Accelerated Stochastic Gradient Descent for Minimizing Finite Sums

Atsushi Nitanda
Tokyo Institute of Technology and NTT DATA Mathematical Systems Inc.
Tokyo, Japan
nitanda@msi.co.jp

Abstract

We propose an optimization method for minimizing the finite sums of smooth convex functions. Our method incorporates an accelerated gradient descent (AGD) and a stochastic variance reduction gradient (SVRG) in a mini-batch setting. An important feature of the method is that it can be directly applied to general convex and optimal strongly convex problems that is a weaker condition than strong convexity. We show that our method achieves a better overall complexity for the general convex problems and linear convergence for optimal strongly convex problems. Moreover we prove the fast iteration complexity of our method. Our experiments show the effectiveness of our method.

1 Introduction

We consider the minimization problem:

$$\underset{x \in \mathbb{R}^d}{\text{minimize}} \quad f(x) \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} f_i(x), \tag{1}$$

where $f_1, \ldots, f_n$ are smooth convex functions from $\mathbb{R}^d$ to $\mathbb{R}$. In machine learning, we often encounter optimization problems of this type, i.e., empirical risk minimization. For example, given a sequence of training examples $(a_1, b_1), \ldots, (a_n, b_n)$, where $a_i \in \mathbb{R}^d$ and $b_i \in \mathbb{R}$. If we set $f_i(x) = \frac{1}{2}(a_i^T x - b_i)^2$, then we obtain linear regression. If we set $f_i(x) = \log(1+\exp(-b_i x^T a_i))$ $(b_i \in \{-1, 1\})$, then we obtain logistic regression. Each $f_i(x)$ may include smooth regularization terms. In this paper we make the following assumption.

Assumption 1. *Each convex function $f_i(x)$ is L-smooth, i.e., there exists $L > 0$ such that for all $x, y \in \mathbb{R}^d$,*

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|.$$

Several papers recently proposed effective methods (SAG [1, 2], SDCA [3, 4], SVRG [5], S2GD [6], Acc-Prox-SDCA [7], Prox-SVRG [8], MISO [9], SAGA [10], APCG [11], Acc-Prox-SVRG [12], mS2GD [13], SPDC [14]) for solving strongly convex problems. These methods attempt to reduce the variance of the stochastic gradient and achieve the linear convergence rates like a deterministic gradient descent.Moreover, because of the computational efficiency of each iteration, the overall complexities (total number of processed examples to find an ϵ-accurate solution in expectation) of these methods are less than those of the deterministic and stochastic gradient descent methods.

However, many problems are not strongly convex. An advantage of the SAG and SAGA is that they support general convex problems. Although we can apply any of these methods to non-strongly convex functions by adding a slight L_2-regularization, this modification increases the difficulty of model selection. In the general convex case, the overall complexities of SAG and SAGA are $O((n + L)/\epsilon)$. This complexity is less than that of the deterministic gradient descent, which have a complexity of $O(nL/\epsilon)$, and is a trade-off with $O(n\sqrt{L/\epsilon})$, which is the complexity of the AGD.

More recently, [15] showed that Prox-SVRG has linear rate of convergence for optimal strongly convex problems defined as follows:

Assumption 2. *Let C be a subset of $\mathbb{R}^d$ and X_* denote the optimal set. We assume $X_* \neq \phi$. f(x) is μ-optimal-strongly convex on C, i.e., there exists $\mu > 0$ such that for all $x \in C \setminus X_*$,*

$$f_* + \frac{\mu}{2}\|x - \Pi_{X_*}(x)\|^2 \leq f(x),$$

where f_* is the optimal value and Π_{X_*} denotes the projection onto X_*.

Appearing in Proceedings of the 19^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz, Spain. JMLR: W&CP volume 51. Copyright 2016 by the authors.

Clearly, we can see that optimal strong convexity is a weaker condition than strong convexity. Since $f_* + \frac{L}{2}\|x - \Pi_{X_*}(x)\|^2 \geq f(x)$ by L-smoothness, we have $\mu \leq L$. We denote the ratio between L and μ by κ and we call it the *condition number*.

In this paper we propose a new method that incorporates the AGD and SVRG in a mini-batch setting like Acc-Prox-SVRG [12]. The difference between our method and Acc-Prox-SVRG is that our method incorporates [16], which is similar to Nesterov's acceleration [17], whereas Acc-Prox-SVRG incorporates [18]. An important feature of our method is that it can be directly applied to general convex and optimal strongly convex problems. We show that for general convex problems, AMSVRG achieves an overall complexity of

$$\tilde{O}\left(n + \min\left\{\frac{L}{\epsilon}, n\sqrt{\frac{L}{\epsilon}}\right\}\right),$$

where the notation $\tilde{O}$ hides constant and logarithmic terms. This complexity is less than that of SAG, SAGA, and AGD. In the optimal strongly convex case, our method achieves an overall complexity

$$\tilde{O}\left(n + \min\left\{\kappa,\ n\sqrt{\kappa}\ \right\}\right),$$

where κ is the condition number L/μ. Moreover, an iteration complexity (the number of iterations needed to find an ϵ-accurate solution in expectation) is

$$O\left(\sqrt{\kappa}\log\left(\frac{1}{\epsilon}\right)\right),$$

This iteration complexity is the same as that of deterministic acceleration methods, i.e., best iteration complexity. Thus, our method converges quickly for general convex and optimal strongly convex problems.

