Stochastic Smoothing for Nonsmooth Minimizations: Accelerating SGD by Exploiting Structure

Hua Ouyang HOUYANG@CC.GATECH.EDU
Alexander Gray AGRAY@CC.GATECH.EDU
College of Computing, Georgia Institute of Technology

Abstract

In this work we consider the stochastic minimization of nonsmooth convex loss functions, a central problem in machine learning. We propose a novel algorithm called Accelerated Nonsmooth Stochastic Gradient Descent (ANSGD), which exploits the structure of common nonsmooth loss functions to achieve optimal convergence rates for a class of problems including SVMs. It is the first stochastic algorithm that can achieve the optimal $O(1/t)$ rate for minimizing nonsmooth loss functions (with strong convexity). The fast rates are confirmed by empirical comparisons, in which ANSGD significantly outperforms previous subgradient descent algorithms including SGD.

1. Introduction

Nonsmoothness is a central issue in machine learning computation, as many important methods minimize nonsmooth convex functions. For example, using the nonsmooth hinge loss yields sparse support vector machines; regressors can be made robust to outliers by using the nonsmooth absolute loss other than the squared loss; the $l1$-norm is widely used in sparse reconstructions. In spite of the attractive properties, nonsmooth functions are theoretically more difficult to optimize than smooth functions (Nemirovski & Yudin, 1983). In this paper we focus on minimizing nonsmooth functions where the functions are either stochastic (stochastic optimization), or learning samples are provided incrementally (online learning).

Smoothness and strong-convexity are typically certificates of the existence of fast global solvers. Nesterov's deterministic smoothing method (Nesterov, 2005b) deals with the difficulty of nonsmooth functions by approximating them with smooth functions, for which optimal methods (Nesterov, 2004) can be applied. It converges as $f(\mathbf{x}_t) - \min_{\mathbf{x}} f(\mathbf{x}) \leq O(1/t)$ after t iterations. If a nonsmooth function is strongly convex, this rate can be improved to $O(1/t^2)$ using the excessive gap technique (Nesterov, 2005a).

In this paper, we extend Nesterov's smoothing method to the stochastic setting by proposing a stochastic smoothing method for nonsmooth functions. Combining this with a stochastic version of the optimal gradient descent method, we introduce and analyze a new algorithm named Accelerated Nonsmooth Stochastic Gradient Descent (ANSGD), for a class of functions that include the popular ML methods of interest.

To our knowledge ANSGD is the first stochastic first-order algorithm that can achieve the optimal $O(1/t)$ rate for minimizing nonsmooth loss functions without Polyak's averaging (Polyak & Juditsky, 1992). In comparison, the classic SGD converges in $O(\ln t/t)$ for nonsmooth strongly convex functions (Shalev-Shwartz et al., 2007), and is usually not robust (Nemirovski et al., 2009). Even with Polyak's averaging (Bach & Moulines, 2011; Xu, 2011), there are cases where SGD's convergence rate still can not be faster than $O(\ln t/t)$ (Shamir, 2011). Numerical experiments on real-world datasets also indicate that ANSGD converges much faster in comparing with these state-of-the-art algorithms.

A perturbation-based smoothing method is recently proposed for stochastic nonsmooth minimization (Duchi et al., 2011). This work achieves similar iteration complexities as ours, in a parallel computation scenario.

In machine learning, many problems can be cast as minimizing a composition of a loss function and a regularization term. Before proceeding to the algorithm,

Appearing in *Proceedings of the 29th International Conference on Machine Learning*, Edinburgh, Scotland, UK, 2012.

we first describe a different setting of "composite minimizations" that we will pursue in this paper, along with our notations and assumptions.

1.1. A Different "Composite Setting"

In the classic *black-box* setting of first-order stochastic algorithms (Nemirovski et al., 2009), the structure of the objective function $\min_{\mathbf{x}}\{f(\mathbf{x}) = \mathbb{E}_{\boldsymbol{\xi}} f(\mathbf{x}, \boldsymbol{\xi}) : \boldsymbol{\xi} \sim P\}$ is unknown. In each iteration t, an algorithm can only access the first-order stochastic oracle and obtain a subgradient $f'(\mathbf{x}, \boldsymbol{\xi}_t)$. The basic assumption is that $f'(\mathbf{x}) = \mathbb{E}_{\boldsymbol{\xi}} f'(\mathbf{x}, \boldsymbol{\xi})$ for any $\mathbf{x}$, where the random vector $\boldsymbol{\xi}$ is from a fixed distribution P.

The *composite setting* (also known as *splitting* (Lions & Mercier, 1979)) is an extension of the black-box model. It was proposed to exploit the structure of objective functions. Driven by applications of sparse signal reconstruction, it has gained significant interest from different communities (Daubechies et al., 2004; Beck & Teboulle, 2009; Nesterov, 2007a). Stochastic variants have also been proposed recently (Lan, 2010; Lan & Ghadimi, 2011; Duchi & Singer, 2009; Hu et al., 2009; Xiao, 2010). A stochastic composite function $\Phi(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x})$ is the sum of a smooth stochastic convex function $f(\mathbf{x}) = \mathbb{E}_{\boldsymbol{\xi}} f(\mathbf{x}, \boldsymbol{\xi})$ and a nonsmooth (but simple and deterministic) function $g()$. To minimize Φ, previous work construct the model iteratively: $\langle \nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t), \mathbf{x} - \mathbf{x}_t \rangle + \frac{1}{\eta_t} D(\mathbf{x}, \mathbf{x}_t) + g(\mathbf{x})$, where $\nabla f(\mathbf{x}_t, \boldsymbol{\xi}_t)$ is a gradient, $D(\cdot, \cdot)$ is a proximal function (typically a Bregman divergence) and η_t is a stepsize.

A successful application of the composite idea typically relies on the assumption that the above model is easy to minimize. If $g()$ is very simple, e.g. $\|\mathbf{x}\|_1$ or the nuclear norm, it is straightforward to obtain the minimum in analytic forms. However, this assumption does not hold for many other applications in machine learning, where many loss functions (not the regularization term, here the nonsmooth $g()$ becomes the nonsmooth loss function) are nonsmooth, and do not enjoy separability properties (Wright et al., 2009). This includes important examples such as hinge loss, absolute loss, and ϵ-insensitive loss.

