Katyusha X: Practical Momentum Method for Stochastic Sum-of-Nonconvex Optimization

Zeyuan Allen-Zhu [1]

Abstract

The problem of minimizing sum-of-nonconvex functions (i.e., convex functions that are average of non-convex ones) is becoming increasingly important in machine learning, and is the core machinery for PCA, SVD, regularized Newton's method, accelerated non-convex optimization, and more. We show how to provably obtain an accelerated stochastic algorithm for minimizing sum-of-nonconvex functions, by *adding one additional line* to the well-known SVRG method. This line corresponds to momentum, and shows how to directly apply momentum to the finite-sum stochastic minimization of sum-of-nonconvex functions. As a side result, our method enjoys linear parallel speed-up using mini-batch.[1]

1. Introduction

The diverse world of non-convex machine learning tasks have given rise to numerous non-convex optimization problems. Some of them are perhaps as hard as minimizing general non-convex objectives (such as deep learning), but some others may be only slightly harder than convex optimization (such as matrix completion, principal component analysis, dictionary learning, etc). Therefore, it is both interesting and challenging to identify classes of optimization problems that *interplay* between non-convex and convex optimization, and (hopefully) optimally and practically solving them.

At least tracing back to 2015, Shalev-Shwartz (2016) identified a class of functions that are convex, but can be written as finite average of non-convex functions. That is,[2]

$$\min_{x\in\mathbb{R}^d} \left\{ f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x) \right\} \tag{1.1}$$

where each $f_i(x)$ is smooth and non-convex, but their average $f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$ is convex.

We say $f(x)$ a *sum-of-nonconvex* (but convex) function following (Garber et al., 2016; Allen-Zhu & Yuan, 2016).

In this paper, we show how to provably obtain an *accelerated* and *stochastic* method for minimizing (1.1). Our new method is based on adding only *one* additional line to the well-known SVRG (stochastic variance-reduction gradient) method (Johnson & Zhang, 2013; Zhang et al., 2013). This additional line corresponds to momentum. To the best of our knowledge, this explains for the first time how to directly apply momentum to the stochastic minimization of sum-of-nonconvex functions. We hope this new algorithm and the new insight of this paper could facilitate our understanding towards how to correctly and provably apply momentum to non-convex machine learning tasks.

1.1. Motivating Examples

There is an increasing number of machine learning tasks that are found reducible to minimizing sum-of-nonconvex functions. For most such tasks, the only known approach for achieving accelerated stochastic performance is by reformulating them as multiple instances of Problem (1.1) where f is a convex sum of non-convex functions.

Perhaps the most famous example is the *shift-and-invert* approach to solve PCA (Saad, 1992). Let $A = \frac{1}{n}\sum_{i=1}^{n} a_i a_i^\top \in \mathbb{R}^{d\times d}$ be some covariance matrix and λ be its largest eigenvalue. Then, computing A's top eigenvector reduces to applying power method to a new matrix $B = (\mu I - A)^{-1}$ with $\mu = \lambda_{\max}(A)\cdot(1+\delta)$ for some approximation parameter $\delta > 0$ (Garber et al., 2016). In

[1]Microsoft Research AI. Correspondence to: Zeyuan Allen-Zhu <zeyuan@csail.mit.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

[1]This paper is related to the `Katyusha` method (Allen-Zhu, 2017a) which gives direct accelerated method for minimizing a function that is an average of convex functions $f_i(x)$. The new method `KatyushaX` tackles the more challenging case when $f_i(x)$ is non-convex but the average is convex. We borrow the name from `Katyusha` because the two papers tackle questions of the same type, but the algorithms are different.

Full and future versions can be found on `https://arxiv.org/abs/1802.03866`.

[2]In fact, we study a more general composite minimization setting $\min_{x\in\mathbb{R}^d}\left\{F(x) := \psi(x) + \frac{1}{n}\sum_{i\in[n]} f_i(x)\right\}$ where $\psi(x)$ is some proper convex function. In this high-level introduction, we ignore the $\psi(\cdot)$ term for simplicity.

other words, PCA reduces to repeatedly minimizing convex functions $f(x) := \frac{1}{2}x^\top(\mu I - A)x + b^\top x$ for different vectors b. If one defines $f_i(x) := \frac{1}{2}x^\top(\mu I - a_i a_i^\top)x + b^\top x$, then $f_i(x)$ is smooth and non-convex, but $f(x)$ is convex.

