Katyusha: The First Direct Acceleration of Stochastic Gradient Methods

Zeyuan Allen-Zhu
zeyuan@csail.mit.edu
Princeton University / Institute for Advanced Study

March 18, 2016*

Abstract

We introduce `Katyusha`, the first direct, primal-only stochastic gradient method that has a provably accelerated convergence rate in convex optimization. In contrast, previous methods are based on dual coordinate descent which are more restrictive, or based on outer-inner loops which make them "blind" to the underlying stochastic nature of the optimization process. `Katyusha` is the first algorithm that incorporates acceleration directly into stochastic gradient updates.

Unlike previous results, `Katyusha` obtains an optimal convergence rate. It also supports proximal updates, non-Euclidean norm smoothness, non-uniform sampling, and mini-batch sampling. When applied to interesting classes of convex objectives, including smooth objectives (e.g., Lasso, Logistic Regression), strongly-convex objectives (e.g., SVM), and non-smooth objectives (e.g., L1SVM), `Katyusha` improves the best known convergence rates.

The main ingredient behind our result is *Katyusha momentum*, a novel "negative momentum on top of momentum" that can be incorporated into a variance-reduction based algorithm and speed it up. As a result, since variance reduction has been successfully applied to a fast growing list of practical problems, our paper suggests that in each of such cases, one had better hurry up and give Katyusha a hug.

*The first version of this paper appeared on arXiv on this date. The second version in May 2016 included experiments. The third and fourth versions polished writing.

1 Introduction

In large-scale machine learning, the number of data examples is usually very large. To search for the optimal solution, one often uses *stochastic gradient methods* which only require one (or a small batch of) random example(s) per iteration in order to form an *estimator* of the full gradient.

While full-gradient based methods can enjoy an *accelerated* (and optimal) convergence rate if Nesterov's momentum trick is used [31–33], theory for stochastic gradient methods are generally lagging behind and less is known for their acceleration.

At a high level, momentum is *dangerous* if stochastic gradients are present. If some gradient estimator is very inaccurate, then adding it to the momentum and moving further in this direction (for every future iteration) may hurt the convergence performance. In other words, when naively equipped with momentum, stochastic gradient methods are "very prune to error accumulation" [22] and do *not* yield accelerated convergence rates in general.[1]

In this paper, we show that at least for convex optimization purposes, such an issue can be solved with a novel "negative momentum" that can be added on top of momentum. We obtain accelerated and the first optimal convergence rates for stochastic gradient methods, and believe our new insight can deepen our understanding to the theory of accelerated methods.

Problem Definition. Consider the following composite convex minimization problem

$$\min_{x\in\mathbb{R}^d} \Big\{ F(x) \stackrel{\text{def}}{=} f(x) + \psi(x) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^n f_i(x) + \psi(x) \Big\} \ . \tag{1.1}$$

Here, $f(x) = \frac{1}{n}\sum_{i=1}^n f_i(x)$ is a convex function that is a finite average of n convex, smooth functions $f_i(x)$, and $\psi(x)$ is convex, lower semicontinuous (but possibly non-differentiable) function, sometimes referred to as the *proximal* function. We mostly focus on the case when $\psi(x)$ is σ-strongly convex and each $f_i(x)$ is L-smooth. (Both these assumptions can be removed and we shall discuss that later.) We look for approximate minimizers $x \in \mathbb{R}^d$ satisfying $F(x) \le F(x^*) + \varepsilon$, where $x^* \in \arg\min_x \{F(x)\}$.

Problem (1.1) arises in many places in machine learning, statistics, and operations research. All convex *regularized empirical risk minimization (ERM)* problems such as Lasso, SVM, Logistic Regression, fall into this category (see Section 1.2). Efficient stochastic methods for Problem (1.1) also lead to fast algorithms for neural nets [2, 21] as well as SVD, PCA, and CCA [4, 5, 18].

We summarize the history of stochastic gradient methods solving Problem (1.1) into three eras.

The Paleozoic Era: Stochastic Gradient Descent (SGD).

Recall that stochastic gradient methods iteratively perform the following update

$$\text{stochastic gradient iteration:} \qquad x_{k+1} \leftarrow \arg\min_{y\in\mathbb{R}^d} \Big\{ \frac{1}{2\eta}\|y - x_k\|_2^2 + \langle \widetilde{\nabla}_k, y\rangle + \psi(y) \Big\} \ ,$$

where η is the step length and $\widetilde{\nabla}_k$ is a random vector satisfying $\mathbb{E}[\widetilde{\nabla}_k] = \nabla f(x_k)$ and is referred to as the *gradient estimator*. If the proximal function $\psi(y)$ equals zero, the update reduces to $x_{k+1} \leftarrow x_k - \eta\widetilde{\nabla}_k$. A popular choice for the gradient estimator is to set $\widetilde{\nabla}_k = \nabla f_i(x_k)$ for some random index $i \in [n]$ per iteration, and methods based on this choice are known as *stochastic gradient descent (SGD)* [13, 45]. Since computing $\nabla f_i(x)$ is usually n times faster than that of $\nabla f(x)$, SGD enjoys a low per-iteration cost as compared to full-gradient methods; however, SGD cannot converge at a rate faster than $1/\varepsilon$ even if $F(\cdot)$ is strongly convex and smooth.

[1] In practice, experimentalists have observed that momentums could sometimes help if stochastic gradient iterations are used. However, the so-obtained methods (1) sometimes fail to converge in an accelerated rate, (2) become unstable and hard to tune, and (3) have no support theory behind them. See Section 7.1 for an experiment illustrating that.

The Mesozoic Era: Variance Reduction Gives Faster Convergence.

The convergence rate of SGD can be further improved with the so-called variance-reduction technique [12, 15, 21, 29, 30, 38–40, 43, 44]. In these cited results, the authors have shown that SGD converges much faster if one makes a better choice of the gradient estimator $\widetilde{\nabla}_k$ so that its variance reduces as k increases. One way to choose this estimator can be described as follows. Keep a snapshot vector $\widetilde{x} = x_k$ that is updated once every m iterations (where m is some parameter usually around $2n$), and compute the full gradient $\nabla f(\widetilde{x})$ only for such snapshots. Then, set

$$\widetilde{\nabla}_k = \nabla f_i(x_k) - \nabla f_i(\widetilde{x}) + \nabla f(\widetilde{x}) \ . \tag{1.2}$$

This choice of gradient estimator ensures that its variance approaches to zero as k grows. Furthermore, the number of stochastic gradients (i.e., the number of computations of $\nabla f_i(x)$ for some i) required to reach an ε-approximate minimizer of Problem (1.1) is only $O\big(\big(n + \frac{L}{\sigma}\big)\log\frac{1}{\varepsilon}\big)$. Since it is often denoted by $\kappa \stackrel{\text{def}}{=} L/\sigma$ the condition number of the problem, we rewrite the above iteration complexity as $O\big((n+\kappa)\log\frac{1}{\varepsilon}\big)$.