In this paper, we tackle this problem by studying a new stochastic composite setting: $\min_{\mathbf{x}} \Phi(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x})$, where loss function $f()$ is convex and nonsmooth, while $g()$ is convex and L_g-Lipschitz smooth: $g(\mathbf{x}) \leq g(\mathbf{y}) + \langle \nabla g(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle + \frac{L_g}{2} \|\mathbf{x} - \mathbf{y}\|^2$. For clarity, in this paper we focus on unconstrained minimizations. Without loss of generality, we assume that both $f()$ and $g()$ are stochastic: $f(\mathbf{x}) = \mathbb{E}_{\boldsymbol{\xi}} f(\mathbf{x}, \boldsymbol{\xi})$ and $g(\mathbf{x}) = \mathbb{E}_{\boldsymbol{\xi}} g(\mathbf{x}, \boldsymbol{\xi})$, where $\boldsymbol{\xi}$ has distribution P. If either one is deterministic, its $\boldsymbol{\xi}$ is then dropped. To make our algorithm and analysis more general, we assume that $g()$ is μ-strongly convex: $\forall \mathbf{x}, \mathbf{y}$, $g(\mathbf{x}) \geq g(\mathbf{y}) + \langle \nabla g(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle + \frac{\mu}{2} \|\mathbf{x} - \mathbf{y}\|^2$. If it is not strongly convex, one can simply take $\mu = 0$.

The main idea of our algorithm again stems from exploiting the structures of $f()$ and $g()$. In Section 2 we propose to form a smooth stochastic approximation of $f()$, such that the optimal methods (Nesterov, 2004) can be applied to attain optimal convergence rates. The convergence of our proposed algorithm is analyzed in Section 3, and a batch-to-online conversion is also proposed. Two popular machine learning problems are chosen as our examples in Section 4, and numerical evaluations are presented in Section 5. All proofs are provided in a longer version of this paper [1].

2. Approach

2.1. Stochastic Smoothing Method

An important breakthrough in nonsmooth minimization was made by Nesterov in a series of works (Nesterov, 2005b;a; 2007b). By exploiting function structures, Nesterov shows that in many applications, minimizing a well-structured nonsmooth function $f(\mathbf{x})$ can be formulated as an equivalent saddle-point form

$$\min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}) = \min_{\mathbf{x} \in \mathcal{X}} \max_{\mathbf{u} \in \mathcal{U}} \Big[\langle A\mathbf{x}, \mathbf{u} \rangle - Q(\mathbf{u}) \Big], \qquad (1)$$

where $\mathbf{u} \in \mathbb{R}^m$, $\mathcal{U} \subseteq \mathbb{R}^m$ is a convex set, A is a linear operator mapping $\mathbb{R}^D \to \mathbb{R}^m$ and $Q(\mathbf{u})$ is a continuous convex function. Inserting a non-negative ζ-strongly convex function $\omega(\mathbf{u})$ in (1) one obtains a smooth approximation of the original nonsmooth function

$$\hat{f}(\mathbf{x}, \gamma) := \max_{\mathbf{u} \in \mathcal{U}} \Big[\langle A\mathbf{x}, \mathbf{u} \rangle - Q(\mathbf{u}) - \gamma \omega(\mathbf{u}) \Big], \qquad (2)$$

where $\gamma > 0$ is a fixed *smoothness parameter* which is crucial in the convergence analysis. The key property of this approximation is:

Lemma 1. *(Nesterov, 2005b)(Theorem 1) Function $\hat{f}(\mathbf{x}, \gamma)$ is convex and continuously differentiable, and its gradient is Lipschitz continuous with constant $L_{\hat{f}} := \frac{\|A\|^2}{\gamma \zeta}$, where*

$$\|A\| := \max_{\mathbf{x}, \mathbf{u}} \{ \langle A\mathbf{x}, \mathbf{u} \rangle : \|\mathbf{x}\| = 1, \|\mathbf{u}\| = 1 \}. \qquad (3)$$

Nesterov's smoothing method was originally proposed for deterministic optimization. A major drawback of

[1] http://arxiv.org/abs/1205.4481

this method is that the number of iterations N must be known beforehand, such that the algorithm can set a proper smoothness parameter $\gamma = O(\frac{2\|A\|}{N+1})$ to ensure convergence. This makes it unsuitable for algorithms that runs forever, or whose number of iterations is not known. Following his work we propose to extend this smoothing method to stochastic optimization. Our stochastic smoothing differs from the deterministic one in the operator A and smoothness parameter γ, where both will be time-varying.

We assume that the nonsmooth part $f(\mathbf{x}, \boldsymbol{\xi})$ of the stochastic composite function $\Phi()$ is well structured, i.e. for a specific realization $\boldsymbol{\xi}_t$, it has an equivalent form like the max function in (1):

$$f(\mathbf{x}, \boldsymbol{\xi}_t) = \max_{\mathbf{u}\in\mathcal{U}} \Big[\langle A_{\boldsymbol{\xi}_t}\mathbf{x}, \mathbf{u}\rangle - Q(\mathbf{u})\Big], \tag{4}$$

where $A_{\boldsymbol{\xi}_t}$ is a stochastic linear operator associated with $\boldsymbol{\xi}_t$. We construct a smooth approximation of this function as:

$$\hat{f}(\mathbf{x}, \boldsymbol{\xi}_t, \gamma_t) := \max_{\mathbf{u}\in\mathcal{U}} \Big[\langle A_{\boldsymbol{\xi}_t}\mathbf{x}, \mathbf{u}\rangle - Q(\mathbf{u}) - \gamma_t\omega(\mathbf{u})\Big], \tag{5}$$

where γ_t is a time-varying smoothness parameter only associated with iteration index t, and is independent of $\boldsymbol{\xi}_t$. Function $\omega()$ is non-negative and ζ-strongly convex. Due to Lemma 1, $\hat{f}(\mathbf{x}, \boldsymbol{\xi}_t, \gamma_t)$ is $\frac{\|A_{\boldsymbol{\xi}_t}\|^2}{\gamma_t\zeta}$-Lipschitz smooth. It follows that

Lemma 2. $\forall \mathbf{x}, \mathbf{y}, t$, $\mathbb{E}_{\boldsymbol{\xi}}\hat{f}(\mathbf{x}, \boldsymbol{\xi}, \gamma_t) \le \mathbb{E}_{\boldsymbol{\xi}}\hat{f}(\mathbf{y}, \boldsymbol{\xi}, \gamma_t) + \mathbb{E}_{\boldsymbol{\xi}}\langle\nabla\hat{f}(\mathbf{y}, \boldsymbol{\xi}, \gamma_t), \mathbf{x}-\mathbf{y}\rangle + \frac{\mathbb{E}_{\boldsymbol{\xi}}\|A_{\boldsymbol{\xi}}\|^2}{\gamma_t\zeta}\|\mathbf{x}-\mathbf{y}\|^2$.

