Sparse Online Learning via Truncated Gradient: Appendix

1 Proof of Main Results

In the setting of standard online learning, we are interested in sequential prediction problems where for $i = 1, 2, \ldots$:

1. An unlabeled example $x_i = [x_i^1, \ldots, x_i^d] \in R^d$ arrives.
2. We make a prediction $\hat{y}_i$ based on the current weights $w_i = [w_i^1, \ldots, w_i^d] \in R^d$.
3. We observe y_i, let $z_i = (x_i, y_i)$, and incur some known loss $L(w_i, z_i)$ convex in parameter w_i.
4. We update weights according to some rule: $w_{i+1} \leftarrow f(w_i)$.

We want an update rule f that allows us to bound the sum of losses, $\sum_{i=1}^t L(w_i, z_i)$, as well as achieving sparsity. For this purpose, we start with the standard stochastic gradient descent (SGD) rule, which is of the form:

$$f(w_i) = w_i - \eta \nabla_1 L(w_i, z_i), \tag{1}$$

where $\nabla_1 L(a, b)$ is a sub-gradient of $L(a, b)$ with respect to the first variable a. The parameter $\eta > 0$ is often referred to as the learning rate.

In order to achieve sparsity, the most natural method is to round small coefficients (whose magnitudes are below a threshold $\theta > 0$) to zero after every K online steps. That is, if i/K is not an integer, we use the standard SGD rule (1); if i/K is an integer, we modify the rule as:

$$f(w_i) = T_0(w_i - \eta \nabla_1 L(w_i, z_i), \theta), \tag{2}$$

where, the threshold $\theta \geq 0$, $T_0(v, \theta) = [T_0(v_1, \theta), \ldots, T_0(v_d, \theta)]$ for vector $v = [v_1, \ldots, v_d] \in R^d$, $T_0(v_j, \theta) = v_j I(|v_j| < \theta)$, and $I(\cdot)$ is the set-indicator function.

We call the following rule *truncated gradient*, where the amount of shrinkage is controlled by a *gravity* parameter $g_i > 0$:

$$f(w_i) = T_1(w_i - \eta \nabla_1 L(w_i, z_i), \eta g_i, \theta), \tag{3}$$

where for a vector $v = [v_1, \ldots, v_d] \in R^d$, and a scalar $g \geq 0$, $T_1(v, \alpha, \theta) = [T_1(v_1, \alpha, \theta), \ldots, T_1(v_d, \alpha, \theta)]$, with

$$T_1(v_j, \alpha, \theta) = \begin{cases} \max(0, v_j - \alpha) & \text{if } v_j \in [0, \theta] \\ \min(0, v_j + \alpha) & \text{if } v_j \in [-\theta, 0] \\ v_j & \text{otherwise} \end{cases}.$$

Throughout the paper, we use $\|\cdot\|_1$ for 1-norm, and $\|\cdot\|$ for 2-norm. For reference, we make the following assumption regarding the loss function:

Assumption 1.1 *We assume that $L(w, z)$ is convex in w, and there exist non-negative constants A and B such that $(\nabla_1 L(w, z))^2 \leq AL(w, z) + B$ for all $w \in R^d$ and $z \in R^{d+1}$.*

For linear prediction problems, we have a general loss function of the form $L(w, z) = \phi(w^T x, y)$. The following are some common loss functions $\phi(\cdot, \cdot)$ with corresponding choices of parameters A and B (which are not unique), under the assumption that $\sup_x \|x\| \leq C$. All of them can be used for binary classification where $y \in \pm 1$, but the last one is more often used in regression where $y \in R$.

- Logistic: $\phi(p, y) = \ln(1 + \exp(-py))$; $A = 0$ and $B = C^2$.
- SVM (hinge loss): $\phi(p, y) = \max(0, 1 - py)$; $A = 0$ and $B = C^2$.
- Least squares (square loss): $\phi(p, y) = (p - y)^2$; $A = 4C^2$ and $B = 0$.

The following lemma is the essential step in our analysis.