Unfortunately, the iteration complexities of all known variance-reduction based methods have a linear dependence on κ. It was an open question regarding how to obtain

an *accelerated* stochastic gradient method with an optimal $\sqrt{\kappa}$ dependency.

The Cenozoic Era: Acceleration Gives Fastest Convergence.

This open question was partially solved recently by the Catalyst [26] or APPA [17] reductions, both based on an outer-inner loop structure first proposed by Shalev-Shwartz and Zhang [41]. We refer to both of them as Catalyst in this paper. Catalyst solves Problem (1.1) using $O\big(\big(n + \sqrt{n\kappa}\big)\log\kappa\log\frac{1}{\varepsilon}\big)$ stochastic gradient iterations, through a logarithmic number of calls to a variance-reduction method.[2] However, Catalyst is still imperfect for the following reasons:

- Optimality. Catalyst does not match the optimal $\sqrt{\kappa}$ dependence [42] and has an extra $\log\kappa$ factor. For similar reasons, it does not lead to the optimal $1/\sqrt{\varepsilon}$ rate (or equivalently $1/T^2$ rate) if the objective is not strongly convex. It does not lead to the optimal $1/\sqrt{\varepsilon}$ rate (or equivalently $1/T^2$ rate) if the objective is non-smooth. It does not lead to the optimal $1/\varepsilon$ rate (or equivalently $1/T$ rate) if the objective is both non-strongly convex and non-smooth.[3]
- Practicality. Catalyst is not very practical since each of its inner iterations needs to be very accurately executed. This makes the stopping criterion hard to be tuned, and makes Catalyst sometimes run slower than non-accelerated variance-reduction methods [25]. We have also confirmed this in our experiments.
- Generality. Catalyst has a few theoretical limitations for being a reduction-based method. For instance, it does not support non-Euclidean norm smoothness on $f_i(\cdot)$. It cannot be applied to non-convex settings; in contrast, variance reduction has been applied to non-convex objectives (such as training neural nets) both empirically [21] and theoretically [2].

A bit less known is the work of Lan and Zhou [24], where the authors proposed a primal-dual method that also has a $\sqrt{\kappa}\log(\kappa)$ dependency. Their method is subject to the same optimality issue as Catalyst, and requires n times more storage compared with Catalyst for Problem (1.1).

In sum, it is not only desirable but also an open question to develop a *direct* and *primal-only* accelerated stochastic gradient method without using reductions or paying the extra $\log\kappa$ factor. This could have both theoretical and practical impacts to the problems that fall into the general framework of (1.1), and deepen our understanding to acceleration in stochastic settings.

[2] Note that $n + \sqrt{n\kappa}$ is always less than $O(n+\kappa)$.

[3] Obtaining *optimal* rates is one of the main goals of machine learning. For instance, obtaining the optimal $1/T$ rate for online learning was a major breakthrough since the $\log T/T$ rate was discovered [19, 36].

1.1 Our Results and High-Level Ideas

We develop a direct, accelerated stochastic gradient method `Katyusha` for solving Problem (1.1) in

$$O\big(\big(n+\sqrt{n\kappa}\big)\log(1/\varepsilon)\big) \text{ stochastic gradient iterations (see Theorem 3.1)}$$

This gives both optimal dependency on κ and on ε which, to the best of our knowledge, was not obtained before for stochastic gradient methods. In addition, if $F(\cdot)$ is non-strongly convex, `Katyusha` converges to an ε-minimizer in (see Corollary 4.5 and Theorem 5.1)

$$O\big(n\log(1/\varepsilon)+\sqrt{nL/\varepsilon}\cdot\|x_0-x^*\|\big) \text{ stochastic gradient iterations ,}$$

giving an optimal $1/\sqrt{\varepsilon}$ convergence rate. In contrast, Catalyst has a slower $\log^2(1/\varepsilon)/\sqrt{\varepsilon}$ rate.

Our Algorithm. If ignoring the proximal term $\psi(\cdot)$, `Katyusha` iteratively updates:

- $x_{k+1}\leftarrow \tau_1 z_k+\tau_2\widetilde{x}+(1-\tau_1-\tau_2)y_k$;
- $\widetilde{\nabla}_{k+1}\leftarrow \nabla f(\widetilde{x})+\nabla f_i(x_{k+1})-\nabla f_i(\widetilde{x})$ where i is a random index in $[n]$;
- $y_{k+1}\leftarrow x_{k+1}-\frac{1}{3L}\widetilde{\nabla}_{k+1}$, and
- $z_{k+1}\leftarrow z_k-\alpha\widetilde{\nabla}_{k+1}$.

Above, $\widetilde{x}$ is a snapshot point which is updated every n iterations, $\widetilde{\nabla}_{k+1}$ is the gradient estimator defined in the same way as (1.2), $\tau_1,\tau_2\in[0,1]$ are two momentum parameters, and α is a parameter that is equal to $\frac{1}{3\tau_1 L}$. The reason for keeping a sequence of three vectors (x_k,y_k,z_k) is a common ingredient that can be found in all existing accelerated methods.[4]

Our New Technique – Katyusha Momentum. The most surprising part of `Katyusha` is the novel choice of x_{k+1} which is a convex combination of y_k, z_k, and $\widetilde{x}$. Our theory suggests the parameter choices $\tau_2=0.5$ and $\tau_1=\min\{\sqrt{n\sigma/L},0.5\}$ and they work well in practice too. To explain this novel combination, let us recall the classical "momentum" view of accelerated methods.

In a classical accelerated gradient method, x_{k+1} is only a convex combination of y_k and z_k (or equivalently, $\tau_2=0$ in our formulation). At a high level, z_k plays the role of "momentum" which adds a weighted sum of the gradient history into y_{k+1}. As an illustrative example, suppose $\tau_2=0$, $\tau_1=\tau$, and $x_0=y_0=z_0$. Then, one can compute that

$$y_k=\begin{cases} x_0-\frac{1}{3L}\widetilde{\nabla}_1, & k=1;\\ x_0-\frac{1}{3L}\widetilde{\nabla}_2-\big((1-\tau)\frac{1}{3L}+\tau\alpha\big)\widetilde{\nabla}_1, & k=2;\\ x_0-\frac{1}{3L}\widetilde{\nabla}_3-\big((1-\tau)\frac{1}{3L}+\tau\alpha\big)\widetilde{\nabla}_2-\big((1-\tau)^2\frac{1}{3L}+(1-(1-\tau)^2)\alpha\big)\widetilde{\nabla}_1, & k=3.\end{cases}$$

Since α is usually much larger than $1/3L$, the above recursion suggests that the contribution of a fixed gradient $\widetilde{\nabla}_t$ gradually increases as time goes. For instance, the weight on $\widetilde{\nabla}_1$ is increasing because $\frac{1}{3L}<\big((1-\tau)\frac{1}{3L}+\tau\alpha\big)<\big((1-\tau)^2\frac{1}{3L}+(1-(1-\tau)^2)\alpha\big)$. This is known as "momentum" which is at the heart of all accelerated first-order methods.

