Improved SVRG for Non-Strongly-Convex or Sum-of-Non-Convex Objectives

Zeyuan Allen-Zhu
zeyuan@csail.mit.edu
Princeton University

Yang Yuan
yangyuan@cs.cornell.edu
Cornell University

February 5, 2016

Abstract

Many classical algorithms are found until several years later to outlive the confines in which they were conceived, and continue to be relevant in unforeseen settings. In this paper, we show that SVRG is one such method: originally designed for strongly convex objectives, is also very robust under non-strongly convex or sum-of-non-convex settings.

If \( f(x) \) is a sum of smooth, convex functions but \( f \) is not strongly convex (such as Lasso or logistic regression), we propose a variant \( \text{SVRG}^{++} \) that makes a novel choice of growing epoch length on top of \( \text{SVRG} \). \( \text{SVRG}^{++} \) is a direct, faster variant of \( \text{SVRG} \) in this setting.

If \( f(x) \) is a sum of non-convex functions but \( f \) is strongly convex, we show that the convergence of \( \text{SVRG} \) linearly depends on the non-convexity parameter of the summands. This improves the best known result in this setting, and gives better running time for stochastic PCA.

1 Introduction

The fundamental algorithmic problem in optimization is to design efficient algorithms for solving certain classes of problems. By distinguishing between smooth and non-smooth functions, between weakly-convex and strongly-convex functions, between proximal and non-proximal functions, or even between convex and non-convex functions, the number of classes grows exponentially and it may be unrealistic to design a new algorithm for each specific class. Taking into account such “design complexity”, it is beneficial to design a single method the works for multiple classes, or perhaps even more beneficial if this method is already widely used and happens to outlive the confines it was originally designed for. Easier done in practice, providing a support theory unifying the underlying classes for a specific method is particularly exciting, challenging, and sometimes even enlightening: the theoretical findings may further suggest experimentalists regarding how such a method should be best tuned in practice.

In this paper, we revisit the SVRG method by Johnson and Zhang [10] and explore its applications to either a non-strongly convex objective, or a sum-of-non-convex objective, or even both. We show faster convergence results for minimizing such objectives by either directly applying SVRG or modifying it in a novel manner.

Consider the following composite convex minimization:

\[
\min_{x \in \mathbb{R}^d} \left\{ F(x) \overset{\text{def}}{=} f(x) + \Psi(x) \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} f_i(x) + \Psi(x) \right\}.
\]
Here, \( f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x) \) is a convex function that is written as a finite average of \( n \) smooth functions \( f_i(x) \), and \( \Psi(x) \) is a relatively simple (but possibly non-differentiable) convex function, sometimes referred to as the \textit{proximal} function. Suppose we are interested in finding an approximate minimizer \( x \in \mathbb{R}^d \) satisfying \( F(x) \leq F(x^*) + \varepsilon \), where \( x^* \) is a minimizer of \( F(x) \).

\textbf{Examples.} Problems of this form arise in many places in machine learning, statistics, and operations research. For instance, many regularized empirical risk minimization (ERM) problems fall into this category with convex \( f_i(\cdot) \). In such problems, we are given \( n \) training examples \( \{(a_1, \ell_1), \ldots, (a_n, \ell_n)\} \), where each \( a_i \in \mathbb{R}^d \) is the feature vector of example \( i \), and each \( \ell_i \in \mathbb{R} \) is the label of example \( i \). The following classification and regression problems are well-known examples of ERM:

- **Ridge Regression:** \( f_i(x) = \frac{1}{2} ((a_i, x) - \ell_i)^2 + \frac{\sigma}{2} \|x\|_2^2 \) and \( \Psi(x) = 0 \).
- **Lasso:** \( f_i(x) = \frac{1}{2} ((a_i, x) - \ell_i)^2 \) and \( \Psi(x) = \sigma \|x\|_1 \).
- **\( \ell_1 \)-Regularized Logistic Regression:** \( f_i(x) = \log(1 + \exp(-\ell_i (a_i, x))) \) and \( \Psi(x) = \sigma \|x\|_1 \).

Another important problem that falls into this category is the \textit{principle component analysis} (PCA) problem. Suppose we are given \( n \) data vectors \( a_1, \ldots, a_n \in \mathbb{R}^d \), denoting by \( A = \frac{1}{n} \sum_{i=1}^{n} a_i a_i^T \) the normalized covariance matrix, Garber and Hazan [6] showed that approximately finding the principle component of \( A \) is equivalent to minimizing \( f(x) = \frac{1}{2} x^T (\lambda I - A) x \) for some suitably chosen parameter \( \lambda > 0 \). Therefore, defining \( f_i(x) \overset{\text{def}}{=} \frac{1}{2} x^T (\lambda I - a_i a_i^T) x \) and \( \Psi(x) = 0 \), this problem falls into (1.1) with non-convex functions \( f_i(\cdot) \).

\textbf{Background of SVRG.} Full-gradient first-order methods consider the following proximal steps for solving (1.1):

\[
x_{t+1} \leftarrow \arg\min_{y \in \mathbb{R}^d} \left\{ \frac{1}{2\eta} \|y - x_t\|_2^2 + \langle \nabla f(x_t), y \rangle + \Psi(y) \right\}.
\]

Above, \( \eta \) is the step length, and if the proximal function \( \Psi(y) \) equals zero, the update simply reduces to \( x_{t+1} \leftarrow x_t - \eta \nabla f(x_t) \). Since computing the full gradient \( \nabla f(\cdot) \) is usually very expensive, stochastic gradient update rules have been proposed instead:

\[
x_{t+1} \leftarrow \arg\min_{y \in \mathbb{R}^d} \left\{ \frac{1}{2\eta} \|y - x_t\|_2^2 + \langle \xi_t, y \rangle + \Psi(y) \right\},
\]

where \( \xi_t \) is a random vector satisfying \( \mathbb{E}[\xi_t] = \nabla f(x_t) \) and is referred to as the \textit{stochastic gradient}.

Given the “finite average” structure \( f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x) \), a popular choice for the stochastic gradient is to set \( \xi_t = \nabla f_i(x_t) \) for some random index \( i \in [n] \) per iteration. Methods based on this choice are known as \textit{stochastic gradient descent} (SGD) methods [1, 27], and note that the computation of \( \nabla f_i(x) \) is usually \( n \) times faster than that of \( \nabla f(x) \), making SGD suitable for large-scale machine learning tasks.

More recently, the convergence speed of SGD has been further improved with the \textit{variance-reduction} technique [2, 3, 10, 14, 18, 21, 22, 26]. In all of these cited results, the authors have, in one way or another, shown that SGD can converge much faster if one makes a better choice of the stochastic gradient \( \xi_t \), so that its variance \( \mathbb{E}[\|\xi_t - \nabla f_i(x_t)\|_2^2] \) reduces as \( t \) increases.

One particular way to reduce the variance is the SVRG method described as follows [10]. Keep a snapshot \( \tilde{x} = x_t \) after every \( m \) stochastic update steps (where \( m \) is some parameter), and compute

\footnote{In fact, even if each \( f_i(x) \) is not smooth but only Lipschitz continuous, standard smoothing techniques such as Chapter 2.3 of [8] can make each \( f_i(x) \) smooth without sacrificing too much accuracy.}
the full gradient $\nabla f(\tilde{x})$ only for such snapshots. Then, set $\xi_t = \nabla f_i(x_t) - \nabla f_i(\tilde{x}) + \nabla f(\tilde{x})$ as the stochastic gradient. One can verify that, under this choice of $\xi_t$, it satisfies $\mathbb{E}[\xi_t] = \nabla f(x_t)$ and $\lim_{t \to \infty} \mathbb{E}[\|\xi_t - \nabla f(x_t)\|_2^2] = 0$.

Non-Strongly Convex Objectives. Although many variance-reduction based methods have been proposed, most of them, including SVRG, only has convergence guarantee of problem (1.1) when the objective $F(x)$ is strongly convex [3, 10, 21, 22, 26]. However, in many machine learning applications, $F(x)$ is simply not strongly convex. This is particularly true for Lasso [25] and $\ell_1$-Regularized Logistic Regression [17], two cornerstone problems extensively used for feature selections.

One way to get around this is to add a dummy regularizer $\frac{1}{2}\|x\|_2^2$ to $F(x)$, and then apply any of the above methods. However, the weight of this regularizer, $\lambda$, needs to be chosen before the algorithm starts. This adds a lot of difficulty when applying such methods to real life: (1) one needs to tune $\lambda$ by repeatedly executing the algorithm, and (2) the error of the algorithm does not converge to zero as time goes (in fact, it converges to $O(\lambda)$ so one needs to know the desired accuracy before the algorithm starts). Perhaps more importantly, adding the dummy regularizer hurts the performance of the algorithm both in theory and practice.

Another possible solution is to tackle the non-strongly convex case directly [2, 14, 18], without using any dummy regularizer. These methods are the so-called anytime algorithms: they can be interrupted at any time, and the training error tends to zero as the number of iterations increases.

While direct methods are much more convenient for practical uses, existing direct methods are much slower than indirect methods (i.e., methods via dummy regularization) at least in theory. More specifically, if the desired accuracy is $\varepsilon$ and the smoothness of each $f_i(x)$ is $L$, then the gradient complexities \footnote{Throughout this paper, we will use gradient complexity as an effective measure of an algorithm’s running time. Usually, the total running time of an algorithm is $O(d)$ multiplied with its gradient complexity, because each $\nabla f_i(x)$ can be computed in $O(d)$ time.} of the best known direct and indirect methods are respectively

$$O\left(\frac{n + L}{\varepsilon}\right) \quad \text{and} \quad O\left((n + \frac{L}{\varepsilon}) \log \frac{1}{\varepsilon}\right).$$

Therefore in theory, when $n$ is usually dominating, indirect methods are faster but less convenient, while direct methods are slower but more convenient.

In this paper, we propose SVRG++, a new method that solves the non-strongly convex case of problem (1.1) directly with gradient complexity $O(n \log \frac{1}{\varepsilon} + \frac{L}{\varepsilon})$, therefore outperforming both direct and indirect methods. On the practical side, SVRG++ is a direct, anytime method, which is convenient to use. We describe SVRG++ and the main techniques we use in Section 4.

Sum-of-Non-Convex Objectives. If $f(x)$ is $\sigma$-strongly convex while each $f_i(x)$ is non-convex but $L$-smooth, Shalev-Shwartz discovered that the SVRG method admits a gradient complexity of $O\left((n + \frac{L^2}{\sigma^2}) \log \frac{1}{\varepsilon}\right)$ for minimizing $F(x)$ [19] in the case of $\Psi(x) = 0$. A similar result has been independently re-discovered by Garber and Hazan [6] and applied to the PCA problem.

Despite of the proximal term $\Psi(x)$ which is not supported in their analysis, their running time is imperfect for two reasons.

- First, this complexity is not stable: even if we modify only one of $f_i(x)$ from convex to (a little bit) non-convex, the best known gradient complexity for SVRG immediately worsens to $O\left((n + \frac{L^2}{\sigma^2}) \log \frac{1}{\varepsilon}\right)$ from $O\left((n + \frac{L}{\sigma}) \log \frac{1}{\varepsilon}\right)$. In contrast, one should expect a more graceful decay of the performance as a function on the “magnitude” of the non-convexity, or perhaps even a threshold where the performance is totally unaffected if the magnitude is “below” this threshold.
• Second, the complexity does not take into account the asymmetry in smoothness. For instance, in PCA applications, each \( f_i(x) \) can be very non-convex and its Hessian has eigenvalues between \(-l < 0\) and \( L > 0\) where \( l \) can be significantly larger than \( L \). Can we take advantage of this asymmetry to get better running time?

