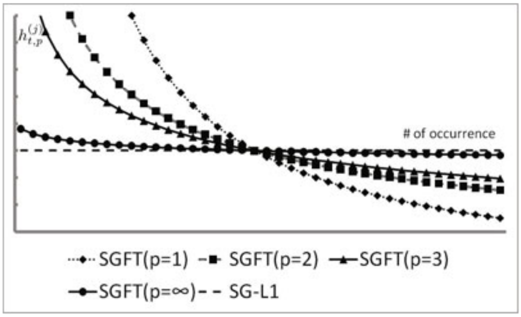

Fig. 1. Comparison of awareness parameter $h_{t,p}^{(j)}$ against parameter p

times larger than that of a feature D, then, $h_{t,p}^{(C)}$ is also 1000 times larger than $h_{t,p}^{(D)}$. Thus, we can keep the truncation of these two features the same in effect.

We can set a wide variety of numbers into parameter p to adjust the importance of rare features. To show how an awareness paramter $h_{t,p}^{(j)}$ is influenced by parameter p, we assume a simple example, in which gradient's value is limited to either 0 or 1, and represent the relationship between the value of $h_{t,p}^{(j)}$ and the count of feature j's occurrence. The example is illustrated in Fig. 1. A horizontal plot describes the count of occurrence in descending order and a vertical plot shows the value of $h_{t,p}^{(j)}$. It indicates that the smaller the value of p, the more slowly a rare feature is truncated. We note that a normal L_1 can be regarded as the algorithm of $h_{t,p}^{(j)} = 1$ for all t, j.

3.1 Subgradient Method with Frequency-Aware Truncation

In the following sections, we focus on the SubGradient method with Frequency-aware Truncation, which we call SGFT.

In SGFT, we can derive the update function as follows:

$$\begin{aligned} w_{t+1}^{(j)} &= sign\left(w_{t+1/2}^{(j)}\right)\left[\left|w_{t+1/2}^{(j)}\right| - \eta_{t+1/2} h_{t,p}^{(j)} \lambda\right]_+ \\ &= sign\left(w_t^{(j)} - \eta_t g_t^{\ell,(j)}\right)\left[\left|w_t^{(j)} - \eta_t g_t^{\ell,(j)}\right| - \eta_{t+1/2} h_{t,p}^{(j)} \lambda\right]_+ . \end{aligned} \quad (6)$$

The process of deriving this updating function is the same as that by Duchi et al.[7]. Equation (6) shows that SGFT can process one piece of data at $O(d)$ computational cost as large as SG-L_1. Fig. 2 illustrates how $h_{t,p}^{(j)}$ affects the updating of $\mathbf{w}_{t+1/2}$ in the regularization step. These figures indicate that the parameter $h_{t,p}^{(j)}$ adjusts the intensity of truncation to retain rarely occurring but informative features.

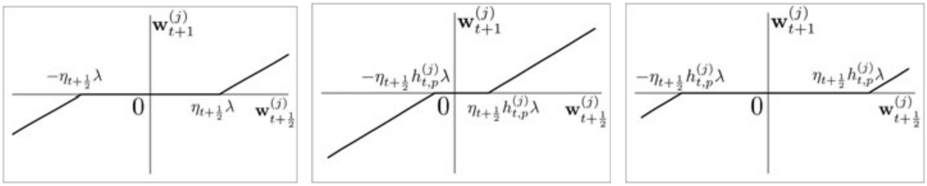

Fig. 2. These figures show how $h_{t,p}^{(j)}$ affects the regularization. Left : Normal SG-L_1 case, Center : Small $h_{t,p}^{(j)}$ case in SGFT, Right : Large $h_{t,p}^{(j)}$ case in SGFT.

3.2 Regret Analysis of SGFT

In SGFT, regularization term r_t is replaced with $r_t(\mathbf{w}_t) = \lambda\|\mathbf{H}_{t,p}\mathbf{w}_t\|_1$ from a normal L_1 norm $r(\mathbf{w}_t) = \lambda\|\mathbf{w}_t\|_1$. When differentiating r_t with respect to a weight vector $\mathbf{w}$ and applying L_2 norm, we obtain

$$\|\partial r_t\|_2 = \lambda\sqrt{\sum_{k=1}^{d}\left(h_{t,p}^{(k)}\right)^2} \; . \tag{7}$$

From equation (7), Lemma 1 is proved.

