Stochastic Coordinate Descent Methods for Regularized Smooth and Nonsmooth Losses

Qing Tao[1,2], Kang Kong[1], Dejun Chu[1], and Gaowei Wu[2]

[1] New Star Research Inst. of Applied Technology, Hefei 230031, P.R. China
ln.kang.kong,fangboc@gmail.com
[2] Inst. of Automation, Chinese Academy of Sciences, Beijing, 1000190, P.R. China
{qing.tao,gaowei.wu}@ia.ac.cn

Abstract. Stochastic Coordinate Descent (SCD) methods are among the first optimization schemes suggested for efficiently solving large scale problems. However, until now, there exists a gap between the convergence rate analysis and practical SCD algorithms for general smooth losses and there is no primal SCD algorithm for nonsmooth losses. In this paper, we discuss these issues using the recently developed structural optimization techniques. In particular, we first present a principled and practical SCD algorithm for regularized smooth losses, in which the one-variable subproblem is solved using the proximal gradient method and the adaptive componentwise Lipschitz constant is obtained employing the line search strategy. When the loss is nonsmooth, we present a novel SCD algorithm, in which the one-variable subproblem is solved using the dual averaging method. We show that our algorithms exploit the regularization structure and achieve several optimal convergence rates that are standard in the literature. The experiments demonstrate the expected efficiency of our SCD algorithms in both smooth and nonsmooth cases.

Keywords: Optimization Algorithms, Coordinate Descent Algorithms, Nonsmooth and smooth Losses, Large-Scale Learning.

1 Introduction

Given a training set $\mathcal{S} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$, where $(\mathbf{x}_i, y_i) \in \mathbb{R}^N \times Y, Y = \{-1, 1\}$, $\mathbf{x}_i$ is independently drawn and identically distributed, and y_i is the label of $\mathbf{x}_i$. The task of regularized learning is usually cast as the following convex optimization problem,

$$F(\mathbf{w}) = \lambda P(\mathbf{w}) + \sum_{i=1}^{m} f_i(\mathbf{w}) \tag{1}$$

where λ is a trade-off parameter, $P(\mathbf{w})$ is a simple regularizer (such as l_1 or l_2 norm) and $f_i(\mathbf{w})$ is the loss caused by $(\mathbf{x}_i, y_i)$. If the gradient of each $f_i(\mathbf{w})$ is Lipschitz continuous, we call (1) a regularized smooth problem. In the literature [3,6,26], $f_i(\mathbf{w}) = \max\{0, 1 - y_i\langle \mathbf{w}, \mathbf{x}_i\rangle\}^2$ is usually referred to as L_2-loss and $f_i(\mathbf{w}) = \max\{0, 1 - y_i\langle \mathbf{w}, \mathbf{x}_i\rangle\}$ is L_1-loss.

P. Flach et al. (Eds.): ECML PKDD 2012, Part I, LNCS 7523, pp. 537–552, 2012.

Using Coordinate Descent (CD) methods to solve optimization problems has a long history and we refer readers to [24] and [26] that summarize previous work and present comparison for various kinds of CD algorithms. In machine learning, the primal CD operates by sequentially drawing all features, one at a time, and adjusting the learning variables using the closed-form solvers that are based on the single feature only. More precisely, the process from $\mathbf{w}^t$ to $\mathbf{w}^{t+1}$ is called an outer iteration. In each outer iteration, there are N inner iterations so that sequentially the component $w_1^t, w_2^t, \cdots, w_N^t$ are updated. As indicated in [18,19], the low computational complexity at per iteration, the inherently fresh information of updated features and cheap computation of coordinate directional derivatives make CD one of the most efficient optimization techniques in dealing with sparse huge scale problems. In the huge-scale problems in the sense that even the problem's data may be only partially available at the moment of evaluating the current test point, going over all dimensions causes an expensive outer iteration. Instead, one can randomly update only one component of $\mathbf{w}$ at each outer iteration. This kind of methods is referred to as Stochastic CD (SCD).

The practical efficiency of CD has been shown by extensive comparison experiments and an important fact is that the dual CD method for linear SVM in [6] performs very well on large scale document data [26]. However, as pointed out in [3], one should use primal CD when the number of features is much smaller than the number of instances. Although the primal CD methods for regularized learning [3,19,20,24,26] has been receiving much attention, some problems still exist. First, for a general smooth loss especially the commonly used L_2-loss, the existing results either just prove convergence rates [20] or only provide practical algorithms [26] due to the lack of the strong convexity. How to fill the gap between the convergence rate analysis and practical efficiency is still an emergent question. Second, for a nonsmooth loss such as the popularly used hinge loss, there is still no primal SCD algorithm due to the lack of the differentiability. In this paper, we discuss these issues using the structural optimization techniques.

Recently, many remarkable achievements have been made in the area of structural convex optimization [16]. It has been shown that the black-box gradient-type optimization approaches can be replaced by optimization techniques based on a clever use of problem's structure. For nonsmooth problems, Nesterov proposed a primal-dual subgradient method [17] to take the place of the classical Projected Subgradient Algorithm (PSA, [1]). In all situations, this method was proved to be optimal from the viewpoint of worst-case black-box lower complexity bounds. Further, if the objective function is smooth, Nesterov presented novel smooth convex optimization algorithms whose rate of convergence achieve the optimal $O(1/t^2)$ convergence rate in a seminal work [13]. Other variants of this method for minimizing composite objective functions of the form are called APG in [23] and FISTA in [2]. By extending the dual averaging scheme in [17] to regularized problems, an online Regularized Dual Averaging (RDA) method for solving regularized problem (1) was obtained in [25]. In the case of l_1 regularization, RDA can particularly exploit the regularization structure and effectively obtain the sparse solutions. More recently, by using PG, Nesterov proposed an efficient SCD scheme for solving huge-scale smooth optimization problems [18].

In this paper, we present a unified framework for developing CD algorithms from the smooth and nonsmooth structural optimization methods. In particular, for smooth losses, we first extend the smooth CD algorithm in [18] to solve regularized smooth problems with nonsmooth regularizers by using PG. Then we consider the important question of decreasing the factor in the convergence rate and derive an adaptive SCD algorithm using the line search technique. Second, for nonsmooth losses, we present a novel SCD algorithm for regularized nonsmooth losses, in which the randomly selected one-variable subproblem is solved using the RDA method. Since we separately treat the regularizer and loss, all our SCD algorithms can effectively exploit the regularization structure. Theoretical analysis shows that we achieve several optimal rates which are standard in the literature especially for convex and strongly convex problems. Our nonsmooth SCD expands the field of SCD and our smooth SCD fills the gap between convergent rate analysis [20] and practical algorithms [26] especially for L_2-loss.

The experiments show that our nonsmooth SCD outperforms the state-of-the-art solvers in [17,4,21,6]. For regularized smooth loss problems, the toy experiments illustrate that our adaptive smooth SCD outperforms the state-of-the-art solver in [2] while the real experiments demonstrate that it has the same practicality as the state-of-the-art solver in [26].