We have the following observation about our composite objective $\Phi()$, which relates the reduction of the original and approximated function values.

Lemma 3. *For any* $\mathbf{x}, \mathbf{x}_t, t$,

$$\begin{aligned}\Phi(\mathbf{x}_t) - \Phi(\mathbf{x}) \le \mathbb{E}_{\boldsymbol{\xi}}\Big[\hat{f}(\mathbf{x}_t, \boldsymbol{\xi}, \gamma_t) + g(\mathbf{x}_t, \boldsymbol{\xi})\Big] - \\ \mathbb{E}_{\boldsymbol{\xi}}\Big[\hat{f}(\mathbf{x}, \boldsymbol{\xi}, \gamma_t) + g(\mathbf{x}, \boldsymbol{\xi})\Big] + \gamma_t D_{\mathcal{U}},\end{aligned} \tag{6}$$

where $D_{\mathcal{U}} := \max_{\mathbf{u}\in\mathcal{U}}\omega(\mathbf{u})$.

2.2. Accelerated Nonsmooth SGD (ANSGD)

We are now ready to present our algorithm ANSGD (Algorithm 1). This stochastic algorithm is obtained by applying Nesterov's optimal method to our smooth surrogate function, and thus has a similar form to that of his original deterministic method (Nesterov, 2004)(p.78). However, our convergence analysis is more straightforward, and does not rely on the concept of estimate sequences. Hence it is easier to identify proper series $\gamma_t, \eta_t, \alpha_t$ and θ_t that are crucial in achieving fast rates of convergence. These series will be determined in our main results (Thm.1 and 2).

Algorithm 1 Accelerated Nonsmooth Stochastic Gradient Descent (ANSGD)

INPUT: series γ_t, η_t, $\theta_t \ge 0$ and $0 \le \alpha_t \le 1$;
OUTPUT: $\mathbf{x}_{t+1}$;
0. Initialize $\mathbf{x}_0$ and $\mathbf{v}_0$;
for $t = 0, 1, 2, \ldots$ **do**
1. $\mathbf{y}_t \leftarrow \frac{(1-\alpha_t)(\mu+\theta_t)\mathbf{x}_t + \alpha_t\theta_t\mathbf{v}_t}{\mu(1-\alpha_t)+\theta_t}$
2. $\hat{f}_{t+1}(\mathbf{x}) \leftarrow \max_{\mathbf{u}\in\mathcal{U}}\Big[\langle A_{\boldsymbol{\xi}_{t+1}}\mathbf{x}, \mathbf{u}\rangle - Q(\mathbf{u}) - \gamma_{t+1}\omega(\mathbf{u})\Big]$
3. $\mathbf{x}_{t+1} \leftarrow \mathbf{y}_t - \eta_t\Big[\nabla\hat{f}_{t+1}(\mathbf{y}_t) + \nabla g_{t+1}(\mathbf{y}_t)\Big]$
4. $\mathbf{v}_{t+1} \leftarrow \frac{\theta_t\mathbf{v}_t + \mu\mathbf{y}_t - \big[\nabla\hat{f}_{t+1}(\mathbf{y}_t) + \nabla g_{t+1}(\mathbf{y}_t)\big]}{\mu+\theta_t}$
end for

3. Convergence Analysis

To clarify our presentation, we use Table 1 to list some notations that will be used throughout the paper.

Table 1. Some notations.

Symbol	Meaning
$\hat{f}_t(\mathbf{x})$, $g_t(\mathbf{x})$	$\hat{f}(\mathbf{x}, \boldsymbol{\xi}_t, \gamma_t)$, $g(\mathbf{x}, \boldsymbol{\xi}_t)$
$\nabla\hat{f}_t(\mathbf{x})$, $\nabla g_t(\mathbf{x})$	$\nabla\hat{f}(\mathbf{x}, \boldsymbol{\xi}_t, \gamma_t)$, $\nabla g(\mathbf{x}, \boldsymbol{\xi}_t)$
L_t	$L_g + \frac{\|A_{\boldsymbol{\xi}_t}\|^2}{\gamma_t\zeta}$
$\sigma_t(\mathbf{x})$	$[\nabla\hat{f}_t(\mathbf{x}) + \nabla g_t(\mathbf{x})] -$ $\mathbb{E}_{\boldsymbol{\xi}_t}[\nabla\hat{f}_t(\mathbf{x}) + \nabla g_t(\mathbf{x})]$
σ^2	$\mathbb{E}\max_t\|\sigma_{t+1}(\mathbf{y}_t)\|^2$
Δ_t	$\mathbb{E}_{\boldsymbol{\xi}_t}\big[\hat{f}_t(\mathbf{x}_t) + g_t(\mathbf{x}_t)\big] -$ $\mathbb{E}_{\boldsymbol{\xi}_t}\big[\hat{f}_t(\mathbf{x}) + g(\mathbf{x})\big]$
Γ_{t+1}	$\langle\sigma_{t+1}(\mathbf{y}_t), \alpha_t\mathbf{x} + (1-\alpha_t)\mathbf{x}_t - \mathbf{y}_t\rangle$
D_t^2	$\frac{1}{2}\mathbb{E}\|\mathbf{x} - \mathbf{v}_t\|^2$

Our convergence rates are based on the following main lemma, which bounds the progressive reduction Δ_t of the smoothed function value. Actually Line 1, 3, and 4 of Alg.1 are also derived from the proof of this lemma.

Lemma 4. *Let* γ_t *be monotonically decreasing. Applying algorithm* ANSGD *to nonsmooth composite function* $\Phi()$, *we have* $\forall\mathbf{x}$ *and* $\forall t \ge 0$,

$$\begin{aligned}&\Delta_{t+1} \le (1-\alpha_t)\Delta_t + (1-\alpha_t)(\gamma_t - \gamma_{t+1})D_{\mathcal{U}} + \\ &\Gamma_{t+1} + \frac{\alpha_t}{2}\Big[\theta_t\|\mathbf{x} - \mathbf{v}_t\|^2 - (\mu+\theta_t)\|\mathbf{x} - \mathbf{v}_{t+1}\|^2\Big] + \\ &\eta_t pq + \left[\frac{\alpha_t}{2(\mu+\theta_t)} + \frac{L_{t+1}}{2}\eta_t^2 - \eta_t\right]q^2\end{aligned} \tag{7}$$

where $p := \|\sigma_{t+1}(\mathbf{y}_t)\|$ *and* $q := \|\nabla\hat{f}_{t+1}(\mathbf{y}_t) + \nabla g_{t+1}(\mathbf{y}_t)\|$.