- HIGH ACCURACY NEEDED. In the above reduction, we need very high accuracy (e.g., 10^{-10}) when minimizing $f(x)$, because we need to apply power method on B so the error blows up. This is very different from classical empirical minimization problems (such as Lasso, SVM), where we only need to minimize the training objective to some accuracy such as 10^{-3}.
- NECESSITY OF PROBLEM (1.1) While there are many algorithms to solve PCA, to the best of our knowledge, the only known stochastic method which gives a provable accelerated rate requires solving Problem (1.1). We discuss more in Related Works.

Other problems that reduce to Problem (1.1) include:

- The accelerated stochastic algorithms for computing top k principle components (k-PCA) and top k singular vectors (k-SVD) require solving Problem (1.1) (Allen-Zhu & Li, 2016).
- The fastest way to compute the near-optimal strategy for the online eigenvector problem (against an adversarial opponent) is by solving Problem (1.1) (Allen-Zhu & Li, 2017b).
- Up to this date, the fastest finite-sum stochastic algorithm for finding approximate local minima of a general non-convex smooth function is either based on cubic regularized Newton's method (Agarwal et al., 2017) or a special reduction (Carmon et al., 2016). Both approaches require solving Problem (1.1).
- In certain parameter regimes, the fastest finite-sum stochastic algorithm for minimizing "approximately convex functions" is based on solving Problem (1.1) (Allen-Zhu, 2017b; Carmon et al., 2016).

1.2. Known Approaches

In the online stochastic setting (i.e., when n is infinite), there is hardly any difference between $f_i(x)$ being convex or not. The stochastic gradient descent (SGD) method gives a $T_{\mathsf{grad}} \propto \varepsilon^{-2}$ rate to Problem (1.1), or $T_{\mathsf{grad}} \propto (\sigma\varepsilon)^{-1}$ rate if $f(x)$ is σ-strongly convex. Both rates are optimal, regardless of $f_i(x)$ being convex or not.

Variance Reduction. In the finite-sum stochastic setting (i.e., when n is finite), it was discovered by Shalev-Shwartz (2016) and Garber et al. (2016) that one can solve (1.1) using variance reduction: for instance, using the SVRG method that was originally designed for convex optimization (Johnson & Zhang, 2013; Zhang et al., 2013).

Specifically, if $f(x)$ is σ-strongly convex and each $f_i(x)$ is L-smooth, then the SVRG method converges to an ε-minimizer of Problem (1.1) using T_{grad} stochastic gradient computations, where

$$\boxed{\begin{aligned} T_{\mathsf{grad}} &= O\Big(\big(n + \frac{\sqrt{n}L}{\sigma}\big)\log\frac{1}{\varepsilon}\Big) \quad \text{if } \sigma > 0 \quad \text{or} \\ T_{\mathsf{grad}} &= O\Big(n\log\frac{1}{\varepsilon} + \frac{\sqrt{n}L}{\varepsilon}\Big) \quad \text{if } \sigma = 0 \quad . \end{aligned}} \tag{1.2}$$

Both rates outperform their corresponding counterparts in the SGD case.

Remark 1.1. The two complexity bounds in (1.2) are better than the original work of Shalev-Shwartz (2016) and Garber et al. (2016). They showed $T_{\mathsf{grad}} = O\big((n + \frac{L^2}{\sigma^2})\log\frac{1}{\varepsilon}\big)$ in the case of $\sigma > 0$; and Allen-Zhu & Yuan (2016) showed $T_{\mathsf{grad}} = O\big(n\log\frac{1}{\varepsilon} + \frac{L^2}{\varepsilon^2}\big)$ in the case of $\sigma = 0$. Both complexity bounds are no better than (1.2). We prove (1.2) as a side result of this paper in the full version.

How to Accelerate. If Nesterov's accelerated gradient method (Nesterov, 2004; 2005) (also known as the *momentum* method) is used, one can achieve $T_{\mathsf{grad}} = O\big(\frac{n\sqrt{L}}{\sqrt{\sigma}}\log\frac{1}{\varepsilon}\big)$ and $T_{\mathsf{grad}} = O\big(\frac{n\sqrt{L}}{\sqrt{\varepsilon}}\big)$ in the two cases. This square-root dependence on σ or ε is known as the *accelerated convergence rate*.

However, Nesterov's method is not stochastic and its T_{grad} linearly scales with n. Can we design a stochastic first-order method that also has an accelerated convergence rate?

Remark 1.2. Obtaining accelerated rates is crucial for Problem (1.1), because high accuracy is usually needed as we discussed in Section 1.1.