In this paper, we prove that if each \( f_i(x) \) is \( L \)-upper smooth and \( l \)-lower smooth (which means the Hessian of \( f_i(x) \) has eigenvalues bounded between \([-l, L]\)), the same SVRG method admits a gradient complexity of \( \tilde{O}\left((n + \frac{L}{\sigma} + \frac{L^2}{\sigma^2}) \log \frac{1}{\varepsilon}\right) \). This resolves both our aforementioned concerns. First, if \( l = O(\sigma) \), our new result suggests that the convergence of SVRG is asymptotically the same as the convex case, meaning there is a threshold \( O(\delta) \) that SVRG allows each \( f_i(x) \) to be non-convex below this threshold for free. Second, in the \( l > L \) case, our result implies a linear dependence on the non-convexity parameter \( l \), rather than the quadratic one \( \tilde{O}\left((n + \frac{L}{\sigma}^2) \log \frac{1}{\varepsilon}\right) \) shown by prior work [6, 19]. To the best of our knowledge, this is the first time that upper and lower smoothness parameters are distinguished in order to prove convergence results for minimizing (1.1).

Our improvement on SVRG immediately leads to faster stochastic algorithms for PCA [6, 24]. Assume that \( A = \frac{1}{n} \sum_{i=1}^{n} a_i a_i^T \) is a normalized covariance matrix where each \( a_i \in \mathbb{R}^d \) has Euclidean norm at most 1. Let \( \lambda \in [0, 1] \) be the largest eigenvalue of \( A \). Garber and Hazan showed that computing the principle component of \( A \) is, up to binary search preprocessing, equivalent to the sum-of-non-convex form of problem (1.1), with upper smoothness \( L = \lambda \) and lower smoothness \( l = 1 \). Garber and Hazan further applied SVRG to minimize this objective and proved an overall running time \( \tilde{O}\left((nd + \frac{d}{\varepsilon^2}) \log \frac{1}{\varepsilon}\right) \). Our result improves this running time to \( \tilde{O}\left((nd + \frac{dL}{\varepsilon^2}) \log \frac{1}{\varepsilon}\right) \). Since \( \lambda \) may be as small as \( 1/n \), this speed up is significant in theory.\(^3\)

Our results above are non-accelerated for the sum-of-non-convex setting. One can apply Catalyst [5, 12] to further improve its running time when \( \sigma \) is very small. Not surprisingly, our performance improvement carries to the accelerated setting as well.

Finally, we also prove that our proposed improvements on SVRG (for non-strongly objectives and for sum-of-non-convex objectives) can be put together, leading to a new algorithm \( \text{SVRG}^{++}_{\text{nc}} \) that works for both non-strongly convex and sum-of-non-convex objectives. This gives faster algorithms and can be found in Appendix D.

**Roadmap.** We discuss related work in Section 2 and provide notational background in Section 3. We state our result for non-strongly convex objectives in Section 4, for sum-of-non-convex objectives in Section 5 and 6. In Section 7 we perform experiments supporting our theory. Most of the technical proofs, as well as our \( \text{SVRG}^{++}_{\text{nc}} \) method for solving both non-strongly convex and sum-of-non-convex objectives, are included in the appendix.

## 2 Other Related Work

If \( f(\cdot) \) is \( L \)-smooth but non-strongly convex, full-gradient gradient descent converges in \( O(L/\varepsilon) \) steps and has a gradient complexity \( O(nL/\varepsilon) \) (see for instance the textbook of Nesterov [16]). This was improved to \( O(n\sqrt{L/\varepsilon}) \) using Nesterov’s accelerated gradient descent [15]. If \( f(\cdot) \) is \( \sigma \)-strongly convex, the gradient complexities of the two cited methods become \( O(nL/\sigma \log(1/\varepsilon)) \) and \( O(n\sqrt{L/\sigma} \log(1/\varepsilon)) \) respectively. However, in the big-data scenario (i.e., with large \( n \)), such performances are often unsatisfactory.

\(^3\)Garber and Hazan also applied acceleration schemes on top of SVRG, and obtained a running time \( \tilde{O}\left(\frac{n^{3/4}L^{1/4}}{\sigma^{1/4}}\right) \). We can do the same thing here and improve their running time to \( \tilde{O}\left(\frac{n^{3/4}L^{1/4}}{\sigma^{1/4}}\right) \) in the accelerated setting.
In the stochastic-gradient setting, if one uses $\xi_t = \nabla f_i(x_t)$, SGD achieves a convergence rate $O(1/\varepsilon^2)$ for non-strongly convex objectives [27, 29] and $O(1/\varepsilon)$ for strongly-convex objectives [9, 20]. Both these rates are quite inefficient when we need a very accurate solution.

In order to improve SGD, in the past three years, several attempts have been made with key idea being (explicitly or implicitly) reducing the variance of the stochastic gradient. The first published method that reduces the variance and overcomes the barrier of SGD is due to SAG [18]. SAG obtains an $O(\log(1/\varepsilon))$ convergence (i.e., linear convergence) for strongly convex and smooth objectives, comparing to the $O(1/\varepsilon)$ rate of standard SGD. This $O(\log(1/\varepsilon))$ rate has also been obtained by several concurrent or subsequent works. For instance, the authors of MISO [14], Finito [3], and SAGA [2] have defined $\xi_t$ to be of a form slightly different from SAG. The authors of SVRG [10] (and its follow-up work Prox-SVRG [26]) have adopted the idea of “epochs” and defined $\xi_t = \nabla_i f(x_t) - \nabla_i f(x_t) + \nabla f(\bar{x})$ like we do in this paper. The algorithm SDCA [22] has also been discovered to be intrinsically performing some “variance reduction” procedure [2, 10, 19].

Among the variance-reduction algorithms mentioned above, only SAG, MISO, and SAGA can provide theoretical guarantees for directly solving non-strongly convex objectives (i.e., are indirect), and run only convex objectives. This class of methods require one to work with the dual of the objective, require adding a dummy regularizer. The best gradient complexity is obtained by several concurrent or subsequent works. For instance, the authors of MISO [14], Finito [3], and SAGA [2] have defined $\xi_t$ to be of a form slightly different from SAG. The authors of SVRG [10] (and its follow-up work Prox-SVRG [26]) have adopted the idea of “epochs” and defined $\xi_t = \nabla_i f(x_t) - \nabla_i f(x_t) + \nabla f(\bar{x})$ like we do in this paper. The algorithm SDCA [22] has also been discovered to be intrinsically performing some “variance reduction” procedure [2, 10, 19].

In the structured case, AccSDCA [23], along with subsequent works APCG [13] and SPDC [28], obtains a slightly better gradient complexity $O((n + \min\{L/\varepsilon, \sqrt{nL/\varepsilon}\}) \log \frac{1}{\varepsilon})$ for non-strongly convex objectives. This class of methods require one to work with the dual of the objective, require one to add dummy regularizer for non-strongly convex objectives (i.e., are indirect), and run only faster than the variance-reduction based methods when $n < \sqrt{L/\varepsilon}$.

3 Notations

Throughout this paper, we denote by $\| \cdot \|$ the Euclidean norm. We assume that each $f_i(\cdot)$ is differentiable and $\Psi(\cdot)$ is convex and lower semicontinuous.

We say that a differentiable function $f_i(\cdot)$ is $L$-smooth (or has $L$-Lipschitz continuous gradient) if:

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\| \quad \forall x, y \in \mathbb{R}^d.$$ 

The above definition has several equivalent forms, and one of them says for all $x, y \in \mathbb{R}^d$:

$$-\frac{L}{2}\|y - x\|^2 \leq f(y) - (f(x) + \langle \nabla f(x), y - x \rangle) \leq \frac{L}{2}\|y - x\|^2.$$ 

In this paper, we say $f_i(\cdot)$ is $L$-upper smooth if it satisfies

$$f(y) - (f(x) + \langle \nabla f(x), y - x \rangle) \leq \frac{L}{2}\|y - x\|^2 \quad \forall x, y \in \mathbb{R}^d,$$

and $f_i(\cdot)$ is $l$-lower smooth if it satisfies

$$f(y) - (f(x) + \langle \nabla f(x), y - x \rangle) \geq -\frac{l}{2}\|y - x\|^2 \quad \forall x, y \in \mathbb{R}^d.$$ 

Let us give a few examples: a convex differentiable function is 0-lower smooth; an $L$-smooth function is $L$-upper and $L$-lower smooth; a convex $L$-smooth function is $L$-upper and 0-lower smooth.

We say a function $f(\cdot)$ is $\sigma$-strongly convex if

$$f(y) - (f(x) + \langle \nabla f(x), y - x \rangle) \geq \frac{\sigma}{2}\|y - x\|^2 \quad \forall x, y \in \mathbb{R}^d.$$
Algorithm 1 $\text{SVRG}^+(x^\phi, m_0, S, \eta)$

1: $\tilde{x}^0 \leftarrow x^\phi$, $x^1_0 \leftarrow x^\phi$
2: for $s \leftarrow 1$ to $S$ do
3: $\tilde{\mu}_{s-1} \leftarrow \nabla f(\tilde{x}^{s-1})$
4: $m_s \leftarrow 2^s \cdot m_0$
5: for $t \leftarrow 0$ to $m_s - 1$ do
6: Pick $i$ uniformly at random in $\{1, \cdots, n\}$.
7: $\xi \leftarrow \nabla f_i(x^s_t) - \nabla f_i(\tilde{x}^{s-1}) + \tilde{\mu}_{s-1}$
8: $x^s_{t+1} = \arg \min_{y \in \mathbb{R}^d} \left\{ \frac{1}{2m_s} \| x^s_t - y \|^2 + \Psi(y) + \langle \xi, y \rangle \right\}$
9: end for
10: $\bar{x}^s \leftarrow \frac{1}{m_s} \sum_{i=1}^{m_s} x^s_t$
11: $x^{s+1}_0 \leftarrow x^s_{m_s}$
12: end for
13: return $\bar{x}^S$.

4 $\text{SVRG}^+$ for Non-Strongly Convex Objectives

In this section we consider the case of (1.1) when each $f_i(x)$ is a convex function and the objective is not necessarily strongly convex. Recall that this class of problems include Lasso and logistic regression as notable examples.

We propose our $\text{SVRG}^+$ algorithm for solving this case, see Algorithm 1. Given an initial vector $x^\phi$, our algorithm is divided into $S$ epochs. The $s$-th epoch consists of $m_s$ stochastic gradient steps (see Line 8 of $\text{SVRG}^+$), where $m_s$ doubles between every consecutive two epochs. This “doubling” feature distinguishes our method from all of the cited variance-reduction based methods.

Within each epoch, similar to SVRG, we compute the full gradient $\tilde{\mu}_{s-1} = \nabla f(\tilde{x}^{s-1})$ where $\tilde{x}^{s-1}$ is the average point of the previous epoch. We then use $\tilde{\mu}_{s-1}$ to define the variance-reduced stochastic gradient $\xi$, see Line 7 of $\text{SVRG}^+$. Unlike SVRG, our starting vector $x^0_0$ of each epoch is set to be the ending vector $x^{s-1}_{m_s}$ of the previous epoch, rather than the average of the previous epoch.$^4$

We state our main result for $\text{SVRG}^+$ as follows:

**Theorem 4.1.** If each $f_i(x)$ is convex in (1.1), then $\text{SVRG}^+(x^\phi, m_0, S, \eta)$ satisfies if $m_0$ and $S$ are positive integers and $\eta = 1/7L$, then

$$\mathbb{E}[F(\bar{x}^S) - F(x^*)] \leq O\left( \frac{F(x^\phi) - F(x^*)}{2^S} + \frac{L\|x^\phi - x^*\|^2}{2^S m_0} \right).$$

(4.1)

In addition, $\text{SVRG}^+$ has a gradient complexity of $O(S \cdot n + 2^S \cdot m_0)$.

As a result, given an initial vector $x^\phi$ satisfying $\|x^\phi - x^*\|^2 \leq \Theta$ and $F(x^\phi) - F(x^*) \leq \Delta$ for parameters $\Theta, \Delta \in \mathbb{R}_+$, by setting $S = \log_2(\Delta/\varepsilon)$, $m_0 = L\Theta/\Delta$, and $\eta = 1/7L$, we obtain an $O(\varepsilon)$ approximate minimizer of $F(\cdot)$ with a total gradient complexity $O(n \log \left( \frac{\Delta}{\varepsilon} \right) + \frac{L\Theta}{\varepsilon})$.