Lemma 1. *We define* $\|\partial f\|$ *as* $\sup_{\mathbf{g}\in\partial f(\mathbf{w})}\|\mathbf{g}\|_2$*. If* $\|\partial \ell_t\| \leq G$*,* $\eta_t = \eta_{t+1/2} = c/\sqrt{t}$ *using a scalar* $c > 0$*, and* $h_{t,p}^{(k)}$ *is* L_p *norm where* $p > 2$*, a scalar* U *exists that satisfies inequality (8).*

$$\lim_{t\to\infty}\|\partial r_t\| < U \; . \tag{8}$$

In the Appendix, we prove formula (8).

In the case of $p \leq 2$, we redefine the diagonal matrix $\mathbf{H}_t$ as $H_t^{(k,k)} = \min(h_{t,p}^{(k)}, V)$ using a scalar V. In this paper, $H^{(i,j)}$ represents the (i,j)-th entry of the matrix $\mathbf{H}$. In the case of $p \leq 2$, we can prove that the upper bound of $\|\partial r_t\|$ is $\sqrt{d}\lambda V$. Thus, there is a scalar U where $\lim_{t\to\infty}\|\partial r_t\| \leq U$. In this case, we can prove that the regret bound of the SGFT is $O(\sqrt{T})$. The proof is in the Appendix.

Theorem 1. *We define the matrix* $\mathbf{H}_{t,p}$ *as*

$$H_{t,p}^{(k,k)} = \begin{cases} \min(h_{t,p}^{(k)}, V) & p \leq 2 \\ h_{t,p}^{(k)} & p > 2 \end{cases} \tag{9}$$

In addition, assume both the loss function and regularization term are convex functions, and that they satisfy $\forall \mathbf{w}_t \;\; \|\mathbf{w}_t - \mathbf{w}^*\|_2 \leq D$*,* $\|\partial \ell_t\| \leq U$*,* $\|\partial r_t\| \leq U$ *where setting* $\eta_t = \eta_{t+1/2} = c/\sqrt{t}$ *using scalars* D*,* U*, and* $c > 0$*.*

In this case, the regret bound of SGFT satisfies formula (10).

$$R_{\ell+r}(T) \leq 2UD + \left(D^2/2c + 8U^2c\right)\sqrt{T} = O(\sqrt{T}) \; , \tag{10}$$

where

$$R_{\ell+r}(T) = \sum_{t=1}^{T} \left\{ \ell_t(\mathbf{w}_t) + r_t(\mathbf{w}_t) - \ell_t(\mathbf{w}^*) - r_t(\mathbf{w}^*) \right\},$$

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_{t=1}^{T} \{\ell_t(\mathbf{w}) + r_t(\mathbf{w})\} .$$

3.3 Lazy Update

SGFT allows us to truncate parameters in a lazy fashion. We do not need to penalize the weights of features that do not occur in the current sample, thus, we can postpone applying the penalty at each iteration. This updating scheme enables faster calculation when the dimension of instances is large and we have sparse samples.

We define the absolute value of the total L_1 penalty from $t = 1$ to n as u_n.

$$u_n = \lambda \sum_{t=1}^{n} \eta_{t+1/2} . \tag{11}$$

At each instance, before we update the parameter in step 4, we apply the L_1 penalty to features that are used in the input.

$$w_{t+1/2}^{(j)} = \begin{cases} \max\left(0, w_t^{(j)} - (u_{t-1} - u_{s-1}) h_{s,p}^{(j)}\right) & w_t^{(j)} \geq 0 \\ \min\left(0, w_t^{(j)} + (u_{t-1} - u_{s-1}) h_{s,p}^{(j)}\right) & w_t^{(j)} < 0 \end{cases} , \tag{12}$$

where s is the sample number that feature j is used at the end. If the value of u_{s-1} is calculated at s-th update, which is the last update of the weight of feature j, only u_t, u_{t-1} must be derived at iteration t and the value from u_1 to u_{t-2} does not have to be preserved.