In `Katyusha`, we put a "magnet" around $\widetilde{x}$, where we choose $\widetilde{x}$ to be essentially "the average x_t of the most recent n iterations". Whenever we compute the next x_{k+1}, it will be attracted by the magnet $\widetilde{x}$ with weight $\tau_2=0.5$. This is a strong magnet: it ensures that x_{k+1} is not too far away from $\widetilde{x}$ so the gradient estimator remains "accurate enough". This can be viewed as a "negative momentum" component, because the magnet retracts x_{k+1} back to $\widetilde{x}$ and this can be understood as "counteracting a fraction of the positive momentum incurred from earlier iterations."

We call it the Katyusha momentum.

This summarizes the high-level idea behind our `Katyusha` method. We remark here if $\tau_1=\tau_2=0$, `Katyusha` becomes almost identical to SVRG [21] which is a variance-reduction based method.

[4] One can of course rewrite the algorithm and keep track of only two vectors per iteration during implementation. This will make the algorithm statement less clean so we refrain from doing so in this paper.

1.2 Applications: Optimal Rates for Empirical Risk Minimization

Suppose we are given n feature vectors $a_1, \dots, a_n \in \mathbb{R}^d$ corresponding to n data samples. Then, the *empirical risk minimization (ERM)* problem is to study Problem (1.1) when each $f_i(x)$ is "rank-one" structured: that is, $f_i(x) \stackrel{\text{def}}{=} g_i(\langle a_i, x\rangle)$ for some loss function $g_i \colon \mathbb{R} \to \mathbb{R}$. Slightly abusing notation, we also write $f_i(x) = f_i(\langle a_i, x\rangle)$. (Assuming "rank-one" simplifies the notations; all of the results stated in this subsection generalize to constant-rank structured functions $f_i(x)$.)

In such a case, Problem (1.1) becomes as

$$\text{ERM:} \quad \min_{x\in\mathbb{R}^d} \left\{ F(x) \stackrel{\text{def}}{=} f(x) + \psi(x) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^n f_i(\langle a_i, x\rangle) + \psi(x) \right\} . \tag{1.3}$$

Without loss of generality, we assume each a_i has norm 1 because otherwise one can scale $f_i(\cdot)$ accordingly. As summarized for instance in [1], there are four interesting cases of ERM problems, and all of them can be trivially written in the form of (1.3):

Case 1: $\psi(x)$ is σ strongly convex and $f_i(x)$ is L-smooth. Examples: ridge regression, elastic net;

Case 2: $\psi(x)$ is non-strongly convex and $f_i(x)$ is L-smooth. Examples: Lasso, logistic regression;

Case 3: $\psi(x)$ is σ strongly convex and $f_i(x)$ is non-smooth. Examples: support vector machine;

Case 4: $\psi(x)$ is non-strongly convex and $f_i(x)$ is non-smooth. Examples: ℓ_1-SVM.

Known Results. For all of the four ERM cases above, accelerated stochastic methods were introduced in the literature, most notably AccSDCA [41], APCG [27], SPDC [46]. However, all known accelerated methods have suboptimal convergence rates for Case 2, 3 and 4.[5] In particular, the best known convergence rate was $\frac{\log(1/\varepsilon)}{\sqrt{\varepsilon}}$, $\frac{\log(1/\varepsilon)}{\sqrt{\varepsilon}}$, and $\frac{\log(1/\varepsilon)}{\varepsilon}$ respectively for Case 2, 3, and 4, and this is a factor $\log(1/\varepsilon)$ worse than the optimal rate for each of the three classes [42].

It is an open question also in the optimization community to design a stochastic gradient method with optimal convergence for such problems. In particular, Dang and Lan [14] provided an interesting attempt to remove such log factors but using a non-classical notion of convergence.[6]

Besides the log factor loss in the running time,[7] the aforementioned methods are dual-based and therefore suffer from several other issues. First, they only apply to ERM problems but not to the more general Problem (1.1). Second, they require proximal updates with respect to the Fenchel conjugate $f_i^*(\cdot)$ which is sometimes unpleasant to work with. Third, their performances cannot benefit from the implicit strong convexity in $f(\cdot)$. All of these issues together make dual-based accelerated methods sometimes even outperformed by primal-only non-accelerated ones, such as SAGA [15] or SVRG [21, 44].

Our Results. `Katyusha` simultaneously closes the gap for all of the three classes of problems with the help from the optimal reductions developed by Allen-Zhu and Hazan [1]. We obtain an ε-approximate minimizer for Case 2 in $O\big(n\log\frac{1}{\varepsilon} + \frac{\sqrt{nL}}{\sqrt{\varepsilon}}\big)$ iterations, for Case 3 in $O\big(n\log\frac{1}{\varepsilon} + \frac{\sqrt{n}}{\sqrt{\sigma\varepsilon}}\big)$ iterations, and for Case 4 in $O\big(n\log\frac{1}{\varepsilon} + \frac{\sqrt{n}}{\varepsilon}\big)$ iterations. In contrast, none of the existing accelerated methods can lead to such optimal rates even if the optimal reductions of [1] are used.

[5] In fact, they also have the suboptimal dependence on the condition number L/σ for Case 1.

[6] Dang and Lan work in a primal-dual $\phi(x,y)$ formulation of Problem (1.1), and produce a primal-dual pair (x,y) so that for every fixed (u,v), the expectation $\mathbb{E}[\phi(x,v) - \phi(u,y)] \le \varepsilon$. Unfortunately, to ensure x is an ε-approximate minimizer of Problem (1.1), one needs the stronger $\mathbb{E}[\max_{(u,v)} \phi(x,v) - \phi(u,y)] \le \varepsilon$ to hold.

[7] In fact, dual-based methods have to suffer from a log factor loss in the convergence rate. This is so because even for Case 1 of Problem (1.3), converting an ε-maximizer for the dual objective to the primal, one only obtains an $n\kappa\varepsilon$-minimizer on the primal objective. As a result, algorithms like APCG who directly work on the dual, algorithms like SPDC who maintain both primal and dual variables, and algorithms like RPDG [24] that are primal-like but still use dual analysis, have to suffer from a log loss in the convergence rates.