Our proof of Theorem 4.1 is included in Appendix A.$^4$

$^4$The theoretical convergence of SVRG relies on its Option II, that is to set the beginning vector of each epoch to be the average (or a random) vector of the previous epoch. However, the authors of SVRG conduct their experiment using the last vector rather than the average because it is more “natural”. This present paper partially shows that this natural choice also has competitive performance, and therefore confirms the empirical finding of SVRG. (Similar result can also be obtained for the strongly convex case, which we exclude for simplicity.)
High-Level Techniques. Our proof is based on a new way to telescope regret inequalities that is specially designed for a growing-epoch method. Unlike the analysis of SVRG, we telescope not only across iterations, see (A.2), but also across epochs, see (A.3). In contrast, the original SVRG has to rely on the strong convexity of \( f(\cdot) \) in order to combine different epochs — this is why SVRG cannot directly solve non-strongly convex objectives. Our technique is also significantly different from known direct methods such as SAG or SAGA: these methods can be analogously viewed as SVRG with epoch length \( n \), because each stochastic gradient in SAG or SAGA is updated once every \( n \) iterations in average, so their epoch length is intrinsically \( O(n) \) and cannot be doubled. Finally, it is the telescoping across all epochs and all iterations that requires the starting vector of an epoch to be the last one from the previous epoch (which is different from SVRG). We shall demonstrate in our experiment section that these modifications on top of SVRG are also useful in practice.

4.1 Additional Improvements

Inspired by SVRG++, we also introduce SVRG_Auto_Epoch, a variant of SVRG++ where epoch length is automatically determined instead of doubled every epoch. Auto epoch is an attractive feature in practice because it enables the algorithm to perform well for different types of objectives.

The criterion we use to determine the termination of an epoch \( s \) in SVRG_Auto_Epoch is based on the quality of the snapshot full gradient \( \nabla f(\bar{x}^{s-1}) \). Intuitively, if epoch length is too long, an algorithm may move too far from the snapshot point, meaning that the gradient estimator \( \xi \) may have a large variance. Following this intuition, for every iteration \( t \), we record \( \text{diff}_t = \| \nabla f_t(x_t) - \nabla f_t(\bar{x}^{s-1}) \|_2^2 \) because \( \mathbb{E}[\text{diff}_t] \) is a very tight upper bound on the variance of the gradient estimator (see the proof of Lemma A.2). Under this notion, we decide the epoch termination of SVRG_Auto_Epoch as follows. Each epoch has a minimum length of \( n/4 \). From iteration \( t = n/4 \) onwards, we keep track of the average \( \text{diff}_t \) in the last \( m \) iterations, i.e., \( \sum_{j=t-n/4+1}^t \text{diff}_j \).

If this quantity is greater than half of the average \( \text{diff}_j \) recorded from the previous epoch, we terminate the current epoch and start a new one. SVRG_Auto_Epoch shows good performance in our experiments, and we leave it as an open question to prove a complexity result for this method.

In addition to auto epoch, SVRG++ can also be combined with other enhancements proposed for SVRG. For example, [7] saves the time to compute full gradients at snapshot points by making them less accurate in the first a few epochs. [11] uses mini-batch gradients per iteration to further decrease the variance. These ideas are orthogonal to our proposed techniques and therefore can be applied to further improve the performance of SVRG++.

5 SVRG for Sum-of-Non-Convex Objectives I: Small Lower Smoothness

In this section we consider problem (1.1) when each \( f_i(x) \) is non-convex, \( L \)-upper smooth, and \( l \)-lower smooth for some \( 0 \leq l \leq L \). We assume that \( f(\cdot) \) is \( \sigma \)-strongly convex. For this class of objectives, the best known gradient complexity for stochastic gradient methods is \( O((n + \frac{L}{\sigma}) \log \frac{1}{\varepsilon}) \) due to SVRG [19].

This gradient complexity is essentially a factor \( L/\sigma \) greater than that for the convex case, that is \( O((n + \frac{L}{\sigma}) \log \frac{1}{\varepsilon}) \). Following the intuition discussed in the introduction, we improve it to \( O((n + \frac{L}{\sigma} + \frac{L^2}{\sigma^2}) \log \frac{1}{\varepsilon}) \), a quantity that is asymptotically the same as the convex setting when \( l \leq O(\sigma) \), and linearly degrades as \( l \) increases.

\[^{5}\text{We always set the first epoch to be of length } n/4 \text{ and the second to be of length } n/2.\]
Algorithm 2 SVRG($x^\phi$, m, S, $\eta$) [10]

1: \(\tilde{x}^0 \leftarrow x^\phi, x_0^1 \leftarrow x^\phi\)
2: for \(s \leftarrow 1\) to \(S\) do
3: \(\bar{\mu}_{s-1} \leftarrow \nabla f(\tilde{x}^{s-1})\)
4: for \(t \leftarrow 0\) to \(m-1\) do
5: \(\xi \leftarrow \nabla f_i(x^s_t) - \nabla f_i(\tilde{x}^{s-1}) + \bar{\mu}_{s-1}\)
6: \(x^s_{t+1} \leftarrow \arg \min_{y \in \mathbb{R}^d} \{\frac{1}{2\eta} \|x^s_t - y\|^2 + \Psi(y) + \langle \xi, y \rangle\}\)
7: end for
8: \(\tilde{x}^s \leftarrow \frac{1}{m} \sum_{t=1}^{m} x^s_t\)
9: \(x_0^{s+1} \leftarrow \tilde{x}^s\)
10: end for
11: return \(\tilde{x}^S\).

Recall that the original SVRG (Option II) works as follows (see Algorithm 2 for completeness). Given an initial vector \(x^\phi\), SVRG is divided into \(S\) epochs, each of length \(m\) for the same \(m\) across epochs. Within each epoch, SVRG computes the full gradient \(\bar{\mu}_{s-1} = \nabla f(\tilde{x}^{s-1})\) where \(\tilde{x}^{s-1}\) is the average point of the previous epoch. Then, SVRG uses \(\bar{\mu}_{s-1}\) to define the variance-reduced version of the stochastic gradient \(\xi\), see Line 6 of Algorithm 2. The starting vector \(x^s_0\) of each epoch is set to be the average vector of the previous epoch.\(^6\)

We state our main result for SVRG in this section as follows:

**Theorem 5.1.** If each \(f_i(x)\) is \(L\)-upper and \(l\)-lower smooth in (1.1) for \(l \in [0, L]\), \(f(x)\) is \(\sigma\)-strongly convex, \(\eta = \min\{\frac{1}{2L}, \frac{\sigma}{6\eta L}\}\) and \(m \geq \frac{10}{\eta l} = \Omega(\max\{\frac{L}{\sigma}, \frac{L}{\sigma^2}\})\), then SVRG($x^\phi$, \(m, S, \eta\)) satisfies

\[
\mathbb{E}[F(\tilde{x}^s) - F(x^*)] \leq \frac{3}{4}(F(\tilde{x}^{s-1}) - F(x^*)) .
\]

Therefore, by setting \(S = \log_{4/3} \left(\frac{F(x^\phi) - F(x^*)}{\varepsilon}\right)\), in a total gradient complexity of

\[
O\left((n + \frac{L}{\sigma} \max\{1, \frac{l}{\sigma}\}) \log \frac{F(x^\phi) - F(x^*)}{\varepsilon}\right),
\]
we obtain an output \(\tilde{x}^S\) satisfying \(\mathbb{E}[F(\tilde{x}^s) - F(x^*)] \leq \varepsilon\).

Our technique for proving this theorem depends on the following new upper bound on the variance. Denoting by \(\xi_t^s\) the stochastic gradient \(\xi\) at epoch \(s\) and iteration \(t\), we have

**Lemma 5.2.**

\[
\mathbb{E}_t \left[\|\xi_t^s - \nabla f(x^s_t)\|^2\right]
\leq 4(L + l) \cdot (F(x^s_t) - F(x^*) + F(\tilde{x}^{s-1}) - F(x^*))
+ (8l^2 + 4L) \left(\|x^s_t - x^*\|^2 + \|\tilde{x}^{s-1} - x^*\|^2\right).
\]

This is different from Section 4.1 of [19], where the author only provided a weaker upper bound \(O(L^2) \cdot (\|x^s_t - x^*\|^2 + \|\tilde{x}^{s-1} - x^*\|^2)\). In the event that \(l\) is very small, our new upper bound reduces to the variance upper bound in the convex setting, see for instance Eq. (8) of [10]. The full proof of Theorem 5.1 is included in Appendix B.

\(^6\) This choice of the starting vector is different from SVRG++, but was the originally choice made by SVRG. Similar result can also be obtained using the choice from SVRG++.
6 SVRG for Sum-of-Non-Convex Objectives II: Large Lower Smoothness

In this section we consider problem (1.1) when each \( f_i(x) \) is a non-convex, \( L \)-upper smooth, and \( l \)-lower smooth function for some \( l \geq L \). We assume \( f(\cdot) \) is \( \sigma \)-strongly convex. For this class of objectives, the best known gradient complexity for stochastic gradient methods is \( O((n + \frac{l^2}{\sigma^2}) \log \frac{1}{\varepsilon}) \) due to SVRG [19].

This known gradient complexity is essentially a factor \( l^2/L^2 \geq 1 \) worse than that of the symmetric case (i.e., the case when \( l = L \)). In this section, we improve this factor to \( l/L \) which is quadratically faster than \( l^2/L^2 \). As we have explained in the introduction, this result improves the convergence for the best known stochastic algorithm for PCA.

We state our main result for SVRG in this section as follows:

**Theorem 6.1.** If each \( f_i(x) \) is \( L \)-upper and \( l \)-lower smooth in (1.1) for \( l \geq L \), \( f(x) \) is \( \sigma \)-strongly convex, \( \eta = \frac{\sigma}{25Ll} \) and \( m \geq \frac{4}{\sigma \eta} = \Omega(\frac{L}{l^2}) \), then SVRG \((x^\phi, m, S, \eta)\) satisfies

\[
\mathbb{E}[F(\bar{x}^S) - F(x^*)] \leq \frac{3}{4} \left(F(\bar{x}^{S-1}) - F(x^*)\right).
\]

Therefore, by setting \( S = \log_{4/3} \left( \frac{F(x^\phi) - F(x^*)}{\varepsilon} \right) \), in a total gradient complexity of

\[
O\left(\left(n + \frac{Ll}{\sigma^2}\right) \log \frac{F(x^\phi) - F(x^*)}{\varepsilon}\right),
\]

we obtain an output \( \bar{x}^S \) satisfying \( \mathbb{E}[F(\bar{x}^S) - F(x^*)] \leq \varepsilon \).

Although Theorem 6.1 (for the large \( l \) setting) has the same form as Theorem 5.1 (for the small \( l \) setting), its proof is quite different. In order to provide a variance bound without paying the \( l^2 \) factor as in Lemma 5.2, we negate the objective for analysis purpose only. This is reasonable because \(-f_i(\cdot)\) becomes \( l \) upper smooth but only \( L \) lower smooth for \( L \leq l \). By applying the smoothness lemmas for minimizing \(-f_i(\cdot)\) (and thus maximizing \( f_i(x) \)), we obtain a better variance upper bound without paying the factor \( l^2 \). The details of this proof is included in Appendix C.

7 Experiments on Empirical Risk Minimization

We confirm our theoretical findings using four real-life datasets: (1) the Adult dataset (32,561 examples and 123 features), (2) the Covtype dataset (581,012 examples and 54 features), (3) the Ijcnn1 dataset (49990 examples and 22 features), and (4) the 2nd class of the MNIST dataset (60,000 examples and 780 features) [4]. In order to make easy comparisons between different datasets, we scale each data vector down by the average Euclidean norm of the whole data set. This step is for comparison only and not necessary in practice.

We perform 3 classification tasks: Lasso, ridge regression, and \( \ell_1 \)-regularized logistic regression. As described in the introduction, Lasso and logistic regression do not admit strongly convex objectives, while the ridge objective is strongly convex. We consider four different values \( \sigma \in \{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\} \), where \( \sigma \) is either the weight in regularizer \( \frac{\sigma}{2} \|x\|_2^2 \) for ridge, or that in regularizer \( \sigma \|x\|_1^2 \) for Lasso and logistic regression.