Second, we perform a subgradient method using $\mathbf{w}_{t+1/2}$ in step 4 and derive $\mathbf{w}_{t+1}$. Finally, we skip step 5 to finish. In the lazy update version of SGFT, we can compute the update at the speed of O(number of features that occur).

4 SGFT with Cumulative Penalty

Tsuruoka et al.[15] proposed a cumulative penalty model for SG-L_1. The normal SG-L_1 has a problem where a solution is often obtained that is significantly affected by the last few instances. This is because the weight easily moves away from zero when a feature is used in the last few instances. The main idea of the cumulative penalty model is to keep track of the total penalty. Then, we apply a cumulative L_1 penalty to smooth the effect of the update fluctuation and move away from zero unless the updating sum exceeds the cumulative penalty. In addition, this model can smooth the effect of the update and also suppress noisy data. In this section, we propose a method combining our models with the cumulative penalty model.

We begin by introducing $q_t^{(j)}$ as an already applied cumulative L_1 penalty of feature j at the t-th instance. We initialize $q_0^{(j)} = 0$ for all j. In this setting, at step 5, we update the weight vector whose feature is used in the current instance as follows:

$$w_{t+1}^{(j)} = \begin{cases} \max\left(0, w_{t+1/2}^{(j)} - (h_{t,p}^{(j)} u_t + q_t^{(j)})\right) & w_{t+1/2}^{(j)} \geq 0 \\ \min\left(0, w_{t+1/2}^{(j)} + (h_{t,p}^{(j)} u_t - q_t^{(j)})\right) & w_{t+1/2}^{(j)} < 0 \end{cases} . \quad (13)$$

Then, we update the parameter $q_t^{(j)}$ if the feature j is used in the current instance as follows:

$$q_t^{(j)} = q_{t-1}^{(j)} + (w_{t+1}^{(j)} - w_{t+1/2}^{(j)}) . \quad (14)$$

In a cumulative penalty setting, we rewrite the optimization problem as if returning to the previous iteration and applying the new frequency-aware adaptation parameter $h_{t,p}^{(j)}$. This reformalization makes the update function simple and reduce space complexity. The same as Tsuruoka et al. did, the whole frequency-aware L_1 penalty is applied at once if the following two types of weight vectors reside within the same orthant: 1) the weight vector that had been updated by the true gradient with the latest penalty and 2) the weight vector calculated with the cumulative form of L_1 normalization.

SGFT with cumulative penalty takes O(number of features that occur) computational time at each iteration.

5 Evaluation

We evaluated our proposed frequency-aware truncated methods using classification tasks. In the experiment, we used three datasets.

First, we used sentiment classification tasks[2] for Amazon.com goods reviews. Classification tasks classify whether a positive or negative opinion is noted in each review. In this dataset, we used the books and dvd categories.

Second, we used the 20 Newsgroups dataset (news20)[8]. The news20 is a news categorization task in which a learning algorithm predicts to what category each news article is assigned. This dataset consists of about 20,000 news articles. Each article is assigned to one of 20 predetermined categories. We used four subsets of news20: ob-2-1, sb-2-1, ob-8-1, and sb-8-1[11]. In each subset, the number of categories and the closeness among categories differed. For the first letter of each subset name, 'o' indicates 'overlap' and 's' denotes 'separated'. Classifying categories correctly is more difficult with an 'overlap' dataset. The second letter of the subset names means the heterogeneity among the categories and there is no difference in the instance number among the categories. The middle number is the number of categories.

Last, we used the Reuters-21578 [10] dataset. The Reuters-21578 also consists of news articles and we used a dataset for a 20-category classification task

Table 2. Dataset specifications

	# of instances	# of features	# of categories
books	4,465	332,441	2
dvd	3,586	282,901	2
ob-2-1	1,000	5,942	2
sb-2-1	1,000	6,276	2
ob-8-1	4,000	13,890	8
sb-8-1	4,000	16,282	8
reut20	7,800	34,488	20

(reut20) from the Reuters-21578. In Table 2, we provide the specifications of each dataset, including the number of features, instances, and categories.