The rest of this paper is organized as follows. Section 2 and Section 3 discuss SCD algorithms for smooth and nonsmooth losses respectively. Experimental results are reported in Section 4 and conclusions are made in the last section.

2 SCD Algorithms for Smooth Losses

It is well-known that the complexity of a convex optimization problem is closely linked with its level of smoothness on the first derivatives of functional components [16]. If more information on the smoothness about f is available, higher performance algorithms are expected. In this section, we assume that ∇f is Lipschitz continuous. The Lipschitz condition of ∇f measures how well f is approximated at some point by its linearization.

For a primal CD method, the optimization process starts from an initial point $\mathbf{w}^0$ and generates a sequence of vectors $\{\mathbf{w}^t\}$. Each outer iteration generates vectors $\mathbf{w}^{t,i} \in \mathbb{R}^N$, $i = 1, 2, \cdots, N+1$, such that $\mathbf{w}^{t,1} = \mathbf{w}^t$, $\mathbf{w}^{t,N+1} = \mathbf{w}^{t+1}$ and $\mathbf{w}^{t,i} = [w_1^{t+1}, \cdots, w_{i-1}^{t+1}, w_i^t, \cdots, w_N^t]^T, \forall i = 2, \cdots, N$. The concerned one-variable sub-problem is $\min_z \lambda P(\mathbf{w}^{t,i} + z\mathbf{e}_i) + a_i(z)$, where $a_i(z) = f(\mathbf{w}^{t,i} + z\mathbf{e}_i)$ and $\mathbf{e}_i = [0, \cdots, 0, 1, 0, \cdots, 0]^T$.

As L_2-loss is only Lipschitz differentiable but not twice differentiable and $P(\mathbf{w}) = \|\mathbf{w}\|_1$ is non-differentiable, certain special considerations in generalizing the second derivative are given in [3]. First, in order to derive the closed-form solution of single-variable sub-problem for L_2-loss, the following second-order approximation of the loss term is adopted

$$\min_z a_i'(0)z + \frac{1}{2}a_i''(0)z^2 + \lambda p(w_i^{t,i} + z) - \lambda p(w_i^{t,i}) \tag{2}$$

where $a_i''(0)$ is the generalized second derivative defined in [26]and [3], $p(\omega) = \omega^2$ when $P(\mathbf{w}) = \|\mathbf{w}\|_2^2$ and $p(\omega) = |\omega|$ when $P(\mathbf{w}) = \|\mathbf{w}\|_1$. On the other hand, to ensure the convergence of l_1 regularized CD algorithms, the line search strategy in [24] is modified in [26] to find γ,

$$F(\mathbf{w}^{t,i} + \gamma d\mathbf{e}_i) - F(\mathbf{w}^{t,i}) \leq \sigma\gamma[a_i'(0)d + \lambda|w_i^{t,i} + d| - \lambda|w_i^{t,i}|] \tag{3}$$

where d is the solution of (2), σ is any constant in $(0, 1)$, $\beta \in (0, 1)$, and $\gamma = \max\{1, \beta, \beta^2, \ldots,\}$ such that γd satisfies (3). By solving (2) and (3), the inner update from $\mathbf{w}^{t,i}$ to $\mathbf{w}^{t,i+1}$ is $w_i^{t,i+1} = w_i^{t,i} + \gamma d$.

In the following, we will construct CD based on the gradient methods for smooth functions. Obviously, there exist a constant $L(f) > 0$ such that $\|\nabla f(\mathbf{w}) - \nabla f(\mathbf{u})\| \leq L(f)\|\mathbf{w} - \mathbf{u}\|, \forall \mathbf{w}, \mathbf{u} \in \mathbb{R}^N$. Then, for any $L \geq L(f)$,

$$f(\mathbf{w}) \leq f(\mathbf{u}) + \langle \mathbf{w} - \mathbf{u}, \nabla f(\mathbf{u})\rangle + (L/2)\|\mathbf{w} - \mathbf{u}\|^2 \tag{4}$$

for every $\mathbf{w}, \mathbf{u} \in \mathbb{R}^N$. Let $l_f(\mathbf{w}, \mathbf{u}) = f(\mathbf{u}) + \langle \mathbf{w} - \mathbf{u}, \nabla f(\mathbf{u})\rangle$. The key operation in PG [23] is

$$\mathbf{w}^{t+1} = \arg\min_{\mathbf{w}}\{l_f(\mathbf{w}, \mathbf{w}^t) + \lambda P(\mathbf{w}) + (L/2)\|\mathbf{w} - \mathbf{w}^t\|^2\} \tag{5}$$

It is easy to find that (5) is separable and has an entry-wise closed-form solution. Motivated by the CD for L_2-loss in [3], we can solve (5) in only one of the N components at each step. To be more precise, the randomly selected one-variable sub-problem now is

$$w_i^{t+1} = \arg\min_{\omega}\left\{\omega(g_i^t - Lw_i^t) + \lambda p(\omega) + (L/2)\omega^2\right\} \tag{6}$$

If d is the solution of (6), the update from $\mathbf{w}^t$ to $\mathbf{w}^{t+1}$ is $\mathbf{w}^{t+1} = \mathbf{w}^t + d\mathbf{e}_i$. We describe our SCD algorithm for regularized smooth losses in Algorithm 1.

Algorithm 1. Smooth SCD

```
Initialize a weight vector $\mathbf{w}^1$.
repeat
    1. Choose $i \in \{1, 2, \ldots, N\}$ uniformly at random
    2. calculate $g_i^t$
    3. solve (6) and update $\mathbf{w}^{t+1}$
    4. $t := t + 1$
until a stopping condition is satisfied
```

If we let $\lambda = 0$ or $P(\mathbf{w})$ be the indicator function of a closed convex set, Smooth SCD recovers the SCD scheme including both unconstrained and constrained minimizations in [18]. After t iterations, Nonsmooth SCD generates a random output $[\mathbf{w}^t, F(\mathbf{w}^t)]$. Obviously, $[\mathbf{w}^t, F(\mathbf{w}^t)]$ depends on the random variable $\xi_t = \{i_0, i_1, i_2, \ldots i_t\}$, where i_t is independently and randomly chosen from the set $\{1, 2, \cdots, N\}$ with probability $1/N$. We denote $\phi_t = \mathbf{E}_{\xi_{t-1}} F(\mathbf{w}^t)$. In the following, we extend the convergence theorem in [18] to regularized learning problems. Even when $\lambda = 0$, our proof is new and rather concise (see Appendix).

Theorem 2.1. *Let $\mathbf{w}^*$ be the optimal solution of (1) in $\mathbb{F}$. Assume $\{\mathbf{w}^t\}$ is generated by Smooth SCD. Then*
i) If $P(\mathbf{w}) = \|\mathbf{w}\|_1$, $\phi_t - F(\mathbf{w}^*) \le O(L/t)$.
ii) If $P(\mathbf{w}) = \|\mathbf{w}\|_2^2$, there exists a $0 < \mu < 1$ such that $\phi_t - F(\mathbf{w}^*) \le O(\mu^t)$.