Lemma 1.1 *For update rule (3) applied to weight vector w on example $z = (x, y)$ with gravity parameter $g_i = g$, resulting in a weight vector w'. If Assumption 1.1 holds, then for all $\bar{w} \in R^d$, we have*

$$\begin{aligned} &(1 - 0.5A\eta)L(w, z) + g\|w' \cdot I(|w'| \leq \theta)\|_1 \\ \leq &L(\bar{w}, z) + g\|\bar{w} \cdot I(|w'| \leq \theta)\|_1 + \frac{\eta}{2}B + \frac{\|\bar{w} - w\|^2 - \|\bar{w} - w'\|^2}{2\eta}. \end{aligned}$$

PROOF. Consider any target vector $\bar{w} \in R^d$ and let $\tilde{w} = w - \eta\nabla_1 L(w,z)$. We have $w' = T_1(\tilde{w}, g\eta, \theta)$. Let

$$u(\bar{w}, w') = g\|\bar{w} \cdot I(|w'| \le \theta)\|_1 - g\|w' \cdot I(|w'| \le \theta)\|_1.$$

Then the update equation implies the following:

$$\begin{aligned}
&\|\bar{w} - w'\|^2 \\
\le&\|\bar{w} - w'\|^2 + \|w' - \tilde{w}\|^2 \\
=&\|\bar{w} - \tilde{w}\|^2 - 2(\bar{w} - w')^T(w' - \tilde{w}) \\
\le&\|\bar{w} - \tilde{w}\|^2 + 2\eta u(\bar{w}, w') \\
=&\|\bar{w} - w\|^2 + \|w - \tilde{w}\|^2 + 2(\bar{w} - w)^T(w - \tilde{w}) + 2\eta u(\bar{w}, w') \\
=&\|\bar{w} - w\|^2 + \eta^2\|\nabla_1 L(w,z)\|^2 + 2\eta(\bar{w} - w)^T\nabla_1 L(w,z) + 2\eta u(\bar{w}, w') \\
\le&\|\bar{w} - w\|^2 + \eta^2\|\nabla_1 L(w,z)\|^2 + 2\eta(L(\bar{w},z) - L(w,z)) + 2\eta u(\bar{w}, w') \\
\le&\|\bar{w} - w\|^2 + \eta^2(AL(w,z) + B) + 2\eta(L(\bar{w},z) - L(w,z)) + 2\eta u(\bar{w}, w').
\end{aligned}$$

Here, the first and second equalities follow from algebra, and the third from the definition of $\tilde{w}$. The first inequality follows because a square is always non-negative. The second inequality follows because $w' = T_1(\tilde{w}, g\eta, \theta)$, which implies that $(w' - \tilde{w})^T w' = -g\eta\|w' \cdot I(|w'| \le \theta)\|_1$ and $|w'_j - \tilde{w}_j| \le g\eta I(|w'_j| \le \theta)$. Therefore

$$\begin{aligned}
-(\bar{w} - w')^T(w' - \tilde{w}) =& -\bar{w}^T(w' - \tilde{w}) + {w'}^T(w' - \tilde{w}) \\
\le& \sum_{j=1}^d |\bar{w}_j||w'_j - \tilde{w}_j| + (w' - \tilde{w})^T w' \\
\le& g\eta \sum_{j=1}^d |\bar{w}_j| I(|w'_j| \le \theta) + (w' - \tilde{w})^T w' = \eta u(\bar{w}, w').
\end{aligned}$$

The third inequality follows from the definition of sub-gradient of a convex function, which implies that

$$(\bar{w} - w)^T\nabla_1 L(w,z) \le L(\bar{w},z) - L(w,z)$$

for all w and $\bar{w}$. The fourth inequality follows from Assumption 1.1. Rearranging the above inequality leads to the desired bound. □

Our main result is Theorem 1.1 that is parameterized by A and B. Specializing it to particular loss functions yields several corollaries.