We have implemented the following algorithms:

- SVRG++ with initial epoch length \( m_0 = n/4 \).
For each algorithm above except SDCA, we have tuned the best step length carefully from the set \{a \times 10^{-k} : a \in \{1, 2, \ldots, 9\}, k \in \mathbb{Z}\} for each plot. For SDCA on Lasso and logistic regression, we

- **SVRG_Auto_Epoch** as we described in Section 4.1.

- **SVRG** [10, 26] with (their suggested) epoch length \(m = 2n\).

Recall that, in theory, SVRG is not designed for non-strongly convex objectives and \(F(\cdot)\) needs to be added by a dummy regularizer for Lasso and logistic regression. However, in our experiments, we observed that this dummy regularizer is not necessary, so have neglected the regularized version of SVRG for a clean comparison.

- **SAGA** [2].

- **SDCA** [21, 22] with Option I (steepest descent). Since SDCA works only with strongly convex objectives, a dummy regularizer has to be introduced for Lasso and Logistic regression.

For each algorithm above except SDCA, we have tuned the best step length carefully from the set \{a \times 10^{-k} : a \in \{1, 2, \ldots, 9\}, k \in \mathbb{Z}\} for each plot. For SDCA on Lasso and logistic regression, we
Figure 2: Performance analysis on sum-of-non-convex objectives. Note that the curves for \( \delta = 0.001, 0.01, 0.02 \) have overlapped in (a).

We have enumerated the weight of its dummy regularizer from the set \( \{10^{-k}, 2 \times 10^{-k}, 5 \times 10^{-k} : k \in \mathbb{Z}\} \), and picked the best one for each plot.

**Performance Comparison.** We have picked a representative regularizer weight \( \sigma \) for each of the eight analysis tasks (lasso or logistic regression on one of the four datasets), and presented the performance plots in Figure 1. For the results on other values of \( \sigma \) as well as those for ridge regression, see Figure 3, 4, 5, and 6 in the appendix.

In the legend of each plot, we use SDCA\((r = r_0)\) to denote that \( r_0 \) is the weight of the best-tuned dummy regularizer. For every other algorithm, we use Alg\((\eta)\) to denote that \( \eta \) is the best-tuned step length for algorithm Alg.

We make the following observations from this experiment:

- **SVRG++ and SVRG Auto_Epoch** consistently outperform SVRG in all the plots, indicating that they do improve over SVRG in non-strongly convex settings.
- **SVRG++ and SVRG Auto_Epoch** outperform SAGA in most cases, and are at least comparable to SAGA in the rest cases. This is not surprising because SAGA is also a direct algorithm for non-strongly convex objectives.
- **SVRG++ and SVRG Auto_Epoch** significantly outperform indirect methods via dummy regularization (i.e., SDCA) in the non-strongly convex settings. But for ridge regression (which is strongly convex), the performance of SDCA is comparable to other methods (see the figures in the appendix).

## 8 Experiments for Sum-of-Non-Convex Objectives

To verify our theoretical findings in Section 5 and 6, we run SVRG on a sum-of-non-convex objective built from synthetically generated data.

We generate \( n = 500 \) random vectors \( a_1, \ldots, a_{500} \in \mathbb{R}^d \) with Euclidean norm 1 each and \( d = 200 \). Define the covariance matrix \( A \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} a_i a_i^T \), and we consider the minimization problem

\[
\min_{x \in \mathbb{R}^d} \left\{ f(x) \overset{\text{def}}{=} \frac{x^T A x}{2} + b x \right\}
\]
for some randomly generated vector $b$.

The matrix $A$ we generated has minimum eigenvalue equal to $7.02 \times 10^{-4}$, and thus $f(x)$ is strongly convex with parameter $7.02 \times 10^{-4}$. Next, we decompose $f(x)$ into an average of $f_i(x)$, each being non-convex with upper and lower smoothness parameters that we can control.

More specifically, given $n$ diagonal matrices $D_1, \ldots, D_n$ satisfying $D_1 + \cdots + D_n = 0$, by setting $f_i(x) \overset{\text{def}}{=} \frac{x^T(a_i^T a_i + D_i)x}{2} + bx$, we have $f(x) = \frac{1}{n} \sum_i f_i(x)$. Under this construction, each $f_i$ is non-convex if $D_i$ has negative entries in the diagonals. We now consider two different ways to build $D_1, \ldots, D_n$.

Our first experiment is parameterized by a given value $\delta \in [0, 1]$. For each $j \in [d]$, we randomly select half of the indices $i \in [n]$ and assign its $j$-th diagonal $(D_i)_{jj}$ to be $\delta$; for the other half of the indices $i$ we assign $(D_i)_{jj}$ to be $-\delta$. In this way, we satisfy $D_1 + \cdots + D_n = 0$ and for each $i \in [n]$, we have $-\delta I \leq \nabla^2 f_i(x) \leq (1 + \delta)I$. In other words, each function $f_i(x)$ is $L \approx 1$ upper smooth and exactly $l = \delta$ lower smooth. This corresponds to the $l \leq L$ regime studied by Section 5.

Our second experiment is parameterized by a given value $k \in [1, n]$. For each $j \in [d]$, consider the $j$-th diagonal entry of all the matrices, $(D_1)_{jj}, (D_2)_{jj}, \ldots, (D_n)_{jj}$. We randomly select one of these entries and set it to be $-k$, and the rest $n - 1$ of them to be $\frac{k}{n-1}$. Under this definition, we have $D_1 + \cdots + D_n = 0$ and for each $i \in [n]$, we have $-kI \leq \nabla^2 f_i(x) \leq (1 + k/(n-1))I$. In other words, each function $f_i(x)$ is approximately $L \approx 1$ upper smooth and $l = k$ lower smooth. This corresponds to the $l \geq 1$ regime studied by Section 6.

We run SVRG (with the best tuned step length $\eta$) for both these experiments, and present the convergence performance in Figure 2. We make the following observations from our results:

- In Figure 2(a), we observe that the performance SVRG is approximately linearly dependent on $lL = O(\delta)$ for large $\delta$, as compared to $L^2 = O(1)$ from prior work. More importantly, SVRG is robust against small non-convexity (i.e., small lower smoothness parameter $l$). Indeed, for $l = \delta \leq 0.02$, the convergence of SVRG is as fast as the convex case (i.e., $\delta = 0$ case). This confirms our theoretical finding in Section 5 that shows there is a threshold around $O(\sigma)$ where the performance of SVRG only starts to degrade when $l$ exceeds this threshold.

- In Figure 2(b), we see that the performance of SVRG is approximately linearly proportional to $lL = O(k)$, as compared to $l^2 = O(k^2)$ from prior work. This confirms our finding in Section 6.

References

[1] Léon Bottou. Stochastic gradient descent. http://leon.bottou.org/projects/sgd, 2007.

[2] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives. In Advances in Neural Information Processing Systems, NIPS 2014, 2014.

[3] Aaron J. Defazio, Tibério S. Caetano, and Justin Domke. Finito: A Faster, Permutable Incremental Gradient Method for Big Data Problems. In Proceedings of the 31st International Conference on Machine Learning, ICML 2014, 2014.

[4] Rong-En Fan and Chih-Jen Lin. LIBSVM Data: Classification, Regression and Multi-label, 2011. Accessed: 2015-06.
[5] Roy Frostig, Rong Ge, Sham M. Kakade, and Aaron Sidford. Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In ICML, volume 37, pages 1–28, 2015.

[6] Dan Garber and Elad Hazan. Fast and simple PCA via convex optimization. ArXiv e-prints, September 2015.

[7] Reza Harikandeh, Mohamed Osama Ahmed, Alim Virani, Mark Schmidt, Jakub Konečný, and Scott Sallinen. Stopwasting my gradients: Practical svrg. In C. Cortes, N.D. Lawrence, D.D. Lee, M. Sugiyama, and R. Garnett, editors, Advances in Neural Information Processing Systems 28, pages 2242–2250. Curran Associates, Inc., 2015.

[8] Elad Hazan. DRAFT: Introduction to online convex optimization. Foundations and Trends in Machine Learning, XX(XX):1–168, 2015.

[9] Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. Machine Learning, 69(2-3):169–192, August 2007.

[10] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In Advances in Neural Information Processing Systems, NIPS 2013, pages 315–323, 2013.

[11] Jakub Konečný, Jie Liu, Peter Richtárik, and Martin Takáč. ms2gd: Mini-batch semi-stochastic gradient descent in the proximal setting. CoRR, abs/1410.4744, 2014.

[12] Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. A Universal Catalyst for First-Order Optimization. In NIPS, 2015.

[13] Qihang Lin, Zhaosong Lu, and Lin Xiao. An Accelerated Proximal Coordinate Gradient Method and its Application to Regularized Empirical Risk Minimization. In Advances in Neural Information Processing Systems, NIPS 2014, pages 3059–3067, 2014.

[14] Julien Mairal. Incremental Majorization-Minimization Optimization with Application to Large-Scale Machine Learning. SIAM Journal on Optimization, 25(2):829–855, April 2015. Preliminary version appeared in ICML 2013.

[15] Yurii Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. In Doklady AN SSSR (translated as Soviet Mathematics Doklady), volume 269, pages 543–547, 1983.

[16] Yurii Nesterov. Introductory Lectures on Convex Programming Volume: A Basic course, volume I. Kluwer Academic Publishers, 2004.

[17] Andrew Y. Ng. Feature selection, L1 vs. L2 regularization, and rotational invariance. In Proceedings of the 21st International Conference on Machine Learning, ICML 2004, page 78. ACM, 2004.

[18] Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. arXiv preprint arXiv:1309.2388, pages 1–45, 2013. Preliminary version appeared in NIPS 2012.

[19] Shai Shalev-Shwartz. SDCA without Duality. arXiv preprint arXiv:1502.06177, pages 1–7, 2015.
[20] Shai Shalev-Shwartz and Yoram Singer. Logarithmic regret algorithms for strongly convex repeated games. Technical report, The Hebrew University, 2007.

[21] Shai Shalev-Shwartz and Tong Zhang. Proximal Stochastic Dual Coordinate Ascent. arXiv preprint arXiv:1211.2717, pages 1–18, 2012.

[22] Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. Journal of Machine Learning Research, 14:567–599, 2013.

[23] Shai Shalev-Shwartz and Tong Zhang. Accelerated Proximal Stochastic Dual Coordinate Ascent for Regularized Loss Minimization. In Proceedings of the 31st International Conference on Machine Learning, ICML 2014, pages 64–72, 2014.

[24] Ohad Shamir. A Stochastic PCA and SVD Algorithm with an Exponential Convergence Rate. In Proceedings of The 32nd International Conference on Machine Learning, ICML 2015, pages 144—153, 2015.

[25] Robert Tibshirani. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological), pages 267–288, 1996.

[26] Lin Xiao and Tong Zhang. A Proximal Stochastic Gradient Method with Progressive Variance Reduction. SIAM Journal on Optimization, 24(4):2057—2075, 2014.

[27] Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In Proceedings of the 21st International Conference on Machine Learning, ICML 2004, 2004.

[28] Yuchen Zhang and Lin Xiao. Stochastic Primal-Dual Coordinate Method for Regularized Empirical Risk Minimization. In Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, 2015.