This can be partially answered by the APPA (Frostig et al., 2015) and Catalyst (Lin et al., 2015) reduction, that we both refer to as Catalyst.[3] Mathematically, Catalyst turns any non-accelerated method into an accelerated one. For instance, when $\sigma > 0$, Catalyst on SVRG (often referred to as AccSVRG) uses the following logic:

— Define a "new problem" $\arg\min_x \big\{f(x) + \frac{L'}{2}\|x - \widehat{x}\|^2\big\}$ for some $\widehat{x} \in \mathbb{R}^d$ and $L' = L/\sqrt{n}$.

— Use Nesterov's method to minimize $f(x)$, which requires solving the "new problem" $\widetilde{O}\big(1 + \frac{\sqrt{L'}}{\sqrt{\sigma}}\big)$ times.

— Solve each "new problem" by SVRG: (1.2) gives $T'_{\mathsf{grad}} = \widetilde{O}\big(\big(n + \frac{\sqrt{n}L}{L'}\big)\big) = \widetilde{O}(n)$.

In total, this requires T_{grad} stochastic gradient computations for $T_{\mathsf{grad}} = \widetilde{O}\big(\big(1 + \frac{\sqrt{L'}}{\sqrt{\sigma}}\big) \times T'_{\mathsf{grad}}\big)$. In other words,

[3] Both reductions are based on an outer-inner loop structure first proposed by Shalev-Shwartz & Zhang (2014). The application of Catalyst to solving Problem (1.1) first appeared in (Shalev-Shwartz, 2016; Garber et al., 2016) in the context of PCA.

AccSVRG requires[4]

$$T_{\mathsf{grad}} = O\Big((n+\frac{n^{3/4}\sqrt{L}}{\sqrt{\sigma}})\log^2\frac{1}{\varepsilon}\Big) \quad \text{if } \sigma>0 \quad \text{or}$$
$$T_{\mathsf{grad}} = O\Big((n+\frac{n^{3/4}\sqrt{L}}{\sqrt{\varepsilon}})\log\frac{1}{\varepsilon}\Big) \quad \text{if } \sigma=0 \quad . \quad (1.3)$$

Unfortunately, the practicality of AccSVRG remains somewhat unsettled. To mention a few issues:

- Since error propagates, one needs to run each SVRG until a very accurate point is obtained.
- To optimize the complexity, one needs to terminate each call of SVRG at a different accuracy.
- One needs to tune three parameters: (1) the regularizer weight L', (2) the learning rate of SVRG, and (3) the weight of the momentum.

When all of these factors are putting together, the practical performance of AccSVRG may be even worse than the non-accelerated SVRG.[5]

1.3. Our Main Result

In this paper, we propose a new method KatyushaX which, copying the original SVRG method but adding only one additional line, achieves the accelerated stochastic convergence rate. We give two different specifications, KatyushaX$^{\mathsf{s}}$ and KatyushaX$^{\mathsf{w}}$, where

- KatyushaX$^{\mathsf{s}}$ needs a momentum parameter in addition to the learning rate for SVRG; and
- KatyushaX$^{\mathsf{w}}$ needs no additional parameter whatsoever on top of SVRG.

We explain how they work below, and Figure 1(a) gives a quick performance comparison between SVRG, KatyushaX$^{\mathsf{s}}$ and KatyushaX$^{\mathsf{w}}$ on some synthetic dataset.

We first recall how SVRG works. Each *epoch* of SVRG consists of n iterations. Each epoch starts with a point w_0 (known as the *snapshot*) where the full gradient $\nabla f(w_0)$ is computed exactly. Then, in each iteration $t=0,1,\ldots,n-1$ of this epoch, SVRG updates $w_{t+1}\leftarrow w_t-\eta\widetilde{\nabla}_t$ where the gradient estimator $\widetilde{\nabla}_t=\nabla f_i(w_t)-\nabla f_i(w_0)+\nabla f(w_0)$ for some random $i\in[n]$, and $\eta>0$ is the learning rate.

If we summarize the above one epoch process of SVRG as $\mathtt{SVRG}^{\mathsf{1ep}}(w_0,\eta):=w_n$, then

- The classical SVRG method can be described by the iterative update

$$\boxed{x_{k+1}=\mathtt{SVRG}^{\mathsf{1ep}}(x_k,\eta)\ .}$$

- Our KatyushaX$^{\mathsf{s}}$, parameterized by a momentum parameter $\tau\in(0,1)$, is

$$\boxed{\begin{aligned} y_k &\leftarrow \mathtt{SVRG}^{\mathsf{1ep}}(x_k,\eta)\\ x_{k+1} &\leftarrow \frac{\frac{3}{2}y_k+\frac{1}{2}x_k-(1-\tau)y_{k-1}}{1+\tau}\ . \end{aligned}}$$

- Our KatyushaX$^{\mathsf{w}}$ is

$$\boxed{\begin{aligned} y_k &\leftarrow \mathtt{SVRG}^{\mathsf{1ep}}(x_k,\eta)\\ x_{k+1} &\leftarrow \frac{(3k+1)y_k+(k+1)x_k-(2k-2)y_{k-1}}{2k+4}\ . \end{aligned}}$$

Remark 1.3. When choosing $\tau=1/2$, KatyushaX$^{\mathsf{s}}$ is exactly identical to SVRG and $y_k\equiv x_{k+1}$.