3.1. How to Choose Stepsizes η_t

In the RHS of (7), nonnegative scalars $p, q \geq 0$ are data-dependent, and could be arbitrarily large. Hence we need to set proper stepsizes η_t such that the last two terms in (7) are non-positive. One might conjecture that: there exist a series $c_t \geq 0$ such that

$$\eta_t pq + \left[\frac{\alpha_t}{2(\mu+\theta_t)} + \frac{L_{t+1}}{2}\eta_t^2 - \eta_t\right] q^2 \leq c_t p^2. \quad (8)$$

It is easy to verify that if we take $\eta_t = \frac{\alpha_t}{\mu+\theta_t}$ and any series $c_t \geq \frac{\alpha_t}{2(\mu+\theta_t-\alpha_t L_{t+1})} \geq 0$, then (8) is satisfied. To retain a tight bound, we take

$$c_t = \frac{\alpha_t}{2(\mu+\theta_t-\alpha_t L_{t+1})}. \quad (9)$$

Taking expectation on both sides of (7) and noticing that $\mathbb{E}_{\boldsymbol{\xi}_{t+1}|\boldsymbol{\xi}_{[t]}}\Gamma_{t+1} = 0$, $\mathbb{E}_{\boldsymbol{\xi}_{t+1}} c_t \leq \frac{\alpha_t}{2(\mu+\theta_t-\alpha_t \mathbb{E}_{\boldsymbol{\xi}_{t+1}} L_{t+1})}$ due to Jensen's inequality, we have

Lemma 5. *$\forall \mathbf{x}$ and $\forall t \geq 0$,*

$$\mathbb{E}\Delta_{t+1} \leq (1-\alpha_t)\mathbb{E}\Delta_t + \alpha_t\theta_t D_t^2 - \alpha_t(\mu+\theta_t)D_{t+1}^2 + \frac{\alpha_t}{2(\mu+\theta_t-\alpha_t\mathbb{E}L_{t+1})}\sigma^2 + (1-\alpha_t)(\gamma_t-\gamma_{t+1})D_{\mathcal{U}}, \quad (10)$$

The optimal convergence rates of our algorithm differs according to the fact of μ (positive or not). They are presented separately in the following two subsections, where the choices of γ_t, θ_t, α_t will also be determined.

3.2. Optimal Rates for Composite Minimizations when $\mu = 0$

When $\mu = 0$, $g()$ is only convex and L_g-Lipschitz smooth, but not assumed to be strongly convex.

Theorem 1. *Take $\alpha_t = \frac{2}{t+2}$, $\gamma_{t+1} = \alpha_t$, $\theta_t = L_g\alpha_t + \frac{\Omega}{\sqrt{\alpha_t}} + \frac{\mathbb{E}\|A_{\boldsymbol{\xi}}\|^2}{\zeta}$ and $\eta_t = \frac{\alpha_t}{\theta_t}$ in Alg.1, where Ω is a constant. We have $\forall \mathbf{x}$ and $\forall t \geq 0$, $\mathbb{E}\left[\Phi(\mathbf{x}_{t+1}) - \Phi(\mathbf{x})\right] \leq$*

$$\frac{4L_g D_0^2}{(t+2)^2} + \frac{2\mathbb{E}\|A_{\boldsymbol{\xi}}\|^2 D_0^2/\zeta + 4D_{\mathcal{U}}}{t+2} + \frac{\sqrt{2}(\Omega D_0^2 + \sigma^2/\Omega)}{\sqrt{t+2}}. \quad (11)$$

In this result, the variance bound is optimal up to a constant factor (Agarwal et al., 2012). The dominating factor is still due to the stochasticity, but not affected by the nonsmoothness of $f()$. Taking the parameter $\Omega = \sigma/D_0$, this last term becomes $\frac{2\sqrt{2}D_0\sigma}{\sqrt{t+2}}$. This bound is better than that of stochastic gradient descent or stochastic dual averaging (Dekel et al., 2010) for minimizing L-Lipschitz smooth functions, whose rate is $O\left(\frac{LD_0^2}{t} + \frac{D_0^2+\sigma^2}{\sqrt{t}}\right)$; without the smooth function $g()$, our bound is of the same order as it, keeping in mind that our rate is for nonsmooth minimizations. This fact underscores the potential of using stochastic optimal methods for nonsmooth functions.

The diminishing smoothness parameter $\gamma_t = \frac{2}{t+2}$ indicates that initially a smoother approximation is preferred, such that the solution does not change wildly due to the nonsmoothness and stochasticity. Eventually the approximated function should be closer and closer to the original nonsmooth function, such that the optimality can be reached. Some concrete examples are given in Fig.1.

The $\mathbb{E}\|A_{\boldsymbol{\xi}}\|^2$ in our bound is a theoretical constant. In Sec.4 we demonstrate a sampling method, and it turns out to work quite well in estimating $\mathbb{E}\|A_{\boldsymbol{\xi}}\|^2$.

3.3. Nearly Optimal Rates for Strongly Convex Minimizations

When $\mu > 0$, $g()$ is strongly convex, and the convergence rate of ANSGD can be improved to $O(1/t)$.

Theorem 2. *Take $\alpha_t = \frac{2}{t+1}$, $\gamma_{t+1} = \alpha_t$, $\theta_t = L_g\alpha_t + \frac{\mu}{2\alpha_t} + \frac{\mathbb{E}\|A_{\boldsymbol{\xi}}\|^2}{\zeta} - \mu$ and $\eta_t = \frac{\alpha_t}{\mu+\theta_t}$ in Alg.1. Denote*

$$C := \max\left\{\frac{4\mathbb{E}\|A_{\boldsymbol{\xi}}\|^2}{\zeta\mu}, 2\left(\frac{L_g}{\mu}\right)^{1/3}\right\}. \quad (12)$$

We have $\forall \mathbf{x}$ and $\forall t \geq 0$, $\mathbb{E}\left[\Phi(\mathbf{x}_{t+1}) - \Phi(\mathbf{x})\right] \leq$

$$\frac{6.58L_g\tilde{D}^2}{t(t+1)} + \mathcal{B} + \frac{4D_{\mathcal{U}}}{t+1} + \frac{\sigma^2}{\mu(t+1)}, \quad (13)$$

where

$$\mathcal{B} := \begin{cases} \frac{2\mathbb{E}\|A_{\boldsymbol{\xi}}\|^2\tilde{D}^2/\zeta}{t+1} & \text{if } 0 \leq t < C, \\ \frac{2(C-2)\mathbb{E}\|A_{\boldsymbol{\xi}}\|^2\tilde{D}^2/\zeta}{t(t+1)} & \text{if } t \geq C, \end{cases} \quad (14)$$

and $\tilde{D}^2 := \max_{0\leq i\leq \min\{t,C\}} D_i^2$.