By fixing the expected accuracy of solution and the confidence level, we can also derive the same orders of convergence rates of Smooth SCD with high probability as that in [21] and [25]. When $P(\mathbf{w}) = \|\mathbf{w}\|_2^2$, Theorem 2.1 indicates that we have achieved optimal convergence rates that are standard in the literature for strongly convex optimization [18]. If the loss function is smooth but not strongly convex such as L_2-loss, we obtain the convergence rate $O(L/t)$ for its l_1 regularization. However, the rate $O(L/t)$ is not optimal for general convex smooth losses. In a series of work, Nesterov proposed several methods to accelerate convergence of PG. They obtain the optimal convergence rate $O(L/t^2)$ that are standard in the literature [14,15]. As an extension of Nesterov's accelerated method [13], a shrinkage-thresholding Accelerated Proximal Gradient (APG) algorithm was recently proposed in [2]. This accelerated scheme for gradient methods can be done also for the SCD schemes, and several variants that can reach convergence rate $O(L/t^2)$ has been discussed in [18]. Unfortunately for some applications, as pointed out in [18], the complexity of one iteration of the accelerated scheme is rather high since the computation of full-dimensional vectors has to be concerned. As the focus in this paper is only on stochastic algorithms, we will not discuss the accelerated SCD. However, we will compare with the batch APG [2] in the experiments.

A possible drawback of the above scheme especially for stochastic learning is that the Lipschitz constant $L(f)$ is not always known or computable. In the optimization process, $L(f)$ plays a dominant part as the stepsize. The selection of stepsize severely affects the performance of optimization methods even in the stochastic setting. It has been indicated in [26] that SCD is much slower than the corresponding deterministic methods when a too large upper bound of the second derivative is used. This fact indicates that the factor L in convergence rate $O(L/t)$ is extremely useful in practice. We therefore analyze an adaptive SCD algorithm with compact Lipschitz constants in the following.

To ensure the holding of Theorem 2.1, the local Lipschitz condition should be satisfied in each iteration, i.e., for any $\mathbf{w} \in [\mathbf{w}^t, \mathbf{w}^{t+1}]$, there exists a constant L_t such that

$$f(\mathbf{w}) \le f(\mathbf{w}^t) + \langle \mathbf{w} - \mathbf{w}^t, \nabla f(\mathbf{w}^t) \rangle + (L_t/2)\|\mathbf{w} - \mathbf{w}^t\|^2 \tag{7}$$

with $\inf_{t \ge 1}\{L_t\} > 0$. Obviously, L_t can be roughly selected as an upper bound on the second derivative of the loss. To find a more compact L_t at each step, it is intuitive to use the second-order derivative of f at $\mathbf{w}^t$ as an initial point for line search. Specifically, let $a_i(z) = f(\mathbf{w}^t + z\mathbf{e}_i)$, we solve the following randomly selected single-variable sub-problem

$$\min_z a_i'(0)z + \frac{1}{2}a_i''(0)z^2 + \lambda p(w_i^t + z) - \lambda p(w_i^t) \tag{8}$$

where $a_i''(0)$ is a subgradient of $a_i'(0)$. To get the solution of (8) in closed-form, we restrict

$$a_i''(0) = \epsilon \quad if \qquad a_i''(0) < \epsilon \tag{9}$$

where ϵ is a predefined sufficiently small positive number. We use the following backtracking line search strategy to find γd,

$$f(\mathbf{w}^t + \gamma d\mathbf{e}_i) - f(\mathbf{w}^t) \leq \gamma[a_i'(0)d + (\gamma/2)d^2] \tag{10}$$

where d is the solution of (8), $\beta \in (1, \infty)$ and $\gamma = \min\{1, \beta, \beta^2, \ldots, \}a_i''(0)$ such that γd satisfies (10). By solving (8) and (10), the update from $\mathbf{w}^t$ to $\mathbf{w}^{t+1}$ is $\mathbf{w}^{t+1} = \mathbf{w}^t + \gamma d\mathbf{e}_i$. We describe our adaptive smooth SCD in Algorithm 2, which is a coordinate-wise version of ISTA with backtracking in [2]. As discussed in [2], Algorithm 2 has the same order of convergence rate as Algorithm 1.

Algorithm 2. Adaptive Smooth SCD

```
Initialize a weight vector $\mathbf{w}^1$ and choose $\beta \in (1, \infty)$.
repeat
  1. Choose $i \in \{1, 2, \ldots, N\}$ uniformly at random
  2. calculate $a_i'(0)$ and $a_i''(0)$ according to (9)
  3. calculate $d$ via (8)
  4. compute $\gamma = \min\{1, \beta, \beta^2, \ldots, \}a_i''(0)$ such that $\gamma d$ satisfies (10) and update $\mathbf{w}^t$
  5. $t := t + 1$
until a stopping condition is satisfied
```

In [20], an $O(L/t)$ convergence rate for SCD was indeed obtained, but it didn't discuss the linear convergence rate for strongly convex objective functions and only focused on several specific data sets with fixed Lipschitz constants. In [26], a practical CD method using one-dimensional Newton direction (CDN) to minimize the second-order approximation in (2) was proposed. CDN uses the line search strategy (3) to find the parameter γ and the optimal linear convergence rate has been obtained when dealing with strongly convex objective functions (see also [3,6]), but its convergence rate for L_2-loss with l_1 regularizer was not explicitly described in [26]. At first sight, our line search strategy (10) is only a little different from the strategy (3). Nevertheless, our strategy looks rather natural and flexible. Further, the principles behind (10) and (3) are quite different, i.e., the goal of the former is to guarantee the effectiveness of a local Lipschitz expansion and decrease the factor in convergence rates while that of the latter is only to ensure sufficient decrease of the objective function. At this point, we have established a close link between the practical CD algorithms in [24,26] and principled structural optimization techniques. Note that (10) has the same computational cost as (3), thus we believe that SCD algorithms developed in this section can fill the gap between convergence rate analysis and practical efficiency. Compared with the SCD in [20] and CDN in [26], our method is interesting in both theory and practice.

3 SCD Algorithms for Nonsmooth Losses

In this section, we only assume that f is only continuous and convex. Since the generalized second derivative of f doesn't exist, we can not discuss CD algorithms along the lines in Section 2. This is the main obstacle to establish CD algorithms for nonsmooth losses. Note that many SCD algorithms as well as their convergence heavily depend on the associated corresponding full-gradient algorithms [18,20]. This fact motivates us to start from the nonsmooth deterministic optimization method.