[29] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In Proceedings of the 20th International Conference on Machine Learning, ICML 2003, pages 928–936, 2003.
A Convergence Analysis for Section 4

For each outer iteration $s \in [S]$ and inner iteration $t \in \{0, 1, \ldots, m_s-1\}$ of SVRG++, we denote by $i_t^s$ the selected random index $i \in [n]$ and $\xi_t^s$ the stochastic gradient $\xi^s = \nabla f_{i_t^s}(x_t^s) - \nabla f_{i_t^s}(\bar{x}^{s-1}) + \bar{\mu}_{s-1}$. Then, using the convexity and smoothness of our objective, as well as the definition of our stochastic gradient step, we obtain the following lemma:

**Lemma A.1.** For every $u \in \mathbb{R}^d$ and $t \in \{0, 1, \ldots, m_s-1\}$, fixing $x_t^s$ and letting $i = i_t^s$ be the random variable, we have

$$
\mathbb{E}_{i_t^s} \left[ F(x_{t+1}^s) - F(u) \right] \leq \mathbb{E}_{i_t^s} \left[ \frac{\eta}{2(1 - \eta L)} \|\xi_t^s - \nabla f(x_t^s)\|^2 + \frac{\|x_t^s - u\|^2 - \|x_{t+1}^s - u\|^2}{2\eta} \right].
$$

**Proof.** We first upper bound the left hand side:

$$
\mathbb{E}_{i_t^s} \left[ F(x_{t+1}^s) - F(u) \right] = \mathbb{E}_{i_t^s} \left[ f(x_{t+1}^s) - f(u) + \Psi(x_{t+1}^s) - \Psi(u) \right]
\leq \mathbb{E}_{i_t^s} \left[ \|\nabla f(x_t^s), x_t^{s+1} - x_t^s\| - \frac{L}{2} \|x_t^s - x_{t+1}^s\|^2 - f(u) + \Psi(x_{t+1}^s) - \Psi(u) \right]
\leq \mathbb{E}_{i_t^s} \left[ \|\nabla f(x_t^s), x_t^{s+1} - x_t^s\| + \|\nabla f(x_t^s), x_t^{s+1} - x_t^s\| + \frac{L}{2} \|x_t^s - x_{t+1}^s\|^2 + \Psi(x_{t+1}^s) - \Psi(u) \right]
\leq \mathbb{E}_{i_t^s} \left[ \langle \xi_t^s, x_t^s - u \rangle + \|\nabla f(x_t^s), x_t^{s+1} - x_t^s\| + \frac{L}{2} \|x_t^s - x_{t+1}^s\|^2 + \Psi(x_{t+1}^s) - \Psi(u) \right].
$$

(A.1)

Above, inequalities 1 and 2 are respectively due to the smoothness and convexity of $f(\cdot)$, and 3 is because $\mathbb{E}_{i_t^s} [\xi_t^s] = \nabla f(x_t^s)$. Next, using the definition of $x_{t+1}^s$ we have

$$
\langle \xi_t^s, x_t^s - u \rangle + \Psi(x_{t+1}^s) - \Psi(u) = \langle \xi_t^s, x_t^s - x_{t+1}^s \rangle + \langle \xi_t^s, x_{t+1}^s - u \rangle + \Psi(x_{t+1}^s) - \Psi(u)
\leq \langle \xi_t^s, x_t^s - x_{t+1}^s \rangle + \frac{1}{\eta} \langle x_{t+1}^s - x_t^s, x_{t+1}^s - u \rangle
\leq \langle \xi_t^s, x_t^s - x_{t+1}^s \rangle + \frac{\|x_t^s - u\|^2}{2\eta} - \frac{\|x_{t+1}^s - u\|^2}{2\eta} - \frac{\|x_{t+1}^s - x_t^s\|^2}{2\eta}.
$$

Above, inequality 4 holds for the following reason. Recall that the minimality of $x_{t+1}^s = \arg \min_{y \in \mathbb{R}^d} \{ \frac{1}{2\eta} \|y - x_t^s\|^2 + \Psi(y) \}$ implies the existence of some subgradient $g \in \partial \Psi(x_{t+1}^s)$ which satisfies $\frac{1}{\eta} (x_{t+1}^s - x_t^s) + \xi_t^s + g = 0$. Combining this with $\Psi(u) - \Psi(x_{t+1}^s) \geq \langle g, u - x_{t+1}^s \rangle$, which is due to the convexity of $\Psi(\cdot)$, we immediately have $\Psi(u) - \Psi(x_{t+1}^s) + \langle \frac{1}{\eta} (x_{t+1}^s - x_t^s) + \xi_t^s, u - x_{t+1}^s \rangle \geq \langle \frac{1}{\eta} (x_{t+1}^s - x_t^s) + \xi_t^s + g, u - x_{t+1}^s \rangle = 0$. This gives inequality 4. In addition, 5 can be verified by expanding the Euclidean norms.

Combining the above two inequalities, we have

$$
\mathbb{E}_{i_t^s} \left[ F(x_{t+1}^s) - F(u) \right]
\leq \mathbb{E}_{i_t^s} \left[ \langle \xi_t^s, x_t^s - x_{t+1}^s \rangle - \frac{1 - \eta L}{2\eta} \|x_t^s - x_{t+1}^s\|^2 + \frac{\|x_t^s - u\|^2 - \|x_{t+1}^s - u\|^2}{2\eta} \right]
\leq \mathbb{E}_{i_t^s} \left[ \frac{\eta}{2(1 - \eta L)} \|\xi_t^s - \nabla f(x_t^s)\|^2 + \frac{\|x_t^s - u\|^2 - \|x_{t+1}^s - u\|^2}{2\eta} \right].
$$

Above, 6 is by the Cauchy-Schwarz inequality. □
The next lemma is classical and analogous to most of the variance reduction literatures (cf. [2, 10, 26]). We include it here for the sake of completeness.

**Lemma A.2.** $E_{i_t} [\|\xi^s_t - \nabla f(x^s_t)\|^2] \leq 4L \cdot \left( F(x^s_t) - F(x^*) + F(x^s_t) - F(x^*) \right)$

*Proof.* The proof of this lemma is classical and is analogous to most of the variance reduction literatures (cf. [2, 10, 26]). Indeed,

$$
E_{i_t} [\|\xi^s_t - \nabla f(x^s_t)\|^2] = E_{i_t} [\|\nabla f_{i_t}(x^s_t) - \nabla f_{i_t}(\tilde{x}^{s-1})\|^2] \\
= E_{i_t} [\|\nabla f_{i_t}(x^s_t) - \nabla f_{i_t}(x^*)\|^2] \\
\leq 2 \cdot E_{i_t} [\|\nabla f_{i_t}(x^s_t)\|^2 + \|\nabla f_{i_t}(\tilde{x}^{s-1}) - \nabla f_{i_t}(x^*)\|^2].
$$

Above, $\xi^s_t$ is because for any random vector $\zeta \in \mathbb{R}^d$, it holds that $E(\zeta - E\zeta|^2 = E(\zeta|^2 - E\zeta|^2)$, and $\eta$ is because for any two vectors $a, b \in \mathbb{R}^d$, it holds that $\|a - b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$.

Next, the classical smoothness assumption on a function $f_i$ yields (see for instance Theorem 2.1.5 in the textbook [16]) $\|\nabla f_i(x) - \nabla f_i(x^*)\|^2 \leq 2L \left( f_i(x) - f_i(x^*) - \langle \nabla f_i(x^*), x - x^* \rangle \right)$. Plugging this into the above inequality, we have

$$
E_{i_t} [\|\xi^s_t - \nabla f(x^s_t)\|^2] \\
\leq 4L \cdot E_{i_t} [f_{i_t}(x^s_t) - f_{i_t}(x^*) - \langle \nabla f_{i_t}(x^*) , x^s_t - x^* \rangle + f_{i_t}(\tilde{x}^{s-1}) - f_{i_t}(x^*) - \langle \nabla f_{i_t}(x^*) , \tilde{x}^{s-1} - x^* \rangle] \\
= 4L \cdot \left( f(x^s_t) - f(x^*) - \langle \nabla f(x^*) , x^s_t - x^* \rangle + f(\tilde{x}^{s-1}) - f(x^*) - \langle \nabla f(x^*) , \tilde{x}^{s-1} - x^* \rangle \right) \\
= 4L \cdot \left( f(x^s_t) - f(x^*) + \langle g^s , x^s_t - x^* \rangle + f(\tilde{x}^{s-1}) - f(x^*) + \langle g^s , \tilde{x}^{s-1} - x^* \rangle \right) \\
\leq 4L \cdot \left( f(x^s_t) - f(x^*) + \langle g^s , x^s_t - x^* \rangle + f(\tilde{x}^{s-1}) - f(x^*) + \langle \tilde{g}^s , \tilde{x}^{s-1} - x^* \rangle \right) \\
= 4L \cdot \left( F(x^s_t) - F(x^*) + F(\tilde{x}^{s-1}) - F(x^*) \right).
$$

Above, $g^s \in \partial \Psi(x^*)$ is the subgradient of $\Psi$ at $x^*$ that satisfies $\nabla f(x^*) + g^s = 0$. \hfill $\square$

We are now ready to prove the main theorem for the convergence of SVRG++:

**Proof of Theorem 4.1.** Combining Lemma A.1 with $u = x^*$ and Lemma A.2, we have

$$
E_{i_t} \left[ F(x_{t+1}^s) - F(x^*) \right] \leq \frac{2\eta L}{(1 - \eta L)} \left( \frac{m_{s-1}}{m_s} \cdot F(x^s_t) - F(x^*) + F(\tilde{x}^{s-1}) - F(x^*) \right) + \frac{\|x^s_t - x^*\|^2 - E_{i_t} [x_{t+1}^s - x^*]^2}{2\eta m_s}. 
$$

Choosing $\eta = 1/7L$ in the above inequality, summing it up over $t = 0, 1, \ldots, m_s - 1$, and dividing both sides by $m_s$, we arrive at

$$
E \left[ \sum_{t=0}^{m_s-1} \frac{F(x^s_{t+1})}{m_s} - F(x^*) \right] \leq E \left[ \frac{1}{3} \sum_{t=0}^{m_s-1} \frac{F(x^s_t)}{m_s} - F(x^*) + F(\tilde{x}^{s-1}) - F(x^*) \right] + \frac{\|x^s_0 - x^*\|^2 - \|x^* - x^s_{m_s}\|^2}{2\eta m_s}. 
$$

(A.2)

After rearranging, this yields

$$
2E \left[ \sum_{t=0}^{m_s-1} \frac{F(x^s_{t+1})}{m_s} - F(x^*) \right] \leq E \left[ \frac{(F(x^s_0) - F(x^*)) - (F(x^s_{m_s}) - F(x^*))}{m_s} + F(\tilde{x}^{s-1}) - F(x^*) \right] \\
+ \frac{\|x^s_0 - x^*\|^2 - \|x^* - x^s_{m_s}\|^2}{2\eta m_s}.
$$

16
Next, using the fact that $F(\tilde{x}^s) \leq \sum_{t=0}^{m_s-1} \frac{F(x_{s+1}^t)}{m_s}$ due to the convexity of $F$ and the definition $	ilde{x}^s = \sum_{t=0}^{m_s-1} \frac{x_{s+1}^t}{m_s}$, as well as the choice $x_{m_s}^s = x_{0}^{s+1}$, we rewrite the above inequality as

$$2\mathbb{E}[F(\tilde{x}^s) - F(x^s)] \leq \mathbb{E} \left[ \frac{F(x_0^s) - F(x^s)}{m_s} \right] + \left[ \frac{\|x_0^s - x^s\|^2 - \|x^s - x_{m_s}^{s+1}\|^2}{2\eta/3 \cdot m_s} \right]. \quad (A.3)$$

After rearranging and using the fact $m_s = 2m_{s-1}$, we conclude that

$$2\mathbb{E}[F(\tilde{x}^s) - F(x^s)] + \frac{\|x^s - x_0^{s+1}\|^2}{4\eta/3 \cdot m_s} + \frac{F(x_0^{s+1}) - F(x^s)}{2m_s} \leq \mathbb{E} \left[ F(\tilde{x}^{s-1}) - F(x^s) \right] + \frac{\|x_0^s - x^s\|^2}{4\eta/3 \cdot m_{s-1}} + \frac{F(x_0^s) - F(x^s)}{2m_{s-1}}.$$

In sum, after telescoping for $s = 1, 2, \ldots, S$, we have\footnote{We can perform telescoping because we set our starting vector $x_0^{s+1}$ of each epoch to equal the ending vector $x_{m_s}^s$ of the previous epoch. This is different from \textsc{svrg}, which chooses the average of the previous epoch as the starting vector. This difference is also beneficial in practice (see Section 7).}

$$\mathbb{E}[F(\tilde{x}^S) - F(x^s)] \leq 2^{-S} \cdot \left( \frac{F(x_0^S) - F(x^s)}{4\eta/3 \cdot m_0} + \frac{F(x_0^1) - F(x^s)}{2m_0} \right) \leq \frac{F(x_0^S) - F(x^s)}{2^{S-1}} + \frac{\|x_0^S - x^s\|^2}{2^S \cdot 4\eta m_0 / 3}.$$