In Section 2, we discuss an optimal strongly convex function. In Section 3, we review the recently proposed accelerated gradient method [16] and the stochastic variance reduction gradient [5]. In Section 4, we describe the general scheme of our method and prove an important lemma that gives us a novel insight for constructing specific algorithms. Moreover, we derive an algorithm that is applicable to general convex and optimal strongly convex problems and show its quickly converging complexity. Our method is a multi-stage scheme like SVRG, but it can be difficult to decide when we should restart a stage. Thus, in Section 5, we introduce some heuristics for determining the restarting time. In Section 6, we present experiments that show the effectiveness of our method.

2 Optimal Strongly Convex

The main differences between strong convexity and optimal strong convexity are that the latter condition admits an infinite number of solutions and linear parts of the function. Thus, optimal strongly convex is a very large class.

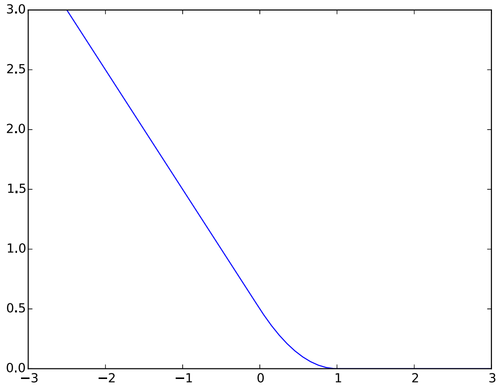

Figure 1: Smoothed hinge loss function.

Two quantities $\frac{1}{2}\|x - \Pi_{X_*}(x)\|^2$ and $f(x) - f_*$ are optimality measures and continuous functions, so that the ratio between these two values: $\mu(x) = \frac{2(f(x)-f_*)}{\|x-\Pi_{X_*}(x)\|^2}$ is positive continuous on the complement of X_*. Let $C \subset \mathbb{R}^d$ be a compact subset. Then, $\mu \overset{\mathrm{def}}{=} \inf_{x \in C \setminus X_*} \mu(x)$ gives the optimal strong convexity parameter on C. Since $C \setminus U$ is also compact, where U is an arbitrary small open neighborhood of X_*, $\mu(x)$ has positive minimum values on $C \setminus U$. This means that whether Assumption 2 is satisfied or not depend on the behavior of f around the boundary of $C \cap X_*$. Therefore, many problems belong to the class of optimal strongly convex on compact set.

A smoothed hinge loss function (Fig. 1)

$$f(x) = \left\{\begin{array}{ll} \frac{1}{2} - x & (x \leq 0), \\ \frac{1}{2}(1-x)^2 & (0 < x \leq 1), \\ 0 & (1 < x), \end{array}\right.$$

is a simple example of optimal strongly function on a bounded region. Let $C = [-a, a]$ $(a > 1)$ be a bounded range. Since $X_* = [1, \infty)$ and $\Pi_{X_*}(x) = 1 \ for \ x \notin X_*$, we can easily see that optimal strong convexity parameter on C for smoothed hinge loss is

$$\mu = \inf_{x \in [-a,1)} \mu(x) = \frac{2f(-a)}{|1+a|^2} = \frac{1+2a}{|1+a|^2} > 0.$$

Here, we checked the value of μ, but we can conclude the positivity of μ by the fact that $C = [-a, a]$ is compact and f is quadratic around $\partial(C \cap X_*) = \{1\}$.

In our analyses, for optimal strongly convex problems we assume that points generated by algorithm is contained in C. For monotonic algorithms (generating decreasing sequence $f(w_s)_{s=1,2,\ldots}$), we may consider the case where C is the sublevel set $\{x \in \mathbb{R}^d; f(x) \leq c\}$ and this assumption holds for sufficiently large $c \geq f_*$. Here, we give the condition of compactness of sublevel set.

Proposition 1. *Let f be C^1 class convex function and X_* be the optimal set of f. If X_* is compact, then for $c \geq f_*$, the sublevel set $\{x \in \mathbb{R}^d; f(x) \leq c\}$ is also compact.*

Thus, by the above discussion, the monotonic algorithm deals with many problems as optimal strongly convex problems and potentially converge fast. We propose such a method later.

3 Preparation

In this section, we review the recently proposed accelerated gradient method and the stochastic variance reduction gradient to introduce our new method.

3.1 Accelerated Gradient Descent

We first introduce some notations. In this section, $\|\cdot\|$ denotes the general norm on $\mathbb{R}^d$. Let $d(x) : \mathbb{R}^d \to \mathbb{R}$ be a distance generating function (i.e., 1-strongly convex smooth function with respect to $\|\cdot\|$). Accordingly, we define the Bregman divergence by

$$V_x(y) = d(y) - (d(x) + (\nabla d(x), y - x)), \quad \forall x, \forall y \in \mathbb{R}^d,$$

where $(,)$ is the Euclidean inner product. The accelerated method proposed in [16] uses a gradient step and mirror descent steps and takes a linear combination of these points. That is,

$$\begin{aligned}
&(Convex\ Combination)\\
&x_{k+1} \leftarrow \tau_k z_k + (1-\tau_k) y_k,\\
&(Gradient\ Descent)\\
&y_{k+1} \leftarrow x_{k+1} - \eta \nabla f(x_{k+1}),\\
&(Mirror\ Descent)\\
&z_{k+1} \leftarrow \arg\min_{z \in \mathbb{R}^d} \{\ \alpha_{k+1}(\nabla f(x_{k+1}), z - z_k) + V_{z_k}(z)\ \}.
\end{aligned}$$

Then, with appropriate parameters, $f(y_k)$ converge to the optimal value as fast as the Nesterov's accelerated methods [17, 18] for non-strongly convex problems. Moreover, in the strongly convex case, we obtain the same fast convergence as Nesterov's methods by restarting this entire procedure.