In [17], a dual averaging method was presented for different types of nonsmooth problems only requiring the subgradient information. At each iteration, the learning variables are adjusted by solving a simple minimization problem that involves running average of all past subgradients that emphasizes more recent gradients. RDA is an extension of this method, which can solve online regularized learning problems [25]. More specifically, the key iteration of batch RDA takes the form

$$\mathbf{w}^{t+1} = \arg\min_{\mathbf{w}} \left\{ \langle \bar{\mathbf{g}}^t, \mathbf{w} \rangle + \lambda P(\mathbf{w}) + (\beta_t/t) h(\mathbf{w}) \right\} \tag{11}$$

where $\bar{\mathbf{g}}^t = \frac{1}{t}\sum_{j=1}^{t} \nabla f(\mathbf{w}^j)$, $\nabla f(\mathbf{w}^j)$ is a subgradient of f at $\mathbf{w}^j$, $h(\mathbf{w})$ is an auxiliary strongly convex function, and $\{\beta_t\}_{t\geq 1}$ is a nonnegative and nondecreasing input sequence. We describe RDA for batch learning in Algorithm 3.

Algorithm 3. Batch RDA

```
Initialize a weight vector $\mathbf{w}^1 = \mathbf{0}$ and $\bar{\mathbf{g}}^0 = \mathbf{0}$.
repeat
    1. compute $\mathbf{g}^t = \nabla f(\mathbf{w}^t)$
    2. update $\bar{\mathbf{g}}^t = [(t-1)\bar{\mathbf{g}}^{t-1} + \mathbf{g}^t]/t$
    3. compute $\mathbf{w}^{t+1}$ via (11)
    4. $t := t + 1$
until a stopping condition is satisfied
```

For simplicity, we choose $h(\mathbf{w}) = (1/2)\|\mathbf{w}\|_2^2$ throughout this paper. According to (11), $\mathbf{w}^{t+1}$ can be found in closed-form with little effort. This is the main reason that RDA methods can successfully deal with large scale problems and exploit the regularization structure. As optimization problem (11) is separable, we can get $\mathbf{w}^{t+1}$ by solving each w_i^{t+1} independently, i.e.,

$$w_i^{t+1} = \arg\min_{\omega} \left\{ \omega \bar{g}_i^t + \lambda p(\omega) + (\beta_t/2t)\omega^2 \right\} \tag{12}$$

where $\bar{g}_i^t$ denotes the i-th component of $\frac{1}{t}\sum_{j=1}^t \nabla f(\mathbf{w}^j)$.

To conduct convergence analysis, we gather the following assumptions from [25], although some of them don't appear explicitly in Theorem 3.1 and 3.2.

Assumption 3.1. *There exists a constant* $M > 0$ *such that* $\|\nabla_i f(\mathbf{w})\| \leq M$, $\forall \mathbf{w} \in \mathbb{R}^N$, $1 \leq \forall i \leq N$ and $\max\{\sigma_1, \beta_1\} > 0$, where σ_1 is dedicated to the convexity parameter of $P(\mathbf{w})$. If $P(\mathbf{w}) = \|\mathbf{w}\|_1$, β_t is order exactly $\sqrt{t}$. If $P(\mathbf{w}) = \|\mathbf{w}\|_2^2$, $\beta_t \leq O(\ln t)$.

[25] gave several precise regret bounds of the RDA method for solving regularized online problems. The convergence rates for stochastic learning problems can be established based on these regret bounds. If the regularizer is general convex such as $P(\mathbf{w}) = \|\mathbf{w}\|_1$, the online RDA has an $O(\sqrt{t})$ regret bound. If the regularization term is strongly convex such as $P(\mathbf{w}) = \|\mathbf{w}\|_2^2$, the online RDA has an $O(\ln t)$ regret bound. As a direct consequence of regret analysis in [25], we can get

Theorem 3.1. *Let* $\mathbb{F}_D = \{h(\mathbf{w}) \leq D^2\}$ *and* $\mathbf{w}^*$ *be the optimal solution of (1) in* $\mathbb{F}_D$. *Assume* $\{\mathbf{w}^t\}$ *is generated by Batch RDA and* $\bar{\mathbf{w}}^t = \frac{1}{t}\sum_{j=1}^t \mathbf{w}^j$. *Then*
i) If $P(\mathbf{w}) = \|\mathbf{w}\|_1$, $F(\bar{\mathbf{w}}^t) - F(\mathbf{w}^*) \leq O(\frac{MD\sqrt{t}}{t})$.
ii) If $P(\mathbf{w}) = \|\mathbf{w}\|_2^2$, $F(\bar{\mathbf{w}}^t) - F(\mathbf{w}^*) \leq O(\frac{(2D^2+\frac{M^2}{4})(1+lnt)}{t})$.

In order to derive CD algorithms for regularized nonsmooth losses, we at each step only solve (11) on one component which is randomly selected from total N components instead of separately going over all the components in the batch setting. In particular, if d is the solution of the randomly selected one-variable subproblem in the form of (12), the update from $\mathbf{w}^t$ to $\mathbf{w}^{t+1}$ becomes $\mathbf{w}^{t+1} = \mathbf{w}^t + d\mathbf{e}_i$. We describe our primal SCD algorithm for nonsmooth losses in Algorithm 4.

Algorithm 4. Nonsmooth SCD

```
Initialize a weight vector $\mathbf{w}^1 = \mathbf{0}$ and $\bar{\mathbf{g}}^0 = \mathbf{0}$.
repeat
    1. Choose $i \in \{1, 2, \ldots, N\}$ uniformly at random
    2. let $g_i^t$ be the $i$-th element of $\nabla f(\mathbf{w}^t)$
    3. update $\bar{g}_i^t = [(t-1)\bar{g}_i^{t-1} + g_i^t]/t$
    4. solve (12) and update $\mathbf{w}^{t+1}$
    5. $t := t + 1$
until a stopping condition is satisfied
```

To measure the stochastic quality of the solutions $\mathbf{w}^1, \ldots, \mathbf{w}^t$, we first prove (in Appendix)

Lemma 3.1. *Assume* $\mathbf{w}^t$ *is generated by Nonsmooth CD.* $\forall \mathbf{w} = (w_1, w_2, \ldots, w_N)^T \in \mathbb{F}_D$, *define* $\delta_t(\mathbf{w}) = \sum_{\tau=1}^t \{g_{i_\tau}^\tau (w_{i_\tau}^\tau - w_{i_\tau}) + \lambda p(w_{i_\tau}^\tau)\} - \frac{1}{N} t\lambda P(\mathbf{w})$ *and* $R_t(\mathbf{w}) = \sum_{\tau=1}^t \{\phi_\tau - F(\mathbf{w})\}$. *Then* $R_t(\mathbf{w}) \leq N\mathbf{E}_{\xi_t}\delta_t(\mathbf{w})$