This finishes the proof of (4.1) due to the choice of $\eta = 1/7L$. Finally, \textsc{svrg}++ computes $S$ times the full gradient $\nabla f(\cdot)$, and $\sum_{s=1}^{S} m_s = O(2^S m_0)$ times the gradient $\nabla f_i(\cdot)$. This gives a total gradient complexity $O(S \cdot n + 2^S \cdot m_0)$. \hfill \square

\section{Convergence Analysis for Section 5}

As in Section 4, for each outer iteration $s \in [S]$ and inner iteration $t \in \{0, 1, \ldots, m - 1\}$ of \textsc{svrg}, we denote by $i_t^s$ the selected random index $i \in [n]$ and $\xi_t^s$ the stochastic gradient $\xi = \nabla f_{i_t^s}(x_t^s) - \nabla f_{i_t^s}(\tilde{x}^{s-1}) + \tilde{\mu}_{s-1}$. Then, the following lemma is a counterpart of Lemma A.1 where the only difference is the use of the strong convexity parameter $\sigma$:

\textbf{Lemma B.1.} For every $u \in \mathbb{R}^d$ and $t \in \{0, 1, \ldots, m - 1\}$, fixing $x_t^s$ and letting $i = i_t^s$ be the random variable, we have

$$\mathbb{E}_{i_t^s} \left[ F(x_{t+1}^s) - F(u) \right] \leq \mathbb{E}_{i_t^s} \left[ \frac{\eta}{2(1 - \eta L)} \|\xi_t^s - \nabla f(x_t^s)\|^2 + \frac{1 - \sigma \eta}{2\eta} \|x_t^s - u\|^2 - \|x_{t+1}^s - u\|^2 \right].$$
Proof. We first upper bound the left hand side using the strong convexity and smoothness of $f(\cdot)$:

\[
E_{i_t} \left[ F(x_{s+1}^t) - F(u) \right] \\
= E_{i_t} \left[ f(x_{s+1}^t) - f(u) + \Psi(x_{s+1}^t) - \Psi(u) \right] \\
\leq E_{i_t} \left[ f(x_{s+1}^t) + \langle \nabla f(x_{s+1}^t), x_{s+1}^t - x_i^t \rangle + \frac{L}{2} \| x_{s+1}^t - x_i^t \|^2 - f(u) + \Psi(x_{s+1}^t) - \Psi(u) \right] \\
\leq E_{i_t} \left[ \langle \nabla f(x_{s+1}^t), x_{s+1}^t - x_i^t \rangle + \frac{L}{2} \| x_{s+1}^t - x_i^t \|^2 + \Psi(x_{s+1}^t) - \Psi(u) \right] \\
= E_{i_t} \left[ \langle \xi_{s+1}^t, x_i^t - u \rangle - \frac{\sigma}{2} \| x_i^t - u \|^2 \right] + \langle \nabla f(x_{s+1}^t), x_{s+1}^t - x_i^t \rangle + \frac{L}{2} \| x_{s+1}^t - x_i^t \|^2 + \Psi(x_{s+1}^t) - \Psi(u) \]
\]

(B.1)

Above, the term $\frac{\sigma}{2} \| x_i^t - u \|^2$ is due to the $\sigma$-strong convexity of $f(\cdot)$, and this is the only difference between the inequalities (B.1) and (A.1). Therefore, Lemma B.1 can be proven using exactly the identical rest of the proof of Lemma A.1.

We next state and prove a counterpart of Lemma A.2.

Lemma 5.2.

\[
E_{i_t} \left[ \| \xi_{s+1}^t - \nabla f(x_{s+1}^t) \|^2 \right] \leq 4(L + l) \cdot (F(x_{s+1}^t) - F(x^*) + F(\bar{x}^s - 1) - F(x^*)) \\
+ (8l^2 + 4Ll)(\| x_{s+1}^t - x^* \|^2 + \| \bar{x}^s - 1 - x^* \|^2) \] .
\]

Before we prove this lemma let us make a few remarks. First, if $l = 0$ then Lemma 5.2 is identical to Lemma A.2. In general, the second term in the above upper bound has a factor $8l^2 + 4Ll$ in the front which increases as $l$ increases. We can also compare Lemma 5.2 to that obtained by Shalev-Shwartz for sum-of-non-convex objectives: he showed $\| \xi_t - \nabla f(x_t) \|^2 \leq O(L^2) \cdot (\| x_t - x^* \|^2 + \| \bar{x}^s - 1 - x^* \|^2)$ in [19] which is suboptimal to ours and exactly why the $L^2$ factor shows up in his final gradient complexity.

Proof of Lemma 5.2. The first step of the proof of this lemma is analogous to most of the variance reduction literatures (cf. [2, 10, 26]):

\[
E_{i_t} \left[ \| \xi_{s+1}^t - \nabla f(x_{s+1}^t) \|^2 \right] = E_{i_t} \left[ \| \langle \nabla f_{i_t}(x_{s}^t) - \nabla f_{i_t}(\bar{x}^s) \rangle \rangle - \nabla f(x_{s+1}^t) - \nabla f(\bar{x}^s) \| ^2 \right] \\
\leq E_{i_t} \left[ \| \nabla f_{i_t}(x_{s}^t) - \nabla f_{i_t}(\bar{x}^s) \| ^2 \right] \\
= E_{i_t} \left[ \| \langle \nabla f_{i_t}(x_{s}^t) - \nabla f_{i_t}(x^*) \rangle - \nabla f_{i_t}(\bar{x}^s) \| \right] \| \nabla f_{i_t}(\bar{x}^s) \|^2 \\
\leq 2 \cdot E_{i_t} \left[ \| \nabla f_{i_t}(x_{s}^t) - \nabla f_{i_t}(x^*) \| ^2 \right] + \| \nabla f_{i_t}(\bar{x}^s) \| ^2 .
\]

(B.2)

Above, $\|$ is because for any random vector $\zeta \in \mathbb{R}^d$, it holds that $E\| \zeta - E\zeta \|^2 = E\| \zeta \|^2 - \| E\zeta \|^2$, and $\|$ is because for any two vectors $a, b \in \mathbb{R}^d$, it holds that $\| a - b \|^2 \leq 2\| a \|^2 + 2\| b \|^2$.

For analysis-purpose only, we define $\phi_i(y) \triangleq f_i(y) - \langle \nabla f_i(x^*), y \rangle + \frac{1}{2} \| y - x^* \|^2$ for each $i \in [n]$. It is clear that $\phi_i(y)$ is a convex, $(L + l)$-smooth function that has a minimizer $y = x^*$ (which can be seen by taking the derivative). For this reason, we claim that

\[
\phi_i(x^*) \leq \phi_i(y) - \frac{1}{L + l} \| \nabla \phi_i(y) \|^2 ,
\]

(B.3)
for each $y$, and this inequality is classical for smooth functions (see for instance Theorem 2.1.5 in
the textbook [16]). By expanding out the definition of $\phi_i(\cdot)$ in (B.3), we immediately have

$$f_i(x^*) - \langle \nabla f_i(x^*), x^* \rangle \leq f_i(y) - \langle \nabla f_i(x^*), y \rangle + \frac{l}{2} \| y - x^* \|^2 - \frac{1}{2(L + l)} \| \nabla f_i(y) - \nabla f_i(x^*) + l(y - x^*) \|^2$$

which then implies

$$\| \nabla f_i(y) - \nabla f_i(x^*) \|^2 \leq 2\| \nabla f_i(y) - \nabla f_i(x^*) + l(y - x^*) \|^2 + 2\| l(y - x^*) \|^2$$

$$\leq 2(L + l)(f_i(y) - f_i(x^*) - \langle \nabla f_i(x^*), y - x^* \rangle) + (4l^2 + 2Ll)\| y - x^* \|^2 .$$

(B.4)

Now, by choosing $y = x_i^*$ and $i = i_t^*$ in (B.4), we have

$$\mathbb{E}_{i_t^*} [\| \nabla f_i(x_{t+1}^*) - \nabla f_i(x^*) \|^2] \leq \mathbb{E}_{i_t^*} [2(L + l)(f_i(x_{t+1}^*) - f_i(x^*) - \langle \nabla f_i(x^*), x_{t+1}^* - x^* \rangle)] + (4l^2 + 2Ll)\| x_{t+1}^* - x^* \|^2$$

$$= 2(L + l)(f(x_{t+1}^*) - f(x^*) + \langle g^*, x_{t+1}^* - x^* \rangle) + (4l^2 + 2Ll)\| x_{t+1}^* - x^* \|^2$$

$$\leq 2(L + l)(f(x_{t+1}^*) - f(x^*) - \psi(x_{t+1}^*) + \psi(x^*)) + (4l^2 + 2Ll)\| x_{t+1}^* - x^* \|^2$$

$$= 2(L + l)(F(x_{t+1}^*) - F(x^*)) + (4l^2 + 2Ll)\| x_{t+1}^* - x^* \|^2 .$$

(B.5)

Above, $g^* \in \partial \Psi(x^*)$ is the subgradient of $\Psi$ at $x^*$ that satisfies $\nabla f(x^*) + g^* = 0.$

Similarly, by choosing $y = \bar{x}^{s-1}$ and $i = i_t^*$ in (B.4), we have

$$\mathbb{E}_{i_t^*} [\| \nabla f_i(\bar{x}^{s-1}) - \nabla f_i(x^*) \|^2] \leq 2(L + l)(F(\bar{x}^{s-1}) - F(x^*)) + (4l^2 + 2Ll)\| \bar{x}^{s-1} - x^* \|^2 .$$

(B.6)

Finally, putting together (B.2), (B.5) and (B.6) we finish the proof of the desired lemma. 

Finally, we are ready to prove our main theorem of this section:

**Proof of Theorem 5.1.** Combining Lemma A.1 with $u = x^*$, Lemma 5.2, as well as the assumption that $l \leq L$, we have

$$\mathbb{E}_{i_t^*} [F(x_{t+1}^*) - F(x^*)] \leq \frac{4\eta L}{(1 - \eta L)}(F(x_{t+1}^*) - F(x^*) + F(\bar{x}^{s-1}) - F(x^*) + \frac{3l}{2} \| x_{t+1}^* - x^* \|^2 + \frac{3l}{2} \| \bar{x}^{s-1} - x^* \|^2)$$

$$+ \frac{(1 - \sigma \eta)}{2\eta} \| x_{t+1}^* - x^* \|^2 - \mathbb{E}_{i_t^*} [x_{t+1}^* - x^*] \|^2 .$$

Choosing $\eta = \min\{\frac{1}{2L}, \frac{\sigma}{63L}\}$ in the above inequality, we conclude that

$$\mathbb{E}_{i_t^*} [F(x_{t+1}^*) - F(x^*)] \leq \frac{1}{5}(F(x_{t+1}^*) - F(x^*) + F(\bar{x}^{s-1}) - F(x^*)) + \frac{\sigma}{10} \| \bar{x}^{s-1} - x^* \|^2$$

$$+ \frac{\| x_{t+1}^* - x^* \|^2 - \mathbb{E}_{i_t^*} [x_{t+1}^* - x^*] \|^2}{2\eta} .$$

Summing it up over $t = 0, 1, \ldots, m - 1$, and dividing both sides by $m$, we arrive at

$$\mathbb{E} \left[ \sum_{t=0}^{m-1} \frac{F(x_{t+1}^*)}{m} - F(x^*) \right] \leq \mathbb{E} \left[ \frac{1}{5} \left( \sum_{t=0}^{m-1} \frac{F(x_{t+1}^*)}{m} - F(x^*) + F(\bar{x}^{s-1}) - F(x^*) \right) + \frac{\| x_{t+1}^* - x^* \|^2}{2\eta \cdot m} + \frac{\sigma}{10} \| \bar{x}^{s-1} - x^* \|^2 \right] .$$

19
After rearranging we have
\[
4E \left[ \sum_{t=0}^{m-1} \frac{F(x_{t+1}^s)}{m} - F(x^*) \right] \leq E \left[ (F(x_0^s) - F(x^*)) - \frac{(F(x_m^s) - F(x^*))}{m} \right] + F(\tilde{x}^{s-1}) - F(x^*)
\]
\[
\leq \left( 1 + \frac{1}{m} \right) (F(\tilde{x}^{s-1}) - F(x^*)) + \left( \frac{5}{\sigma \eta m} + 1 \right) (F(\tilde{x}^{s-1}) - F(x^*))
\]
\[
\leq \left( 1 + \frac{1}{m} \right) (F(\tilde{x}^{s-1}) - F(x^*)) + (4L^2 + 2Ll) \|y - x^*\|^2.
\]

Above, the last inequality uses the fact that \(x^*\) is a minimizer of \(F(\cdot)\) as well as our choice \(x_0^s = \tilde{x}^{s-1}\). Using the convexity of \(F(\cdot)\) we have \(F(\tilde{x}^s) \leq \frac{1}{m} \sum_{i=1}^m F(x_i^s)\) and therefore the above inequality gives
\[
E[F(\tilde{x}^s) - F(x^*)] \leq \frac{2}{4} \cdot \frac{1}{m} + \frac{5}{4} \sigma \eta m \cdot (F(\tilde{x}^{s-1}) - F(x^*))
\]
\[
= \frac{2}{4} \cdot \frac{1}{m} + \frac{5}{4} \sigma \eta m \cdot (F(\tilde{x}^{s-1}) - F(x^*))
\]
\[
\square
\]

C Convergence Analysis for Section 6

This section is devoted to proving Theorem 6.1.

