2009 Ninth IEEE International Conference on Data Mining

Inverse Time Dependency in Convex Regularized Learning

Zeyuan Allen Zhu[12*], Weizhu Chen[2], Chenguang Zhu[23], Gang Wang[2], Haixun Wang[2], Zheng Chen[2]

[1]Fundamental Science Class, Department of Physics, Tsinghua University
zhuzeyuan@hotmail.com

[2]Microsoft Research Asia
{v-zezhu, wzchen, v-chezhu, gawa, haixunw, zhengc}@microsoft.com

[3]Department of Computer Science and Technology, Tsinghua University
zcg.cs60@gmail.com

***Abstract*—In the conventional regularized learning, training time increases as the training set expands. Recent work on L_2 linear SVM challenges this common sense by proposing the inverse time dependency on the training set size. In this paper, we first put forward a Primal Gradient Solver (PGS) to effectively solve the convex regularized learning problem. This solver is based on the stochastic gradient descent method and the Fenchel conjugate adjustment, employing the well-known online strongly convex optimization algorithm with logarithmic regret. We then theoretically prove the inverse dependency property of our PGS, embracing the previous work of the L_2 linear SVM as a special case and enable the ℓ_p -norm optimization to run within a bounded sphere, which qualifies more convex loss functions in PGS. We further illustrate this solver in three examples: SVM, logistic regression and regularized least square. Experimental results substantiate the property of the inverse dependency on training data size.**

Keywords – Primal Gradient Solver; inverse time dependency; Fenchel conjugate; regularized learning; online convex optimization

I. INTRODUCTION

In the regularized learning theory, in order to minimize the sum of the *regularization* part and the *loss* part, most of the research works are interested in the *generalization objective* rather than the *empirical objective* [12] [1]. The generalization objective, also known as the *stochastic objective*, is given with respect to a linear predictor $\boldsymbol{w} \in S$, where $S \subset \mathbb{R}^n$ is the domain of $\boldsymbol{w}$:

$$F_\sigma(\boldsymbol{w}) = \sigma \cdot r(\boldsymbol{w}) + l(\boldsymbol{w}) = \sigma \cdot r(\boldsymbol{w}) + \mathbb{E}_{\boldsymbol{\theta}\sim Dist}[l(\langle \boldsymbol{w}, \boldsymbol{\theta}\rangle; \boldsymbol{\theta})] \quad (1)$$

where $r(\boldsymbol{w})$ is the regularizer with a positive weight σ, and $l(\langle \boldsymbol{w}, \boldsymbol{\theta}\rangle; \boldsymbol{\theta})$ is a mapping that calculates the cost or regret by the linear predicting value $\langle \boldsymbol{w}, \boldsymbol{\theta}\rangle$. The expectation is based on a random selection of the sample $\boldsymbol{\theta}$ over the entire sample distribution $Dist$.

Note that the form $\boldsymbol{\theta}$ is used in order to ensure the generality. As an example, $\boldsymbol{\theta}$ can be in the form of $(\boldsymbol{x}, y)$ where $\boldsymbol{x}$ is a vector of features and y is the class identity, adapting (1) to classifications. The loss function l can be for example the SVM hinge loss

$$l(\langle \boldsymbol{w}, \boldsymbol{x}\rangle, y) = \max\{0, 1 - y\langle \boldsymbol{w}, \boldsymbol{x}\rangle\}$$

Practically, an optimization approach for this sort of problem becomes to minimize the *empirical objective* $\hat{F}_\sigma(\boldsymbol{w})$[1] instead, where the average loss over a set of m training samples is used to approximate the generalization loss.

$$\hat{F}_\sigma(\boldsymbol{w}) = \sigma \cdot r(\boldsymbol{w}) + \hat{l}(\boldsymbol{w}) = \sigma \cdot r(\boldsymbol{w}) + \frac{1}{m}\sum_{i=1}^{m} l(\langle \boldsymbol{w}, \boldsymbol{\theta}_i\rangle; \boldsymbol{\theta}_i) \quad (2)$$

The accuracy of a given predictor on some unknown prediction set is strongly associated with equation (1). This naturally leads to a two-step research work: connect (1) and (2) as the step 1, and effectively solve (2) as the step 2.

Step 1. Recently, Léon Bottou *et al* [1] studied the correlation between stochastic and empirical but unregularized objectives and divided the tradeoff into three parts, namely, the approximation, estimation and optimization tradeoff. For regularized learning, Karthik Sridharan *et al* [12] stated that $\hat{F}_\sigma(\boldsymbol{w})$ converges with a rate of $1/m$ to $F_\sigma(\boldsymbol{w})$ for strongly convex objectives.

Step 2. In 2004, T. Zhang [13] introduced the stochastic gradient descent (SGD) algorithm to solve large scale linear prediction problems. It proves that a constant learning rate will numerically achieve some good accuracy, and states the correlation between SGD and online learning. In 2006, Hazan *et al* [3] introduced a framework with logarithmic regret to solve online strongly convex problems, which is the tightest known regret bound for online optimization. Utilizing this result, Shai Shalev-Shwartz *et al* [10] proposed an ℓ_2-norm linear SVM algorithm called PEGASOS.

On the basis of the above two steps, Shai Shalev-Shwartz *et al* [11] presented a surprising result for PEGASOS: assuming the endurable accuracy is given and fixed, the training time has an inverse dependency on the size of the training data, i.e. the larger the dataset is, the faster the program runs to achieve this given accuracy. He claimed that, for example, if we get a predictor with accuracy 95% by training one thousand samples, we can use the extra nine thousand samples to train and get a predictor also with accuracy of 95%, but in less time.

* This work was done when the first author was visiting Microsoft Research Asia. The first author is supported by the National Innovation Research Project for Undergraduates (NIRPU).

[1] In order to distinguish between the two – generalized and empirical, throughout this paper we will use ˆ to denote the empirical functions.

1550-4786/09 $26.00 © 2009 IEEE
DOI 10.1109/ICDM.2009.28

TABLE I. SUMMARY OF TERMINOLOGY

Sample	θ	Generalization objective	$F_\sigma(w) = \sigma \cdot r(w) + l(w)$
Training sample space	$\Psi = \{\theta_1, \dots \theta_m\}$	Regularizer	$r(w)$
The domain of predictor w	S	Generalization loss	$l(\boldsymbol{w}) = \mathbb{E}_{\boldsymbol{\theta} \sim Dist}[l(\langle \boldsymbol{w}, \boldsymbol{\theta} \rangle; \boldsymbol{\theta})]$
Population optimum	$\boldsymbol{w}^* = \text{argmin}_{\boldsymbol{w} \in S} F_\sigma(\boldsymbol{w})$	Empirical objective	$\hat{F}_\sigma(\boldsymbol{w}) = \sigma \cdot r(\boldsymbol{w}) + \hat{l}(\boldsymbol{w})$
Empirical optimum	$\widehat{\boldsymbol{w}} = \text{argmin}_{\text{w} \in \text{S}} \widehat{F_\sigma}(w)$	Empirical loss	$\hat{l}(\boldsymbol{w}) = \frac{1}{m} \sum_{i=1}^{m} l(\langle \boldsymbol{w}, \boldsymbol{\theta_i} \rangle; \boldsymbol{\theta_i})$
Reference predictor	$\boldsymbol{w_0}$	Temporal objective at iter. t	$c_t(\boldsymbol{w}) = \sigma \cdot r(\boldsymbol{w}) + g_t(\boldsymbol{w})$
Our generated predictor	$\widetilde{\boldsymbol{w}}$	Temporal loss at iter. t	$g_t(\boldsymbol{w}) = \frac{1}{\lvert A_t \rvert} \sum_{\boldsymbol{\theta} \in A_t} l(\langle \boldsymbol{w}, \boldsymbol{\theta} \rangle; \boldsymbol{\theta})$
Average number of non-zero features per sample is	d	Optimization error	ϵ_{acc}, satisfies $\hat{F}_\sigma(\mathbf{w}) \leq \hat{F}_\sigma(\widehat{\boldsymbol{w}}) + \epsilon_{acc}$
Dimension of feature space	n	Generalization error	ϵ, satisfies $\forall \boldsymbol{w_0} \in S, l(\widetilde{\boldsymbol{w}}) - l(\boldsymbol{w_0}) \leq \epsilon$

Notice that Shai focuses solely on the ℓ_2-norm linear SVM problem, partially because the ℓ_2-norm is naturally a strongly convex function and the hinge loss in SVM is easy to be handled. However, applying this inverse dependency property into more general problems, like ℓ_p-norm, other loss functions, or other machine learning algorithms, is very desirable, but it is an under-explored research problem.