Remark 1.4. In KatyushaX$^{\mathsf{s}}$, if we replace $\frac{3}{2}y_k+\frac{1}{2}x_k$ with $2y_k$, then the update becomes $x_{k+1}\leftarrow y_k+\frac{1-\tau}{1+\tau}(y_k-y_{k-1})$. This corresponds to a classical momentum scheme by Nesterov (2004). The smaller $\tau>0$ is the "stronger" the momentum behaves.

Remark 1.5. KatyushaX$^{\mathsf{w}}$ is in fact KatyushaX$^{\mathsf{s}}$ with $\tau=\frac{2}{k+2}$ decreasing in k. See full version for the details.

Our main theorems are the following:

Theorem 1 (informal). *If $f(x)$ is σ-strongly convex and each $f_i(x)$ is L-smooth, then KatyushaX$^{\mathsf{s}}$ with $\eta=\Theta(\frac{1}{\sqrt{nL}})$ and $\tau=\min\{\frac{1}{2},\Theta(\frac{n^{1/4}\sqrt{\sigma}}{\sqrt{L}})\}$ outputs a point x with $\mathbb{E}[f(x)-f(x^*)]\le\varepsilon$ using*

$$T_{\mathsf{grad}}=O\Big((n+\frac{n^{3/4}\sqrt{L}}{\sqrt{\sigma}})\log\frac{1}{\varepsilon}\Big)$$

stochastic gradient computations.

Theorem 2 (informal). *If $f(x)$ is convex and each $f_i(x)$ is L-smooth, then KatyushaX$^{\mathsf{w}}$ with $\eta=\Theta(\frac{1}{\sqrt{nL}})$ outputs a point x with $\mathbb{E}[f(x)-f(x^*)]\le\varepsilon$ using*

$$T_{\mathsf{grad}}=O\Big(n+\frac{n^{3/4}\sqrt{L}}{\sqrt{\varepsilon}}\Big)$$

stochastic gradient computations.

In sum, we have not only tightened the complexity bounds by removing a logarithmic factor each (comparing to AccSVRG (1.3)), but also obtained a much *simpler, practical* acceleration (i.e., momentum) scheme for minimizing sum-of-nonconvex functions stochastically.

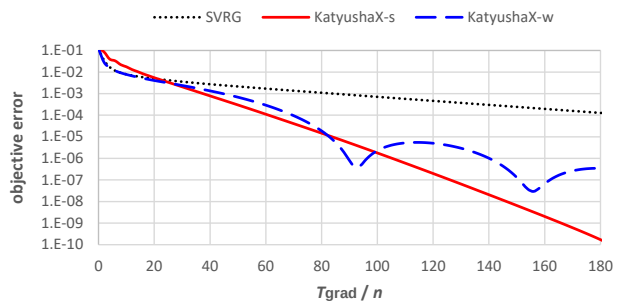
1.4. Our Side Results

To demonstrate the strength of KatyushaX, we prove our main theorem in several more general settings.

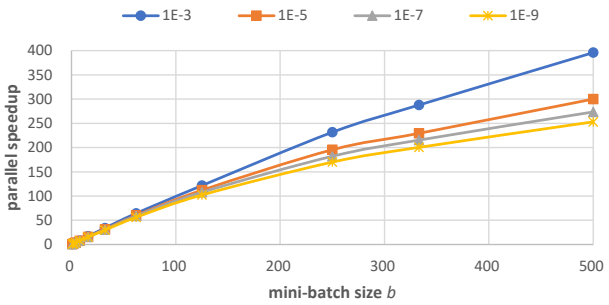
(1) Upper and Lower Smoothness. For non-convex func-

[4] For why these logarithmic factors show up, we refer readers to the journal version of Catalyst (Lin et al., 2017). In particular, one of the two log factors in the case of $\sigma>0$ is because SVRG is a randomized algorithm.