In the rest of the paper, we only consider the Euclidean norm, i.e., $\|\cdot\| = \|\cdot\|_2$.

3.2 Stochastic Variance Reduction Gradient

To ensure the convergence of stochastic gradient descent (SGD), the learning rate must decay to zero so that we can reduce the variance effect of the stochastic gradient. This slows down the convergence. Variance reduction techniques [5, 6, 8, 13] such as SVRG have been proposed to solve this problem. We review SVRG in a mini-batch setting [12,13]. SVRG is a multi-stage scheme. During each stage, this method performs m SGD iterations using the following direction,

$$v_k = \nabla f_{I_k}(x_k) - \nabla f_{I_k}(\tilde{x}) + \nabla f(\tilde{x}),$$

where $\tilde{x}$ is a starting point at stage, k is an iteration index, $I_k = \{i_1, \ldots, i_b\}$ is a uniformly randomly chosen size b subset of $\{1, 2, \ldots, n\}$, and $f_{I_k} = \frac{1}{b}\sum_{j=1}^{b} f_{i_j}$. Note that v_k is an unbiased estimator of gradient $\nabla f(x_k)$: $\mathbb{E}_{I_k}[v_k] = \nabla f(x_k)$, where $\mathbb{E}_{I_k}$ denotes the expectation with respect to I_k. A bound on the variance of v_k is given in the following lemma, which is proved in the Supplementary Material.

Lemma 1. *Suppose Assumption 1 holds, and let $x_* = \arg\min_{x \in \mathbb{R}^d} f(x)$. Conditioned on x_k, we have*

$$\begin{aligned}
&\mathbb{E}_{I_k}\|v_k - \nabla f(x_k)\|^2\\
&\quad \leq 4L\frac{n-b}{b(n-1)}\left(f(x_k) - f_* + f(\tilde{x}) - f_*\right). \qquad (2)
\end{aligned}$$

Due to this lemma, SVRG with $b = 1$ achieves a complexity of $O((n+\kappa)\log\frac{1}{\epsilon})$.

4 Algorithms

We now introduce our *Accelerated efficient Minibatch SVRG (AMSVRG)* which incorporates AGD and SVRG in a mini-batch setting. Our method is a multi-stage scheme similar to SVRG. During each stage, this method performs several APG-like [16] iterations combining stochastic gradient descent (SGD) and stochastic mirror descent (SMD) steps with SVRG direction in a mini-batch setting. Each stage of AMSVRG is described in Figure 2.

4.1 Convergence analysis of the single stage of AMSVRG

Before we introduce the multi-stage scheme, we show the convergence of single-stage version Algorithm 1. The following lemma is the key to the analysis of our method and gives us an insight on how to construct algorithms.

Lemma 2. *Consider Algorithm 1 in Figure 2 under Assumption 1. We set $\delta_k = \frac{n-b_k}{b_k(n-1)}$. Let $x_* \in$*

```
**Algorithm 1**$(y_0,\ z_0,\ m,\ \eta,\ (\alpha_{k+1})_{k\in\mathbb{Z}_+},\ (b_{k+1})_{k\in\mathbb{Z}_+},\ (\tau_k)_{k\in\mathbb{Z}_+})$

$\tilde{v} \leftarrow \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(y_0)$
**for** $k \leftarrow 0$ **to** $m$
    $x_{k+1} \leftarrow (1-\tau_k)y_k + \tau_k z_k$
    Randomly pick subset $I_{k+1} \subset \{1,2,\ldots,n\}$ of size $b_{k+1}$
    $v_{k+1} \leftarrow \nabla f_{I_{k+1}}(x_{k+1}) - \nabla f_{I_{k+1}}(y_0) + \tilde{v}$
    $y_{k+1} \leftarrow \arg\min_{y\in\mathbb{R}^d}\left\{\ \eta(v_{k+1}, y - x_{k+1}) + \frac{1}{2}\|y - x_{k+1}\|^2\ \right\}$   (SGD step)
    $z_{k+1} \leftarrow \arg\min_{z\in\mathbb{R}^d}\left\{\ \alpha_{k+1}(v_{k+1}, z - z_k) + V_{z_k}(z)\ \right\}$   (SMD step)
**end for**
**Option I-a:** $w \leftarrow y_{m+1}$,
**Option I-b:** $w \leftarrow \frac{1}{m+1}\sum_{k=1}^{m+1} x_k$,
**Option II:** If $f(y_1) < f(w)$, then $w \leftarrow y_1$,
Return $w$
```

Figure 2: Each stage of AMSVRG

$\arg\min_{x\in\mathbb{R}^d} f(x)$. If $\eta = \frac{1}{L}$, then we have,

$$\sum_{k=0}^{m}\alpha_{k+1}\left(\frac{1}{\tau_k} - (1+4\delta_{k+1})L\alpha_{k+1}\right)\mathbb{E}[f(x_{k+1}) - f_*)]$$
$$+L\alpha_{m+1}^2\mathbb{E}[f(y_{m+1}) - f_*]$$
$$\leq V_{z_0}(x_*) + \sum_{k=1}^{m}\left(\alpha_{k+1}\frac{1-\tau_k}{\tau_k} - L\alpha_k^2\right)\mathbb{E}[f(y_k) - f_*]$$
$$+\left(\alpha_1\frac{1-\tau_0}{\tau_0} + 4L\sum_{k=0}^{m}\alpha_{k+1}^2\delta_{k+1}\right)(f(y_0) - f_*).$$

To prove Lemma 2, the following additional lemmas are required.