In this paper, we introduce the Primal Gradient Solver (PGS), which employs the following regularizer:

$$r(\boldsymbol{w}) = \frac{1}{2(p-1)} \|\boldsymbol{w}\|_p^2, p \in (1,2] \tag{3}$$

where the coefficient of $1/2(p-1)$ is to maintain the strong convexity of $r(\boldsymbol{w})$. At the same time, we consider the arbitrary Lipschitz continuous and convex loss function $l(\langle \boldsymbol{w}, \boldsymbol{\theta} \rangle; \boldsymbol{\theta})$. We prove that for a fixed accuracy, our Primal Gradient Solver algorithm can achieve the inverse time dependency on the training data size. This conclusion is also verified in the experiments. We summarize the contributions of this paper as below:

- It proposes a Primal Gradient Solver (PGS) and proves its inverse dependency property. This work generalizes the state-of-the-art ℓ_2-SVM result [11] to ℓ_p-norm and convex loss functions. Notice that the generalization is non-trivial, since the mathematical analysis utilizes a Fenchel conjugate of the regularizer, which lacks an explicit expression in most circumstances.
- By bounding S (the domain of $\boldsymbol{w}$), PGS is able to support more loss functions. For example, Least Square Loss is ineligible for $S = \mathbb{R}^n$ because of its unbounded gradient, but is proved to be acceptable for $S = \{\boldsymbol{w}: \|\boldsymbol{w}\|_p \leq B\}$, where B is a constant large enough to embrace the optimal solution of $\boldsymbol{w}^*$ in S.
- It firstly demonstrates that both logistic loss and least square loss can be adopted into the proposed solver and achieve the inverse dependency property. Extensive experimental results on two machine learning algorithms, logistic regression and regularized least square, substantiate the conclusion.

The reminder of this paper is organized as follows. We first provide mathematical backgrounds on convex optimization theory in Section II. Next in Section III, we propose our main result by introducing our Primal Gradient Solver, and analyzing its inverse dependency property. We further demonstrate our solver in SVM, logistic regression and regularized least square in Section IV, and present experimental results in Section V to substantiate our findings. We provide the theoretical proofs of our main theorems in Section VI. We then raise some discussions in Section VII, and conclude the paper in Section VIII.

II. MATH BACKGROUND AND TERMINOLOGY

Throughout this paper we assume norms to be p-norms, where $p \in [1, \infty) \cup \{\infty\}$. We also summarize the notations used in this paper in TABLE I. Considering the boundedness of some vector x, we will stick to the expression "$\|x\|_p$ is bounded" instead of "x is bounded" for some explicit p.[2] Next in this section, we introduce some definitions frequently used in convex optimization, and a proposition to be used later.

Definition 1: A function $f: S \to \mathbb{R}$ is called L-Lipschitz continuous w.r.t a norm $\|.\|$ if $\forall \boldsymbol{w_1}, \boldsymbol{w_2} \in S$,

$$|f(\boldsymbol{w_1}) - f(\boldsymbol{w_2})| \leq L \cdot \|\boldsymbol{w_1} - \boldsymbol{w_2}\| \tag{4}$$

Definition 2: A function $f: S \to R$ is called σ-strongly convex w.r.t a norm $\|.\|$ if $\forall \boldsymbol{w_1}, \boldsymbol{w_2} \in S, \alpha \in [0,1]$,

$$\begin{aligned} & f(\alpha \boldsymbol{w_1} + (1-\alpha)\boldsymbol{w_2}) \\ & \leq \alpha f(\boldsymbol{w_1}) + (1-\alpha) F(\boldsymbol{w_2}) \\ & - \frac{\sigma}{2} \alpha (1-\alpha) \|\boldsymbol{w_1} - \boldsymbol{w_2}\| \end{aligned} \tag{5}$$

Definition 3: The Fenchel conjugate of a function $f: S \to R$ is defined as:

[2] This is because although in finite dimension, norms are pair-wise bounded $\forall p, q \in [1, \infty) \cup \{\infty\}, \exists C \in \mathbb{R}^+, \forall x, \|x\|_p \leq C \cdot \|x\|_q$ however, the bounding C may hide a constant up to n.

1. INPUT: training sample space $\Psi = \{\theta_1, \dots \theta_m\}$ p, σ, T, k
2. INITIALIZE: $\boldsymbol{w_0} \leftarrow 0, \boldsymbol{\lambda} \leftarrow 0, q \leftarrow 1/(1-1/p)$
3. FOR $t = 1,2, \dots, T$
4. Choose $A_t \subset \Psi$ satisfying $|A_t| = k$
5. Set $g_t(\boldsymbol{w}) \leftarrow \frac{1}{|A_t|}\sum_{\boldsymbol{\theta} \in A_t} l(\langle \boldsymbol{w}, \boldsymbol{\theta} \rangle; \boldsymbol{\theta})$
6. Choose $\boldsymbol{\lambda_t} \in \partial g_t(\boldsymbol{w_{t-1}})$
7. Let $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} - \boldsymbol{\lambda_t}$
8. Define $\boldsymbol{w_t} \leftarrow \nabla r^* \left(\frac{\boldsymbol{\lambda}}{(t+1)\sigma}\right)$ where $r(\boldsymbol{w}) = \frac{1}{2(p-1)}\|\boldsymbol{w}\|_p^2$
9. Return a random $\boldsymbol{w_i} \in \{\boldsymbol{w_1}, \dots \boldsymbol{w_T}\}$ as linear predictor

Figure 1: The Primal Gradient Solver.

$$f^*(\boldsymbol{\theta}) = \sup_{\boldsymbol{w} \in S} \langle \boldsymbol{w}, \boldsymbol{\theta} \rangle - f(\boldsymbol{w}) \tag{6}$$

Example 1: When $S = \mathbb{R}^n$, for $p \in (1,2]$, the function $f(\boldsymbol{w}) = \frac{1}{2(p-1)}\|\boldsymbol{w}\|_p^2$ is 1-strongly convex w.r.t the ℓ_p norm, and its Fenchel conjugate $f^*(\boldsymbol{\theta}) = \frac{1}{2(q-1)}\|\boldsymbol{\theta}\|_q^2$. Here $\frac{1}{p} + \frac{1}{q} = 1$. Proofs can be found in [8] [2]. The strong convexity does not hold for $p > 2$.

Definition 4: The dual norm of the ℓ_p -norm $\|\boldsymbol{x}\|_p = (\sum_i |x|^p)^{1/\mathrm{p}}$ is the ℓ_q -norm $\|\boldsymbol{x}\|_q = (\sum_i |x|^q)^{1/\mathrm{q}}$ if $1/\mathrm{p} + 1/\mathrm{q} = 1$. As a special case, $\|\boldsymbol{x}\|_1 = \sum_i |x_i|$ is dual to $\|\boldsymbol{x}\|_\infty = \max_i |x_i|$.

Definition 5: A vector $\boldsymbol{\lambda}$ is a sub-gradient of a function f at $\boldsymbol{w}$ if for all $\boldsymbol{w'} \in S$ we have $f(\boldsymbol{w'}) - f(\boldsymbol{w}) \geq \langle \boldsymbol{w'} - \boldsymbol{w}, \boldsymbol{\lambda} \rangle$. The differential set of f at $\boldsymbol{w}$ consists of all the sub-gradients and is denoted by $\partial f(\boldsymbol{w})$. When f is differentiable at $\boldsymbol{w}$, $\partial f(\boldsymbol{w})$ contains exactly one element $\partial f(\boldsymbol{w}) = \{\nabla f(\boldsymbol{w})\}$.

Proposition 1: If a function $f: S \to \mathbb{R}$ is L-Lipschitz continuous w.r.t norm $\|.\|_p$, then $\forall \boldsymbol{w} \in S$, the sub- gradient at $\boldsymbol{w}$ is bounded: $\|\boldsymbol{\lambda}\|_q \leq L, \forall \boldsymbol{\lambda} \in \partial f(\boldsymbol{w})$, where $1/\mathrm{p} + 1/\mathrm{q} = 1$.