[5] This is so already in the easier case where each $f_i(x)$ is convex (Allen-Zhu, 2017a). In this simpler case, direct acceleration (without applying the Catalyst reduction) is more practical and already known (see Katyusha (Allen-Zhu, 2017a) and DASVRDA (Murata & Suzuki, 2017)).



(a) SVRG vs. KatyushaX^s vs. KatyushaX^w



(b) mini-batch performance of KatyushaX^s

Figure 1: A simple illustration on minimizing $f(x) = \frac{1}{2}(\mu I - BB^\top)$ where $B \in \mathbb{R}^{1000\times 1000}$ is a random ± 1 matrix, and $\mu = \lambda_1(BB^\top) + 0.5\big(\lambda_1(BB^\top) - \lambda_2(BB^\top)\big)$. Such $f(x)$ is a typical instance in stochastic PCA (Garber et al., 2016).
Remark 1. In SVRG, the best learning rate is $\eta = 0.4/L$ after tuning.
Remark 2. We used $\eta = 0.4/L$ for KatyushaX^w. We used $\eta = 0.4/L$ and $\tau = 0.1$ for KatyushaX^s.
Remark 3. In the mini-batch experiment, we used $\eta = \frac{0.4b}{L}$ and $\tau = 0.1$. The parallel speed-up is in terms of achieving objective error $10^{-3}, 10^{-5}, 10^{-7}, 10^{-9}$.

tions, its upper and lower smoothness parameters (i.e., maximum eigenvalue vs. negated minimum eigenvalue of Hessian) may be very different. This is especially true for all PCA and SVD related applications (Garber et al., 2016; Allen-Zhu & Li, 2016; 2017a), where $f_i(x) = \frac{\mu}{2}\|x\|^2 - \langle a_i, x\rangle^2$ so its upper smoothness is μ and lower smoothness is $2\|a_i\|^2$. (It usually happens that $\|a_i\|^2 \gg \mu$.)

Assuming $f_i(x)$ is ℓ_1-upper and ℓ_2-lower smooth (and $\ell_2 \geq \ell_1$), it is known that by simply changing the learning rate of SVRG, its runs in a worst-case complexity proportional to $\sqrt{\ell_1\ell_2}$ instead of L (Allen-Zhu & Yuan, 2016).

We show KatyushaX enjoys this speed up as well. It runs in a complexity proportional to $(\ell_1\ell_2)^{1/4}$ instead of $L^{1/2}$.

(2) Composite Minimization. Consider objective $F(x) = \psi(x) + \frac{1}{n}\sum_{i=1}^n f_i(x)$ where $\psi(x)$ is some proper convex (not necessarily smooth) function, usually referred to as the proximal term.[6] Then, most stochastic gradient methods can be extended to minimize composite objectives, if we replace the update $w_{t+1} \leftarrow w_t - \eta\widetilde{\nabla}_t$ with $w_{t+1} \leftarrow \arg\min_{z\in\mathbb{R}^d}\big\{\frac{1}{2\eta}\|z - w_t\|_2^2 + \langle\widetilde{\nabla}_t, z\rangle + \psi(z)\big\}$.

We show that KatyushaX also extends to the composite minimization setting.

(3) Parallelism / Mini-batch. Instead of using a single stochastic gradient $\nabla f_i(\cdot)$ per iteration, for any stochastic method, one can replace it with the average of b stochastic gradients $\frac{1}{b}\sum_{i\in S}\nabla f_i(\cdot)$ where S is a random subset of $[n]$ with cardinality b. This is known as the *mini-batch* technique and it allows the stochastic gradients to be computed in a distributed manner, using up to b processors.

[6] Examples of proximal terms include $\psi(x) = \|x\|_1$ or $\psi(x) = \begin{cases} 0, & x \in \mathcal{X}; \\ +\infty, & x \notin \mathcal{X}. \end{cases}$ for some convex set $\mathcal{X} \subseteq \mathbb{R}^d$.

Our KatyushaX methods extends to this mini-batch setting too. Using mini-batch size b, the worst-case number of parallel iterations of KatyushaX reduces by

$$\begin{cases} O(b), & \text{if } b \leq \sqrt{n}; \\ O(\sqrt{b}), & \text{if } b \in [\sqrt{n}, n]. \end{cases}$$

Therefore, at least for small $b \in \{1, 2, \ldots, \lceil\sqrt{n}\rceil\}$, KatyushaX enjoys a *linear speed-up* in the parallel worst-case running time. We do not find such result recorded before this work.

(4) Non-Uniform Sampling. When functions $f_i(x)$ are of non-uniform hardness (say, with different smoothness parameters), instead of sampling each function $f_i(x)$ uniformly at random, one can sample i with a probability proportional to its "hardness." This can improve the performance of stochastic gradient methods.