From the proof of Lemma 3.1,$N\mathbf{E}_{\xi_t}\delta_t(\mathbf{w}) = \mathbf{E}_{\xi_t}\sum_{\tau=1}^{t}[\mathbf{g}^{\tau}(\mathbf{w}^{\tau} - \mathbf{w}) + \lambda P(\mathbf{w}^{\tau}) - \lambda P(\mathbf{w})]$. In online learning, the bound of $\sum_{\tau=1}^{t}\{\mathbf{g}^{\tau}(\mathbf{w}^{\tau} - \mathbf{w}) + \lambda P(\mathbf{w}^{\tau}) - \lambda P(\mathbf{w})\}$ has been analyzed in [25]. The regret, primal variable and dual average can be bounded based on this bound. Following similar arguments, we can derive

Theorem 3.2. *Let $\mathbf{w}^*$ be the optimal solution of (1) in $\mathbb{F}_D$. Assume $\{\mathbf{w}^t\}$ is generated by Stochastic Nonsmooth CD. Then*
i) If $P(\mathbf{w}) = \|\mathbf{w}\|_1$, $\frac{1}{t}\sum_{\tau=1}^{t}\phi_{\tau} - F(\mathbf{w}^*) \leq O(\frac{MD\sqrt{t}}{t})$.
ii) If $P(\mathbf{w}) = \|\mathbf{w}\|_2^2$, $\frac{1}{t}\sum_{\tau=1}^{t}\phi_{\tau} - F(\mathbf{w}^*) \leq O(\frac{(2D^2+\frac{M^2}{4})(1+lnt)}{t})$.

In addition to convergence in expectation, we can also derive the same orders of convergence rates with high probability. Theorem 3.2 indicates that we have achieved the optimal convergence rates that are standard in the literature for convex and strongly convex nonsmooth losses minimization. At first sight, online RDA in [25] and our Nonsmooth SCD share the same idea in principle, i.e., both of them approach the solution of (1) by optimizing the regularized dual averaging objective function defined in [17]. However, the former accomplishes the task of sparse online learning and the latter expands the field of SCD algorithms.

4 Experiments

In this section, we will present experiments to validate our theoretical analysis and demonstrate the performance of our algorithms. Typically, we consider one toy data set and four large scale data sets. The toy data set with 800 samples in $\mathbb{R}^{800}$ is generated like [4], i.e., we choose a $\mathbf{w}$ with entries distributed normally with 0 mean and unit variance and randomly zeroed 50% of the vector, the data matrix $\mathbf{X} \in \mathbb{R}^{800\times800}$ was random with entries also normally distributed, and we set $\mathbf{y} = \mathbf{X}\mathbf{w} + \mathbf{v}$, where the components of $\mathbf{v}$ were also distributed normally at random. The four real data sets are described in Table 1. We do not include the bias term for all the solvers. All algorithms are implemented in C++ and all the experiments are run on a Sun Ultra 45 Workstation with 1.6GHz UltraSPARC IIIi processor and 4GB of main memory under Solaris 10. The trade-off parameter λ is chosen by using the cross validation strategy. To have a fair comparison, each stochastic algorithm is run 10 times and the reported are averaged results. We find that SCD achieves consistently better test accuracy than other solvers. For clarity, we category the experiments into nonsmooth and smooth loss problems.

Nonsmooth Loss Problems. We first consider the hinge loss with l_1 regularizer (l_1-R-L_1) problem. It has been shown in [4] that the variants of stochastic gradient projection methods augmented with L1 efficient projection procedures outperform many optimization techniques such as exponentiated gradient algorithm. Specifically, we can employ the efficient projection algorithm in [4] to implement the PSA for hinge loss (L1-PSA) ([22]). Since few papers study large scale l_1-R-L_1 problems and they are excluded from the comparison in [26], to illustrate the scalability of our Nonsmooth SCD, we choose to compare

Table 1. Real Data-sets where the split describes the size of a train/test set

Data-set	Dimension	Split
Astro-Physics	99,757	29,882/32,487
CCAT	47,236	23,149/781,265
A9a	123	24,703/7,858
Covtype	54	522,911/58,101

our Nonsmooth SCD with L1-PSA and Batch RDA. In the experiments, Nonsmooth SCD obtains the same level of sparsity as Batch RDA. The relationship between$|(F(\mathbf{w}^t) - F(\mathbf{w}^*)|/|F(\mathbf{w}^*)|$ vs. CPU time is illustrated in Fig. 1, Fig. 2, Fig. 3 and Fig. 4.

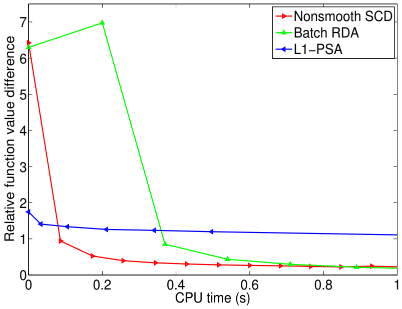

Fig. 1. l_1-R-L_1 on Astro-ph

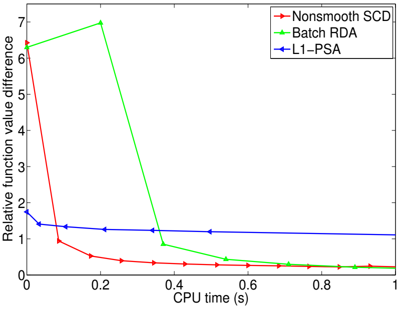

Fig. 2. l_1-R-L_1 on CCAT

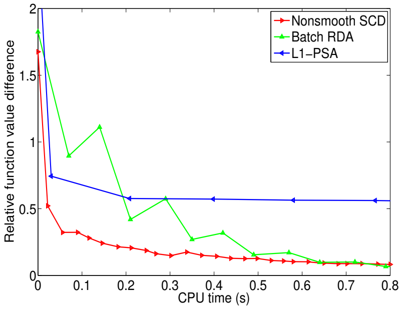

Fig. 3. l_1-R-L_1 on A9a

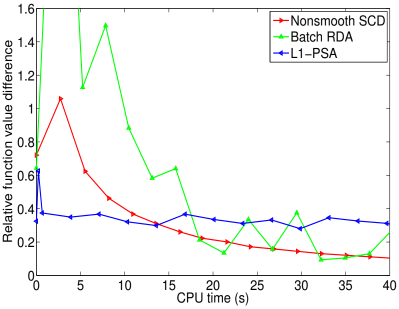

Fig. 4. l_1-R-L_1 on Covertype

We then consider the hinge loss with l_2 regularizer (l_2-R-L_1). For this problem, one of the most efficient primal algorithms is Pegasos in [21] and the state-of-the-art dual algorithm is Dual SCD in [6]. In particular, the experiments in [6] indicate that Dual CD is much faster than many solvers such as Pegasos, TRON [9], SVM$^{\text{perf}}$ [7]. To illustrate the scalability of our Nonsmooth SCD, we choose to compare with Pegasos ($\lambda_{\text{pegasos}} = 2\lambda_{\text{SCD}}/m$, [6]), Dual SCD and Batch RDA.