Proof: By the definition of differential set and Lipschitz continuity, we have for any $\boldsymbol{w'} \in S$,

$$\langle \boldsymbol{w'} - \boldsymbol{w}, \boldsymbol{\lambda} \rangle \leq f(\boldsymbol{w'}) - f(\boldsymbol{w}) \leq L \cdot \|\boldsymbol{w'} - \boldsymbol{w}\|_p$$

By the knowledge of Hölder inequality there exists a $\boldsymbol{w'} \in S$ such that $\langle \boldsymbol{w'} - \boldsymbol{w}, \boldsymbol{\lambda} \rangle = \|\boldsymbol{w'} - \boldsymbol{w}\|_p \|\boldsymbol{\lambda}\|_q$, and combining the above two we arrive at $\|\boldsymbol{\lambda}\|_q \leq L$. ■

III. MAIN RESULT

In this section we first propose a Primal Gradient Solver and state the requirements for the regularizer and the loss function; we then use two theorems to reveal the inverse time dependency, that is, the required running time decreases as the number of samples increases, when achieving a fixed generalization error.

1. INPUT: $\boldsymbol{\lambda}, p, S$. Let n be the feature dimension.
2. FOR $i = 1,2, \dots, n$
3. $w_t^{(i)} \leftarrow \frac{1}{q-1}\left(\sum_j \left|\frac{\lambda^{(j)}}{(t+1)\sigma}\right|^q\right)^{\frac{2}{q}-1} \cdot \left|\frac{\lambda^{(j)}}{(t+1)\sigma}\right|^{q-1} \cdot \operatorname{sgn} \lambda^{(j)}$
4. IF $S = \mathbb{R}^n$, RETURN $\boldsymbol{w_t}$
5. IF $S = \{\boldsymbol{w}: \|\boldsymbol{w}\|_p \leq B\}$
6. IF $\|\boldsymbol{w_t}\|_p > B$ THEN, $\boldsymbol{w_t} \leftarrow \frac{B}{\|\boldsymbol{w_t}\|_q}\boldsymbol{w_t}$
7. RETURN $\boldsymbol{w_t}$

Figure 2: Explicit calculation for $\boldsymbol{w_t} = \nabla \mathrm{r}^*(\boldsymbol{\lambda}/(\mathrm{t}+1)\sigma)$. We use the superscript of the form $^{(j)}$ to denote the j$^{\text{th}}$ coordinate of a vector

A. Primal Gradient Solver

We first introduce the Primal Gradient Solver for the ℓ_p regularized convex optimization problem, assuming $p \in (1,2]$. By substituting the regularizer (3) into (2), we have:

$$\hat{F}_\sigma(\boldsymbol{w}) = \frac{\sigma}{2(p-1)}\|\boldsymbol{w}\|_p^2 + \frac{1}{m}\sum_{i=1}^{m} l(\langle \boldsymbol{w}, \boldsymbol{\theta_i} \rangle; \boldsymbol{\theta_i}) \tag{7}$$

In this paper, we concentrate on the loss function that satisfies the following two assumptions:

- **Convexity**: $l(\langle \boldsymbol{w}, \boldsymbol{\theta} \rangle; \boldsymbol{\theta})$ satisfies the convexity w.r.t. $\boldsymbol{w}$ in S. Pay attention that we do not require the strong convexity here.
- **Lipschitz Continuity**: $l(\langle \boldsymbol{w}, \boldsymbol{\theta} \rangle; \boldsymbol{\theta})$ satisfies L - Lipschitz continuity w.r.t. $\boldsymbol{w}$ and $\|.\|_p$ norm in S, where L is a constant.

We notice that with the help of Proposition 1, the sub-gradient $\partial_{\boldsymbol{w}} l(\langle \boldsymbol{w}, \boldsymbol{\theta} \rangle; \boldsymbol{\theta})$ is bounded w.r.t. $\|.\|_q$. This property will be used later.

Inspired by the work of PEGASOS [10], we propose a Primal Gradient Solver (Figure 1). We take four parameters: the norm parameter p, the weight of the regularizer σ, the number of iterations T, and a given positive integer k. Initially we set $\boldsymbol{w_0} = 0$ and a working vector $\boldsymbol{\lambda} = 0$. At iteration t we randomly choose a set $A_t \subset \Psi, |A_t| = k$, and consider a temporal loss function $g_t(\boldsymbol{w})$ to approximate the empirical loss $\hat{l}(\boldsymbol{w})$:

$$g_t(\boldsymbol{w}) = \frac{1}{|A_t|}\sum_{\boldsymbol{\theta} \in A_t} l(\langle \boldsymbol{w}, \boldsymbol{\theta} \rangle; \boldsymbol{\theta}) \tag{8}$$

The solver then picks up an arbitrary sub-gradient $\boldsymbol{\lambda_t} \in \partial g_t(\boldsymbol{w_{t-1}})$, and subtract it from $\boldsymbol{\lambda}$ by $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} - \boldsymbol{\lambda_t}$. The next $\boldsymbol{w_t}$ is calculated according to the gradient of the Fenchel conjugate (see Section II for definition):

$$\boldsymbol{w_t} = \nabla r^* \left(\frac{\boldsymbol{\lambda}}{(t+1)\sigma}\right) \tag{9}$$

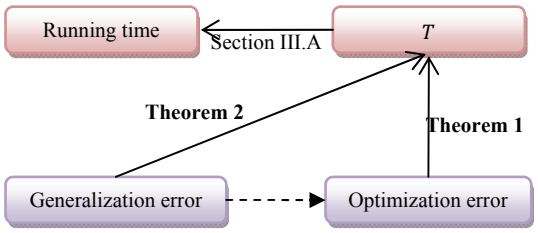

Figure 3: Outline of the proof.

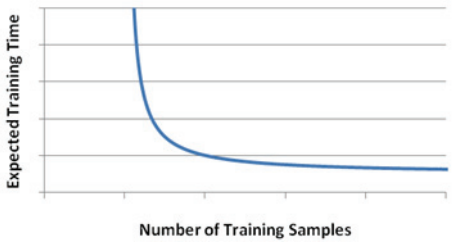

Figure 4: Inverse time dependency with fixed generalization loss

The above process is organized in Figure 1. In Figure 2, we write the explicit formula of Equation (9) for the two cases $S = \mathbb{R}^n$ and $S = \{\boldsymbol{w}: \|\boldsymbol{w}\|_p \leq B\}$. We will show that these two cases cover most of the circumstances in the applications.

- If $S = \mathbb{R}^n$, we recall Example 1 in Section II, and calculate the gradient of $r^*(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_q^2/2(q-1)$ directly. The explicit form is shown on Line 3 in Figure 2.
- If $S = \{\boldsymbol{w}: \|\boldsymbol{w}\|_p \leq B\}$ is bounded, we actually calculate $\boldsymbol{w_t}$ in the same way, but project it back to the p-norm sphere S if it lies outside S (Line 6 of Figure 2). The proof of this can be found in the Appendix, by comparing the results of Corollary 2 and Corollary 3.

Assume the dimension of the feature space, i.e., the dimension of $\boldsymbol{w}$, is n, and the average number of non- zero features per sample is d. If the sub-gradient $\partial g_t(\boldsymbol{w})$ can be computed efficiently in $O(dk)$, the time complexity for the Primal Gradient Solver is $O(T(dk+n))$ since calculating the gradient of r^* costs $O(n)$, as shown in in Figure 2.

Notice that the calculation in Figure 2 gains a speed-up in the special case of $p = 2$, since the term $(\sum_j \dots)^{\frac{2}{q}-1}$ degenerates to 1:

$$w_t^{(\mathrm{i})} = \left|\frac{\lambda^{(j)}}{(t+1)\sigma}\right|^{q-1} \cdot \operatorname{sgn}\lambda^{(j)} \Rightarrow \boldsymbol{w_t} = \frac{\boldsymbol{\lambda}}{(t+1)\sigma} \tag{10}$$

In this case we no longer need $O(n)$ to calculate ∇r^*, as we can use a variable to store the coefficient in front of $\boldsymbol{\lambda}$ and update it in $O(1)$ time, leaving the overall complexity $O(Tdk)$.

B. Inverse Dependency on Training Data Size

In the previous sub-section, we introduced a Primal Gradient Solver for ℓ_p regularized convex optimization, and estimated the running time in terms of the number of iterations. Now we state the correlation between the optimization error and the number of iterations T, which will give us a running time in terms of the optimization error (see Figure 3).