We show that KatyushaX also enjoys non-uniform sampling benefits. If each function $f_i(x)$ is L_i-smooth, one can sample i with probability proportional to L_i^2. If we denote by $\overline{L} = (\sum_i L_i^2/n)^{1/2}$, then the worst-case complexities can be improved to depend on $\overline{L}$ instead of $\max_i\{L_i\}$.

1.5. Other Related Works

Since sum-of-nonconvex optimization is closely related to PCA, let us mention the most standard variants of PCA and their relationships to Problem (1.1).

Offline Stochastic PCA. In the offline setting, we assume $A = \frac{1}{n}\sum_{i=1}^n a_i a_i^\top$ and a stochastic method can compute $\langle a_i, x\rangle$ for some vector x in each iteration.

To approximate the top eigenvector of A (i.e., 1-PCA), the first variance reduction method is by Shamir (2015) and does not need sum-of-nonconvex optimization. Unfortu-

nately, his method is not gap-free[7] and not accelerated. Garber et al. (2016) obtained a stochastic, gap-free, and accelerated method by reducing 1-PCA to Problem (1.1) using shift-and-invert. To this date, this seems to be the only approach to obtain a stochastic *and* accelerated method for 1-PCA.

To approximate the top k eigenvectors of A (i.e., k-PCA), the first variance reduction method is by Shamir (2016) and does not need sum-of-nonconvex optimization. His method is not gap-free, not accelerated, and has a slow worst-case complexity. Allen-Zhu & Li (2016) obtained a stochastic, gap-free, and accelerated method for k-PCA by reducing the problem to Problem (1.1). To this date, this seems to be the only approach to obtain a stochastic *and* accelerated method for k-PCA.

Online Stochastic PCA. In the online setting, we assume $A = \mathbb{E}_i[a_i a_i^\top]$ where there may be infinitely many vectors a_i so the complexity of the stochastic method cannot depend on n.

In the case of online 1-PCA, the optimal algorithm is Oja's method (Oja, 1982), whose first optimal analysis was due to (Jain et al., 2016) and first optimal gap-free analysis was due to (Allen-Zhu & Li, 2017b). In the case of online k-PCA, the optimal algorithm is a block variant of Oja's method, whose first optimal analysis was due to (Allen-Zhu & Li, 2017a).

In the online stochastic setting, due to information-theoretic lower bounds (Allen-Zhu & Li, 2017a;b), one cannot apply variance reduction or acceleration to improve the worst-case complexity.

Online Adversarial PCA. In an online learning scenario where the player chooses a unit vector v_t at round t and the adversary chooses a matrix A_t, one can design regret-minimizing strategy for the player in terms of maximizing $\sum_t v_t^\top A_t v^t$. In this game, the regret-optimal strategy for the player is follow-the-regularized-leader (FTRL) but it runs very slow; an efficient strategy for the player is follow-the-perturbed-leader (FTPL) but it gives a poor regret (Ma et al., 2015). The recent result follow-the-compressed-leader (FTCL) gives a strategy that is both regret near-optimal and efficiently computable (Allen-Zhu & Li, 2017b). Both strategies FTCL and FTPL rely on solving multiple instances of Problem (1.1).

Stochastic Nonconvex Optimization. In the harder problem where $f(x) = \frac{1}{n}\sum_{i=1}^n f_i(x)$ is also non-convex, variance reduction is also proven useful, both in terms of finding approximate stationary points and approximate local minima.

In the finite-sum case (i.e., when n is finite), the SVRG method finds an ε-approximate stationary point in $T_{\mathsf{grad}} = O(n^{2/3}/\varepsilon^2)$ where in contrast SGD needs $O(1/\varepsilon^4)$ and full gradient descent needs $O(n/\varepsilon^2)$.[8] This was obtained independently by Reddi et al. (2016) and Allen-Zhu & Hazan (2016a). To find an ε-approximate local minima, one needs an additional second-order smoothness assumption, and the two independent works Agarwal et al. (2017); Carmon et al. (2016) need $T_{\mathsf{grad}} = O\big(\frac{n}{\varepsilon^{1.5}} + \frac{n^{3/4}}{\varepsilon^{1.75}}\big)$ (both these algorithms reduce the task to solving Problem (1.1)).