After this paper appeared on arXiv, Woodworth and Srebro [42] proved the tightness of our results. They showed lower bounds $\Omega\big(n+\frac{\sqrt{nL}}{\sqrt{\varepsilon}}\big)$, $\Omega\big(n+\frac{\sqrt{n}}{\sqrt{\sigma\varepsilon}}\big)$, and $\Omega\big(n+\frac{\sqrt{n}}{\varepsilon}\big)$ for Cases 2, 3, and 4 respectively at least for small ε.[8]

1.3 Other Extensions

Mini-batch. `Katyusha` trivially extends to minibatch scenarios. Instead of using a single $\nabla f_i(\cdot)$ per iteration, one can use the average of b stochastic gradients $\frac{1}{b}\sum_{j\in S}\nabla f_i(\cdot)$ where S is a random subset of $[n]$ with cardinality b. Our theorems extend to this setting, where the only change needed is to re-compute the snapshot every n/b iterations rather than every n iterations.

Non-Uniform Sampling. If each $f_i(\cdot)$ has a different smooth parameter L_i, one can select the random index i from a non-uniform distribution in order to obtain an even faster running time. This can be done using the same techniques proposed in [11], and we include the details in Section 6.1.

Non-Euclidean Norms. If the smoothness of each $f_i(x)$ is with respect to a non-Euclidean norm (such as the well known ℓ_1 norm case over the simplex), our results in this paper still hold. Our update on the y_{k+1} side becomes the non-Euclidean norm gradient descent, and our update on the z_{k+1} side becomes the non-Euclidean norm mirror descent. We include such details in Section 6.2. In contrast, to the best of our knowledge, Catalyst, AccSDCA and APCG do not work with non-Euclidean norms. SPDC can be revised to work with non-Euclidean norms, see [7].

Katyusha Momentum Weight. To provide the simplest proof, we choose $\tau_2=1/2$ which also works well in practice. Our same proof trivially generalizes to all constant values of $\tau_2\in(0,1)$ and it could be beneficial to tune it for different datasets in practice. However, for a stronger comparison, we refrain from doing so in this paper: by fixing $\tau_2=1/2$ and thus without increasing parameter tuning difficulties, `Katyusha` can already beat most of the state-of-the-arts.

1.4 Related Work

For smooth convex minimization problems, (full) gradient descent converges at a rate $\frac{L}{\varepsilon}$ —or $\frac{L}{\sigma}\log\frac{1}{\varepsilon}$ if the objective is σ-strongly convex. This is not optimal among the class of first-order methods. Nesterov showed that the optimal rate should be $\frac{\sqrt{L}}{\sqrt{\varepsilon}}$ —or $\frac{\sqrt{L}}{\sqrt{\sigma}}\log\frac{1}{\varepsilon}$ if the objective is σ-strongly convex— and this was achieved by his celebrated accelerated (full) gradient descent method [31].

Randomized Coordinate Descent. Another way to define gradient estimator is to set $\widetilde{\nabla}_k = d\nabla_j f(x_k)$ where j is a random coordinate. This is *(randomized) coordinate descent* as opposed to stochastic gradient descent. Designing accelerated methods for coordinate descent is significantly easier than designing that for stochastic gradient descent, and has indeed been done in many previous results including [11, 27, 28, 34].[9] The state-of-the-art accelerated coordinate descent method is NUACDM [11]. Coordinate descent *cannot* be applied to solve Problem (1.1) because in our setting, only one copy $\nabla f_i(\cdot)$ is computed in a stochastic iteration.

Hybrid Accelerated and Stochastic Methods. Several recent results study hybrid methods with convergence rates that are generally *non-accelerated* and only accelerated in *extreme cases*.

[8] More precisely, their lower bounds for Cases 3 and 4 are $\Omega\big(\min\big\{\frac{1}{\sigma\varepsilon}, n+\frac{\sqrt{n}}{\sqrt{\sigma\varepsilon}}\big\}\big)$ and $\Omega\big(\min\big\{\frac{1}{\varepsilon^2}, n+\frac{\sqrt{n}}{\varepsilon}\big\}\big)$. However, since the vanilla SGD requires $O(\frac{1}{\sigma\varepsilon})$ and $O(\frac{1}{\varepsilon^2})$ iterations for Cases 3 and 4, such lower bounds are matched by combining the best between `Katyusha` and SGD.

[9] The reason behind it can be understood as follows. If a function $f(\cdot)$ is L smooth with respect to coordinate j, then a coordinate descent step $x'\leftarrow x-\frac{1}{L}\nabla_j f(x)\mathbf{e}_j$ always decreases the objective, i.e., $f(x+\frac{1}{L}\nabla_j f(x)\mathbf{e}_j)<f(x)$. In contrast, this is *false* for stochastic gradient descent, because $f(x_k-\eta\widetilde{\nabla}_k)$ may be even larger than $f(x_k)$.

The authors of [20, 23] obtained running time of the form $O(L/\sqrt{\varepsilon} + \sigma/\varepsilon^2)$ in the presence of stochastic gradient with variance σ. While the first term $L/\sqrt{\varepsilon}$ is an accelerated rate (for non-strongly convex but smooth functions), the second term is non-accelerated. For Problem (1.1), these algorithms do not give faster running time than `Katyusha` unless σ is very very small.

Nitanda's method adds momentum to the non-accelerated variance-reduction method in a naive manner [35] and thus corresponds to this paper but *without* Katyusha momentum (i.e., $\tau_2 = 0$). The theoretical running time of [35] is always slower than this paper and cannot even outperform SVRG [21] unless $\kappa > n^2$ —which is usually *false* in practice (see page 7 of [35]).[10] We have anyways included an experiment in Section 7.1 to illustrate why Katyusha momentum is necessary.

Linear Coupling. In a recent work by Allen-Zhu and Orecchia, the authors have proposed a new framework called *linear coupling* that facilitates the design of accelerated gradient methods [8]. Their new framework not only reconstructs Nesterov's accelerated (full-)gradient method [8], provides even faster accelerated coordinate descent method [11], but also leads to many recent breakthroughs for designing accelerated methods on non-smooth problems (such as positive LP [9, 10] and positive SDP [3]) or even general non-convex problems [2]. This present paper also falls into this linear-coupling framework.

1.5 Roadmap

- In Section 2, we provide necessary notations and useful preliminaries .
- In Section 3, we state and prove our theorem on `Katyusha` for the strongly convex case.
- In Section 4, we apply `Katyusha` to non-strongly convex or non-smooth cases *using reductions*.
- In Section 5, we provide a *direct* algorithm for the non-strongly case.
- In Section 6, we adapt our algorithms and theorems to work in extended settings.
- In Section 7, we provide an empirical evaluation to illustrate the necessity of Katyusha momentum, and the practical performance of `Katyusha` comparing to the start-of-the-arts.

2 Preliminaries

Throughout this paper (except Section 6), we denote by $\|\cdot\|$ the Euclidean norm. We denote by $\nabla f(x)$ the full gradient of function f if it is differentiable, or the subgradient if f is only Lipschitz continuous. Recall some classical definitions on strong convexity (SC) and smoothness.