Lemma 3. *(Stochastic Gradient Descent). Suppose Assumption 1 holds, and let $\eta = \frac{1}{L}$. Conditioned on x_k, it follows that for $k \geq 1$,*

$$\mathbb{E}_{I_k}[f(y_k)] \leq$$
$$f(x_k) - \frac{1}{2L}\|\nabla f(x_k)\|^2 + \frac{1}{2L}\mathbb{E}_{I_k}\|v_k - \nabla f(x_k)\|^2. \quad (3)$$

Lemma 4. *(Stochastic Mirror Descent). Conditioned on x_k, we have that for arbitrary $u \in \mathbb{R}^d$,*

$$\alpha_k(\nabla f(x_k), z_{k-1} - u)$$
$$\leq V_{z_{k-1}}(u) - \mathbb{E}_{I_k}[V_{z_k}(u)] + \frac{1}{2}\alpha_k^2\|\nabla f(x_k)\|^2$$
$$+\frac{1}{2}\alpha_k^2\mathbb{E}_{I_k}\|v_k - \nabla f(x_k)\|^2. \quad (4)$$

These lemmas are proved in the Supplementary Material.

From now on we consider Algorithm 1 with Option I-a and we set η, α_{k+1}, and τ_k as follows: For $k = 0, 1, \ldots$ we set

$$\eta = \frac{1}{L}, \quad \alpha_{k+1} = \frac{1}{4L}(k+2), \quad \frac{1}{\tau_k} = L\alpha_{k+1} + \frac{1}{2}. \quad (5)$$

Theorem 1. *Consider Algorithm 1 with Option I-a under Assumption 1. For $p \in \left(0, \frac{1}{2}\right]$, we choose $b_{k+1} \in \mathbb{Z}_+$ such that $4L\delta_{k+1}\alpha_{k+1} \leq p$. Then, we have*

$$\mathbb{E}[f(w) - f_*] \leq \mathbb{E}[f(y_{m+1}) - f_*]$$
$$\leq \frac{16L}{(m+2)^2}V_{z_0}(x_*) + \frac{5}{2}p(f(y_0) - f_*).$$

Moreover, if $m \geq 4\sqrt{\frac{LV_{z_0}(x_*)}{q(f(y_0)-f_*)}}$ for $q > 0$, then it follows

$$\mathbb{E}[f(w)-f_*] \leq \mathbb{E}[f(y_{m+1})-f_*] \leq \left(q + \frac{5}{2}p\right)(f(y_0)-f_*).$$

Proof. Using Lemma 2 and

$$\tau_0 = 1, \quad \frac{1}{\tau_k} - (1+4\delta_{k+1})L\alpha_{k+1} \geq 0,$$
$$\alpha_{k+1}\frac{1-\tau_k}{\tau_k} - L\alpha_k^2 = L\alpha_{k+1}^2 - \frac{1}{2}\alpha_{k+1} - L\alpha_k^2$$
$$= -\frac{1}{16L} < 0,$$

we have

$$L\alpha_{m+1}^2\mathbb{E}[f(y_{m+1}) - f_*]$$
$$\leq V_{z_0}(x_*) + 4L\sum_{k=0}^{m}\alpha_{k+1}^2\delta_{k+1}(f(y_0) - f_*).$$

This proves the theorem because $4L\sum_{k=0}^{m}\alpha_{k+1}^2\delta_{k+1} \leq p\sum_{k=0}^{m}\alpha_{k+1} \leq \frac{5p}{32L}(m+2)^2$. □

Let $b_{k+1}, m \in \mathbb{Z}_+$ be the minimum values satisfying the assumption of Theorem 1 for $p = q = \epsilon$, i.e., $b_{k+1} = \left\lceil\frac{n(k+2)}{\epsilon(n-1)+k+2}\right\rceil$ and $m = \left\lceil 4\sqrt{\frac{LV_{z_0}(x_*)}{\epsilon(f(y_0)-f_*)}}\right\rceil$. Then, from Theorem 1, we have an upper bound on the overall complexity (total number of processed examples to obtain ϵ-accurate solution in expectation):

$$O\left(n+\sum_{k=0}^{m} b_{k+1}\right) \le O\left(n+m\frac{nm}{\epsilon n+m}\right)$$
$$= O\left(n+\frac{nL}{\epsilon^2 n+\sqrt{\epsilon L}}\right),$$

where we used the monotonicity of b_{k+1} with respect to k for the first inequality. Note that the notation O also hides $V_{z_0}(x_*)$ and $f(y_0)-f_*$.

4.2 Multi-Stage Scheme

In this subsection, we introduce and analyze AMSVRG, as described in Figure 3.

If we run Algorithm 1 with Option II in AMSVRG, it follows that $f(w) \le f(y_1)$. Since $x_1 = y_0 = z_0$, the step to obtain y_1 corresponds to the deterministic gradient descent from the starting point at each stage. This means that AMSVRG (with Option II) is monotonic that generates decreasing sequence $\{f(w_s)\}_{s=0,1,\ldots}$. Note that Option II requires computational cost for computing function values of $O(n)$ but the order of overall complexity does not change.

4.2.1 General Convex

We consider the convergence of AMSVRG for general convex problems under the following boundedness assumption which has been used in a several papers to analyze incremental and stochastic methods (e.g., [19, 20]).

Assumption 3. *(Boundedness) There is a compact subset $\Omega \subset \mathbb{R}^d$ such that the sequence $\{w_s\}$ generated by AMSVRG is contained in Ω.*

Note that, if we change the initialization of $z_0 \leftarrow w_s$ to $z_0 \leftarrow z$: *constant*, the above method with this modification will achieve the same convergence for general convex problems without the boundedness assumption (c.f. supplementary materials). However, for the strongly convex case, this modified version is slower than the above scheme. Therefore, we consider the version described in Figure 3.