We use the same notation as in Section 5 and Lemma B.1 remains true here. We replace Lemma 5.2 with the following:

Lemma C.1.
\[
E_{i_t} \left[ \|\xi_t^s - \nabla f(x_t^s)\|^2 \right] \leq (8L^2 + 4Ll)(\|x_t^s - x^*\|^2 + \|\tilde{x}^{s-1} - x^*\|^2)
\]

Proof. We begin the proof by first recalling (B.2) from the proof of Lemma 5.2.
\[
E_{i_t} \left[ \|\xi_t^s - \nabla f(x_t^s)\|^2 \right] \leq 2 \cdot E_{i_t} \left[ \|\nabla f_{i_t}(x_t^s) - \nabla f_{i_t}(x^*)\|^2 + \|\nabla f_{i_t}(\tilde{x}^{s-1}) - \nabla f_{i_t}(x^*)\|^2 \right].
\]

This time, we define \(\phi_i(y) \overset{\text{def}}{=} -f_i(y) + \langle \nabla f_i(x^*), y \rangle + \frac{L}{2} \|y - x^*\|^2\) for each \(i \in [n]\). It is clear that \(\phi_i(y)\) is a convex, \((L + l)\)-smooth function that has a minimizer \(y = x^*\) (which can be seen by taking the derivative). For this reason, we claim that
\[
\phi_i(x^*) \leq \phi_i(y) - \frac{1}{L + l} \|\nabla \phi_i(y)\|^2,
\]
for each \(y\), and this inequality is classical for smooth functions (see for instance Theorem 2.1.5 in the textbook [16]). By expanding out the definition of \(\phi_i(\cdot)\) in (C.1), we immediately have
\[
-f_i(x^*) + \langle \nabla f_i(x^*), x^* \rangle \leq -f_i(y) + \langle \nabla f_i(x^*), y \rangle + \frac{L}{2} \|y - x^*\|^2 - \frac{1}{2(L + l)} \|\nabla f_i(y) - \nabla f_i(x^*) - L(y - x^*)\|^2
\]
which then implies that
\[
\|\nabla f_i(y) - \nabla f_i(x^*)\|^2 \leq 2 \|\nabla f_i(y) - \nabla f_i(x^*) - L(y - x^*)\|^2 + 2 \|l(y - x^*)\|^2
\]
\[
\leq 2(L + l)(f_i(x^*) - f_i(y) + \langle \nabla f_i(x^*), y - x^* \rangle) + (4L^2 + 2Ll) \|y - x^*\|^2.
\]
Now by choosing $y = x_t^s$ and $i = i_t^s$ in (C.2), we have

\[
\mathbb{E}_{i_t^s}[\|\nabla f_{i_t^s}(x_t^s) - \nabla f_{i_t^s}(x^s)\|^2] \leq \mathbb{E}_{i_t^s}[2(L + l)(f_{i_t^s}(x^s) - f_{i_t^s}(x_t^s) + \langle \nabla f_{i_t^s}(x^s) , x_t^s - x^s \rangle ) + (4L^2 + 2LL)\|x_t^s - x^s\|^2 \\
= 2(L + l)(f(x^s) - f(x_t^s) + \langle \nabla f(x^s) , x_t^s - x^s \rangle ) + (4L^2 + 2LL)\|x_t^s - x^s\|^2 \\
\leq (4L^2 + 2LL)\|x_t^s - x^s\|^2 .
\]  

(C.3)

Above, the second inequality uses the convexity of $f(\cdot)$. Similarly, by choosing $y = \bar{x}^{s-1}$ and $i = i_t^s$ in (C.2), we have

\[
\mathbb{E}_{i_t^s}[\|\nabla f_{i_t^s}(\bar{x}^{s-1}) - \nabla f_{i_t^s}(x^s)\|^2] \leq (4L^2 + 2LL)\|\bar{x}^{s-1} - x^s\|^2 .
\]

(C.4)

Finally, putting together (B.2), (C.3) and (C.4) we finish the proof of the desired lemma.

Finally, we are ready to prove our main theorem of this section:

**Proof of Theorem 6.1.** Combining Lemma A.1 with $u = x^s$, Lemma C.1, as well as the assumption that $L \leq l$, we have

\[
\mathbb{E}_{i_t^s}[F(x_{t+1}^s) - F(x^s)] \leq \frac{12\eta l}{(1 - \eta L)} \frac{1}{2} \|x_t^s - x^s\|^2 + \frac{1}{2} \|\bar{x}^{s-1} - x^s\|^2 \\
+ \frac{(1 - \sigma \eta)\|x_t^s - x^s\|^2 - \mathbb{E}_{i_t^s}\|x_{t+1}^s - x^s\|^2}{2\eta} .
\]

Choosing $\eta = \frac{\sigma}{25Ll} \leq \frac{1}{25L}$ in the above inequality, we obtain that

\[
\mathbb{E}_{i_t^s}[F(x_{t+1}^s) - F(x^s)] \leq \frac{\sigma}{4} \|\bar{x}^{s-1} - x^s\|^2 + \frac{\|x_t^s - x^s\|^2 - \mathbb{E}_{i_t^s}\|x_{t+1}^s - x^s\|^2}{2\eta} .
\]

Summing it up over $t = 0, 1, \ldots, m - 1$, and dividing both sides by $m$, we arrive at

\[
\mathbb{E}\left[\frac{\sum_{t=0}^{m-1} F(x_{t+1}^s)}{m} - F(x^s)\right] \leq \mathbb{E}\left[\frac{\|x_0^s - x^s\|^2}{2\eta \cdot m} + \frac{\sigma}{4} \|\bar{x}^{s-1} - x^s\|^2 \right] .
\]

Finally, using our choice $x_0^s = \bar{x}^{s-1}$, using the convexity of $F(\cdot)$ which tells us $F(\bar{x}^{s-1}) \leq \frac{1}{m} \sum_{t=1}^{m} F(x_t^s)$, and using the strong convexity of $F(\cdot)$ which tells us $\frac{\sigma}{2} \|\bar{x}^{s-1} - x^s\|^2 \leq F(\bar{x}^{s-1}) - F(x^s)$, we conclude from the above inequality that

\[
\mathbb{E}[F(\bar{x}^s) - F(x^s)] \leq \frac{2 + \frac{4\eta m}{\sigma}}{4} (F(\bar{x}^{s-1}) - F(x^s)) .
\]

**D** SVRG++ for Non-Strongly Convex AND Sum-of-Non-Convex Objectives

In this section we show that our improvements for (1) non-strongly convex objectives in Section 4 and for (2) sum-of-non-convex objectives in Section 5 and 6 can be non-trivially put together. That is, we consider the case of (1.1) when each $f_i(x)$ is a not-necessarily convex function but $L$-upper and $l$-lower smooth for $l \geq 0$. We assume that $f$, the average of functions $f_i$, is simply convex but not necessarily strongly convex.
Algorithm 3 SVRG$_{nc}^{++}$($x^\phi, m_0, S, \eta$)

1: $\tilde{x}_0 \leftarrow x^\phi$, $x_0^1 \leftarrow x^\phi$
2: for $s \leftarrow 1$ to $S$ do
3:   $\tilde{\mu}_{s-1} \leftarrow \nabla f(\tilde{x}^{s-1})$
4:   $m_s \leftarrow 2^s \cdot m_0$
5:   $k \leftarrow 0$ and $T \leftarrow m_1 + \cdots + m_S$
6:   for $t \leftarrow 0$ to $m_s - 1$ do
7:     Pick $i$ uniformly at random in $\{1, \ldots, n\}$.
8:     $\xi \leftarrow \nabla f_i(x_t^s) - \nabla f_i(\tilde{x}^{s-1}) + \tilde{\mu}_{s-1}$
9:     $k \leftarrow k + 1$ and $\eta_{t+1} \leftarrow \eta \cdot \sqrt{T - k}$
10:    $x_{t+1}^s = \arg \min_{y \in \mathbb{R}^d} \left\{ \frac{1}{2m_{t+1}} \|x_t^s - y\|^2 + \Psi(y) + \langle \xi, y \rangle \right\}$
11:   end for
12:   $\tilde{x}^s \leftarrow \frac{1}{m_s} \sum_{i=0}^{m_s-1} x_t^s$
13:   $x_0^{s+1} \leftarrow x_{m_s}^s$
14: end for
15: return $\tilde{x}^S$.

For this class of objectives, if one applies a classical regularization (by adding a dummy $\frac{\sigma}{2} \|x\|^2$ regularizer for $\sigma \overset{\text{def}}{=} \frac{\varepsilon}{\|x_0 - x^*\|^2}$) reduction to that of Shalev-Shwartz SVRG [19], we can obtain a gradient complexity of essentially $O((n + \frac{T^2}{\varepsilon}) \log \frac{1}{\varepsilon})$. If one applies the same reduction to our new analysis in Section 5 and 6, we can obtain a gradient complexity of essentially $O((n + \frac{T^2}{\varepsilon} + \frac{T^3}{\varepsilon^2}) \log \frac{1}{\varepsilon})$. In this section, we propose a direct algorithm SVRG$_{nc}^{++}$ for solving this class of objectives with a gradient complexity of $O(n \log \frac{1}{\varepsilon} + \frac{T^2}{\varepsilon} + \frac{T^3}{\varepsilon^2})$.

Our SVRG$_{nc}^{++}$ algorithm for this case is analogous to SVRG$^{++}$ in Section 4. Given an initial vector $x^\phi$, our algorithm is divided into $S$ epochs. The $s$-th epoch consists of $m_s$ stochastic gradient steps, where $m_s$ doubles between every consecutive two epochs. As before, within each epoch we compute the full gradient $\tilde{\mu}_{s-1} = \nabla f(\tilde{x}^{s-1})$ where $\tilde{x}^{s-1}$ is the average point of the previous epoch. We use also $\tilde{\mu}_{s-1}$ to define the variance-reduced version of the stochastic gradient $\xi$. Unlike SVRG$^{++}$, for analysis purpose the step length $\eta$ is no longer a constant throughout the iterations. However, it will almost remain a constant.