Theorem 1 (To be proved in Section VI.A): If $r(w) = \frac{\sigma}{2(p-1)}\|\boldsymbol{w}\|_p^2$, $g_t(w) = \frac{1}{m}\sum_{i=1}^m l(\langle \boldsymbol{w}, \phi(\boldsymbol{\theta_i})\rangle; \boldsymbol{\theta_i})$, the loss satisfies the convexity and Lipschitz continuity, then $\forall \delta \in (0,1)$, with probability of at least $1-\delta$ over the choices of $A_1, \dots A_T$ and the index i, we have:

$$\hat{F}_\sigma(\boldsymbol{w_i}) \leq \hat{F}_\sigma(\hat{\boldsymbol{w}}) + \frac{C\log T}{\sigma T\delta} \tag{11}$$

Based on the above theorem, if the endurable optimization error is ϵ_{acc}, and satisfies $\hat{F}_\sigma(\boldsymbol{w_i}) \leq \hat{F}_\sigma(\hat{\boldsymbol{w}}) + \epsilon_{acc}$, the algorithm needs $T = \tilde{O}\left(\frac{1}{\sigma\delta\epsilon_{acc}}\right)$ iterations ignoring logarithmic factors.

The optimization error ϵ_{acc} functions as a bridge to the study of the generalization error. We state that if $\widetilde{\boldsymbol{w}}$ is some predictor, optimized by our Primal Gradient Solver, the most immediate reflection of its accuracy is the generalization error ϵ. In some other words, $\forall \boldsymbol{w_0} \in S$, $l(\widetilde{\boldsymbol{w}}) - l(\boldsymbol{w_0}) \leq \epsilon$. The following theorem actually bases on Theorem 1 to further give us a correlation between the generalization error and the number of iterations.

Theorem 2 (To be proved in Section VI.B): Suppose $\widetilde{\boldsymbol{w}}$ is the predictor optimized by the Primal Gradient Solver. If the desired error rate ϵ obeys $l(\widetilde{\boldsymbol{w}}) \leq l(\boldsymbol{w_0}) + \epsilon, \forall \boldsymbol{w_0} \in S$, then the required number of iterations satisfies:

$$T = O\left(\frac{1/\delta}{\frac{2\epsilon^2(p-1)}{\|\boldsymbol{w_0}\|_p^2} - \tilde{O}\left(\frac{1}{m}\right)}\right) \tag{12}$$

Choosing[3] $k = 1$ and integrating (12) into the complexity of the Primal Gradient Solver, we conclude that:

- $p = 2$, the time complexity is $O\left(\frac{d/\delta}{\frac{2\epsilon^2(p-1)}{\|\boldsymbol{w_0}\|_p^2} - \tilde{O}\left(\frac{1}{m}\right)}\right)$
- $p \in (1,2)$, the time complexity is $O\left(\frac{n/\delta}{\frac{2\epsilon^2(p-1)}{\|\boldsymbol{w_0}\|_p^2} - \tilde{O}\left(\frac{1}{m}\right)}\right)$

As illustrated in Figure 4, the time complexity derived from above, decreases as the sample count m increases. This is called the property of inverse time dependency on the training data size. This conclusion confirms the theoretical result in [11] which proves the inverse dependency in the special case of $p = 2$ with the SVM hinge loss.

[3] We will discuss how to choose the best k in the Section VII.

We state that this result comes from the perfect wedding of the following two: when the number of training samples increases

- We expect a smaller gap between the empirical objective and the generalization objective.
- We approximate the loss function more accurately using the random sampling.

IV. Applications

In this section we utilize the Primal Gradient Solver on three specific loss functions. We first consider the binary classification problem with instance-label pairs $\boldsymbol{\theta} = (\boldsymbol{x}, y)$ where $y \in \{-1,1\}$, we have the following two famous demonstrations of the loss functions.

- The SVM hinge loss:
$$l(\langle \boldsymbol{w}, \boldsymbol{\theta} \rangle; \boldsymbol{\theta}) = \max\{0, 1 - y\langle \boldsymbol{w}, \boldsymbol{x} \rangle\}$$
- The Logistic loss:
$$l(\langle \boldsymbol{w}, \boldsymbol{\theta} \rangle; \boldsymbol{\theta}) = \log\left(1 + e^{-y\langle \boldsymbol{w}, \boldsymbol{x} \rangle}\right)$$

If we consider the regression problem with instance-value pairs $\boldsymbol{\theta} = (\boldsymbol{x}, y)$ where $y \in \mathbb{R}$, we have

- The Least Square loss:
$$l(\langle \boldsymbol{w}, \boldsymbol{\theta} \rangle; \boldsymbol{\theta}) = (\langle \boldsymbol{w}, \boldsymbol{x} \rangle - y)^2$$

The convexity of the three loss functions above and the Lipschitz continuity of first two loss functions can be easily verified mathematically, w.r.t the entire space $S = \mathbb{R}^n$. Now we consider the Lipschitz continuity of the Least Square loss. It can be checked this property does not hold in the entire space $S = \mathbb{R}^n$, but we may constrain the space to $S = \{w: \|w\|_p \leq C\}$. For any $\boldsymbol{w_1}, \boldsymbol{w_2} \in S$, using Hölder's inequality we deduce that

$$\begin{aligned} & l(\langle \boldsymbol{w_1}, \boldsymbol{\theta} \rangle; \boldsymbol{\theta}) - l(\langle \boldsymbol{w_2}, \boldsymbol{\theta} \rangle; \boldsymbol{\theta}) \\ & = \langle \boldsymbol{w_1} - \boldsymbol{w_2}, \boldsymbol{x} \rangle (\langle \boldsymbol{w_1}, \boldsymbol{x} \rangle + \langle \boldsymbol{w_2}, \boldsymbol{x} \rangle - 2y) \\ & \leq \|\boldsymbol{w_1} - \boldsymbol{w_2}\|_p \|\boldsymbol{x}\|_q \left(2C\|\boldsymbol{x}\|_q + 2|y|\right) \leq \|\boldsymbol{w_1} - \boldsymbol{w_2}\|_p \cdot L \end{aligned}$$

the last inequality holds for the reason that the sample space is fixed and thus $\|\boldsymbol{x}\|_q$ and $|y|$ are naturally bounded. All we left to do is to further verify the empirical optimum solution $\widehat{\boldsymbol{w}}$ must lie in $S = \{\boldsymbol{w}: \|\boldsymbol{w}\|_p \leq B\}$. This is because $r(\boldsymbol{w}^*) \leq F_\sigma(\boldsymbol{w}^*) \leq F_\sigma(0) \leq (\max|y|)^2$ is bounded, where $\max|y|$ is the upper bound for $|y|$.

Considering the algorithmic framework in Figure 1, we write down $\boldsymbol{\lambda_t}$:

- SVM hinge loss:
$\lambda_t = \frac{1}{|A_t|} \sum_{(\boldsymbol{x},y) \in A_t, y\langle \boldsymbol{x}, \boldsymbol{w_t} \rangle < 1} y \cdot \boldsymbol{x}$
- Logistic loss:
$\lambda_t = \frac{1}{|A_t|} \sum_{(\boldsymbol{x},y) \in A_t} \frac{-y \cdot e^{-y\langle \boldsymbol{x}, \boldsymbol{w_t} \rangle}}{1 + e^{-y\langle \boldsymbol{x}, \boldsymbol{w_t} \rangle}} \boldsymbol{x}$
- Least Square loss:
$\lambda_t = \frac{1}{|A_t|} \sum_{(\boldsymbol{x},y) \in A_t} 2(\langle \boldsymbol{w}, \boldsymbol{x} \rangle - y_i)\boldsymbol{x}$

Therefore, our solver can be properly adapted to these three loss functions. Note that the Lipschitz continuity of the loss function is an important requirement in the deduction (see Section VI). If this requirement is not met, we need to restrict S to some bounded sphere just like we did for the Least Square loss. We emphasize that the introduction of bounded S enables more kinds of convex and continuous functions to be included as loss functions.

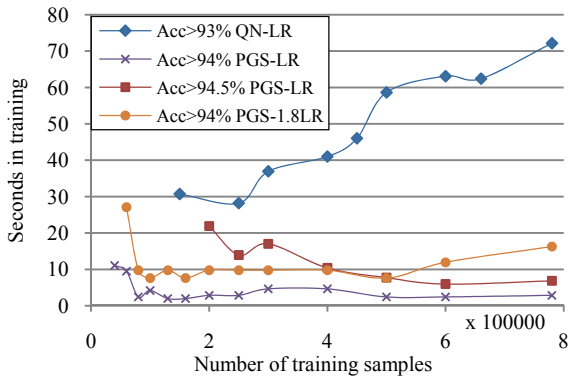

Figure 5: Running time required to achieve given accuracy on CCAT for optimal σ.