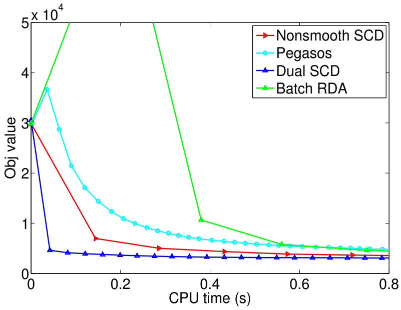

Fig. 5. l_2-R-L_1 on Astro-ph

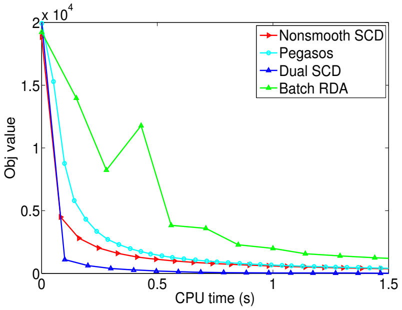

Fig. 6. l_2-R-L_1 on CCAT

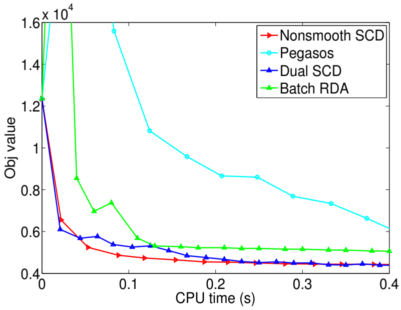

Fig. 7. l_2-R-L_1 on A9a

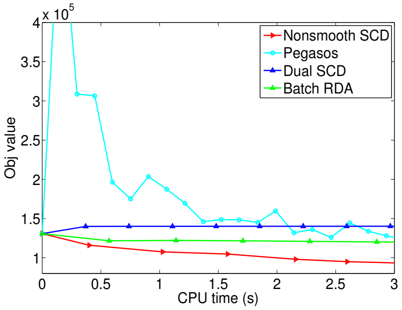

Fig. 8. l_2-R-L_1 on Covertype

The relationship between $F(\mathbf{w}^t) - F(\mathbf{w}^*)$ vs. CPU time is illustrated in Fig. 5, Fig. 6, Fig. 7 and Fig. 8.

In l_1 and l_2 regularized experiments, three kinds of phenomena are observed: 1) our Nonsmooth SCD converges faster than Batch RDA. 2) our primal Nonsmooth SCD is faster than the two other primal algorithms L1-PSA and Pegasos. 3) the performance of primal CD degrades as the datasets get larger and this can be seen when the size of the training set is greater than the dimension ($m < N$). For example, on Astro-physics and CCAT, our primal Nonsmooth SCD is a little slower than the state of the art Dual SCD in [6] (Fig. 5 and 6). 4) the performance of primal CD upgrade as the dimensions get larger and this can be seen when the size of the training set is less than the dimension ($m > N$). For example, our primal Nonsmooth SCD has similar performance as the state of the art Dual on CD A9a and outperforms it on Convertype (Fig. 7 and 8).

Based the above experimental results, SCD methods outperform their corresponding deterministic approaches and Nonsmooth SCD is among the first optimization schemes suggested for efficiently solving large scale nonsmooth primal learning problems. We conclude that our Nonsmooth SCD has achieved all the expected effects that a primal SCD algorithm should have.

Smooth Loss Problems. Our Adaptive Smooth SCD algorithm can deal with many learning problems such as the popular regularized squared and logistic loss problems considered in [20]. To illustrate our main contribution in smooth cases, we only consider L_2-loss.

In [26], many scalable algorithms for regularized smooth losses were compared. These solvers include CDN, SCD [20], CGDGS [27], IPM [8], Lassplore [10] and GLMNET [5]. The extensive experiments sufficiently illustrate that CDN is the fastest. Therefore, to illustrate the scalability of our Adaptive Smooth SCD, we only focus on comparing with the stochastic CDN in [26]. The relationships between $F(\mathbf{w}^t)$ and CPU time are illustrated in Fig. 9, Fig. 10, Fig. 11 and Fig. 12. In addition to obtaining the same level of both sparsity and test accuracy, from these figures, it is easy to find that our Adaptive smooth SCD has almost the same practicality as the stochastic CDN in [26].

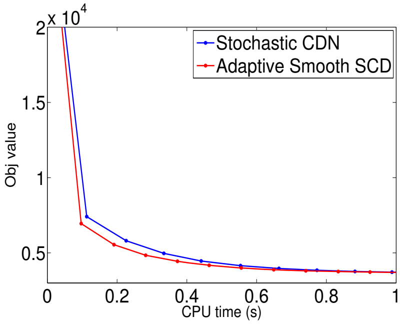

Fig. 9. l_1-R-L_2 on Astro-ph

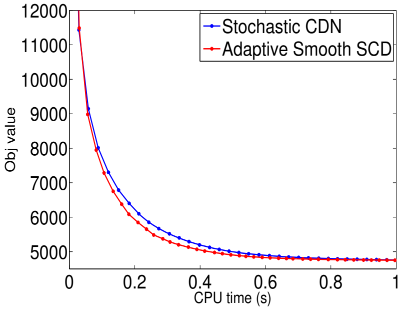

Fig. 10. l_1-R-L_2 on CCAT

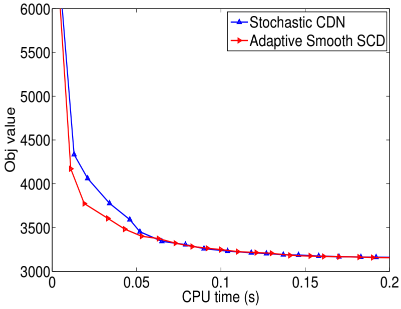

Fig. 11. l_1-R-L_2 on A9a

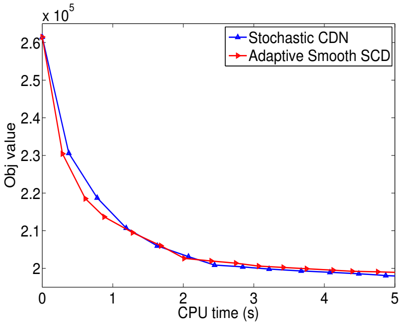

Fig. 12. l_1-R-L_2 on Covertype

To further illustrate the effectiveness of our line search strategy in (10), we also do a toy experiment. One purpose is to compare the performance of PG, APG and Adaptive Smooth SCD when the global Lipschitz constant of ∇f is known, and the other is to test if our line search strategy (10) can really select a more aggressive local Lipschitz constant.

On the toy data set, we follow the strategy in [11] and [12] to calculate the global Lipschitz constant of ∇f. We use this Lipschitz constant to implement PG and APG. The objective values vs. CPU time are illustrated in Fig. 13. From Fig. 13, we can see that our Adaptive Smooth SCD converges much faster than PG. This fact shows that SCD methods with line search strategy outperform their corresponding deterministic approaches in smooth loss cases. What is more, our Adaptive Smooth SCD converges even faster than APG.