Definition 2.1 (smoothness and strong convexity)**.** *For a convex function $f\colon \mathbb{R}^n \to \mathbb{R}$,*
- *f is σ-strongly convex if $\forall x, y \in \mathbb{R}^n$, it satisfies $f(y) \geq f(x) + \langle \nabla f(x), y - x\rangle + \frac{\sigma}{2}\|x - y\|^2$.*
- *f is L-smooth if $\forall x, y \in \mathbb{R}^n$, it satisfies $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$.*

We also need to use the following definition of the HOOD property:

Definition 2.2 ([1])**.** *An algorithm solving the strongly convex case of Problem (1.1) satisfies the* homogenous objective decrease (HOOD) *property with time $\mathsf{Time}(L, \sigma)$, if for every starting point x_0, it produces an output x' satisfying $\mathbb{E}\big[F(x')\big] - F(x^*) \leq \frac{F(x_0) - F(x^*)}{4}$ in time at most $\mathsf{Time}(L, \sigma)$.*

The authors of [1] designed three reductions AdaptReg, AdaptSmooth, and JointAdaptRegSmooth to convert an algorithm satisfying the HOOD property to solve the following three cases:

[10] Nitanda's method is usually not considered as an accelerated method, since it requires mini-batch size to be very large in order to be accelerated. If mini-batch is large then one can use full-gradient method directly and acceleration is trivial there. This is also confirmed by the authors of [22] at the beginning of their Section IV.F. In contrast, our acceleration in this paper holds even if mini-batch size is 1.

Theorem 2.3. *Given an algorithm satisfying* HOOD *with* $\mathsf{Time}(L,\sigma)$ *and a starting vector* x_0.

- NonSC+Smooth. *For Problem (1.1) where* $f(\cdot)$ *is* L*-smooth,* AdaptReg *outputs* x *satisfying* $\mathbb{E}\big[F(x)\big] - F(x^*) \le O(\varepsilon)$ *in time*

$$\textstyle\sum_{t=0}^{T-1} \mathsf{Time}(L, \frac{\sigma_0}{2^t}) \text{ where } \sigma_0 = \frac{F(x_0)-F(x^*)}{\|x_0-x^*\|^2} \text{ and } T = \log_2 \frac{F(x_0)-F(x^*)}{\varepsilon}.$$

- SC+NonSmooth. *For Problem (1.3) where* $\psi(\cdot)$ *is* σ*-SC and each* $f_i(\cdot)$ *is* G*-Lipschitz continuous,* AdaptSmooth *outputs* x *satisfying* $\mathbb{E}\big[F(x)\big] - F(x^*) \le O(\varepsilon)$ *in time*

$$\textstyle\sum_{t=0}^{T-1} \mathsf{Time}(\frac{2^t}{\lambda_0}, \sigma) \text{ where } \lambda_0 = \frac{F(x_0)-F(x^*)}{G^2} \text{ and } T = \log_2 \frac{F(x_0)-F(x^*)}{\varepsilon}.$$

- NonSC+NonSmooth. *For Problem (1.3) where each* $f_i(\cdot)$ *is* G*-Lipschitz continuous, then* JointAdaptRegSmooth *outputs* x *satisfying* $\mathbb{E}\big[F(x)\big] - F(x^*) \le O(\varepsilon)$ *in time*

$$\textstyle\sum_{t=0}^{T-1} \mathsf{Time}(\frac{2^t}{\lambda_0}, \frac{\sigma_0}{2^t}) \text{ where } \lambda_0 = \frac{F(x_0)-F(x^*)}{G^2},\ \sigma_0 = \frac{F(x_0)-F(x^*)}{\|x_0-x^*\|^2} \text{ and } T = \log_2 \frac{F(x_0)-F(x^*)}{\|x_0-x^*\|^2}.$$

We shall verify in later that `Katyusha` satisfies HOOD so the above reductions can be applied.

3 Katyusha in the Strongly Convex Setting

We formally introduce our `Katyusha` algorithm in Algorithm 1. It follows from our high-level description in Section 1.1, and we make several remarks here behind our specific design.

- `Katyusha` *is divided into epochs each consisting of* m *iterations. In theory,* m *can be anything linear in* n*. We let snapshot* $\widetilde{x}$ *be a weighted average of* y_k *in the most recent epoch.*

  $\widetilde{x}$ and $\widetilde{\nabla}_k$ correspond to a standard design on variance-reduced gradient estimators.[11] The practical recommendation is $m = 2n$ [21]. Our choice $\widetilde{\nabla}_k$ is independent from our acceleration techniques, and we expect our result continues to apply to other choices of gradient estimators.

- τ_1 *and* α *are standard parameters already present in Nesterov's full-gradient method [8].*

  We choose $\alpha = 1/3\tau_1 L$ to present the simplest proof, and recall it was $\alpha = 1/\tau_1 L$ in the original Nesterov's full-gradient method.[12] In practice, like other accelerated methods, it suffices to fix $\alpha = 1/3\tau_1 L$ and only tune τ_1 (usually known as the learning rate).

- *The parameter* τ_2 *is our novel weight for the Katyusha momentum. Any constant in* $(0,1)$ *works for* τ_2*, and we simply choose* $\tau_2 = 1/2$ *for our theoretical and experimental results.*

We state our main theorem for `Katyusha` as follows:

Theorem 3.1. *If each* $f_i(x)$ *is convex,* L*-smooth, and* $\psi(x)$ *is* σ*-strongly convex in Problem (1.1), then* $\texttt{Katyusha}(x_0, S, \sigma, L)$ *satisfies*

$$\mathbb{E}\big[F(\widetilde{x}^S)\big] - F(x^*) \le \begin{cases} O\Big(\big(1+\sqrt{\sigma/(3Lm)}\big)^{-Sm}\Big) \cdot \big(F(x_0) - F(x^*)\big), & \text{if } \frac{m\sigma}{L} \le \frac{3}{4}; \\ O\big(1.5^{-S}\big) \cdot \big(F(x_0) - F(x^*)\big), & \text{if } \frac{m\sigma}{L} > \frac{3}{4}. \end{cases}$$

In other words, choosing $m = \Theta(n)$, `Katyusha` *achieves an* ε*-additive error (i.e.,* $\mathbb{E}\big[F(\widetilde{x}^S)\big] - F(x^*) \le \varepsilon$*) using at most* $O\big(\big(n+\sqrt{nL/\sigma}\big) \cdot \log \frac{F(x_0)-F(x^*)}{\varepsilon}\big)$ *iterations.*[13]

[11] That is, the SVRG estimator [21]. We choose $\widetilde{x}$ to be a weighted average (rather than the last iterate) of the most recent epoch because it was reported to work better in practice [12].