From Theorem 1, we can see that for small p and q (e.g. $p = 1/10,\ q = 1/4$), the expected value of the objective function is halved at every stage under the assumptions of Theorem 1. Hence, running AMSVRG for $O(\log(1/\epsilon))$ outer iterations achieves an ϵ-accurate solution in expectation. Here, we consider the complexity at stage s to halve the expected objective value. Let $b_{k+1}, m_s \in \mathbb{Z}_+$ be the minimum values satisfying the assumption of Theorem 1, i.e., $b_{k+1} = \left\lceil \frac{n(k+2)}{p(n-1)+k+2} \right\rceil$ and $m_s = \left\lceil 4\sqrt{\frac{LV_{w_s}(x_*)}{q(f(w_s)-f_*)}} \right\rceil$. If the initial objective gap $f(w_s) - f_*$ in stage s is larger than ϵ, then the complexity at stage is

$$O\left(n+\sum_{k=0}^{m_s} b_{k+1}\right) \le O\left(n+\frac{nm_s^2}{n+m_s}\right)$$
$$= O\left(n+\frac{nL}{n(f(w_s)-f_*)+\sqrt{(f(w_s)-f_*)L}}\right)$$
$$\le O\left(n+\frac{nL}{\epsilon n+\sqrt{\epsilon L}}\right),$$

where we used the monotonicity of b_{k+1} with respect to k for the first inequality. Note that by Assumption 3, $\{V_{w_s}(x_*)\}_{s=1,2,\ldots}$ are uniformly bounded and notation O also hides $V_{w_s}(x_*)$. The above analysis implies the following theorem.

Theorem 2. *Consider AMSVRG under Assumptions 1 and 3. We set η, α_{k+1}, and τ_k as in (5). Let $b_{k+1} = \left\lceil \frac{n(k+2)}{p(n-1)+k+2} \right\rceil$ and $m_s = \left\lceil 4\sqrt{\frac{LV_{w_s}(x_*)}{q(f(w_s)-f_*)}} \right\rceil$, where p and q are small values described above. Then, the overall complexity to run AMSVRG for $O(\log(1/\epsilon))$ outer iterations or to obtain an ϵ-accurate solution is*

$$O\left(\left(n+\frac{nL}{\epsilon n+\sqrt{\epsilon L}}\right)\log\left(\frac{1}{\epsilon}\right)\right).$$

4.2.2 Optimal Strongly Convex

Next, we consider the optimal strongly convex case. We assume that f is a μ-optimal-strongly convex function on $C \subset \mathbb{R}^d$. In this case, we choose the distance generating function $d(x) = \frac{1}{2}\|x\|^2$, so that the Bregman divergence becomes $V_x(y) = \frac{1}{2}\|x-y\|^2$. Let the parameters be the same as in Theorem 2 with $x_* = \Pi_{X_*}(w_s)$ at stage s. Then, the expected value of the objective function is halved at every stage. Moreover, we assume that $\{w_s\}_{s=0,1,\ldots} \subset C$. As mentioned in Section 2, for monotonic methods, we may consider the case where C is the sublevel set $\{x \in \mathbb{R}^d; f(x) \le c\}$ and this assumption holds for sufficiently large level. Since, by definition of optimal strong convexity, we have $m_s = \left\lceil 4\sqrt{\frac{L\|w_s-\Pi_{X_*}(w_s)\|^2}{2q(f(w_s)-f_*)}} \right\rceil \le \left\lceil 4\sqrt{\frac{\kappa}{q}} \right\rceil$, the complexity at each stage is

$$O\left(n+\sum_{k=0}^{m_s} b_{k+1}\right) \le O\left(n+\frac{n\kappa}{n+\sqrt{\kappa}}\right).$$

Thus, we have the following theorem.

Theorem 3. *Consider AMSVRG under Assumptions 1 and 2. Let parameters $\eta, \alpha_{k+1}, \tau_k, m_s$, and b_{k+1} be the same as those in Theorem 2 with $x_* = \Pi_{X_*}(w_s)$ at stage s. If $\{w_s\}_{s=0,1,\ldots} \subset C$, then the overall complexity for obtaining ϵ-accurate solution in expectation*

```
Algorithm 2($w_0,\ (m_s)_{s\in\mathbb{Z}_+},\ \eta,\ (\alpha_{k+1})_{k\in\mathbb{Z}_+},\ (b_{k+1})_{k\in\mathbb{Z}_+},\ (\tau_k)_{k\in\mathbb{Z}_+}$)

for $s \leftarrow 0,\ 1, \ldots$
    $y_0 \leftarrow w_s,\ \ z_0 \leftarrow w_s$
    $w_{s+1} \leftarrow$ Algorithm1($y_0,\ z_0,\ m_s,\ \eta,\ (\alpha_{k+1})_{k\in\mathbb{Z}_+},\ (b_{k+1})_{k\in\mathbb{Z}_+},\ (\tau_k)_{k\in\mathbb{Z}_+}$)
end
```

Figure 3: Accelerated efficient Mini-batch SVRG

is

$$O\left(\left(n+\frac{n\kappa}{n+\sqrt{\kappa}}\right)\log\left(\frac{1}{\epsilon}\right)\right),$$

and its iteration complexity is

$$O\left(\sqrt{\kappa}\log\left(\frac{1}{\epsilon}\right)\right).$$

Table 1 lists the overall complexities of the AGD, SAG, SVRG, Acc-Prox-SVRG, Acc-SDCA, APCG, SPDC, and AMSVRG. The notation $\tilde{O}$ hides constant and logarithmic terms. By simple calculations, we see that

$$\frac{n\kappa}{n+\sqrt{\kappa}} = \frac{1}{2}H(\kappa, n\sqrt{\kappa}\ ),$$

$$\frac{nL}{\epsilon n+\sqrt{\epsilon L}} = \frac{1}{2}H\left(\frac{L}{\epsilon}, n\sqrt{\frac{L}{\epsilon}}\right),$$

where $H(\cdot,\cdot)$ is the harmonic mean whose order is the same as $\min\{\cdot,\cdot\}$. Thus, as shown in Table 1, the complexity of AMSVRG is less than or equal to that of other methods in general convex and optimal strongly convex.