Note that C is the smallest iteration index for which one can retain $1/t^2$ rates for the $\mathbb{E}\|A_{\boldsymbol{\xi}}\|^2$ part ($\mathcal{B}$). Without any knowledge about L_g, μ and $\mathbb{E}\|A_{\boldsymbol{\xi}}\|^2$, one can set a parameter Ω and take $\theta_t = L_g\alpha_t + \frac{\mu}{2\alpha_t} + \frac{\mathbb{E}\|A_{\boldsymbol{\xi}}\|^2}{\Omega\zeta} - \mu$ in the algorithm. In our experiments, we observe that one can take Ω fairly large (of $O(\mathbb{E}\|A_{\boldsymbol{\xi}}\|^2)$), meaning that C can be very small (O(1)), and $\mathcal{B}$ is $O(\frac{1}{t^2})$ for *all* t. In this sense, strongly convex ANSGD is almost parameter-free. Without the $O(1/t)$ rate of $D_{\mathcal{U}}$, all terms in our bound are optimal. This is why our rate is called "nearly" optimal. In practice, $D_{\mathcal{U}}$ is usually small, and it will be dominated by the last term $\frac{\sigma^2}{\mu(t+1)}$.

3.4. Batch-to-Online Conversion

The performance of an online learning (online convex minimization) algorithm is typically measured by *regret*, which can be expressed as

$$R(t) := \sum_{i=0}^{t-1} \left[\Phi(\mathbf{x}_i, \boldsymbol{\xi}_{i+1}) - \Phi(\mathbf{x}_t^*, \boldsymbol{\xi}_{i+1})\right], \tag{15}$$

where $\mathbf{x}_t^* := \arg\min_{\mathbf{x}} \sum_{i=0}^{t-1} [\Phi(\mathbf{x}, \boldsymbol{\xi}_{i+1})]$. In the learning theory literature, many approaches are proposed which use online learning algorithms for batch learning (stochastic optimization), called "online-to-batch" (O-to-B) conversions. For convex functions, many of these approaches employ an "averaged" solution as the final solution.

On the contrary, we show that stochastic optimization algorithms can also be used *directly* for online learning. This "batch-to-online" (B-to-O) conversion is almost free of any additional effort: under i.i.d. assumptions of data, one can use any stochastic optimization algorithm for online learning.

Proposition 1. *For any* $t \geq 0$, $\mathbb{E}_{\boldsymbol{\xi}_{[t]}} R(t) \leq$

$$\sum_{i=0}^{t-1} \mathbb{E}_{\boldsymbol{\xi}_{[i]}} \left[\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}^*)\right] + \mathbb{E}_{\boldsymbol{\xi}_{[t]}} \sum_{i=0}^{t-1} \left[\Phi(\mathbf{x}_t^*) - \Phi(\mathbf{x}_t^*, \boldsymbol{\xi}_{i+1})\right] \tag{16}$$

where $\mathbf{x}^* := \arg\min_{\mathbf{x}} \Phi(\mathbf{x})$ *and* $\mathbf{x}_t^* := \arg\min_{\mathbf{x}} \sum_{i=0}^{t-1} [\Phi(\mathbf{x}, \boldsymbol{\xi}_{i+1})]$.

When $\Phi()$ is convex, the second term in (16) can be bounded by applying standard results in uniform convergence (e.g. (Boucheron et al., 2005)): $\sum_{i=1}^{t-1} \Phi(\mathbf{x}_t^*) - \Phi(\mathbf{x}_t^*, \boldsymbol{\xi}_{i+1}) = O(\sqrt{t})$. Together with summing up the RHS of (11), we can obtain an $O(\sqrt{t})$ regret bound. When $\Phi()$ is strongly convex, the second term in (16) can be bounded using (Shalev-Shwartz et al., 2009): $\sum_{i=1}^{t-1} \Phi(\mathbf{x}_t^*) - \Phi(\mathbf{x}_t^*, \boldsymbol{\xi}_{i+1}) = O(\ln t)$. Together with summing up the RHS of (13), an $O(\ln t)$ regret bound is achieved. The $O(\sqrt{t})$ and $O(\ln t)$ regret bounds are known

Using our proposed ANSGD for online learning by B-to-O achieves the same (optimal) regret bounds as state-of-the-art algorithms designated for online learning. However, using O-to-B, one can only retain an $O(\ln t/t)$ rate of convergence for stochastic strongly convex optimization. From this perspective, O-to-B is inferior to B-to-O. The sub-optimality of O-to-B is also discussed in (Hazan & Kale, 2011).

4. Examples

In this section, two nonsmooth functions are given as examples. We will show how these functions can be stochastically approximated, and how to calculate parameters used in our algorithm.

4.1. Hinge Loss SVM Classification

Hinge loss is a convex surrogate of the $0-1$ loss. Denote a sample-label pair as $\boldsymbol{\xi} := \{\mathbf{s}, l\} \sim P$, where $\mathbf{s} \in \mathbb{R}^D$ and $l \in \mathbb{R}$. Hinge loss can be expressed as $f_{\text{hinge}}(\mathbf{x}) := \max\{0, 1 - l\mathbf{s}^T\mathbf{x}\}$. It has been widely used for SVM classifiers where the objective is $\min \Phi(\mathbf{x}) = \min \mathbb{E}_{\boldsymbol{\xi}} f_{\text{hinge}}(\mathbf{x}) + \frac{\lambda}{2}\|\mathbf{x}\|^2$. Note that the regularization term $g(\mathbf{x}) = \frac{\lambda}{2}\|\mathbf{x}\|^2$ is λ-strongly convex, hence according to Thm.2, ANSGD enjoys $O(1/(\lambda t))$ rates. Taking $\omega(\mathbf{u}) = \frac{1}{2}\|\mathbf{u}\|^2$ in (5), it is easy to check that the smooth stochastic approximation of hinge loss is

$$\hat{f}_{\text{hinge}}(\mathbf{x}, \boldsymbol{\xi}_t, \gamma_t) = \max_{0 \leq u \leq 1} \left\{ u\left(1 - l_t \mathbf{s}_t^T \mathbf{x}\right) - \gamma_t \frac{u^2}{2} \right\}. \tag{17}$$