Taking the SVM loss with $p = 2$ and $S = \{\boldsymbol{w}: \|\boldsymbol{w}\| \leq 1/\sqrt{\sigma}\}$ as an example, our solver immediately gives the algorithm called PEGASOS [10]. In that paper the proof of the accuracy bound depends on the boundedness of S. However, we used a slightly different Lemma 1 which tells us that even in $S = \mathbb{R}^n$ case our algorithm can still run efficiently. It answers the question in footnote 2 of [11] on why the projection step can be skipped.

V. Experiments

In this section we further strengthen our theoretical result proposed in the previous section by presenting the experimental results. We test our solver in three regularizer-loss pairs: ℓ_2-Logistic, $\ell_{1.8}$-Logistic and ℓ_2-LeastSquare. We do not use the SVM loss here since its ℓ_2-norm counterpart has been well-tested in [11]. All the following works are conducted on a computer with 2.4 GHz AMD Opteron Processor 852 and 32G RAM. We first introduce the dataset in the experiments:

- The binary classification set CCAT retrieved from RCV1 collection [5]. We used 781,265 samples in training and performed prediction on 23,149 testing samples. A total of 47,236 features are in this dataset and with sparsity 0.16%.
- Three toy binary classification sets with 200,000 samples are used where the number of features is 10, 20, and 40 separately. The samples with positive label and with negative label are generated from two Gaussian distributions with different means but the same covariance. Thus, the optimal separating plane is a linear function characterized by a unit vector $\boldsymbol{w}^*$, and can be pre-calculated. Assume the program returns a unit predictor $\boldsymbol{w}$, we will use the error $\|\boldsymbol{w}^* - \boldsymbol{w}\|_2$ to verify its correctness.

Throughout this section, for a given training sample count m, we first choose an optimal $\sigma(m)$ according to the maximal achievable accuracy on the testing set, and then re-run the program to retrieve the required running time to obtain some benchmark accuracies, like 93%, 94%, etc. We

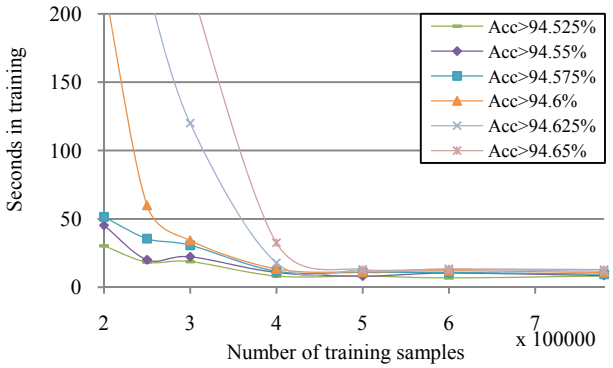

Figure 6: Inverse dependency experiment of 2-norm logistic regression, on CCAT dataset

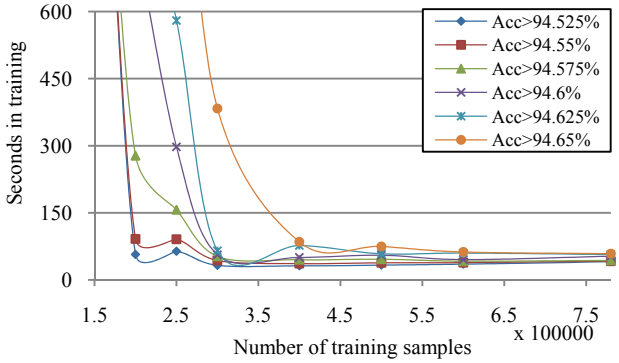

Figure 8: Inverse dependency experiment of 1.8-norm logistic regression, on CCAT dataset

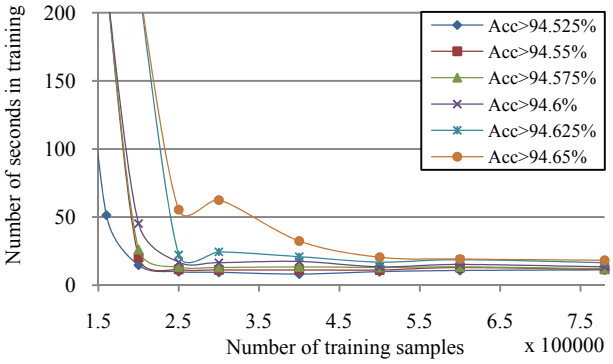

Figure 7: Inverse dependency experiment of 2-norm regularized least square, on CCAT dataset

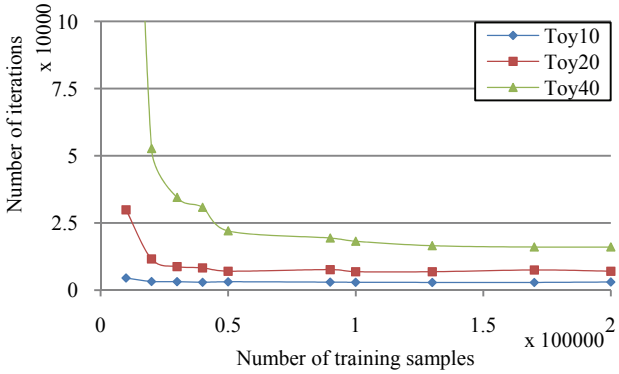

Figure 9: Inverse dependency experiment of 2-norm logistic regression, on toy dataset

remark here that choosing a best σ according to the test data is not scientific [4]. See further discussion in Section VII for details.

In the first experiment we compare our Primal Gradient Solver (PGS) for ℓ_2 and $\ell_{1.8}$ -regularized Logistic Regression (LR) against the L-BFGS Quasi-Newton (QN) method [7] for LR. The latter has been proved to be superior in training large-scale ℓ_2-regularized Logistic Regression by [6]. In PGS, we choose $k = 1$ for $p = 2$ and $k = 300$ for $p = 1.8 \in (1,2)$. The reason for this selection is discussed in Section VII.

As one can see from Figure 5, except for the Quasi-Newton algorithm, the running time of our Primal Gradient Solver does not increase as the sample size m increases, for both $p = 2$ and $p = 1.8 \in (1,2)$. Although QN can achieve an accuracy of the same level as PGS, namely, higher than 94.5%, its running time is above 600 seconds and we ignore it in Figure 5 for the sake of simplicity. It is worth noting that, in the experiment of QN, we also discover the number of iterations inversely dependent on m. However, because each iteration in QN has a time complexity related to m, the total running time of QN still increases. On the contrary, PGS is profited by its stochastic behavior. Not only its number of iterations inversely dependent on m, the time complexity of a single iteration in PGS is also independent on m. It is the combination of these two properties that contributes to the final inverse time dependency.

In the second experiment we check the inverse time dependency for different sets of regularizer-loss pairs against both CCAT data and our toy data. We run our program against a set of distinct sample sizes and record the number of seconds required to reach each accuracy benchmark. Due to the randomness of our Primal Gradient Solver we test our program at least 20 times and choose the median. Notice that although Equation (12) theoretically studies an upper bound in the training time, the decreasing of upper bound does not directly suggest the real-time inverse dependency. Nevertheless, the experimental results in Figure 6, Figure 7, Figure 8 and Figure 9 all confirm the property in (12).

Similar to the first experiment, we set for ℓ_2-norm $k = 1$ and for $\ell_{1.8}$ -norm $k = 300$. The median of 20 runs are used for Figure 6, Figure 7 and Figure 8. Figure 9 demonstrates the number of iterations required for PGS of ℓ_2 Logistic Regression to train our toy data to achieve an error $\|\boldsymbol{w}^* - \boldsymbol{w}\|_2$ of 0.05. The median of 150 runs are used. We state that the time complexity at each iteration is constant and independent on the number of training samples m, so we use the number of iterations to be the y-axis for a better illustration in Figure 9.

In the third experiment, we test our program in CCAT dataset against the optimal solution generated by Quasi-Newton algorithm. We run the QN program with sufficient

TABLE II. THE RUNNING TIME AND ACCURACY OF OUR PRIMAL GRADIENT SOLVER USING AN OPTIMAL σ ON CCAT.