4.3 Fast Iteration Complexity and its Benefits

We consider the optimal strongly convex case. It is well known that for deterministic optimization problems, Nesterov's acceleration achieves the best iterations complexity, while for stochastic optimization problems, complexity reduction by acceleration is slight due to the variance of stochastic gradient. There are several observations: (i) acceleration + small mini-batching may have the almost same convergence rate as that of SGD as indicated by [21], (ii) acceleration + SVRG without mini-batching has the same iteration complexity as SVRG by [12], (iii) Mini-batching + SVRG has the almost same iteration complexity as deterministic gradient descent because SVRG is a stochastic variant of it. On the other hand, by combining three techniques: acceleration, mini-batching, and SVRG, AMSVRG achieves the same iteration complexity as Nesterov's acceleration, as shown in the previous subsection. These observations mean that we may need not only SVRG but also mini-batching to obtain sufficiently small variance for the acceleration scheme. This feature of our acceleration scheme leads to some advanteges: effective parallelization and better performance for linear-model on a sparse dataset without using sparse structure.

For strongly convex problems, AMSVRG is slower than optimal methods: Acc-SDCA, APCG, and SPDC. However, the gradient evaluations for the mini-batch can be parallelized [25–27]. Let's consider the case where f is strongly convex and $n \geq \sqrt{\kappa}$. Although the overall complexity of AMSVRG is the same as that of SAG and SVRG, AMSVRG has better iteration complexity $\tilde{O}(\sqrt{\kappa})$ than the others thanks to Nesterov-like acceleration. This means that mini-batch parallelization scheme of AMSVRG leads to a significant improvement. Let P be a number of processors and $P \leq O(\sqrt{\kappa})$: maximum mini-batch size. If we ignore the communication delay, then the overall complexity at each processor is at most $\tilde{O}(n/P+\kappa/P)$. If we can use $P = O(\sqrt{\kappa})$, it is $\tilde{O}(n/\sqrt{\kappa}+\sqrt{\kappa})$. Hence AMSVRG is very scalable with respect to P and potentially becomes faster than some of optimal methods. Let's consider SPDC [14] that is one of the optimal method for strongly convex and support mini-batch setting. The complexity of SPDC at each processor is $\tilde{O}(n/P+\sqrt{\kappa n/P})$, when P is equal to mini-batch size (best choice for SPDC). Thus, if $\kappa/P < n$, AMSVRG may be faster than SPDC.

Next, we discuss the performance of AMSVRG for linear model that takes a form of $f_i(x) = l(a_i^T x)$ on a sparse dataset $\{a_i\}_{i=1,\ldots,n}$. Since $\nabla f_i(x) = l'(a_i^T x)a_i$, some algorithms such as SGD and SVRG can be updated efficiently by using sparsity of a_i. It is unclear whether AMSVRG can be also implemented efficiently, but our acceleration scheme reduces the number of dense computations, consequently, AMSVRG has the same complexity as sparse implementation of SVRG, without using sparse structure for problems with large condition number. Let d_0 be the maximum number of non-zero elements of a_i. Then, the overall complexity including d and d_0 of AMSVRG is as follows:

$$\tilde{O}\left(nd_0+m(b_m d_0+d)\right) \leq \tilde{O}\left(nd_0+\kappa\left(d_0+\frac{d}{\sqrt{\kappa}}\right)\right),$$

Table 1: Comparison of overall complexity.

Algorithm	General Convex	Optimal Strongly Convex	Strongly Convex
AGD	$\tilde{O}\left(n\sqrt{\frac{L}{\epsilon}}\right)$	$\tilde{O}\left(n\sqrt{\kappa}\right)$	$\tilde{O}\left(n\sqrt{\kappa}\right)$
SAG	$\tilde{O}\left(\frac{n+L}{\epsilon}\right)$	$\tilde{O}\left(\frac{n+L}{\epsilon}\right)$	$\tilde{O}\left(\max\{n,\kappa\}\right)$
SVRG	—	$\tilde{O}\left(n+\kappa\right)$	$\tilde{O}\left(n+\kappa\right)$
Acc-SVRG	—	—	$\tilde{O}\left(n+\min\left\{\kappa,\ n\sqrt{\kappa}\ \right\}\right)$
Acc-SDCA, APCG, SPDC	—	—	$\tilde{O}\left(n+\min\left\{\kappa,\ \sqrt{n\kappa}\ \right\}\right)$
AMSVRG	$\tilde{O}\left(n+\min\left\{\frac{L}{\epsilon}, n\sqrt{\frac{L}{\epsilon}}\ \right\}\right)$	$\tilde{O}\left(n+\min\left\{\kappa,\ n\sqrt{\kappa}\ \right\}\right)$	$\tilde{O}\left(n+\min\left\{\kappa,\ n\sqrt{\kappa}\ \right\}\right)$

where we used $m \le O(\sqrt{\kappa})$ and $b_m \le O(m) = O(\sqrt{\kappa})$. Hence, if the condition number is sufficiently large: $d/d_0 \le \sqrt{\kappa}$, the overall complexity is

$$\tilde{O}\left(d_0(n+\kappa)\right).$$

Therefore, AMSVRG efficiently performs on sparse datasets without an implementation trick.