This maximization is simple enough such that we can obtain an equivalent smooth representation: $\hat{f}_{\text{hinge}}(\mathbf{x}, \boldsymbol{\xi}_t, \gamma_t) =$

$$\begin{cases} 0 & \text{if } l_t \mathbf{s}_t^T \mathbf{x} \geq 1, \\ \frac{(1 - l_t \mathbf{s}_t^T \mathbf{x})^2}{2\gamma_t} & \text{if } 1 - \gamma_t \leq l_t \mathbf{s}_t^T \mathbf{x} < 1, \\ 1 - l_t \mathbf{s}_t^T \mathbf{x} - \frac{\gamma_t}{2} & \text{if } l_t \mathbf{s}_t^T \mathbf{x} < 1 - \gamma_t. \end{cases} \tag{18}$$

Several examples of $\hat{f}_{\text{hinge}}$ with varying γ_t are plotted in Fig.1(left) in comparing with the hinge loss.

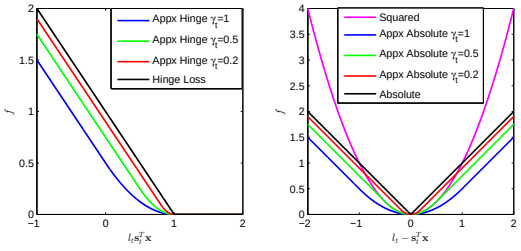

Figure 1. Left: Hinge loss and its smooth approximations. Right: Absolute loss and its smooth approximations.

Here $\mathbf{u}$ is a scalar, hence it is straightforward to calculate $\frac{\mathbb{E}\|A_{\boldsymbol{\xi}}\|^2}{\zeta}$, which will be used to generate sequences θ_t. In binary classification, suppose $l \in \{1, -1\}$. Using definition (3), one only needs to calculate $\mathbb{E}(\max_{\|\mathbf{x}\|=1} \mathbf{s}_t^T \mathbf{x})^2$. Practically one can take a small subset of k random samples $\mathbf{s}_i$ (e.g. $k = 100$), and calculate the sample average of the squared norms $\frac{1}{k}\sum_{i=1}^{k} \|\mathbf{s}_i\|^2$. This yields $\frac{1}{k}\sum_{i=1}^{k} (\max_{\|\mathbf{x}\|=1} \mathbf{s}_i^T \mathbf{x})^2$, an estimate of $\mathbb{E}\|A_{\boldsymbol{\xi}}\|^2$.

4.2. Absolute Loss Robust Regression

Absolute loss is an alternative to the popular squared loss for robust regressions (Hastie et al., 2009). Using same notations as Sec.4.1 it can be expressed as

$f_{abs}(\mathbf{x}) := |l - \mathbf{s}^T\mathbf{x}|$. Taking $\omega(\mathbf{u}) = \frac{1}{2}\|\mathbf{u}\|^2$ in (5), its smooth stochastic approximation can be expressed as

$$\hat{f}_{\text{abs}}(\mathbf{x}, \boldsymbol{\xi}_t, \gamma_t) = \max_{-1\le u\le 1} \left\{ u(l_t - \mathbf{s}_t^T\mathbf{x}) - \gamma_t \frac{u^2}{2} \right\}. \quad (19)$$

Solving this maximization wrt u we obtain an equivalent form: $\hat{f}_{\text{abs}}(\mathbf{x}, \boldsymbol{\xi}_t, \gamma_t) =$

$$\begin{cases} l_t - \mathbf{s}_t^T\mathbf{x} - \frac{\gamma_t}{2} & \text{if } l_t - \mathbf{s}_t^T\mathbf{x} \ge \gamma_t, \\ \frac{(l_t - \mathbf{s}_t^T\mathbf{x})^2}{2\gamma_t} & \text{if } -\gamma_t \le l_t - \mathbf{s}_t^T\mathbf{x} < \gamma_t, \\ -(l_t - \mathbf{s}_t^T\mathbf{x}) - \frac{\gamma_t}{2} & \text{if } l_t - \mathbf{s}_t^T\mathbf{x} < -\gamma_t. \end{cases} \quad (20)$$

This approximation looks similar to the well-studied Huber loss (Huber, 1964), though they are different. Actually they share the same form only when $\gamma_t = 0.5$ (green curve in Fig.1 Right). The parameter $\mathbb{E}\|A_{\boldsymbol{\xi}}\|^2$ can be estimated in a similar way as in Sec.4.1.

5. Experimental Results

In this section, five publicly available datasets from various application domains will be used to evaluate the efficiency of ANSGD. Datasets "svmguide1", "real-sim", "rcv1" and "alpha" are for binary classifications, and "abalone" is for robust regressions.[2]

Following our examples in Sec.4, we will evaluate our algorithm using approximated hinge loss for classifications, and approximated absolute loss for regressions. Exact hinge and absolute losses will be used for subgradient descent algorithms that we will compare with, as described in the following section. All losses are squared-$l2$-norm-regularized. The regularization parameter λ is shown on each figure. When assuming strong-convexity, we take $\mu = \lambda$.

5.1. Algorithms for Comparison and Parameters

We compare ANSGD with three state-of-the-art algorithms. Each algorithm has a data-dependent tuning parameter, denoted by Ω (although they have different physical meanings). The best values of Ω are found based on a tuning subset of samples. Note that when assuming strong-convexity, our ANSGD is almost parameter-free. As discussed after Thm.2, our experiments indicate that the optimal Ω is taken such that $\frac{\mathbb{E}\|A_{\boldsymbol{\xi}}\|^2}{\Omega\zeta} \approx 1$, meaning that one can simply take $\theta_t = L_g\alpha_t + \frac{\mu}{2\alpha_t} + 1 - \mu$.

SGD. The classic stochastic approximation (Robbins & Monro, 1951) is adopted: $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \eta_t f'(\mathbf{x}_t)$, where $f'(\mathbf{x}_t)$ is the subgradient. When only assuming convexity ($\mu = 0$), we use stepsize $\eta_t = \frac{\Omega}{\sqrt{t}}$. When assuming strong-convexity, we follow the stepsize used in SGD2 (Bottou): $\eta_t = \frac{1}{\mu(t+\Omega)}$.