In this experiment, we used the hinge-loss function as a loss function. When there are more than two categories, it is not possible to use hinge-loss directly because the hinge-loss function was developed for binary categorization. In our experiment, we defined a weight vector as $\mathbf{w} \in \hat{W} \subset \mathbf{R}^{d \times K}$, where K was the number of classes, and a feature vector as $\Phi(\mathbf{x}, y)$, mapped from the Cartesian product $X \times Y$, where Y was the set of labels in the 1-of-K scheme. Moreover, we set the loss function as formula (15).

$$\ell_t(\mathbf{w}_t) = [1 - \langle \mathbf{w}_t, \Phi(\mathbf{x}_t, y_t) \rangle + \max_{z_t \in Y \setminus y_t} \langle \mathbf{w}_t, \Phi(\mathbf{x}_t, z_t) \rangle]_+ \ , \tag{15}$$

where y_t is a correct label at t. We can process the multi-class classification tasks as define above. In the experiment, we examined SGFT, SG-L_1, and RDA[16] to compare the precision and sparseness rates. From a family of frequency-aware truncated methods, we selected the algorithms of $p = 1, 2, 3, \infty$.

The step size η_t was set at $\eta_t = \eta_{t+1/2} = 1/\sqrt{t}$ to satisfy the restriction of the regret bound in SGFT and SG-L_1. Moreover, in SGFT where $p = 1, 2$, we set $V = 500$ to satisfy the regret bound restriction[4]. In contrast, in RDA, we set $h(\mathbf{w}) = 1/2\|\mathbf{w}\|_2^2$ and $\beta_t = \sqrt{t}$. In our experiment, we evaluated the performance of our methods using a ten-fold cross-validation to achieve the highest precision rate by adjusting the parameter λ. We set the number of iterations to 20.

The experimental results of SGFT against the change of parameter p are shown in Table 3. The figure in $[\cdot]$ means the standard deviation, and the figure in $(\cdot)$ denotes the sparseness rate. Moreover, the highest precision rates among all the algorithms are written in **bold** font.

Table 3 indicates that SGFT with $p = 2$ achieves the best performance in the four datasets. Moreover, in the other three datasets, SGFT $p = 2$ has the second highest precision, indicating SGFT $p = 2$ is an efficient learning method in the SGFT family. Table 3 also shows that SGFT has a tendency of increasing sparsity responding to increase of parameter p.

[4] In this experiment, the value of $h_{t,p}^{(j)}$ did not exceed 500, thus the value of V did not influence the result.

Table 3. SGFT's precision (sparseness) rate against parameter p (Iterations : 20)

	SGFT ($p = 1$)	SGFT ($p = 2$)	SGFT ($p = 3$)	SGFT ($p = \infty$)
books	85.23[1.52] (34.52)	**85.52**[1.24] (48.26)	85.14[1.33] (49.58)	85.05[1.41] (69.39)
dvd	82.49[1.68] (37.46)	84.75[1.75] (59.72)	**85.03**[2.28] (63.74)	84.02[1.66] (67.19)
ob-2-1	97.00[1.73] (42.78)	**97.10**[1.14] (56.73)	96.90[1.87] (59.03)	96.80[1.94] (59.78)
sb-2-1	**98.90**[0.83] (60.13)	98.40[0.80] (70.32)	98.40[1.11] (71.99)	98.10[1.14] (72.69)
ob-8-1	92.25[1.14] (62.83)	**93.10**[1.41] (62.84)	93.00[1.29] (64.64)	91.45[1.33] (77.78)
sb-8-1	90.90[1.72] (68.26)	92.55[1.85] (68.49)	**93.78**[2.44] (70.23)	91.25[1.44] (83.53)
reut20	95.23[0.65] (89.11)	**96.04**[0.56] (90.38)	95.91[0.55] (90.21)	94.80[0.67] (91.05)

Table 4. Precision (sparseness) rate (Iterations : 20)

	SGFT ($p = 2$)	SG-L_1	RDA
books	85.52[1.24] (48.26)	84.98[1.61] (48.28)	**86.57**[1.16] (34.65)
dvd	84.75[1.75] (59.72)	83.91[1.55] (79.57)	**86.36**[2.08] (37.08)
ob-2-1	97.10[1.14] (56.73)	96.40[1.96] (49.23)	**97.60**[1.80] (39.83)
sb-2-1	**98.40**[0.80] (70.32)	97.20[1.78] (84.25)	98.20[0.75] (56.67)
ob-8-1	93.10[1.41] (62.84)	90.63[1.64] (87.90)	**93.78**[1.21] (50.52)
sb-8-1	92.55[1.85] (68.49)	90.53[1.61] (67.46)	**95.45**[0.95] (60.46)
reut20	96.04[0.56] (90.38)	95.53[0.63] (89.29)	**96.27**[0.63] (86.67)

Table 4 illustrates the results of SG-L_1 and RDA as compared with SGFT $p = 2$, which showed the most efficient performance in Table 3.