5 Restart Scheme

The parameters of AMSVRG are essentially η, m_s, and b_{k+1} (*i.e.*, p) because the appropriate values of both α_{k+1} and τ_k can be expressed by $\eta = 1/L$ as in (5). It may be difficult to choose an appropriate m_s which is the restart time for Algorithm 1. So, we propose heuristics for determining the restart time.

First, we suppose that the number of components n is sufficiently large such that the complexity of our method becomes $O(n)$. That is, for appropriate m_s, $O(n)$ is an upper bound on $\sum_{k=0}^{m_s} b_{k+1}$ (which is the complexity term). Therefore, we estimate the restart time as the minimum index $m \in \mathbb{Z}_+$ that satisfies $\sum_{k=0}^{m} b_{k+1} \ge n$. This estimated value is upper bound on m_s (in terms of the order). In this paper, we call this restart method *R1*.

Second, we propose an adaptive restart method using SVRG. In a strongly convex case, we can easily see that if we restart the AGD for general convex problems every $\sqrt{\kappa}$, then the method achieves a linear convergence similar to that for strongly convex problems. The drawback of this restart method is that the restarting time depends on an unknown parameter κ, so several papers [22–24] have proposed effective adaptive restart methods. Moreover, [23] showed that this technique also performs well for general convex problems. Inspired by their study, we propose an SVRG-based adaptive restart method called *R2*. That is, if

$$(v_{k+1}, y_{k+1} - y_k) > 0,$$

then we return y_k and start the next stage.

Third, we propose the restart method *R3*, which is a combination of the above two ideas. When $\sum_{k=0}^{m} b_{k+1}$ exceeds $10n$, we restart Algorithm 1, and when

$$(v_{k+1}, y_{k+1} - y_k) > 0 \quad \wedge \quad \sum_{k=0}^{m} b_{k+1} > n,$$

we return y_k and restart Algorithm 1.

6 Numerical Experiments

In this section, we compare AMSVRG with SVRG and SAGA. We ran an $L2$-regularized multi-class logistic regularization on *mnist* and *covtype* and ran an $L2$-regularized binary-class logistic regularization on *rcv1*. The datasets and their descriptions can be found at the LIBSVM website[1]. In these experiments, we vary regularization parameter λ in $\{0,\ 10^{-7},\ 10^{-6},\ 10^{-5}\}$. We ran AMSVRG using some values of η from $[10^{-2},\ 5\times 10]$ and p from $[10^{-1},\ 10]$, and then we chose the best η and p.

The results are shown in Figure 4. The horizontal axis is the number of single-component gradient evaluations. Our methods performed well and outperformed the other methods in some cases. For mnist and covtype, AMSVRG R1 and R3 converged quickly, and for rcv1, AMSVRG R2 worked very well. This tendency was more remarkable when the regularization parameter λ was small.