[12] Any α that is constant factor smaller than $1/\tau_1 L$ works in theory, and we use $1/3$ to provide the simplest proof.

Algorithm 1 `Katyusha`(x_0, S, σ, L)

1: $m \leftarrow 2n$; ◇ *epoch length*
2: $\tau_2 \leftarrow \frac{1}{2}$, $\tau_1 \leftarrow \min\{\frac{\sqrt{m\sigma}}{\sqrt{3L}}, \frac{1}{2}\}$, $\alpha \leftarrow \frac{1}{3\tau_1 L}$; ◇ *parameters*
3: $y_0 = z_0 = \widetilde{x}^0 \leftarrow x_0$; ◇ *initial vectors*
4: **for** $s \leftarrow 0$ **to** $S-1$ **do**
5: $\quad \mu^s \leftarrow \nabla f(\widetilde{x}^s)$; ◇ *compute the full gradient once every m iterations*
6: $\quad$ **for** $j \leftarrow 0$ **to** $m-1$ **do**
7: $\quad\quad k \leftarrow (sm)+j$;
8: $\quad\quad x_{k+1} \leftarrow \tau_1 z_k + \tau_2 \widetilde{x}^s + (1-\tau_1-\tau_2) y_k$;
9: $\quad\quad \widetilde{\nabla}_{k+1} \leftarrow \mu^s + \nabla f_i(x_{k+1}) - \nabla f_i(\widetilde{x}^s)$ where i is random from $\{1, 2, \ldots, n\}$;
10: $\quad\quad z_{k+1} = \arg\min_z \{\frac{1}{2\alpha}\|z - z_k\|^2 + \langle \widetilde{\nabla}_{k+1}, z\rangle + \psi(z)\}$;
11: $\quad\quad$ Option I: $y_{k+1} \leftarrow \arg\min_y \{\frac{3L}{2}\|y - x_{k+1}\|^2 + \langle \widetilde{\nabla}_{k+1}, y\rangle + \psi(y)\}$;
12: $\quad\quad$ Option II: $y_{k+1} \leftarrow x_{k+1} + \tau_1(z_{k+1} - z_k)$ ◇ *we analyze only I but II also works*
13: $\quad$ **end for**
14: $\quad \widetilde{x}^{s+1} \leftarrow \big(\sum_{j=0}^{m-1}(1+\alpha\sigma)^j\big)^{-1} \cdot \big(\sum_{j=0}^{m-1}(1+\alpha\sigma)^j \cdot y_{sm+j+1}\big)$; ◇ *compute snapshot $\widetilde{x}$*
15: **end for**
16: **return** $\widetilde{x}^S$.

The proof of Theorem 3.1 is included in Section 3.1 and 3.2. As discussed in Section 1.1, the main idea behind our theorem is the negative momentum that helps reduce the error occurred from the stochastic gradient estimator.

Remark 3.2. Because $m = 2n$, each iteration of `Katyusha` computes only 1.5 stochastic gradients $\nabla f_i(\cdot)$ in the amortized sense, the same as non-accelerated methods such as SVRG [21]. Therefore, the per-iteration cost of `Katyusha` is dominated by the computation of $\nabla f_i(\cdot)$, the proximal update in Line 10 of Algorithm 1, plus an overhead $O(d)$. If $\nabla f_i(\cdot)$ has at most $d' \le d$ non-zero entries, this overhead $O(d)$ is improvable to $O(d')$ using a sparse implementation of `Katyusha`.[14]

For ERM problems defined in Problem (1.3), the amortized per-iteration complexity of `Katyusha` is $O(d')$ where d' is the sparsity of feature vectors, the same as the per-iteration complexity of SGD.

3.1 One-Iteration Analysis

In this subsection, we first analyze the behavior of `Katyusha` in a single iteration (i.e., for a fixed k). We view y_k, z_k and x_{k+1} as fixed in this section so the only randomness comes from the choice of i in iteration k. We abbreviate in this subsection by $\widetilde{x} = \widetilde{x}^s$ where s is the epoch that iteration k belongs to, and denote by $\sigma_{k+1}^2 \stackrel{\text{def}}{=} \|\nabla f(x_{k+1}) - \widetilde{\nabla}_{k+1}\|^2$ so $\mathbb{E}[\sigma_{k+1}]$ is the variance of the gradient estimator $\widetilde{\nabla}_{k+1}$ in this iteration.

Our first lemma lower bounds the expected objective decrease $F(x_{k+1}) - \mathbb{E}[F(y_{k+1})]$. Our $\mathsf{Prog}(x_{k+1})$ defined below is a non-negative, classical quantity that would be a lower bound on the amount of objective decrease if $\widetilde{\nabla}_{k+1}$ were equal to $\nabla f(x_{k+1})$ [8]. However, since the variance σ_{k+1}^2 is non-zero, this lower bound must be compensated by a negative term that depends on $\mathbb{E}[\sigma_{k+1}^2]$.

[13] Like in all stochastic first-order methods, one can apply a Markov inequality to conclude that with probability at least 2/3, `Katyusha` satisfies $F(\widetilde{x}^S) - F(x^*) \le \varepsilon$ in the same stated asymptotic running time.

[14] This requires to defer a coordinate update to the moment it is accessed. Update deferral is a standard technique used in sparse implementations of all stochastic gradient methods, including SVRG, SAGA, APCG [15, 21, 27].

Lemma 3.3 (proximal gradient descent)**.** *If*

$$y_{k+1} = \arg\min_y \big\{ \frac{3L}{2}\|y - x_{k+1}\|^2 + \langle \widetilde{\nabla}_{k+1}, y - x_{k+1}\rangle + \psi(y) - \psi(x_{k+1}) \big\} \ , \qquad \textit{and}$$

$$\mathsf{Prog}(x_{k+1}) \stackrel{\text{def}}{=} -\min_y \big\{ \frac{3L}{2}\|y - x_{k+1}\|^2 + \langle \widetilde{\nabla}_{k+1}, y - x_{k+1}\rangle + \psi(y) - \psi(x_{k+1}) \big\} \geq 0 \ ,$$

we have (where the expectation is only over the randomness of $\widetilde{\nabla}_{k+1}$)

$$F(x_{k+1}) - \mathbb{E}\big[F(y_{k+1})\big] \geq \mathbb{E}\big[\mathsf{Prog}(x_{k+1})\big] - \frac{1}{4L}\mathbb{E}\big[\sigma_{k+1}^2\big] \ .$$

Proof.