From Table 4, SGFT is confirmed to outperform SG-L_1 in all the datasets. At the same time, SGFT does not necessarily have a smaller sparsity rate than SG-L_1. This result shows that frequency-aware truncation improves the accuracy of precision, without degrading sparsity. From the experimental results, note that frequency-aware truncation could improve the accuracy by retaining rarely occurring but important features and dropping unimportant features. Thus, in the setting of sparse online learning, frequency-aware truncation is a useful method compared with the normal L_1-regularization for these datasets.

We also evaluated the experimental results of RDA. The results showed that RDA obtained the highest precision rate in these tasks except for sb-2-1, but, the rate of sparsity was smaller than SGFT. This indicates that RDA is a sophisticated algorithm for precise learning; however, to obtain a sparse solution, frequency-aware truncation methods are also efficient for learning. We consider that the margin between these two methods occur partly because RDA[16] has the smaller regret bound than FOBOS[7] and SGFT in terms of the coefficient.

6 Conclusion

We analyzed a new family of truncated methods for retaining rarely occurring features in an online setting. These methods integrate the sum of updates in the loss minimization steps into the regularization step to adjust the intensity of truncation. In this way, we can solve the problem where rarely used features are truncated on a priority basis. Specifically, we proved the computational cost and theoretical guarantees of SGFT, which is also known as a frequency-aware truncated method. In addition, we provided possible extensions of our work, such as lazy-update and cumulative-penalty schemes. Finally, we evaluated the performance of our methods in experiments. The experimental results showed that frequency-aware truncated methods could retain rarely occurring but important features without loss of sparsity.

A few discussions for further research remain in connection with our proposed methods. The first is the integration of frequency-aware methods into a primal-dual averaging framework. We assume that a frequency-aware scheme could be connected with dual-averaging methods with a minor change of frequency-aware term's definition. This extension would also enable us to give the same regret bound and computational time as those for dual-averaging methods and expect the higher performance than RDA. The second issue is whether we can optimize parameter p in an online setting. We aim to investigate these questions and further extensions of our proposed methods.

References

1. Bertsekas, D.P.: Nonlinear Programming. Athena Scientific (1999)
2. Blitzer, J., Dredze, M., Pereira, F.: Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In: Association for Computational Linguistics (ACL), pp. 440–447 (2007), `http://www.cs.jhu.edu/~mdredze/datasets/sentiment`
3. Carpenter, B.: Lazy sparse stochastic gradient descent for regularized multinomial logistic regression. Technical report, Alias-i (2008)
4. Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., Singer, Y.: Online Passive-Aggressive Algorithms. Journal of Machine Learning Research 7, 551–585 (2006)
5. Dredze, M., Crammer, K.: Confidence-weighted linear classification. In: ICML, pp. 264–271 (2008)
6. Duchi, J., Hazan, E., Singer, Y.: Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. In: COLT, pp. 257–269 (2010)
7. Duchi, J., Singer, Y.: Efficient Online and Batch Learning Using Forward Backward Splitting. Journal of Machine Learning Research 10, 2899–2934 (2009)
8. Lang, K.: Newsweeder: Learning to filter netnews. In: International Conference on Machine Learning (ICML), pp. 331–339 (1995), `http://mlg.ucd.ie/datasets`
9. Langford, J., Li, L., Zhang, T.: Sparse Online Learning via Truncated Gradient. J. Mach. Learn. Res. 10, 777–801 (2009)
10. Lewis, D.D.: Reuters-21578, `http://www.daviddlewis.com/resources/testcollections/reuters21578`
11. Matsushima, S., Shimizu, N., Yoshida, K., Ninomiya, T., Nakagawa, H.: Exact Passive-Aggressive Algorithm for Multiclass Classification Using Support Class. In: SDM, pp. 303–314 (2010)
12. Nesterov, F.: Primal-Dual subgradient methods for convex problems. Mathematical Programming 120(1), 221–259 (2009)
13. Rosenblatt, F.: The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. Psychological Review 65, 386–408 (1958)

14. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information Processing and Management 24(5), 513–523 (1988)
15. Tsuruoka, Y., Tsujii, J., Ananiadou, S.: Stochastic Gradient Descent Training for L1-regularized Log-linear. In: ACL-IJCNLP, pp. 477–485 (2009)
16. Xiao, L.: Dual averaging methods for regularized stochastic learning and online optimization. In: Advances in Neural Information Processing Systems, vol. 23 (2009)
17. Zinkevich, M.: Online convex programming and generalized infinitesimal gradient ascent. In: International Conference on Machine Learning (ICML), pp. 928–936 (2003)

Appendix

Proof of Lemma 1

Let $\boldsymbol{\eta}_t$ be a vector of $(\eta_1, \eta_2, \ldots, \eta_t)$. If $\|\partial \ell_t\| \leq G$ for all t, we can derive

$$h_{t,p}^{(k)} \leq \|G\boldsymbol{\eta}_t\|_p = G\|\boldsymbol{\eta}_t\|_p \ . \tag{16}$$

The first inequality follows from the inequality below.

$$\forall t, k \quad |g_t^{\ell,(k)}| \leq \|\mathbf{g}_t^\ell\|_2 \leq \|\partial \ell_t\| \leq G \ .$$

From the definition of L_p norm, we can rewrite equation (16) as

$$\|\boldsymbol{\eta}_t\|_p = \sqrt[p]{\sum_{k=1}^{t} |\eta_k|^p} \ . \tag{17}$$

Thus, if we substitute η_k with $c/\sqrt{k}$, we obtain equation (18).

$$\|\boldsymbol{\eta}_t\|_p = c\sqrt[p]{\sum_{k=1}^{t} k^{-\frac{p}{2}}} \ . \tag{18}$$

$\sum_{t=1}^{T} t^{-\frac{p}{2}}$ is a zeta function. From the characteristics of zeta functions, if $-\frac{p}{2} < -1$, that is, $p > 2$, $\sum_{k=1}^{t} k^{-\frac{p}{2}}$ has a upper bound and thus converges as $t \longrightarrow \infty$. We set the upper limit value to S, obtaining $\|\boldsymbol{\eta}_t\|_p = cS^{\frac{1}{p}}$. Then, there is a scalar U which satisfies equation (19).

$$\|\partial r_t(\mathbf{w})\| \leq \lambda G \sqrt{\sum_{l=1}^{d} (cS^{\frac{1}{p}})^2} = c\lambda G S^{\frac{1}{p}} \sqrt{d} \leq U \ . \tag{19}$$

Therefore, we can prove Lemma 1. However, in the case of $p \leq 2$, we cannot bound $h_{t,p}^{(k)}$ because the zeta function does not converge.

Proof of Theorem 1

The procedure of the proof is similar to that by Duchi et al.[7], but, there is a small difference because, in our methods, the regularization term depends on t which is the number of iterations. First, we prove Lemma 2.

Lemma 2. *Assume both loss function ℓ_t and regularization term r_t have convexity and satisfy equation (20).*

$$\|\partial \ell_t(\mathbf{w})\|^2 \leq G^2, \|\partial r_t(\mathbf{w})\|^2 \leq G^2 \ . \tag{20}$$

Let step size η_t satisfy $\eta_{t+1} \leq \eta_{t+1/2} \leq \eta_t$ and $\eta_t \leq 2\eta_{t+1}$. In this case, we can prove equation (21).

$$\begin{aligned} \forall \mathbf{w}^* \quad \exists c \leq 5 \qquad & 2\eta_t \ell_t(\mathbf{w}_t) - 2\eta_t \ell_t(\mathbf{w}^*) + 2\eta_{t+1/2} r_t(\mathbf{w}_{t+1}) - 2\eta_{t+1/2} r_t(\mathbf{w}^*) \\ & \leq \|\mathbf{w}_t - \mathbf{w}^*\|_2^2 - \|\mathbf{w}_{t+1} - \mathbf{w}^*\|_2^2 + 8\eta_t^2 G^2 \ . \end{aligned} \tag{21}$$