[1] http://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/

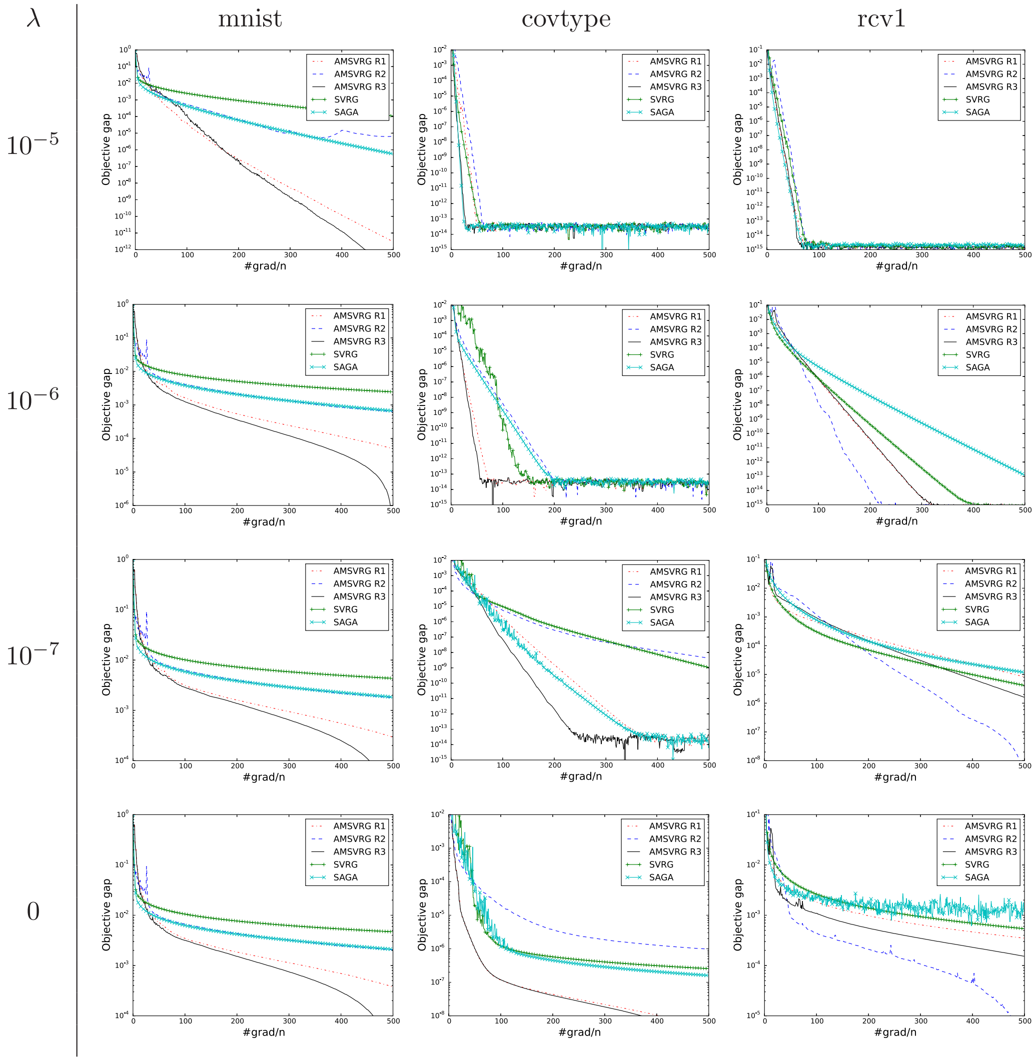

Figure 4: Comparison of algorithms applied to $L2$-regularized multi-class logistic regularization (left: mnist, middle: covtype), and $L2$-regularized binary-class logistic regularization (right: rcv1).

7 Conclusion

We propose method that incorporates accelerated gradient method and the SVRG in the increasing mini-batch setting. We showed that our method achieves a fast convergence complexity for general convex and optimal strongly convex problems.

References

[1] N. Le Roux, M. Schmidt, and F. Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. *Advances in Neural Information Processing System 25*, pages 2672-2680, 2012.

[2] M. Schmidt, N. Le Roux, and F. Bach. Minimizing finite sums with the stochastic average gradient. *arXiv:1309.2388*, 2013.

[3] S. Shalev-Shwartz and T. Zhang. Proximal stochastic dual coordinate ascent. *arXiv:1211.2717*, 2012.

[4] S. Shalev-Shwartz and T. Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research 14*, pages 567-599, 2013.

[5] R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in Neural Information Processing System 26*, pages 315-323, 2013.

[6] J. Konečný and P. Richtárik. Semi-stochastic gradient descent methods. *arXiv:1312.1666*, 2013.

[7] S. Shalev-Shwartz and T. Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. *Proceedings of the 31th International Conference on Machine Learning*, pages 64-72, 2014.

[8] L. Xiao and T. Zhang. A proximal stochastic gradient method with progressive variance reduction. *arXiv:1403.4699*, 2014.

[9] J. Mairal. Incremental majorization-minimization optimization with application to large-scale machine learning. *SIAM Journal on Optimization*, 25(2), pages 829-855, 2015.

[10] A. Defazio, F. Bach, and S. Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in Neural Information Processing System 27*, pages 1646-1654, 2014.

[11] Q. Lin, Z. Lu, and L. Xiao. An accelerated proximal coordinate gradient method and its application to regularized empirical risk minimization. *Advances in Neural Information Processing System 27*, pages 3059-3067, 2014.

[12] A. Nitanda. Stochastic proximal gradient descent with acceleration techniques. *Advances in Neural Information Processing System 27*, pages 1574-1582, 2014.

[13] J. Konečný, J. Liu, P. Richtárik, and M. Takáč. Mini-batch semi-stochastic gradient descent in the proximal setting. *arXiv:1504.04407*, 2015.

[14] Y. Zhang and L. Xiao. Stochastic primal-dual coordinate method for regularized empirical risk minimization. *Proceedings of the 32th International Conference on Machine Learning*. pages 353-361, 2015.

[15] P. Gong and J. Ye. Linear convergence of variance-reduced stochastic gradient without strong convexity. *arXiv:1406.1102v2*, 2015.

[16] Z. Allen-Zhu and L. Orecchia. Linear coupling of gradient and mirror descent: A novel, simple interpretation of Nesterov's accelerated method. *arXiv:1407.1537*, 2015.

[17] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1), pages 127-152, 2005.

[18] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course.* Kluwer, Boston, 2004.

[19] L. Bottou and Y. LeCun. On-line learning for very large datasets. *Applied Stochastic Models in Business and Industry*, 21(2), pages 137-151, 2005.

[20] M. Gürbüzbalaban, A. Ozdaglar, and P. Parrilo. A globally convergent incremental Newton method. *arXiv:1410.5284*, 2014.

[21] A. Cotter, O. Shamir, N. Srebro, and K. Sridharan. Better mini-batch algorithms via accelerated gradient methods. *Advances in Neural Information Processing System 24*, pages 1647-1655, 2011.

[22] B. O'Donoghue and E. J. Candès. Adaptive restart for accelerated gradient schemes. *Foundations of Computational Mathematics*, pages 1-18, 2013.

[23] P. Giselsson and S. Boyd. Monotonicity and restart in fast gradient methods. *In 53rd IEEE Conference on Decision and Control*, pages 5058-5063, 2014.

[24] W. Su, S. Boyd, and E. J. Candès. A differential equation for modeling Nesterov's accelerated gradient method: theory and insights. *Advances in Neural Information Processing System 27*, pages 2510-2518, 2014.

[25] A. Agarwal and J. Duchi. Distributed delayed stochastic optimization. *Advances in Neural Information Processing System 24*, pages 873-881, 2011.

[26] O. Dekel, R. Gilad-Bachrach, O. Shamir, and L. Xiao. Optimal distributed online prediction using mini-batches. *Journal of Machine Learning Research 13*, pages 165-202, 2012.

[27] S. Shalev-Shwartz and T. Zhang. Accelerated mini-batch stochastic dual coordinate ascent. *Advances in Neural Information Processing System 26*, pages 378-385, 2013.