Regularizer	Loss	Optimal σ	QN Accuracy	PGS Accuracy	PGS Training Time
ℓ_2	LogisticRegression	1E-6	0.94799	0.94735±0.00042	55sec
$\ell_{1.8}$	LogisticRegression	4E-7	0.94808	0.94763±0.00035	576sec
ℓ_2	Least Square	2E-5	0.94687	0.94615±0.00060	22sec

The program has been run 20 times and the accuracy is given by "median ± standard" deviation in the table.

number of iterations to reach the convergent solution that minimizes the objective (it takes more than 2 hours). Results in TABLE II. indicate that our Primal Gradient Solver can obtain the accuracy on the same level as Quasi-Newton, while the training time is within 1 minute for ℓ_2 regularized ones, and within 10 minutes for the $\ell_{1.8}$ regularized one.

VI. PROOF OF THE MAIN THEOREMS

In this section we put forward the detailed proofs of the two theorems in Section III.B, using the best known logarithmic regret [4] for online convex optimization [15], and the Oracle inequality in decomposing generalization loss [12].

A. Proof of Theorem 1

According to (8), our temporal objective at iteration t is given by

$$c_t(\boldsymbol{w}) = \sigma \cdot r(\boldsymbol{w}) + g_t(\boldsymbol{w}) \tag{13}$$

We state that $r(\boldsymbol{w}) = \|\mathrm{w}\|_\mathrm{p}^2/2(\mathrm{p}-1)$ is 1-strongly convex (Example 1) and a $g_t(\boldsymbol{w})$ is convex according to our requirement to l. This suggests $c_t(\boldsymbol{w})$ be σ-strongly convex, based on the additivity in [8].

We next examine the counterpart of our problem in online convex optimization, introduced by [15]. In such problem, the ultimate purpose is to minimize the *regret*

$$regret \coloneqq \sum_{t=1}^{T} c_t(\boldsymbol{w_t}) - \min_{\mathrm{w}\in\mathrm{S}} \sum_{t=1}^{T} c_t(\boldsymbol{w}) \tag{14}$$

The following lemma gives a bound for the regret of our Primal Gradient Solver (Figure 1). Its proof can be seen in Theorem 2 in [4].

Lemma 1: Let $c_1, \dots c_T$ be a sequence of σ-strongly convex functions over some convex domain S w.r.t the some norm $\|.\|_p$. Assume $\|.\|_q$ is the dual norm of $\|.\|_p$, then the algorithm defined in Figure 1 satisfies:

$$\sum_{t=1}^{T} c_t(\boldsymbol{w_t}) - \min_{\boldsymbol{w}\in S} \sum_{t=1}^{T} c_t(\boldsymbol{w}) \le \frac{1}{2}\sum_{t=1}^{T} \frac{\|\boldsymbol{\lambda_t}\|_q^2}{t\sigma} \tag{15}$$

Corollary 1: If c_t is defined according the requisites of the Primal Gradient Solver, the above regret is further bounded by $\frac{C\log T}{\sigma}$, where C is a constant.

Proof: This boundedness is ensured if $\|\boldsymbol{\lambda_t}\|_q^2$ is bounded by constant. Recall the Lipschitz continuity for $l(\langle \boldsymbol{w}, \boldsymbol{\theta}\rangle; \boldsymbol{\theta})$, which infers the Lipschitz continuity for g_t. Based on Proposition 1, the ℓ_q-norm of $\boldsymbol{\lambda_t} \in \partial g_t(\boldsymbol{w_{t-1}})$ is bounded, arriving at our conclusion. ■

We now start to calculate the expected optimization error, based on the i.i.d. selection of subsets $A_1, \dots A_T$ and the $\boldsymbol{w_i}$ in Line 9 of Figure 1.

$$\mathbb{E}[\epsilon_{acc}] = \mathbb{E}_{A_1,\dots,A_T}\mathbb{E}_{1\le i\le T}\left[\hat{F}_\sigma(\boldsymbol{w_i})\right] - \hat{F}_\sigma(\widehat{\boldsymbol{w}}) \tag{16}$$

where the empirical optimum $\widehat{\boldsymbol{w}} = \mathrm{argmin}_{\mathbf{w}\in\mathrm{S}} \widehat{F_\sigma}(\boldsymbol{w})$

Using a similar technique from [10], we state that

$$\mathbb{E}_{A_1,\dots,A_T}\mathbb{E}_{1\le i\le T}\left[\hat{F}_\sigma(\boldsymbol{w_i})\right] = \mathbb{E}_{A_1,\dots,A_T}\mathbb{E}_{1\le i\le T}[c_i(\boldsymbol{w_i})]$$

and

$$\hat{F}_\sigma(\widehat{\boldsymbol{w}}) = \mathbb{E}_{A_1,\dots,A_T}\left[\frac{1}{T}\min_{\boldsymbol{w}\in S}\sum_{t=1}^{T} c_t(\boldsymbol{w})\right]$$

substituting them into (16) and using the result of Lemma 1 we have

$$\mathbb{E}[\epsilon_{acc}] \le \frac{C\log T}{\sigma T} \tag{17}$$

Now incorporating the Markov inequality, we provide the proof of theorem 1.

Proof of Theorem 1: The random variable $\epsilon_{acc} = \hat{F}_\sigma(\boldsymbol{w_i}) - \hat{F}_\sigma(\widehat{\boldsymbol{w}}) \ge 0$ is non-negative, and we have $\mathbb{E}[\epsilon_{acc}] \le \frac{C\log T}{\sigma T}$, then using the Markov inequality

$$\begin{aligned} &\Pr\left[\epsilon_{acc} \ge \frac{\mathbb{E}[\epsilon_{acc}]}{\delta}\right]\cdot\frac{\mathbb{E}[\epsilon_{acc}]}{\delta} \le \mathbb{E}[\epsilon_{acc}] \\ &\Rightarrow \Pr\left[\epsilon_{acc} \le \frac{C\log T}{\sigma T\delta}\right] \ge 1-\delta \end{aligned} \tag{18}$$

The above inequality shows that with probability at least $1-\delta$, we have $\epsilon_{acc} \le \frac{C\log T}{\sigma T\delta}$. This immediately gives us the statement. ■

B. Proof of Theorem 2

Proof of Theorem 2: Following [12], we decompose the generalization loss into four parts:

$$\begin{aligned} &l(\widetilde{\boldsymbol{w}}) - l(\boldsymbol{w_0}) \\ &= \left(F_\sigma(\widetilde{\boldsymbol{w}}) - F_\sigma(\boldsymbol{w}^*)\right) + \left(F_\sigma(\boldsymbol{w}^*) - F_\sigma(\boldsymbol{w_0})\right) \\ &- \frac{\sigma}{2(p-1)}\|\widetilde{\boldsymbol{w}}\|_p^2 + \frac{\sigma}{2(p-1)}\|\boldsymbol{w_0}\|_p^2 \end{aligned} \tag{19}$$

here $\widetilde{\boldsymbol{w}}$ is the solution given by our Primal Gradient Solver, population optimum $\boldsymbol{w}^* = \mathrm{argmin}_{\boldsymbol{w}\in S} F_\sigma(\boldsymbol{w})$, and generalization loss $l(\boldsymbol{w}) = \mathbb{E}_{\boldsymbol{\theta}\sim D}[l(\langle \boldsymbol{w}, \boldsymbol{\theta}\rangle; \boldsymbol{\theta})]$.

The second and third term of equation (19) is non-positive, while the first term, the generalization objective difference, can be further bounded by the empirical objective difference according to the main result in [12]. Combining the results along with the optimization accuracy studied in the previous section (Theorem 1), we arrive at the following inequality

$$l(\widetilde{\boldsymbol{w}}) - l(\boldsymbol{w_0}) \leq \tilde{O}\left(\frac{1}{\sigma T\delta}\right) + \frac{\sigma}{2(p-1)}\|\boldsymbol{w_0}\|_p^2 + \tilde{O}\left(\frac{1}{\sigma m}\right) \tag{20}$$

If we choose $\sigma = \widetilde{\Theta}\left(\sqrt{\frac{2(p-1)}{\|\boldsymbol{w_0}\|_p^2}\left(\frac{1}{T\delta} + \tilde{O}\left(\frac{1}{m}\right)\right)}\right)$, the right hand side is bounded as following:

$$l(\widetilde{\boldsymbol{w}}) - l(\boldsymbol{w_0}) \leq \tilde{O}\left(\|\boldsymbol{w_0}\|_p \sqrt{\frac{1}{2(p-1)}\left(\frac{1}{T\delta} + \tilde{O}\left(\frac{1}{m}\right)\right)}\right) \tag{21}$$

Let ϵ equal to the right side of this inequality, we immediately arrive at Theorem 2. ■

VII. Further Discussion

In this section we dialectically analyze the limitation of our Primal Gradient Solver and propose some enhancements. We also discuss some problems raised in the previous sections.