Theorem 1.1 *(Sparse Online Regret) Consider sparse online update rule (3) with $w_1 = [0, \ldots, 0]$ and $\eta > 0$. If Assumption 1.1 holds, then for all $\bar{w} \in R^d$ we have*

$$\frac{1 - 0.5A\eta}{T}\sum_{i=1}^T \left[L(w_i, z_i) + \frac{g_i}{1 - 0.5A\eta}\|w_{i+1} \cdot I(w_{i+1} \le \theta)\|_1\right] \le \frac{\eta}{2}B + \frac{\|\bar{w}\|^2}{2\eta T} + \frac{1}{T}\sum_{i=1}^T [L(\bar{w}, z_i) + g_i\|\bar{w} \cdot I(w_{i+1} \le \theta)\|_1],$$

where $I(\cdot)$ is the set-indicator function and for vectors $v = [v_1, \ldots, v_d]$ and $v' = [v'_1, \ldots, v'_d]$, we let

$$\|v \cdot I(|v'| \le \theta)\|_1 = \sum_{j=1}^d |v_j| I(|v'_j| \le \theta).$$

PROOF. Apply Lemma 1.1 to the update on trial i, we have

$$\begin{aligned}
&(1 - 0.5A\eta)L(w_i, z_i) + g_i\|w_{i+1} \cdot I(|w_{i+1}| \le \theta)\|_1 \\
\le& L(\bar{w}, z_i) + \frac{\|\bar{w} - w_i\|^2 - \|\bar{w} - w_{i+1}\|^2}{2\eta} + g_i\|\bar{w} \cdot I(|w_{i+1}| \le \theta)\|_1 + \frac{\eta}{2}B.
\end{aligned}$$

Now summing over $i = 1, 2, \ldots, T$, we obtain

$$\begin{aligned}
& \sum_{i=1}^{T} [(1 - 0.5A\eta)L(w_i, z_i) + g_i \|w_{i+1} \cdot I(|w_{i+1}| \leq \theta)\|_1] \\
\leq & \sum_{i=1}^{T} \left[\frac{\|\bar{w} - w_i\|^2 - \|\bar{w} - w_{i+1}\|^2}{2\eta} + L(\bar{w}, z_i) + g_i \|\bar{w} \cdot I(|w_{i+1}| \leq \theta)\|_1 + \frac{\eta}{2} B \right] \\
= & \frac{\|\bar{w} - w_1\|^2 - \|\bar{w} - w_T\|^2}{2\eta} + \frac{\eta}{2} TB + \sum_{i=1}^{T} [L(\bar{w}, z_i) + g_i \|\bar{w} \cdot I(|w_{i+1}| \leq \theta)\|_1] \\
\leq & \frac{\|\bar{w}\|^2}{2\eta} + \frac{\eta}{2} TB + \sum_{i=1}^{T} [L(\bar{w}, z_i) + g_i \|\bar{w} \cdot I(|w_{i+1}| \leq \theta)\|_1].
\end{aligned}$$

The first equality follows from the telescoping sum and the second inequality follows from the initial condition (all weights are zero) and dropping negative quantities. The theorem follows by dividing with respect to T and rearranging terms. □

Theorem 1.2 *(Stochastic Setting) Consider a set of training data $z_i = (x_i, y_i)$ for $i = 1, \ldots, n$, and let*

$$R(w, g) = \frac{1}{n} \sum_{i=1}^{n} L(w, z_i) + g\|w\|_1$$

be the L_1-regularized loss over training data. Let $\hat{w}_1 = w_1 = 0$, and define recursively for $t = 1, 2, \ldots$

$$w_{t+1} = T(w_t - \eta \nabla_1(w_t, z_{i_t}), g\eta), \qquad \hat{w}_{t+1} = \hat{w}_t + (w_{t+1} - \hat{w}_t)/(t+1),$$

where each i_t is drawn from $\{1, \ldots, n\}$ uniformly at random. If Assumption 1.1 holds, then for all T and $\bar{w} \in R^d$:

$$\mathbf{E}\left[(1 - 0.5A\eta) R\left(\hat{w}_T, \frac{g}{1 - 0.5A\eta}\right)\right] \leq \mathbf{E}\left[\frac{1 - 0.5A\eta}{T} \sum_{i=1}^{T} R\left(w_i, \frac{g}{1 - 0.5A\eta}\right)\right] \leq \frac{\eta}{2} B + \frac{\|\bar{w}\|^2}{2\eta T} + R(\bar{w}, g).$$

PROOF. Note that the recursion of $\hat{w}_t$ implies that

$$\hat{w}_T = \frac{1}{T} \sum_{t=1}^{T} w_t$$

from telescoping the update rule.