More details about the selection of Lipschitz constant are reported in Fig. 14, where the red points represent the local componentwise Lipschitz constant in each iteration of Adaptive Smooth SCD while the blue line is the global Lipschitz constant. From Fig. 14, we find that the local componentwise Lipschitz constant is much smaller than the global Lipschitz constant. This fact means that our line search strategy can decrease the factor of the convergence rates and then improve the performance of smooth SCD. Based on our theoretic analysis in Section 3 and experimental results in this example, we conclude that our Adaptive Smooth SCD has achieved the expected effects in both convergence rates and practicality. Therefore, Adaptive Smooth SCD is a principled and practical method for solving large scale problems.

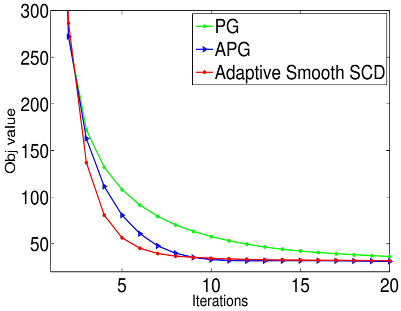

Fig. 13. l_1-R-L_2 on toy data set

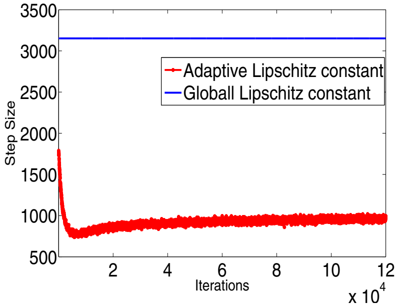

Fig. 14. The adaptive Lipschitz constant

5 Conclusion

In this paper, we have established an interesting framework for developing SCD algorithms for regularized both nonsmooth and smooth losses minimization from structural optimization techniques. We have analyzed how our algorithms are not worse than the state-of-the-art scalable solvers. Experiments confirm the correctness of our theoretical analysis and efficiency of the proposed algorithms. There are several possible extension to this work. For example, the cyclic CD algorithms for regularized nonsmooth losses and the comparison analysis of runtime bounds. These will be included in our future work.

Acknowledgments. The work was supported in part by the NSFC (Grant No. 60835002, 60975040 and 61175050) and the first author is also supported by the Open Project Program of the NLPR. We thank the anonymous reviewers for their helpful comments.

Appendix

To prove Theorem 2.1, we first give the following key lemma [23].
3-Point Property. *Assume*

$$\mathbf{w}^{\tau+1} = \arg\min_{\mathbf{w}} l_f(\mathbf{w}, \mathbf{w}^\tau) + \lambda P(\mathbf{w}) + \tfrac{L}{2}\|\mathbf{w} - \mathbf{w}^\tau\|^2$$

Then $\forall \mathbf{w} \in \mathbb{R}^N$*, we have*
$l_f(\mathbf{w}, \mathbf{w}^\tau) + \lambda P(\mathbf{w}) + \tfrac{L}{2}\|\mathbf{w} - \mathbf{w}^\tau\|^2$
$\geq l_f(\mathbf{w}^{\tau+1}, \mathbf{w}^\tau) + \lambda P(\mathbf{w}^{\tau+1}) + \tfrac{L}{2}\|\mathbf{w}^{\tau+1} - \mathbf{w}^\tau\|^2 + \tfrac{L}{2}\|\mathbf{w} - \mathbf{w}^{\tau+1}\|^2$

Proof of Theorem 2.1
i) Note

$$\boldsymbol{\phi}_\tau = \mathbf{E}_{\xi_\tau} F(\mathbf{w}^{\tau+1}) = \mathbf{E}_{\xi_{\tau-1}} \mathbf{E}_{i_\tau} F(\mathbf{w}^{\tau+1})$$

$\forall \mathbf{w} \in \mathbb{R}_N$, by using the smooth assumption

$$\mathbf{E}_{i_\tau} F(\mathbf{w}^{\tau+1}) \leq \mathbf{E}_{i_\tau}[l_f(\mathbf{w}^{\tau+1}, \mathbf{w}^t) + \lambda P(\mathbf{w}^{\tau+1}) + \tfrac{L}{2}\|\mathbf{w}^{\tau+1} - \mathbf{w}^\tau\|^2]$$

By using the 3-Point Property,
$\mathbf{E}_{i_\tau} F(\mathbf{w}^{\tau+1}) \leq \mathbf{E}_{i_\tau}[l_f(\mathbf{w}, \mathbf{w}^\tau) + \lambda P(\mathbf{w}) + \tfrac{L}{2}\|\mathbf{w} - \mathbf{w}^\tau\|^2 - \tfrac{L}{2}\|\mathbf{w} - \mathbf{w}^{\tau+1}\|^2]$
$\leq F(\mathbf{w}) + \tfrac{L}{2}\|\mathbf{w} - \mathbf{w}^\tau\|^2 - \tfrac{L}{2}\mathbf{E}_{i_\tau}\|\mathbf{w} - \mathbf{w}^{\tau+1}\|^2$
So,
$\boldsymbol{\phi}_t \leq \mathbf{E}_{\xi_{\tau-1}}[F(\mathbf{w}) + \tfrac{L}{2}\|\mathbf{w} - \mathbf{w}^\tau\|^2 - \tfrac{L}{2}\mathbf{E}_{i_\tau}\|\mathbf{w} - \mathbf{w}^{\tau+1}\|^2]$
$\leq F(\mathbf{w}) + \tfrac{L}{2}\mathbf{E}_{\xi_{\tau-1}}\|\mathbf{w} - \mathbf{w}^\tau\|^2 - \tfrac{L}{2}\mathbf{E}_{\xi_\tau}\|\mathbf{w} - \mathbf{w}^{\tau+1}\|^2$
Adding the above inequalities from $\tau = 1$ to $\tau = t$,

$$\sum_{\tau=1}^{t} [\boldsymbol{\phi}_\tau - F(\mathbf{w})] \leq \tfrac{L}{2}\|\mathbf{w} - \mathbf{w}^1\|^2$$

On the other hand,

$$\boldsymbol{\phi}_\tau - F(\mathbf{w}) \leq \boldsymbol{\phi}_{\tau-1} - F(\mathbf{w}),\ t[\boldsymbol{\phi}_t - F(\mathbf{w})] \leq \sum_{\tau=1}^{t} [\boldsymbol{\phi}_\tau - F(\mathbf{w})] \leq \tfrac{L}{2}\|\mathbf{w} - \mathbf{w}^1\|^2$$

This proves i) in Theorem 2.1.

ii) If $P(\mathbf{w}) = \|\mathbf{w}\|_2^2$, F becomes a strongly convex function with Lipschiz constant $L + 2\lambda$. This fact implies that ii) follows from Theorem 2 in [18].