$$\begin{aligned}
\mathsf{Prog}(x_{k+1}) &= -\min_y \Big\{ \frac{3L}{2}\|y - x_{k+1}\|^2 + \langle \widetilde{\nabla}_{k+1}, y - x_{k+1}\rangle + \psi(y) - \psi(x_{k+1}) \Big\} \\
&\stackrel{①}{=} -\Big(\frac{3L}{2}\|y_{k+1} - x_{k+1}\|^2 + \langle \widetilde{\nabla}_{k+1}, y_{k+1} - x_{k+1}\rangle + \psi(y_{k+1}) - \psi(x_{k+1}) \Big) \\
&= -\Big(\frac{L}{2}\|y_{k+1} - x_{k+1}\|^2 + \langle \nabla f(x_{k+1}), y_{k+1} - x_{k+1}\rangle + \psi(y_{k+1}) - \psi(x_{k+1}) \Big) \\
&\quad + \Big(\langle \nabla f(x_{k+1}) - \widetilde{\nabla}_{k+1}, y_{k+1} - x_{k+1}\rangle - L\|y_{k+1} - x_{k+1}\|^2 \Big) \\
&\stackrel{②}{\leq} -\Big(f(y_{k+1}) - f(x_{k+1}) + \psi(y_{k+1}) - \psi(x_{k+1}) \Big) + \frac{1}{4L}\|\nabla f(x_{k+1}) - \widetilde{\nabla}_{k+1}\|^2 \ .
\end{aligned}$$

Above, ① is by the definition of y_{k+1}, and ② uses the smoothness of function $f(\cdot)$, as well as Young's inequality $\langle a, b\rangle - \frac{1}{2}\|b\|^2 \leq \frac{1}{2}\|a\|^2$. Taking expectation on both sides we arrive at the desired result. □

The following lemma provides a novel upper bound on the expected variance of the gradient estimator. Note that all known variance reduction analysis for convex optimization, in one way or another, upper bounds this variance essentially by $4L \cdot (f(\widetilde{x}) - f(x^*))$, the objective distance to the minimizer (c.f. [15, 21]). The recent breakthrough of Allen-Zhu and Hazan [1] upper bounds it by the point distance $\|x_{k+1} - \widetilde{x}\|^2$ for non-convex objectives, which is tighter if $\widetilde{x}$ is close to x_{k+1} but unfortunately not enough for the purpose of this paper.

In this paper, we upper bound it by the tightest possible quantity which is essentially $2L \cdot \big(f(\widetilde{x}) - f(x_{k+1})\big) \ll 4L \cdot \big(f(\widetilde{x}) - f(x^*)\big)$. Unfortunately, this upper bound needs to be compensated by an additional term $\langle \nabla f(x_{k+1}), \widetilde{x} - x_{k+1}\rangle$, which could be positive but we shall cancel it using the introduced Katyusha momentum.

Lemma 3.4 (variance upper bound)**.**

$$\mathbb{E}\big[\|\widetilde{\nabla}_{k+1} - \nabla f(x_{k+1})\|^2\big] \leq 2L \cdot \big(f(\widetilde{x}) - f(x_{k+1}) - \langle \nabla f(x_{k+1}), \widetilde{x} - x_{k+1}\rangle\big) \ .$$

Proof. Each $f_i(x)$, being convex and L-smooth, implies the following inequality which is classical in convex optimization and can be found for instance in Theorem 2.1.5 of the textbook of Nesterov [32].

$$\|\nabla f_i(x_{k+1}) - \nabla f_i(\widetilde{x})\|^2 \leq 2L \cdot \big(f_i(\widetilde{x}) - f_i(x_{k+1}) - \langle \nabla f_i(x_{k+1}), \widetilde{x} - x_{k+1}\rangle\big)$$

Therefore, taking expectation over the random choice of i, we have

$$\begin{aligned}
\mathbb{E}\big[\|\widetilde{\nabla}_{k+1} - \nabla f(x_{k+1})\|^2\big] &= \mathbb{E}\big[\big\|\big(\nabla f_i(x_{k+1}) - \nabla f_i(\widetilde{x})\big) - \big(\nabla f(x_{k+1}) - \nabla f(\widetilde{x})\big)\big\|^2\big] \\
&\stackrel{①}{\leq} \mathbb{E}\big[\big\|\nabla f_i(x_{k+1}) - \nabla f_i(\widetilde{x})\big\|^2\big] \\
&\stackrel{②}{\leq} 2L \cdot \mathbb{E}\big[f_i(\widetilde{x}) - f_i(x_{k+1}) - \langle \nabla f_i(x_{k+1}), \widetilde{x} - x_{k+1}\rangle\big] \\
&= 2L \cdot \big(f(\widetilde{x}) - f(x_{k+1}) - \langle \nabla f(x_{k+1}), \widetilde{x} - x_{k+1}\rangle\big) \ .
\end{aligned}$$

Above, ① is because for any random vector $\zeta \in \mathbb{R}^d$, it holds that $\mathbb{E}\|\zeta - \mathbb{E}\zeta\|^2 = \mathbb{E}\|\zeta\|^2 - \|\mathbb{E}\zeta\|^2$; ② follows from the first inequality in this proof. □

The next lemma is a classical one for proximal mirror descent.

Lemma 3.5 (proximal mirror descent)**.** *Suppose $\psi(\cdot)$ is σ-SC. Then, fixing $\widetilde{\nabla}_{k+1}$ and letting*

$$z_{k+1} = \arg\min_z \big\{ \frac{1}{2}\|z - z_k\|^2 + \alpha\langle \widetilde{\nabla}_{k+1}, z - z_k\rangle + \alpha\psi(z) - \alpha\psi(z_k)\big\} \ ,$$

it satisfies for all $u \in \mathbb{R}^d$,

$$\alpha\langle \widetilde{\nabla}_{k+1}, z_{k+1} - u\rangle + \alpha\psi(z_{k+1}) - \alpha\psi(u) \le -\frac{1}{2}\|z_k - z_{k+1}\|^2 + \frac{1}{2}\|z_k - u\|^2 - \frac{1+\alpha\sigma}{2}\|z_{k+1} - u\|^2 \ .$$

Proof. By the minimality definition of z_{k+1}, we have that

$$z_{k+1} - z_k + \alpha\widetilde{\nabla}_{k+1} + \alpha g = 0$$

where g is *some* subgradient of $\psi(z)$ at point $z = z_{k+1}$. This implies that for every u it satisfies

$$0 = \big\langle z_{k+1} - z_k + \alpha\widetilde{\nabla}_{k+1} + \alpha g, z_{k+1} - u\big\rangle \ .$$