In the online case (i.e., when n is infinite), the SCSG method of Lei et al. (2017) is a variant of SVRG and finds ε-approximate stationary points in $T_{\mathsf{grad}} = O(1/\varepsilon^{3.333})$. To find an ε-approximate local minima, one needs an additional second-order smoothness assumption and the rate can be improved to $T_{\mathsf{grad}} = O(1/\varepsilon^{3.25})$ using variance reduction (Allen-Zhu, 2017b); in contrast, without variance reduction, the best rate (achieved by a variant of SGD) is $T_{\mathsf{grad}} = O(1/\varepsilon^{3.5})$ (Allen-Zhu, 2018). None of these algorithms rely on Problem (1.1), and momentum is not known to be helpful in the online setting.

Stochastic Convex Optimization. Variance reduction was first discovered for the purpose of minimizing the simpler problem $f(x) = \frac{1}{n}\sum_{i=1}^n f_i(x)$ where each $f_i(x)$ is also convex (Schmidt et al., 2013). Its accelerated variant was first discovered by the APPA/Catalyst reduction (Frostig et al., 2015; Lin et al., 2015; Shalev-Shwartz & Zhang, 2014). A number of simpler accelerated schemes have been proposed since then, including (Allen-Zhu, 2017a; Lan & Zhou, 2015), but they do not address the case when each $f_i(x)$ is nonconvex. For more details, see (Allen-Zhu, 2017a) and the references therein.

Relationship to Katyusha. We have borrowed the algorithm name from (Allen-Zhu, 2017a), where the author obtained a direct accelerated stochastic method `Katyusha` for minimizing a function $f(x)$ that is a finite average of *convex* functions $f_i(x)$. The two algorithms are different:

- `Katyusha` applies a momentum step every iteration, but `KatyushaX` applies a momentum step every epoch (i.e., every n iterations).
- `KatyushaX` applies to a more general class of sum-of-nonconvex functions than `Katyusha`.
- `Katyusha` gives a better complexity than `KatyushaX` when restricted to convex $f_i(x)$.

The two works also share some similarity.

- Both works provably and directly add momentum to a stochastic method. This can have practical impacts.
- Both works introduce some "negative momentum" on top of SVRG. In every iteration of `Katyusha`, the

[7] We say a method is gap-free if it does not need an eigengap assumption between the top two eigenvalues.

[8] In this high-level summary, we have hidden the smoothness and variance parameters inside the big-O notion.

point retracts towards the most recent snapshot (this is achieved by a three-point linear coupling (Allen-Zhu, 2017a)). In KatyushaX, after each epoch we applied $x_{k+1} \leftarrow \frac{\frac{3}{2}y_k + \frac{1}{2}x_k - (1-\tau)y_{k-1}}{1+\tau}$ which is different from the classical update $x_{k+1} \leftarrow \frac{2y_k-(1-\tau)y_{k-1}}{1+\tau}$. This can also be viewed as retracting y_k towards the most recent snapshot x_k.

Full Version. For the full and future versions of this paper, see https://arxiv.org/abs/1802.03866.

Acknowledgements

We would like to thank the anonymous referees for useful suggestions on this paper.

References

Agarwal, N., Allen-Zhu, Z., Bullins, B., Hazan, E., and Ma, T. Finding Approximate Local Minima for Nonconvex Optimization in Linear Time. In *STOC*, 2017. Full version available at http://arxiv.org/abs/1611.01146.

Allen-Zhu, Z. Katyusha: The First Direct Acceleration of Stochastic Gradient Methods. In *STOC*, 2017a. Full version available at http://arxiv.org/abs/1603.05953.

Allen-Zhu, Z. Natasha: Faster Non-Convex Stochastic Optimization via Strongly Non-Convex Parameter. In *ICML*, 2017b. Full version available at http://arxiv.org/abs/1702.00763.

Allen-Zhu, Z. How To Make the Gradients Small Stochastically. *ArXiv e-prints*, abs/1801.02982, January 2018. Full version available at http://arxiv.org/abs/1801.02982.

Allen-Zhu, Z. and Hazan, E. Variance Reduction for Faster Non-Convex Optimization. In *ICML*, 2016a. Full version available at http://arxiv.org/abs/1603.05643.

Allen-Zhu, Z. and Hazan, E. Optimal Black-Box Reductions Between Optimization Objectives. In *NIPS*, 2016b.

Allen-Zhu, Z. and Li, Y. LazySVD: Even Faster SVD Decomposition Yet Without Agonizing Pain. In *NIPS*, 2016. Full version available at http://arxiv.org/abs/1607.03463.

Allen-Zhu, Z. and Li, Y. First Efficient Convergence for Streaming k-PCA: a Global, Gap-Free, and Near-Optimal Rate. In *FOCS*, 2017a. Full version available at http://arxiv.org/abs/1607.07837.