Proof of Lemma 3.1

$$\mathbf{E}_{\xi_t}\sum_{\tau=1}^{t}\{g_{i_\tau}^\tau(w_{i_\tau}^\tau - w_{i_\tau}) + \lambda p(w_{i_\tau}^\tau)\} = \sum_{\tau=1}^{t}\mathbf{E}_{\xi_\tau}\{g_{i_\tau}^\tau(w_{i_\tau}^\tau - w_{i_\tau}) + \lambda p(w_{i_\tau}^\tau)\}$$
$$= \sum_{\tau=1}^{t}\mathbf{E}_{\xi_{\tau-1}}\mathbf{E}_{i_\tau}\{g_{i_\tau}^\tau(w_{i_\tau}^\tau - w_{i_\tau}) + \lambda p(w_{i_\tau}^\tau)\}$$

By the definition of expectation in i_τ, we obtain

$$\mathbf{E}_{i_\tau}\{g_{i_\tau}^\tau(w_{i_\tau}^\tau - w_{i_\tau}) + \lambda p(w_{i_\tau}^\tau)\} = \frac{1}{N}[\mathbf{g}^\tau(\mathbf{w}^\tau - \mathbf{w}) + \lambda P(\mathbf{w}^\tau)]$$

According to the definition of subgradient $\langle \mathbf{g}^\tau, \mathbf{w}^\tau - \mathbf{w}\rangle \geq f(\mathbf{w}^\tau) - f(\mathbf{w})$. So,

$$\frac{1}{N}[\mathbf{g}^\tau(\mathbf{w}^\tau - \mathbf{w}) + \lambda P(\mathbf{w}^\tau)] - \frac{1}{N}\lambda P(\mathbf{w}) \geq \frac{1}{N}[f(\mathbf{w}^\tau) - f(\mathbf{w}) + \lambda P(\mathbf{w}^\tau)] - \frac{1}{N}\lambda P(\mathbf{w})$$
$$= \frac{1}{N}[F(\mathbf{w}^\tau) - F(\mathbf{w})]$$

By taking the expectation $\mathbf{E}_{\xi_{\tau-1}}$,

$$\mathbf{E}_{\xi_{\tau-1}}\mathbf{E}_{i_\tau}\{g_{i_\tau}^\tau(w_{i_\tau}^\tau - w_{i_\tau}) + \lambda p(w_{i_\tau}^\tau)\} - \frac{1}{N}\lambda P(\mathbf{w})$$
$$\geq \frac{1}{N}\mathbf{E}_{\xi_{\tau-1}}[F(\mathbf{w}^\tau) - F(\mathbf{w})] \geq \frac{1}{N}[\boldsymbol{\phi}^\tau - F(\mathbf{w})]$$

By adding the above inequalities from $\tau = 1$ to $\tau = t$, the lemma is proved.

References

1. Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge University Press (2004)
2. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM Journal on Imaging Sciences 2(1), 183–202 (2009)
3. Chang, K.W., Hsieh, C.J., Lin, C.J.: Coordinate descent method for large-scale L_2-loss linear support vector machines. Journal of Machine Learning Research 9, 1369–1398 (2008)
4. Duchi, J., Shalev-Shwartz, S., Singer, Y., Chandra, T.: Efficient projections onto the l 1-ball for learning in high dimensions. In: Proceedings of the 25th International Conference on Machine Learning, pp. 272–279 (2008)
5. Friedman, J., Hastie, T., Tibshirani, R.: Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software 33(1), 1–22 (2010)
6. Hsieh, C.J., Chang, K.W., Lin, C.J., Keerthi, S.S., Sundararajan, S.: A dual coordinate descent method for large-scale linear SVM. In: Proceedings of the 25th International Conference on Machine Learning, pp. 408–415 (2008)
7. Joachims, T.: Training linear SVMs in linear time. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 217–226 (2006)
8. Koh, K., Kim, S.J., Boyd, S.: An interior-point method for large-scale l_1-regularized logistic regression. Journal of Machine Learning Research 8, 1519–1555 (2007)
9. Lin, C.J., Weng, R.C., Keerthi, S.S.: Trust region newton method for logistic regression. Journal of Machine Learning Research 9, 627–650 (2008)
10. Liu, J., Ye, J.: Efficient Euclidean projections in linear time. In: Proceedings of the 26th International Conference on Machine Learning, pp. 657–664 (2009)
11. Mangasarian, O.L.: A finite Newton method for classification. Optimization Methods and Software 17(5), 913–929 (2002)
12. Mangasarian, O.L., Musicant, D.R.: Successive overrelaxation for support vector machines. IEEE Trans. Neural Networks 10, 1032–1037 (1999)

13. Nesterov, Y.: A method of solving a convex programming problem with convergence rate $O(1/k^2)$. Soviet Mathematics Doklady 27, 372–376 (1983)
14. Nesterov, Y.: Smooth minimization of non-smooth functions. Mathematical Programming 103(1), 127–152 (2005)
15. Nesterov, Y.: Gradient methods for minimizing composite objective function. CORE Discussion Papers 2007076. Center for Operations Research and Econometrics, CORE (2007)
16. Nesterov, Y.: How to advance in structural convex optimization. OPTIMA: Mathematical Programming Society Newsletter 78, 2–5 (2008)
17. Nesterov, Y.: Primal-dual subgradient methods for convex problems. Mathematical Programming 120(1), 221–259 (2009)
18. Nesterov, Y.: Efficiency of coordinate descent methods on huge-scale optimization problems. Technical report, University catholique de Louvain, Center for Operations Research and Econometrics, CORE (2010)
19. Saha, A., Tewari, A.: On the finite time convergence of cyclic coordinate descent methods. Arxiv preprint arXiv:1005.2146 (2010)
20. Shalev-Shwartz, S., Tewari, A.: Stochastic methods for l_1 regularized loss minimization. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 929–936 (2009)
21. Shalev-Shwartz, S., Singer, Y., Srebro, N.P.: Primal estimated sub-gradient solver for SVM. In: Proceedings of the 24th International Conference on Machine Learning, pp. 807–814 (2007)
22. Tao, Q., Sun, Z., Kong, K.: Developing Learning Algorithms via Optimized Discretization of Continuous Dynamical Systems. IEEE Trans. Syst. Man Cybern. B 42(1), 140–149 (2012)
23. Tseng, P.: On accelerated proximal gradient methods for convex-concave optimization. submitted to SIAM Journal on Optimization (2008)
24. Tseng, P., Yun, S.: A coordinate gradient descent method for nonsmooth separable minimization. Mathematical Programming 117(1), 387–423 (2009)
25. Xiao, L.: Dual averaging methods for regularized stochastic learning and online optimization. Journal of Machine Learning Research 11, 2543–2596 (2010)
26. Yuan, G.X., Chang, K.W., Hsieh, C.J., Lin, C.: A Comparison of Optimization Methods and Software for Large-scale L_1-regularized Linear Classification. Journal of Machine Learning Research 11, 3183–3234 (2010)
27. Yun, S., Toh, K.C.: A coordinate gradient descent method for l_1-regularized convex minimization. To appear in Computational Optimizations and Applications (2009)