From the condition that the loss function is convex, we can derive equation (22) in terms of any subgradient $\mathbf{g}_t^\ell \in \partial \ell_t(\mathbf{w}_t)$.

$$\ell_t(\mathbf{w}^*) \geq \ell_t(\mathbf{w}_t) + \langle \mathbf{g}_t^\ell, \mathbf{w}^* - \mathbf{w}_t \rangle \implies -\langle \mathbf{g}_t^\ell, \mathbf{w}_t - \mathbf{w}^* \rangle \leq \ell_t(\mathbf{w}^*) - \ell_t(\mathbf{w}_t) \ . \tag{22}$$

This is the case with regard to regularization term $r_t(\cdot)$. In this paper, we denote any subgradient of regularization term $r_t(\mathbf{w}_{t+1})$ as $\mathbf{g}_{t+1}^r$.

From the Cauchy-Shwartz inequality and equation (2), we obtain

$$\begin{aligned} \langle \mathbf{g}_{t+1}^r, \mathbf{w}_{t+1} - \mathbf{w}_t \rangle &= \langle \mathbf{g}_{t+1}^r, -\eta_t \mathbf{g}_t^\ell - \eta_{t+1/2} \mathbf{g}_{t+1}^r \rangle \\ &\leq \|\mathbf{g}_{t+1}^r\|_2 \|\eta_t \mathbf{g}_t^\ell + \eta_{t+1/2} \mathbf{g}_{t+1}^r\|_2 \\ &\leq \eta_{t+1/2} \|\mathbf{g}_{t+1}^r\|_2^2 + \eta_t \|\mathbf{g}_{t+1}^r\|_2 \|\mathbf{g}_t^\ell\|_2 \\ &\leq (\eta_{t+1/2} + \eta_t) G^2 \ . \end{aligned} \tag{23}$$

In the first equation above, we use $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \mathbf{g}_t^\ell - \eta_{t+1/2} \mathbf{g}_{t+1}^r$ derived from the derivation of equations (2) and (5).

Then, we proceed to derive the upper bound of the difference between $\mathbf{w}^*$ and $\mathbf{w}_{t+1}$ for obtaining the upper bound of $\ell_t(\mathbf{w}_t) + r_t(\mathbf{w}_t) - \ell_t(\mathbf{w}^*) - r_t(\mathbf{w}^*)$. We can expand the L_2 norm of the difference between $\mathbf{w}^*$ and $\mathbf{w}_{t+1}$ as follows:

$$\begin{aligned} \|\mathbf{w}_{t+1} - \mathbf{w}^*\|_2^2 &= \|\mathbf{w}_t - (\eta_t \mathbf{g}_t^\ell + \eta_{t+1/2} \mathbf{g}_{t+1}^r) - \mathbf{w}^*\|_2^2 \\ &= \|\mathbf{w}_t - \mathbf{w}^*\|_2^2 - 2\left(\eta_t \langle \mathbf{g}_t^\ell, \mathbf{w}_t - \mathbf{w}^* \rangle + \eta_{t+1/2} \langle \mathbf{g}_{t+1}^r, \mathbf{w}_t - \mathbf{w}^* \rangle\right) \\ &\quad + \|\eta_t \mathbf{g}_t^\ell + \eta_{t+1/2} \mathbf{g}_{t+1}^r\|_2^2 \\ &= \|\mathbf{w}_t - \mathbf{w}^*\|_2^2 - 2\eta_t \langle \mathbf{g}_t^\ell, \mathbf{w}_t - \mathbf{w}^* \rangle + \|\eta_t \mathbf{g}_t^\ell + \eta_{t+1/2} \mathbf{g}_{t+1}^r\|_2^2 \\ &\quad - 2\eta_{t+1/2} \left(\langle \mathbf{g}_{t+1}^r, \mathbf{w}_{t+1} - \mathbf{w}^* \rangle - \langle \mathbf{g}_{t+1}^r, \mathbf{w}_{t+1} - \mathbf{w}_t \rangle\right) \ . \end{aligned} \tag{24}$$