Why p-norm? In the Primal Gradient Solver, the strong convexity is a core requisite to ensure the convergence rate of $\boldsymbol{1/T}$. However, few strongly convex functions found up to now are also suitable to be regularizers. In this paper we examined the squared $\boldsymbol{p} \in (\boldsymbol{1}, \boldsymbol{2}]$ norms, and experimental results show that $\boldsymbol{p} = \boldsymbol{1.8}$ does slightly better than others (for instance, TABLE II.). The reason is still unknown and the choice of $\boldsymbol{p}$ may open an interesting field to study, for example: multiple-regularizer learning.

We notice that $\boldsymbol{p} = \boldsymbol{1}$ is not included in this paper for the reason that 1-norm itself has a poor convexity. However, 1-norm has the often desired property of reducing the number of active features. In [8] [9], Shai proposed a substitute $\boldsymbol{r}(\boldsymbol{w}) = \sum_{i=1}^{n}|\boldsymbol{w_i}|\log|\boldsymbol{w_i}|$ that is strongly convex, which also works well in Primal Gradient Solver with advantages in feature selection.

The adoption of Kernel. All the works above are verified under the assumption that $\boldsymbol{w}$ is a linear predictor. When $p = 2$, a common technique is to construct a mapping $\boldsymbol{\phi}$ that maps from the feature space to the *Reproducing Kernel Hilbert Space* (RKHS) space, availing us a non-linear separator. We emphasize that our Primal Gradient Solver can be slightly adjusted to cater for this assumption, as the calculation of $\langle \boldsymbol{w}, \boldsymbol{\phi}(\boldsymbol{x})\rangle$ needs a traverse on the support vector by $\langle \boldsymbol{w}, \boldsymbol{\phi}(\boldsymbol{x})\rangle = \sum_i\left(\alpha_i \mathcal{K}(\boldsymbol{x_i}, \boldsymbol{x})\right)$. However, due to the complexity cost for this inner product, the inverse time dependency property no longer holds. In a counterpart of this paper [14] we studied the performance of such kernel PGS, and the result shows that even without the inverse time dependency, the algorithm overwhelms the state-of-the-art in both efficiency and accuracy.

Incorporate with a biased term. In our algorithm defined above, we have ignored the biased term in the general loss l. The most efficient way to compensate for it is to add this biased term to the loss function like $\log\left(1 + e^{-y\langle \boldsymbol{w}, \boldsymbol{\phi}(\boldsymbol{x})\rangle + b}\right)$, and at the same time modify the regularizer to

$$\frac{1}{2(p-1)}\left(\|w\|_p^p + b^p\right)^{1/p}$$

Doing this allows us to preserve the strong convexity of the regularizer, but runs into a different way as the normal regularizer without this bias term. If we consistently ignore this term in the regularizer, the convergence rate of our solver will reduce to $O\left(1/\sqrt{T}\right)$ like a generalized convex optimization problem [4].

The selection of parameter $\boldsymbol{k}$. From the above discussion we can see the number of selected samples $k = |A_t|$ is never used within the analysis. Actually, in each iteration we may use the Chernoff bound to boost the confidence and give a better bound than $T = \tilde{O}\left(\frac{1}{\sigma\delta\epsilon}\right)$. Both theoretical analysis and experimental results show that in the $p = 2$ case it is worthless to set $k > 1$; as an alternative, we may choose one single sample each iteration and do $k \cdot T$ iterations in total while the time complexity remains the same and the accuracy is raised.

However, for $p \in (1,2)$ it is not the case. As mentioned in Section III.A, if the complexity of Primal Gradient Solver is $O\left(T(dk + n)\right)$, we had better choose $k = O(n/d)$ which keeps the complexity unchanged but boosts the confidence significantly. For the RCV1 dataset where $n = 47{,}236$ and $D \approx 40$, we may choose $k = 300$, which greatly reduces the number of required iterations. Experimental results in Section V have confirmed this analysis and we will investigate the influence of k more theoretically in the future.

The selection of weight $\boldsymbol{\sigma}$. According to Eq.(12), the running time depends on an unknown vector $\boldsymbol{w_0}$ that is the optimal predictor in training. Similarly, the choice of σ also depends on $\boldsymbol{w_0}$ and we never have such a priori knowledge on how to choose it. A validation set does not work because we are optimizing the running speed and not until we actually know σ we cannot run the program at all. Due to this reason, we are currently working on a modified version of the Primal Gradient Solver that will make σ self-adapted.

VIII. Summary

In this paper we analyzed a Primal Gradient Solver for the ℓ_p-norm regularized convex learning problems that can deal with any loss satisfying the convexity and Lipschitz continuity, including the famous SVM loss, Logistic loss and Least Square loss. For all of them the expected running time is proved to be $O(d/\epsilon_{acc}\delta\sigma)$ for $p = 2$ and $O(n/\epsilon_{acc}\delta\sigma)$ for $p \in (1,2)$, where δ is the confidence parameter, σ is the regularization parameter, d is the average number of non-

zero features for a sample, n is the dimension size of the feature space, and ϵ_{acc} is the desired optimization error.

Experimental results on CCAT dataset in Reuters Corpus Volume 1 (RCV1) show that our Primal Gradient Solver, for all of the three loss functions, approaches an accuracy of 94% within 10 seconds for $p = 2$, and 20 seconds for $p = 1.8$, while the L-BFGS Quasi-Newton method needs 600 seconds to obtain the same accuracy.

The most important contribution of this paper is that, based on this Primal Gradient Solver, we proved it is not only more efficient than the traditional algorithms, but also endowed with inverse time dependency property on the number of training samples, for a fixed accuracy.

This result, confirmed by the dataset of RCV1 and three toy sets, reminds us that even a linear time algorithm might not theoretically meet the best efficiency. There might exist some algorithm, like our Primal Gradient Solver, whose time complexity is independent on the number of samples m, and even inversely dependent on m.

ACKNOWLEDGMENT

Zeyuan Allen Zhu wants to thank Shai Shalev-Shwartz of Hebrew University for his valuable discussions. The authors also acknowledge Matt Callcut and all four anonymous reviewers for their fruitful comments.

REFERENCE

[1] Léon Bottou and Olivier Bousquet, "The Tradeoffs of Large Scale Learning," in *NIPS*, 2007.

[2] Stephen Boyd and Lieven Vandenberghe, *Convex Optimization*, 6th ed.: Cambridge University Press, 2008.

[3] Elad Hazan, Adam Kalai, Satyen Kale, and Amit Agarwal, "Logarithmic Regret Algorithms for Online Convex Optimization," in *COLT*, 2006.

[4] Sham Kakade and Shai Shalev-Shwartz, "Mind the Duality Gap: Logarithmic regret algorithms for online optimization," in *NIPS*, 2009.

[5] David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li, "RCV1: A New Benchmark Collection for Text Categorization Research," *Journal of Machine Learning Research*, vol. 5, pp. 361-397, 2004.

[6] Thomas P. Minka, "A comparison of numerical optimizers for logistic regression," Microsoft Research, Technical Report 2003.

[7] Jorge Nocedal and Stephen J. Wright, *Numerical Optimization, Chapter 6-7*, 2nd ed.: Springer.

[8] Shai Shalev-Shwartz, "Online Learning: Theory, Algorithms, and applications," The Hebrew University, PhD Thesis 2007.

[9] Shai Shalev-Shwartz and Yoram Singer, "Logarithmic Regret Algorithms for Strongly Convex Repeated Games," The Hebrew University, Technical Report 2007.

[10] Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro, "Pegasos: Primal Estimated sub-GrAdient SOlver for SVM," in *ICML*, 2007.

[11] Shai Shalev-Shwartz and Nathan Srebro, "SVM Optimization: Inverse Dependence on Training Set Size," in *ICML*, 2008.

[12] Karthik Sridharan, Nathan Srebro, and Shai Shalev-Shwartz, "Fast Rates for Regularized Objectives," in *NIPS*, 2008.

[13] Tong Zhang, "Solving Large Scale Linear Prediction Problems Using Stochastic Gradient Descent Algorithms," in *ICML*, 2004.

[14] Zeyuan Allen Zhu, Weizhu Chen, Gang Wang, Chenguang Zhu, and Zheng Chen, "P-packSVM: Parallel Primal grAdient desCent Kernel SVM," in *ICDM*, 2009.