Because $R(w, g)$ is convex in w, the first inequality follows directly from Jensen's inequality.

In the following we only need to prove the second inequality. Theorem 1.1 implies the following:

$$\frac{1 - 0.5A\eta}{T} \sum_{t=1}^{T} \left[L(w_t, z_{i_t}) + \frac{g}{1 - 0.5A\eta} \|w_t\|_1 \right] \leq g\|\bar{w}\|_1 + \frac{\eta}{2} B + \frac{\|\bar{w}\|^2}{2\eta T} + \frac{1}{T} \sum_{t=1}^{T} L(\bar{w}, z_{i_t}). \tag{4}$$

Observe that

$$\mathbf{E}_{i_t}\left[L(w_t, z_{i_t}) + \frac{g}{1 - 0.5A\eta} \|w_t\|_1 \right] = R\left(w_t, \frac{g}{1 - 0.5A\eta}\right)$$

and

$$g\|\bar{w}\|_1 + \mathbf{E}_{i_1, \ldots, i_T}\left[\frac{1}{T} \sum_{t=1}^{T} L(\bar{w}, z_{i_t}) \right] = R(\bar{w}, g).$$

The second inequality is obtained by taking the expectation with respect to $\mathbf{E}_{i_1, \ldots, i_T}$ in (4). □

2 Truncated Gradient for Least-Squares Regression

The truncated descent update rule (3) can be applied to least-squares regression using square loss, leading to

$$f(w_i) = T_1(w_i - \eta(y_i - \hat{y}_i)x_i, \eta g_i, \theta),$$

where the prediction is given by $\hat{y}_i = \sum_j w_i^j x_i^j$. A complete algorithm description is given in the appendix.

This leads to Algorithm 1 which implements sparsification for square loss using truncated gradient. In the description, we use superscripted symbol w^j to denote the j-th component of vector w (in order to differentiate from w_i, which we have used to denote the i-th weight vector). For clarity, we also drop the index i from w_i. Although we keep the choice of gravity parameters g_i open in the algorithm description, in practice, we only consider the following choice:

$$g_i = \begin{cases} Kg & \text{if } i/K \text{ is an integer} \\ 0 & \text{otherwise} \end{cases}.$$

Algorithm 1 Truncated Gradient

Inputs:

- threshold $\theta \geq 0$
- gravity sequence $g_i \geq 0$
- learning rate $\eta \in (0, 1)$
- example oracle $\mathcal{O}$

initialize weights $w^j \leftarrow 0$ ($j = 1, \ldots, d$)
for trial $i = 1, 2, \ldots$

1. Acquire an unlabeled example $x = [x^1, x^2, \ldots, x^d]$ from oracle $\mathcal{O}$
2. **forall** weights w^j ($j = 1, \ldots, d$)
   (a) **if** $w^j > 0$ and $w^j \leq \theta$ **then** $w^j \leftarrow \max\{w^j - g_i\eta, 0\}$
   (b) **elseif** $w^j < 0$ and $w^j \geq -\theta$ **then** $w^j \leftarrow \min\{w^j + g_i\eta, 0\}$
3. Compute prediction: $\hat{y} = \sum_j w^j x^j$
4. Acquire the label y from oracle $\mathcal{O}$
5. Update weights for all features j: $w^j \leftarrow w^j + 2\eta(y - \hat{y})x^j$

In many online-learning situations (such as web applications), only a small subset of the features have nonzero values for any example x. It is thus desirable to deal with sparsity only in this small subset rather than all features, while simultaneously inducing sparsity on all feature weights. Moreover, it is important to store only features with non-zero coefficients (if the number of features is so large that it cannot be stored in memory, this approach allows us to use a hashtable to track only the nonzero coefficients). We describe how this can be implemented efficiently in the next section.