More precisely, define $T = m_1 + \cdots + m_S \leq 2m_0 \cdot 2^S$ to be the total number of iterations. Then, for some parameter $\eta > 0$ to be chosen later, we define the sequence of step lengths

$$
\left(\eta_0^1, \eta_1^1, \ldots, \eta_{m_s}^1 = \eta_0^2, \eta_1^2, \ldots, \eta_{m_s}^2 = \eta_0^3, \eta_1^3, \ldots, \eta_{m_s}^S \right) \overset{\text{def}}{=} \left(\frac{\eta \sqrt{T}}{\sqrt{2T}}, \frac{\eta \sqrt{T}}{\sqrt{2T - 1}}, \ldots, \frac{\eta \sqrt{T}}{\sqrt{T}}\right).
$$

Note that in the above definition, the last step length $\eta_{m_s}^s$ is chosen as the same as the first step length $\eta_0^{s+1}$ of the next epoch. We also have $\frac{\eta}{\sqrt{2}} \leq \eta_t^s \leq \eta$ for all epochs $s$ and all iterations $t \in \{0, 1, \ldots, m_s\}$. Since for every real $k \geq 1$ we have $\sqrt{k} - \sqrt{k - 1} \geq \frac{1}{2 \sqrt{k}}$, it satisfies that

$$
\frac{1}{\eta_t^{s+1}} - \frac{1}{\eta_t^s} \geq \frac{1}{2 \eta \sqrt{T \cdot 2^2}} = \frac{1}{2 \sqrt{2 \eta T}}.
$$

We state our main convergence result for SVRG$_{nc}^{++}$ in this section as follows:

**Theorem D.1.** Suppose that $f(x)$ is convex, each $f_i(x)$ is $L$-upper and $l$-lower smooth in (1.1) for $l \in [0, L]$, and we are an initial vector $x^\phi$ satisfying $\|x^\phi - x^*\|^2 \leq \Theta$ and $F(x^\phi) - F(x^*) \leq \Delta$ for
parameters $\Theta, \Delta \in \mathbb{R}_+$. Then, $\text{SVRG}^{++}_{nc}(x^0, m_0, S, \eta)$ satisfies if \( \eta = \min \left\{ \frac{1}{13L}, \frac{2\sqrt{2}\Theta Ll}{312\sqrt{2}g_\Theta} \right\} \), \( m_0 = \frac{\Theta}{\eta \Delta} \), and \( S = \log(\Delta/\varepsilon) \), we have

\[
\mathbb{E}[F(\tilde{x}^S) - F(x^*)] \leq O(\varepsilon) .
\]

The total gradient complexity is \( O(S \cdot n + 2^S \cdot m_0) = O(n \log \frac{\Delta}{\varepsilon} + \frac{L\Theta^2}{\varepsilon} + \frac{L\Theta^2}{\varepsilon}) \).

We use the same notations of \( i^*_t \) and \( \xi^*_t \) as in previous sections. The following lemma is exactly Lemma A.1 where the step length \( \eta \) is replaced with \( \eta^*_t \):

**Lemma D.2** (Lemma A.1 revised). For every \( u \in \mathbb{R}^d \) and \( t \in \{0, 1, \ldots, m_s - 1\} \), fixing \( x_t^s \) and letting \( i = i^*_t \) be the random variable, we have

\[
\mathbb{E}_{i^*_t}[F(x_{t+1}^s) - F(u)] \leq \mathbb{E}_{i^*_t}\left[ \frac{\eta^*_t}{2(1 - \eta^*_t + L)} \| \xi^*_t - \nabla f(x^*_t) \|^2 + \frac{\|x_t^s - u\|^2 - \|x_{t+1}^s - u\|^2}{2\eta^*_t} \right].
\]

Also, by combining Lemma 5.2 (for \( F(x) \)) and Lemma C.1 (for \( l \geq L \)), we have that for every \( l \geq 0 \),

**Lemma D.3.**

\[
\mathbb{E}_{i^*_t}[\|\xi^*_t - \nabla f(x^*_t)\|^2] \leq 8L \cdot (F(x_t^s) - F(x^*) + F(\bar{x}^{s-1}) - F(x^*))
\]

\[
+ 12Ll(\|x_t^s - x^*\|^2 + \|\bar{x}^{s-1} - x^*\|^2) .
\]

Now we are ready to prove a lemma that is different from all previous sections.

**Lemma D.4.** If \( m_0 \geq 1 \), \( \eta \leq 1/13L \), and \( \frac{1}{4\sqrt{2}T \eta} \geq 39\eta Ll \), we have

\[
\mathbb{E}[F(\tilde{x}^S) - F(x^*)] \leq \frac{F(x^0) - F(x^*)}{2^{S-1}} + \frac{39\eta L\|x^0 - x^*\|^2}{2^S} + \frac{\|x^0 - x^*\|^2}{2^S \cdot \frac{4\eta L m_0}{3}} . \tag{D.2}
\]

**Proof.** Combining Lemma D.2 with \( u = x^* \) and Lemma D.3, as well as using the fact that \( \eta^*_t \leq \eta \), we have

\[
\mathbb{E}_{i^*_t}[F(x_{t+1}^s) - F(x^*)] \leq \frac{4\eta L}{(1 - \eta L)} (F(x_t^s) - F(x^*) + F(\bar{x}^{s-1}) - F(x^*) + 3l\|x_t^s - x^*\|^2 + 3l\|\bar{x}^{s-1} - x^*\|^2)
\]

\[
+ \frac{\|x_t^s - x^*\|^2 - \mathbb{E}_{i^*_t}\|x_{t+1}^s - x^*\|^2}{2\eta^*_t} .
\]

Choosing \( \eta \leq 1/13L \) in the above inequality, we have

\[
\mathbb{E}_{i^*_t}[F(x_{t+1}^s) - F(x^*)] \leq \frac{1}{3} (F(x_t^s) - F(x^*) + F(\bar{x}^{s-1}) - F(x^*)) + 13\eta L\|x_t^s - x^*\|^2 + 2\|\bar{x}^{s-1} - x^*\|^2)
\]

\[
+ \frac{\|x_t^s - x^*\|^2 - \mathbb{E}_{i^*_t}\|x_{t+1}^s - x^*\|^2}{2\eta^*_t} \leq \frac{1}{3} (F(x_t^s) - F(x^*) + F(\bar{x}^{s-1}) - F(x^*)) + 13\eta L\|x_t^s - x^*\|^2 + 2\|\bar{x}^{s-1} - x^*\|^2)
\]

\[
+ \frac{\|x_t^s - x^*\|^2 - \mathbb{E}_{i^*_t}\|x_{t+1}^s - x^*\|^2}{2\eta^*_t} .
\]

where the last inequality uses (D.1) and the assumption that \( \frac{1}{4\sqrt{2}T \eta} \geq 39\eta Ll \).
Summing it up over \( t = 0, 1, \ldots, m_s - 1 \) and dividing both sides by \( m_s \), we arrive at
\[
\mathbb{E}\left[ \sum_{t=0}^{m_s-1} \frac{F(x_{t+1}^s) + 26\eta L\|x_t^s - x^s\|^2}{m_s} - F(x^s) \right] = \mathbb{E}\left[ \frac{1}{3} \sum_{t=0}^{m_s-1} \frac{F(x_t^s) - F(x) + F(\bar{x}^{s-1}) - F(x^s)}{m_s} \right] + 13 \cdot \|\bar{x}^{s-1} - x^s\|^2 + \frac{\|x_0^s - x^s\|^2}{2\eta_0^s \cdot m_s} - \frac{\|x^s - x_{m_s}^s\|^2}{2\eta_0^s \cdot m_s}.
\]

After rearranging, this yields
\[
2\mathbb{E}\left[ \sum_{t=0}^{m_s-1} \frac{F(x_t^s) + 39\eta L\|x_t^s - x^s\|^2}{m_s} - F(x^s) \right] \leq \mathbb{E}\left[ \frac{3(F(x_0^s) - F(x^s)) - 3(F(x_{m_s}) - F(x^s)) + F(\bar{x}^{s-1}) - F(x^s)}{m_s} \right] + 39 \cdot \|\bar{x}^{s-1} - x^s\|^2 + \frac{\|x_0^s - x^s\|^2}{2\eta_0^s / 3 \cdot m_s} - \frac{\|x^s - x_{m_s}^s\|^2}{2\eta_0^s / 3 \cdot m_s}.
\]

Next, using the fact that \( F(\bar{x}^s) \leq \sum_{t=0}^{m_s-1} \frac{F(x_t^s)}{m_s} \) and \( \|\bar{x}^s - x^s\|^2 \leq \sum_{t=0}^{m_s-1} \|x_t^s - x^s\|^2 \) which follow from convexity and the definition \( \bar{x}^s = \sum_{t=0}^{m_s-1} x_t^s / m_s \), we can we rewrite the above inequality as
\[
2\mathbb{E}\left[ F(\bar{x}^s) - F(x^s) + 39\eta L\|\bar{x}^s - x^s\|^2 \right] \leq \mathbb{E}\left[ \frac{3(F(x_0^s) - F(x^s)) - 3(F(x_{m_s}) - F(x^s)) + F(\bar{x}^{s-1}) - F(x^s)}{m_s} \right] + \frac{39 \cdot \|\bar{x}^{s-1} - x^s\|^2}{\eta_0^s / 3 \cdot m_s - 1} - \frac{\|x^s - x_{m_s}^s\|^2}{2\eta_0^s / 3 \cdot m_s}.
\]

At this point, let us recall choice \( x_{m_s}^s = x_0^{s+1}, \eta_{m_s}^s = \eta_0^{s+1}, \) and \( m_s = 2m_{s-1} \), which yield
\[
2\mathbb{E}\left[ F(\bar{x}^s) - F(x^s) + 39\eta L\|\bar{x}^s - x^s\|^2 + \|x^s - x_0^{s+1}\|^2 \right] \leq \mathbb{E}\left[ \frac{3(F(x_0^s) - F(x^s)) - 3(F(x_{m_s}) - F(x^s)) + F(\bar{x}^{s-1}) - F(x^s)}{2m_{s-1} / 3} \right] + \frac{39 \cdot \|\bar{x}^{s-1} - x^s\|^2}{\eta_0^s / 3 \cdot m_{s-1}} - \frac{\|x^s - x_{m_s}^s\|^2}{2\eta_0^s / 3 \cdot m_{s-1}}.
\]

In sum, after telescoping for \( s = 1, 2, \ldots, S \), we have
\[
\mathbb{P}[F(\bar{x}^S) - F(x^s)] \leq 2^{-S} \cdot \left( F(x^0) - F(x^s) + 39\eta L\|x^0 - x^s\|^2 + \|x^s - x_0^{s+1}\|^2 \right) \leq \frac{F(x^0) - F(x^s)}{2^{S-1}} + \frac{39\eta L\|x^0 - x^s\|^2}{2^S} + \frac{\|x^s - x_0^{s+1}\|^2}{2^S \cdot 3m_0 / \sqrt{2}}.
\]

Finally, the above lemma immediately yields our desired theorem:

**Proof of Theorem D.1.** Under the given parameter choices, we first have
\[
\frac{1}{4\sqrt{2T\eta}} \geq \frac{1}{4\sqrt{2m_0 \cdot 2^S}} = \frac{1}{8\sqrt{2m_0} \cdot \frac{2^S}{\epsilon}} = \frac{\epsilon}{8\sqrt{2\Theta}} = \frac{\epsilon}{312\sqrt{2\Theta}} \geq 39\eta L
\]
so the preassumption of Lemma D.4 holds.
Now we consider the three terms on the right hand side of (D.2). The first term is no more than $\frac{2\Delta}{2^S} \leq 2\varepsilon$. The second term is no more than

$$\frac{39\eta L l \Theta}{2^S} = 39\eta L l \Theta \leq \frac{\varepsilon}{8\sqrt{2}\Delta} \leq \frac{\varepsilon}{8\sqrt{2}} .$$

The third term is no more than

$$\frac{\Theta}{\Delta / \varepsilon \cdot \frac{4\eta m_0}{3\sqrt{2}}} = \frac{\Theta}{4/\varepsilon \cdot \frac{4\eta}{3\sqrt{2}}} = \frac{3\sqrt{2}}{4} \varepsilon .$$

In sum, we conclude that $\mathbb{E}[F(\tilde{x}^S) - F(x^*)] \leq O(\varepsilon)$. □

E Missing Figures in Section 7
Figure 3: Training error comparisons on dataset Adult. The $y$ axis represents the training objective value minus the minimum, and the $x$ axis represents the number of passes to the dataset.
Figure 4: Training error comparisons on dataset Covtype. The y axis represents the training objective value minus the minimum, and the x axis represents the number of passes to the dataset.
Figure 5: Training error comparisons on dataset Ijcnn1. The $y$ axis represents the training objective value minus the minimum, and the $x$ axis represents the number of passes to the dataset.
Figure 6: Training error comparisons on dataset mnist. The y axis represents the training objective value minus the minimum, and the x axis represents the number of passes to the dataset.