Averaged SGD. This is algorithmically the same as SGD, except that the averaged result $\bar{\mathbf{x}} := \frac{1}{t}\sum_{i=1}^t \mathbf{x}_i$ is used for testing. We follow the stepsizes suggested by the recent work on the non-asymptotic analysis of SGD (Bach & Moulines, 2011; Xu, 2011), where it is argued that Polyak's averaging combining with proper stepsizes yield optimal rates. When only assuming convexity, we use stepsizes $\eta_t = \frac{\Omega}{\sqrt{t}}$ (Bach & Moulines, 2011). When assuming strong convexity, the stepsize is taken as $\eta_t = \frac{1}{\Omega(1+\mu t/\Omega)^{3/4}}$(Xu, 2011).

AC-SA. This approach (Lan, 2010; Lan & Ghadimi, 2011) is interesting to compare because like ANSGD, it is another way of obtaining a stochastic algorithm based on Nesterov's optimal method, begging the question of whether it has similar behavior. Theoretically, according to Prop.8 and 9 in (Lan & Ghadimi, 2011), the bound for the nonsmooth part is of $O(1/\sqrt{t})$ for $\mu = 0$ and $O(1/t)$ for $\mu > 0$. In comparison, our nonsmooth part converges in $O(1/t)$ for $\mu = 0$ and $O(1/t^2)$ for $\mu > 0$. Numerically we observe that directly applying AC-SA to nonsmooth functions results in inferior performances.

5.2. Results

Due to the stochasticity of all the algorithms, for each setting of the experiments, we run the program for 10 times, and plot the mean and standard deviation of the results using error bars.

In the first set of experiments, we compare ANSGD with two subgradient-based algorithms SGD and Averaged SGD. Classification results are shown in Fig.2, 3, 4 and 5, and regression results are shown in Fig.6. In each figure, the left column is for algorithms without strongly convex assumptions, while in the right column the algorithms assume strong-convexity and take $\mu = \lambda$. For classification results, we plot function values over the testing set in the first row, and plot testing accuracies in the second row.

It is clear that in all these experiments, ANSGD's function values converges consistently faster than the other two SGD algorithms. In non-strongly convex experi-

[2] Dataset "alpha" is obtained from ftp://largescale.ml.tu-berlin.de/largescale/, and the other four datasets can be accessed via http://www.csie.ntu.edu.tw/~cjlin/libsvmtools. Dataset "rcv1" comes with 20,242 training samples and 677,399 testing samples. For "svmguide1" and "real-sim", we randomly take 60% of the samples for training and 40% for testing. For "alpha" and "abalone", 80% are used for training, and the rest 20% are used for testing.

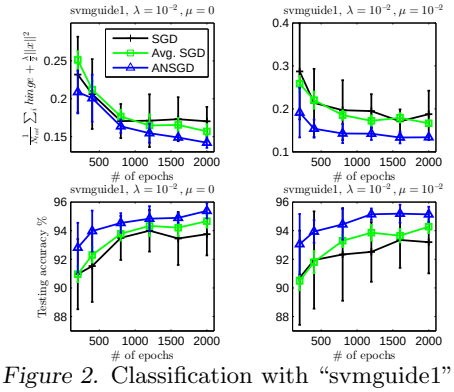

Figure 2. Classification with "svmguide1".

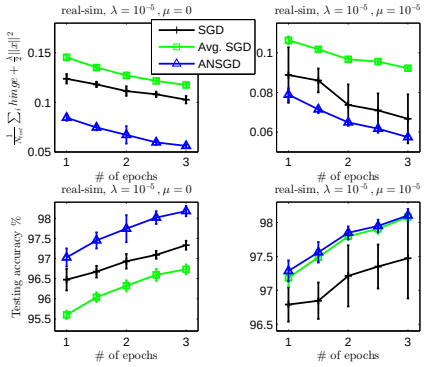

Figure 3. Classification with "real-sim".

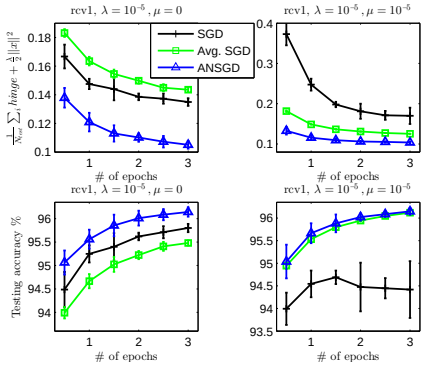

Figure 4. Classification with "rcv1".

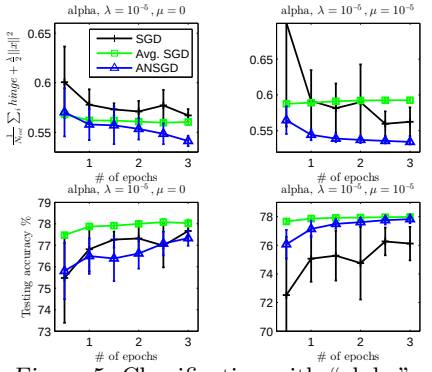

Figure 5. Classification with "alpha".

ments, it converges significantly faster than SGD and its averaged version. In strongly convex experiments, it still out performs, and is more robust than strongly

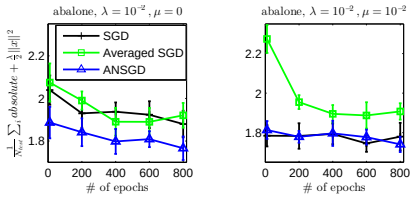

Figure 6. Regression with "abalone".

convex SGD. Averaged SGD performs well in strongly convex settings, in terms of prediction accuracies, although its errors are still higher than ANSGD in the first three datasets. The only exception is in "alpha" (Fig.5), where Averaged SGD retains higher function values than ANSGD, but its accuracies are contradictorily higher in early stages. The reason might be that the inexact solution serves as an additional regularization factor, which cannot be predicted by the analysis of convergence rates.

In the second set of experiments, we compare ANSGD with AC-SA and its strongly convex version. Results are in Fig.7, 8 and 9. In all experiments our ANSGD significantly outperforms AC-SA, and is much more stable. These experiments confirm the theoretically better rates discussed in Sec.5.1.