[15] Martin Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *ICML*, 2003, pp. 928-936.

APPENDIX

Lemma 2: Let f be a closed and strongly convex function over $S \subset \mathbb{R}^n$ with respect to norm $\|.\|$, then f^* is differentiable and

$$\nabla f^*(\boldsymbol{\theta}) = \underset{\boldsymbol{w} \in S}{\operatorname{argmax}} \langle \boldsymbol{w}, \boldsymbol{\theta} \rangle - f(\boldsymbol{w}) \tag{22}$$

Its proof can be seen from Lemma 6 of [9].

Theorem 3: If $S = \{\boldsymbol{w}: \|\boldsymbol{w}\|_p \le B\}$ and $f(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|_p^2$, let $\frac{1}{p} + \frac{1}{q} = 1$, then

$$f^*(\boldsymbol{\theta}) = \begin{cases} \frac{1}{2}\|\boldsymbol{\theta}\|_q^2, & \|\boldsymbol{\theta}\|_q \le B \\ \frac{1}{2}\|\boldsymbol{\theta}\|_q^2 - \frac{1}{2}\left(B - \|\boldsymbol{\theta}\|_q\right)^2, & \|\boldsymbol{\theta}\|_q > B \end{cases} \tag{23}$$

and

$$\left(\nabla f^*(\boldsymbol{\theta})\right)_i = \frac{\min\{B, \|\boldsymbol{\theta}\|_q\}}{\|\boldsymbol{\theta}\|_q^{q/p}} \theta_i^{q/p} \tag{24}$$

Proof: For any given $\boldsymbol{\theta}$, using the Hölder's inequality we have $\langle \boldsymbol{w}, \boldsymbol{\theta} \rangle \le \|\boldsymbol{w}\|_p \cdot \|\boldsymbol{\theta}\|_q$. Subtracting both sides of them by $\frac{1}{2}\|\boldsymbol{w}\|_p^2$, we have

$$\begin{aligned} &\langle \boldsymbol{w}, \boldsymbol{\theta} \rangle - \frac{1}{2}\|\boldsymbol{w}\|_p^2 \le \|\boldsymbol{w}\|_p \cdot \|\boldsymbol{\theta}\|_q - \frac{1}{2}\|\boldsymbol{w}\|_p^2 \\ &= -\frac{1}{2}\left(\|\boldsymbol{w}\|_p - \|\boldsymbol{\theta}\|_q\right)^2 + \frac{1}{2}\|\boldsymbol{\theta}\|_q^2 \end{aligned} \tag{25}$$

and substituting the definition of f^*, i.e. Eq. (6):

$$\begin{aligned} f^*(\boldsymbol{\theta}) &= \sup_{\boldsymbol{w} \in S} \langle \boldsymbol{w}, \boldsymbol{\theta} \rangle - \frac{1}{2}\|\boldsymbol{w}\|_p^2 \\ &\le \sup_{\boldsymbol{w} \in S} \left(-\frac{1}{2}\left(\|\boldsymbol{w}\|_p - \|\boldsymbol{\theta}\|_q\right)^2 + \frac{1}{2}\|\boldsymbol{\theta}\|_q^2 \right) \\ &= \begin{cases} \frac{1}{2}\|\boldsymbol{\theta}\|_q^2, & \|\boldsymbol{\theta}\|_q \le B \\ \frac{1}{2}\|\boldsymbol{\theta}\|_q^2 - \frac{1}{2}\left(B - \|\boldsymbol{\theta}\|_q\right)^2, & \|\boldsymbol{\theta}\|_q > B \end{cases} \end{aligned} \tag{26}$$

Actually, the equality of Eq. (26) holds, because of an explicit construction of $\boldsymbol{w}$ that satisfies the equality sign in Hölder's inequality. For a given norm $C = \min\{B, \|\boldsymbol{\theta}\|_q\}$, we construct $\boldsymbol{w}^*$ by letting $w_i^* = \frac{C}{\|\theta\|_q^{q/p}} \theta_i^{q/p}$. It can be easily checked that $\|\boldsymbol{w}^*\|_p = C$ and $\langle \boldsymbol{w}^*, \boldsymbol{\theta} \rangle = \|\boldsymbol{w}^*\|_p \cdot \|\boldsymbol{\theta}\|_q$, and the latter holds because $((w_1^*)^p, \ldots (w_n^*)^p)$ and $(\theta_1^q, \ldots \theta_n^q)$ are linear dependent. We remark here that we use x^p for the abbreviation of $|x|^p \cdot \operatorname{sgn} x$. This suffices to prove the equality of Eq. (26), i.e. Eq. (23).

Regarding Eq. (24), we adopt Lemma 2 and see that $\nabla f^*(\boldsymbol{\theta}) = \text{argmin}_{\mathbf{w}\in\mathrm{S}}\left(\|\boldsymbol{w}\|_p - \|\boldsymbol{\theta}\|_q\right)^2 = \boldsymbol{w}^*$. ■

Corollary 2: If $S = \{\boldsymbol{w}: \|\boldsymbol{w}\|_p \le B\}$ and $f(\boldsymbol{w}) = \frac{1}{2(p-1)}\|\boldsymbol{w}\|_p^2$, let $\frac{1}{p} + \frac{1}{q} = 1$, then

$$f^*(\boldsymbol{\theta}) = \begin{cases} \frac{1}{2(q-1)}\|\boldsymbol{\theta}\|_q^2 & \|(p-1)\boldsymbol{\theta}\|_q \le B \\ \frac{1}{2(q-1)}\|\boldsymbol{\theta}\|_q^2 - \frac{1}{2(p-1)}\left(B - \|(p-1)\boldsymbol{\theta}\|_q\right)^2 & \|(p-1)\boldsymbol{\theta}\|_q > B \end{cases} \tag{27}$$

and

$$\left(\nabla f^*(\boldsymbol{\theta})\right)_i = \frac{\min\{B, \|(p-1)\boldsymbol{\theta}\|_q\}}{\|\boldsymbol{\theta}\|_q^{q/p}}\theta_i^{q/p} \tag{28}$$

Proof: Assume $g(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|_p^2$ and we have known g^* according to Theorem 3. Now we calculate f^* using g^*:

$$\begin{aligned} f^*(\boldsymbol{\theta}) &= \sup_{\boldsymbol{w}\in}\langle \boldsymbol{w}, \boldsymbol{\theta}\rangle - \frac{1}{p-1}g(\boldsymbol{w}) \\ &= \frac{1}{p-1}\left(\sup_{\boldsymbol{w}\in S}\langle \boldsymbol{w}, (p-1)\boldsymbol{\theta}\rangle - g(\boldsymbol{w})\right) \\ &= \frac{1}{p-1}g^*\left((p-1)\boldsymbol{\theta}\right) \end{aligned}$$

This immediately gives us Eq. (27) after substituting Eq. (23) and noticing that $p - 1 = \frac{1}{q-1}$. Next, regarding Eq. (28), we make the calculation:

$$\begin{aligned} \left(\nabla f^*(\boldsymbol{\theta})\right)_i &= \frac{1}{p-1}\left(\nabla_{\boldsymbol{\theta}} g^*\left((p-1)\boldsymbol{\theta}\right)\right)_i \\ &= \left(\nabla_{(p-1)\boldsymbol{\theta}}\, g^*\left((p-1)\boldsymbol{\theta}\right)\right)_i \\ &= \frac{\min\{B, \|(p-1)\boldsymbol{\theta}\|_q\}}{\|(p-1)\boldsymbol{\theta}\|_q^{q/p}}(p-1)^{q/p}\theta_i^{q/p} \\ &= \frac{\min\{B, \|(p-1)\boldsymbol{\theta}\|_q\}}{\|\boldsymbol{\theta}\|_q^{q/p}}\theta_i^{q/p} \end{aligned}$$

where the third equality is according to Eq. (24). This completes the proof. ■

Corollary 3: If $S = \mathbb{R}^n$ and $f(\boldsymbol{w}) = \frac{1}{2(p-1)}\|\boldsymbol{w}\|_p^2$, then $f^*(\boldsymbol{\theta}) = \frac{1}{2(q-1)}\|\boldsymbol{\theta}\|_q^2$ and $\left(\nabla f^*(\boldsymbol{\theta})\right)_i = \frac{1}{q-1}\|\boldsymbol{\theta}\|_q^{1-q/p} \cdot \theta_i^{q/p}$