Allen-Zhu, Z. and Li, Y. Follow the Compressed Leader: Faster Online Learning of Eigenvectors and Faster MMWU. In *ICML*, 2017b. Full version available at http://arxiv.org/abs/1701.01722.

Allen-Zhu, Z. and Orecchia, L. Linear Coupling: An Ultimate Unification of Gradient and Mirror Descent. In *Proceedings of the 8th Innovations in Theoretical Computer Science*, ITCS '17, 2017. Full version available at http://arxiv.org/abs/1407.1537.

Allen-Zhu, Z. and Yuan, Y. Improved SVRG for Non-Strongly-Convex or Sum-of-Non-Convex Objectives. In *ICML*, 2016.

Carmon, Y., Duchi, J. C., Hinder, O., and Sidford, A. Accelerated Methods for Non-Convex Optimization. *ArXiv e-prints*, abs/1611.00756, November 2016.

Frostig, R., Ge, R., Kakade, S. M., and Sidford, A. Unregularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In *ICML*, 2015.

Garber, D., Hazan, E., Jin, C., Kakade, S. M., Musco, C., Netrapalli, P., and Sidford, A. Robust shift-and-invert preconditioning: Faster and more sample efficient algorithms for eigenvector computation. In *ICML*, 2016.

Jain, P., Jin, C., Kakade, S. M., Netrapalli, P., and Sidford, A. Streaming PCA: Matching Matrix Bernstein and Near-Optimal Finite Sample Guarantees for Oja's Algorithm. In *COLT*, 2016.

Johnson, R. and Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, NIPS 2013, pp. 315–323, 2013.

Lan, G. and Zhou, Y. An optimal randomized incremental gradient method. *ArXiv e-prints*, abs/1507.02000, October 2015.

Lei, L., Ju, C., Chen, J., and Jordan, M. I. Nonconvex Finite-Sum Optimization Via SCSG Methods. In *NIPS*, 2017.

Lin, H., Mairal, J., and Harchaoui, Z. A Universal Catalyst for First-Order Optimization. In *NIPS*, 2015.

Lin, H., Mairal, J., and Harchaoui, Z. Catalyst Acceleration for First-order Convex Optimization: from Theory to Practice. *ArXiv e-prints*, abs/1712.05654, 2017.

Ma, Z., Lu, Y., and Foster, D. Finding linear structure in large datasets with scalable canonical correlation analysis. In *ICML*, pp. 169–178, 2015.

Murata, T. and Suzuki, T. Doubly accelerated stochastic variance reduced dual averaging method for regularized empirical risk minimization. In *NIPS*, 2017.

Nesterov, Y. *Introductory Lectures on Convex Programming Volume: A Basic course*, volume I. Kluwer Academic Publishers, 2004. ISBN 1402075537.

Nesterov, Y. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, December 2005. ISSN 0025-5610. doi: 10.1007/s10107-004-0552-5.

Oja, E. Simplified neuron model as a principal component analyzer. *Journal of mathematical biology*, 15(3):267–273, 1982.

Reddi, S. J., Hefny, A., Sra, S., Poczos, B., and Smola, A. Stochastic variance reduction for nonconvex optimization. In *ICML*, 2016.

Saad, Y. *Numerical methods for large eigenvalue problems*. Manchester University Press, 1992.

Schmidt, M., Le Roux, N., and Bach, F. Minimizing finite sums with the stochastic average gradient. *ArXiv e-prints*, abs/1309.2388, September 2013. Preliminary version appeared in NIPS 2012.

Shalev-Shwartz, S. SDCA without Duality, Regularization, and Individual Convexity. In *ICML*, 2016.

Shalev-Shwartz, S. and Zhang, T. Accelerated Proximal Stochastic Dual Coordinate Ascent for Regularized Loss Minimization. In *Proceedings of the 31st International Conference on Machine Learning*, ICML 2014, pp. 64–72, 2014.

Shamir, O. A Stochastic PCA and SVD Algorithm with an Exponential Convergence Rate. In *ICML*, pp. 144—-153, 2015.

Shamir, O. Fast stochastic algorithms for svd and pca: Convergence properties and convexity. In *ICML*, 2016.

Xiao, L. and Zhang, T. A Proximal Stochastic Gradient Method with Progressive Variance Reduction. *SIAM Journal on Optimization*, 24(4):2057—-2075, 2014.

Zhang, L., Mahdavi, M., and Jin, R. Linear convergence with condition number independent access of full gradients. In *Advances in Neural Information Processing Systems*, pp. 980–988, 2013.