For reference, we present a specialization of Theorem 1.1 in the following corollary for least-squares regression:

Corollary 2.1 *(Sparse Online Regret for Square Loss) Consider truncated gradient with square loss. If there exists $C > 0$ such that $\|x\| \leq C$ for all x, then for all $\bar{w} \in R^d$, we have*

$$\frac{1 - 2C^2\eta}{T} \sum_{i=1}^{T} \left[(w_i^T x_i - y_i)^2 + \frac{g_i}{1 - 2C^2\eta} \|w_i \cdot I(|w_i| \leq \theta)\|_1 \right] \leq \frac{\|\bar{w}\|^2}{2\eta T} + \frac{1}{T} \sum_{i=1}^{T} \left[(\bar{w}^T x_i - y_i)^2 + g_{i+1} \|\bar{w} \cdot I(|w_{i+1}| \leq \theta)\|_1 \right].$$

This corollary explicitly states that the average per-example square loss incurred by the learner (left term) is bounded by the average square loss of the best weight vector $\bar{w}$, plus a term related to the size of $\bar{w}$ which decays as $1/T$ and an additive offset controlled by the sparsity threshold θ and the gravity parameter g_i.

3 Efficient Implementation

We altered an efficient SGD implementation for least-squares regression according to truncated gradient. It optimizes square loss on a linear representation $w \cdot x$ via (1) with a couple caveats:

1. The prediction is normalized by the square root of the number of nonzero entries in a sparse vector, $w \cdot x/|x|_0^{0.5}$. This alteration is just a constant rescaling on dense vectors which is effectively removable by an appropriate rescaling of the learning rate.
2. The prediction is clipped to the interval $[0, 1]$, implying that the loss function is not square loss for unclipped predictions outside of this dynamic range. Instead the update is a constant value, equivalent to the gradient of a linear loss function.

The learning rate is controllable, supporting $1/i$ decay as well as a constant learning rate (and rates in-between). The program operates in an entirely online fashion. Features are hashed instead of being stored explicitly, and weights can be easily inserted into or deleted from the table dynamically. So the memory footprint is essentially just the number of nonzero weights, even when the numbers of data and features are very large.

In many online-learning situations such as web applications, only a small subset of the features have nonzero values for any example x. It is thus desirable to deal with sparsity only in this small subset rather than in all features, while simultaneously inducing sparsity on all feature weights. The approach we took was to store a time-stamp τ_j for each feature j. The time-stamp was initialized to the index of the example where feature j was nonzero for the first time. During online learning, we simply went through all nonzero features j of example i, and could "simulate" the shrinkage of w^j after τ_j in a batch mode. Specifically, instead of using update rule (3) for weight w^j, we shrunk the weights of all nonzero feature j differently by the following:

$$f(w^j) = T_1\left(w^j + 2\eta(y - \hat{y})x^j, \left\lfloor \frac{i - \tau_j}{K} \right\rfloor K\eta g, \theta\right),$$

and τ_j is updated by

$$\tau_j \leftarrow \tau_j + \left\lfloor \frac{i - \tau_j}{K} \right\rfloor K.$$

In the coefficient rounding algorithm (2), for instance, for each nonzero feature j of example i, we can first perform a regular gradient descent on the square loss, and then do the following: if $|w_j|$ is below the threshold θ and $i \geq \tau_j + K$, we round w_j to 0 and set τ_j to i.

This implementation shows that the truncated gradient method satisfies the following requirements needed for solving large scale problems with sparse features.

- The algorithm is computationally efficient: the number of operations per online step is linear in the number of nonzero features, and independent of the total number of features.
- The algorithm is memory efficient: it maintains a list of active features, and a feature can be inserted when observed, and deleted when the corresponding weight becomes zero.

4 Dataset Summary

The datasets used in our experiments are summarized in Table 1.

Table 1: Dataset Summary.

Dataset	#features	#train data	#test data	task
ad	1411	2455	824	classification
crx	47	526	164	classification
housing	14	381	125	regression
krvskp	74	2413	783	classification
magic04	11	14226	4794	classification
mushroom	117	6079	2045	classification
spambase	58	3445	1156	classification
wbc	10	520	179	classification
wdbc	31	421	148	classification
wpbc	33	153	45	classification
zoo	17	77	24	regression
rcv1	38853	781265	23149	classification
Big_Ads	3×10^9	26×10^6	2.7×10^6	classification