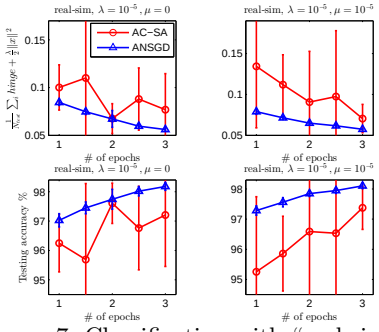

Figure 7. Classification with "real-sim".

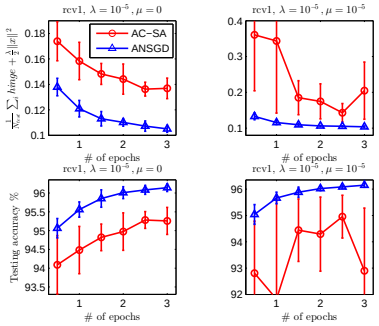

Figure 8. Classification with "rcv1".

6. Conclusions and Future Work

We introduce a different composite setting for nonsmooth functions. Under this setting we propose a stochastic smoothing method and a novel stochastic

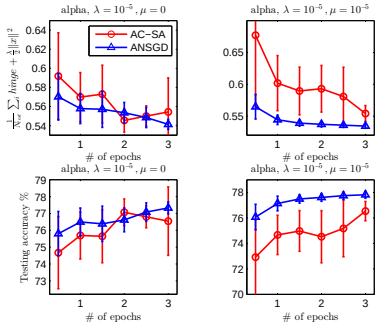

Figure 9. Classification with "alpha".

algorithm ANSGD. Convergence analysis show that it achieves (nearly) optimal rates under both convex and strongly convex assumptions. We also propose a "Batch-to-Online" conversion for online learning, and show that optimal regrets can be obtained.

We will extend our method to constrained minimizations, as well as cases when the approximated function $\hat{f}()$ is not easily obtained by maximizing $\mathbf{u}$. Nesterov's excessive gap technique has the "true" optimal $1/t^2$ bound, and we will investigate the possibility of integrating it in our algorithm. Exploiting links with statistical learning theories may also be promising.

References

Agarwal, Alekh, Bartlett, Peter L., Ravikumar, P., and Wainwright, Martin J. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. *Information Theory, IEEE Trans.*, 2012.

Bach, Francis and Moulines, Eric. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *NIPS*, 2011.

Beck, Amir and Teboulle, Marc. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):193–202, 2009.

Bottou, Leon. Stochastic gradient descent 2.0. URL http://leon.bottou.org/projects/sgd.

Boucheron, Stéphane, Bousquet, Olivier, and Lugosi, Gábor. Theory of classification: A survey of some recent advances. *ESAIM: Probability and Statistics*, 9:323–375, 2005.

Daubechies, I., Defrise, M., and Mol, C. De. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11):14131457, 2004.

Dekel, Ofer, Gilad-Bachrach, Ran, Shamir, Ohad, and Xiao, Lin. Optimal distributed online prediction using mini-batches. *arXiv*, 2010. URL http://arxiv.org/abs/1012.1367.

Duchi, John and Singer, Yoram. Efficient online and batch learning using forward backward splitting. *JMLR*, (10): 2899–2934, 2009.

Duchi, John, Bartlett, Peter L., and Wainwright, Martin J. Randomized smoothing for stochastic optimization. *arXiv*, 2011. URL http://arxiv.org/abs/1103.4296.

Hastie, Trevor, Tibshirani, Robert, and Friedman, Jerome. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd edition, 2009.

Hazan, Elad and Kale, Satyen. Beyond the regret minimization barrier: an optimal algorithm for stochastic strongly-convex optimization. In *COLT*, 2011.

Hu, Chonghai, Kwok, James T., and Pan, Weike. Accelerated gradient methods for stochastic optimization and online learning. In *NIPS 22*, 2009.

Huber, Peter J. Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35(1):73–101, 1964.

Lan, G. and Ghadimi, S. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, i: a generic algorithmic framework. *SIAM J. on Optimization*, 2011.

Lan, Guanghui. An optimal method for stochastic composite optimization. *Mathematical Programming*, 2010. doi: DOI10.1007/s10107-010-0434-y.

Lions, P. L. and Mercier, B. Splitting algorithms for the sum of two nonlinear operators. *SIAM J. on Numerical Analysis*, 16(6):964–979, 1979.

Nemirovski, A. and Yudin, D. *Problem Complexity and Method Efficiency in Optimization*. John Wiley and Sons, 1983.

Nemirovski, A., Juditsky, A., Lan, G., and Shapiro, A. Robust stochastic approximation approach to stochastic programming. *SIAM J. on Optimization*, 19(4):1574–1609, 2009.

Nesterov, Yurii. *Introductory Lectures on Convex Optimization, A Basic Course*. Kluwer Academic Publishers, 2004.

Nesterov, Yurii. Excessive gap technique in nonsmooth convex minimization. *SIAM J. Optim.*, 16(1):235–249, 2005a.

Nesterov, Yurii. Smooth minimization of non-smooth functions. *Math. Program., Ser. A*, 103:127–152, 2005b.

Nesterov, Yurii. Gradient methods for minimizing composite objective function. Technical Report CORE DISCUSSION PAPER 2007/76, 2007a.

Nesterov, Yurii. Smoothing technique and its applications in semidefinite optimization. *Mathematical Programming*, 110(2):245–259, 2007b.

Polyak, Boris T. and Juditsky, Anatoli B. Acceleration of stochastic approximation by averaging. *SIAM J. on Control and Optimization*, 30(4):838–855, 1992.

Robbins, Herbert and Monro, Sutton. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

Shalev-Shwartz, Shai, Singer, Yoram, and Srebro, Nathan. Pegasos: Primal estimated sub-gradient solver for svm. In *ICML*, 2007.

Shalev-Shwartz, Shai, Shamir, Ohad, Srebro, Nathan, and Sridharan, Karthik. Stochastic convex optimization. In *COLT*, 2009.

Shamir, Ohad. Making gradient descent optimal for strongly convex stochastic optimization. In *OPT 2011*, 2011. URL http://arxiv.org/abs/1109.5647.

Wright, S. J., Nowak, R. D., and Figueiredo, M. A. T. Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, 57(7): 2479–2493, 2009.

Xiao, Lin. Dual averaging methods for regularized stochastic learning and online optimization. *JMLR*, 11:2543–2596, 2010.

Xu, Wei. Towards optimal one pass large scale learning with averaged stochastic gradient descent. *arXiv*, 2011. URL http://arxiv.org/abs/1107.2490.