The bound of the third term is derived as

$$\begin{aligned} \|\eta_t \mathbf{g}_t^\ell + \eta_{t+1/2} \mathbf{g}_{t+1}^r\|_2^2 &= \eta_t^2 \|\mathbf{g}_t^\ell\|_2^2 + 2\eta_t \eta_{t+1/2} \langle \mathbf{g}_t^\ell, \mathbf{g}_{t+1}^r \rangle + \eta_{t+1/2}^2 \|\mathbf{g}_{t+1}^r\|_2^2 \\ &\leq 4\eta_t^2 G^2 \ . \end{aligned} \tag{25}$$

The upper bound of equation (24) is obtained by equations (22), (23), and (25).

$$\begin{aligned}\|\mathbf{w}_{t+1}-\mathbf{w}^*\|_2^2 &\leq \|\mathbf{w}_t-\mathbf{w}^*\|_2^2 - 2\eta_t\langle \mathbf{g}_t^\ell, \mathbf{w}_t-\mathbf{w}^*\rangle - 2\eta_{t+1/2}\langle \mathbf{g}_{t+1}^r, \mathbf{w}_{t+1}-\mathbf{w}^*\rangle \\ &\quad +\|\eta_t\mathbf{g}_t^\ell + \eta_{t+1/2}\mathbf{g}_{t+1}^r\|_2^2 + 4\eta_{t+1/2}\eta_t G^2 \\ &\leq \|\mathbf{w}_t-\mathbf{w}^*\|_2^2 + 2\eta_t(\ell_t(\mathbf{w}^*)-\ell_t(\mathbf{w}_t)) \\ &\quad +2\eta_{t+1/2}(r_t(\mathbf{w}^*)-r_t(\mathbf{w}_{t+1})) + 8\eta_t^2G^2 \ . \end{aligned} \tag{26}$$

From equation (26), we finish the proof of Lemma 2.

Next, we prove the upper bound of SGFT using Lemma 2. Zinkevich's regret analysis[17] for online convex programming is effective, thus, we use this method.

From Lemma 2, when we set $\eta_t = \eta_{t+1/2}$, we obtain

$$\begin{aligned}&\ell_t(\mathbf{w}_t)-\ell_t(\mathbf{w}^*)+r_t(\mathbf{w}_{t+1})-r_t(\mathbf{w}^*) \\ &\quad\leq \frac{1}{2\eta_t}\left(\|\mathbf{w}_t-\mathbf{w}^*\|_2^2 - \|\mathbf{w}_{t+1}-\mathbf{w}^*\|_2^2\right) + 4G^2\eta_t \ .\end{aligned} \tag{27}$$

Then, we calculate the sum of equation (27) from $t = 1$ to T and derive

$$\begin{aligned}R_{\ell+r}(T) &\leq 2GD + \sum_{t=1}^{T}\frac{1}{2\eta_t}\left(\|\mathbf{w}_t-\mathbf{w}^*\|_2^2 - \|\mathbf{w}_{t+1}-\mathbf{w}^*\|_2^2\right) + 4G^2\sum_{t=1}^{T}\eta_t \\ &\leq 2GD + \frac{D^2}{2\eta_1} + \frac{D^2}{2}\sum_{t=2}^{T}\left(\frac{1}{\eta_t}-\frac{1}{\eta_{t-1}}\right) + 4G^2\sum_{t=1}^{T}\eta_t \\ &\leq 2GD + \frac{D^2}{2\eta_T} + 4G^2\sum_{t=1}^{T}\eta_t \ ,\end{aligned} \tag{28}$$

from the following restriction

$$\sum_{t=1}^{T}(r_t(\mathbf{w}_t)-r_{t-1}(\mathbf{w}_t)) - r_T(\mathbf{w}_{T+1}) \leq \|\partial r_T(\mathbf{w})\|\|\mathbf{w}\|_2 \leq 2GD \ . \tag{29}$$

The second inequality holds using $\|\mathbf{w}_t-\mathbf{w}^*\|_2 \leq D$. Assuming that $\eta_t = c/\sqrt{t}$, we can prove the upper bound of regret is $O(\sqrt{T})$ from the fact that $\sum_{t=1}^{T}\eta_t \leq 2c\sqrt{T}$. Thus, we have proved Theorem 1.