At this point, using the equality $\langle z_{k+1} - z_k, z_{k+1} - u\rangle = \frac{1}{2}\|z_k - z_{k+1}\|^2 - \frac{1}{2}\|z_k - u\|^2 + \frac{1}{2}\|z_{k+1} - u\|^2$, as well as the inequality $\langle g, z_{k+1} - u\rangle \ge \psi(z_{k+1}) - \psi(u) + \frac{\sigma}{2}\|z_{k+1} - u\|^2$ which comes from the strong convexity of $\psi(\cdot)$, we can write

$$\begin{aligned}
&\alpha\langle \widetilde{\nabla}_{k+1}, z_{k+1} - u\rangle + \alpha\psi(z_{k+1}) - \alpha\psi(u)\\
&= -\langle z_{k+1} - z_k, z_{k+1} - u\rangle - \langle \alpha g, z_{k+1} - u\rangle + \alpha\psi(z_{k+1}) - \alpha\psi(u)\\
&\le -\frac{1}{2}\|z_k - z_{k+1}\|^2 + \frac{1}{2}\|z_k - u\|^2 - \frac{1+\alpha\sigma}{2}\|z_{k+1} - u\|^2 \ .
\end{aligned}$$

□

The following lemma combines Lemma 3.3, Lemma 3.4 and Lemma 3.5 all together, using the special choice of x_{k+1} which is a convex combination of y_k, z_k and $\widetilde{x}$:

Lemma 3.6 (coupling step 1)**.** *If $x_{k+1} = \tau_1 z_k + \tau_2\widetilde{x} + (1 - \tau_1 - \tau_2)y_k$, where $\tau_1 \le \frac{3}{\alpha L}$ and $\tau_2 = \frac{1}{2}$,*

$$\begin{aligned}
&\alpha\langle \nabla f(x_{k+1}), z_k - u\rangle - \alpha\psi(u)\\
&\le \frac{\alpha}{\tau_1}\Big(F(x_{k+1}) - \mathbb{E}\big[F(y_{k+1})\big] + \tau_2 F(\widetilde{x}) - \tau_2\mathbb{E}\big[F(x_{k+1})\big] - \tau_2\langle \nabla f(x_{k+1}), \widetilde{x} - x_{k+1}\rangle\Big)\\
&\quad + \frac{1}{2}\|z_k - u\|^2 - \frac{1+\alpha\sigma}{2}\mathbb{E}\big[\|z_{k+1} - u\|^2\big] + \frac{\alpha(1 - \tau_1 - \tau_2)}{\tau_1}\psi(y_k) - \frac{\alpha}{\tau_1}\psi(x_{k+1}) \ .
\end{aligned}$$

Proof. We first apply Lemma 3.5 and get

$$\begin{aligned}
&\alpha\langle \widetilde{\nabla}_{k+1}, z_k - u\rangle + \alpha\psi(z_{k+1}) - \alpha\psi(u)\\
&= \alpha\langle \widetilde{\nabla}_{k+1}, z_k - z_{k+1}\rangle + \alpha\langle \widetilde{\nabla}_{k+1}, z_{k+1} - u\rangle + \alpha\psi(z_{k+1}) - \alpha\psi(u)\\
&\le \alpha\langle \widetilde{\nabla}_{k+1}, z_k - z_{k+1}\rangle - \frac{1}{2}\|z_k - z_{k+1}\|^2 + \frac{1}{2}\|z_k - u\|^2 - \frac{1+\alpha\sigma}{2}\|z_{k+1} - u\|^2 \ . \qquad (3.1)
\end{aligned}$$

By defining $v \stackrel{\text{def}}{=} \tau_1 z_{k+1} + \tau_2 \widetilde{x} + (1-\tau_1-\tau_2) y_k$, we have $x_{k+1} - v = \tau_1(z_k - z_{k+1})$ and therefore

$$
\begin{aligned}
&\mathbb{E}\Big[\alpha\langle \widetilde{\nabla}_{k+1}, z_k - z_{k+1}\rangle - \frac{1}{2}\|z_k - z_{k+1}\|^2\Big] = \mathbb{E}\Big[\frac{\alpha}{\tau_1}\langle \widetilde{\nabla}_{k+1}, x_{k+1} - v\rangle - \frac{1}{2\tau_1^2}\|x_{k+1} - v\|^2\Big] \\
&= \mathbb{E}\Big[\frac{\alpha}{\tau_1}\Big(\langle \widetilde{\nabla}_{k+1}, x_{k+1} - v\rangle - \frac{1}{2\alpha\tau_1}\|x_{k+1} - v\|^2 - \psi(v) + \psi(x_{k+1})\Big) + \frac{\alpha}{\tau_1}\Big(\psi(v) - \psi(x_{k+1})\Big)\Big] \\
&\stackrel{①}{\le} \mathbb{E}\Big[\frac{\alpha}{\tau_1}\Big(\langle \widetilde{\nabla}_{k+1}, x_{k+1} - v\rangle - \frac{3L}{2}\|x_{k+1} - v\|^2 - \psi(v) + \psi(x_{k+1})\Big) + \frac{\alpha}{\tau_1}\Big(\psi(v) - \psi(x_{k+1})\Big)\Big] \\
&\stackrel{②}{\le} \mathbb{E}\Big[\frac{\alpha}{\tau_1}\Big(F(x_{k+1}) - F(y_{k+1}) + \frac{1}{4L}\sigma_{k+1}^2\Big) + \frac{\alpha}{\tau_1}\Big(\psi(v) - \psi(x_{k+1})\Big)\Big] \\
&\stackrel{③}{\le} \mathbb{E}\Big[\frac{\alpha}{\tau_1}\Big(F(x_{k+1}) - F(y_{k+1}) + \frac{1}{2}\big(f(\widetilde{x}) - f(x_{k+1}) - \langle \nabla f(x_{k+1}), \widetilde{x} - x_{k+1}\rangle\big)\Big) \\
&\qquad + \frac{\alpha}{\tau_1}\Big(\tau_1\psi(z_{k+1}) + \tau_2\psi(\widetilde{x}) + (1-\tau_1-\tau_2)\psi(y_k) - \psi(x_{k+1})\Big)\Big] \ . \qquad (3.2)
\end{aligned}
$$

Above, ① uses our choice $\tau_1 \le \frac{3}{\alpha L}$, ② uses Lemma 3.3, ③ uses Lemma 3.4 together with the convexity of $\psi(\cdot)$ and the definition of v. Finally, noticing that $\mathbb{E}[\langle \widetilde{\nabla}_{k+1}, z_k - u\rangle] = \langle \nabla f(x_{k+1}), z_k - u\rangle$ and $\tau_2 = \frac{1}{2}$, we obtain the desired inequality by combining (3.1) and (3.2). □

The next lemma simplifies the left hand side of Lemma 3.6 using the convexity of $f(\cdot)$, and gives an inequality that relates the objective-distance-to-minimizer quantities $F(y_k) - F(x^*)$, $F(y_{k+1}) - F(x^*)$, and $F(\widetilde{x}) - F(x^*)$ to the point-distance-to-minimizer quantities $\|z_k - x^*\|^2$ and $\|z_{k+1} - x^*\|^2$.

Lemma 3.7 (coupling step 2)**.** *Under the same choices of τ_1, τ_2 as in Lemma 3.6, we have*

$$
\begin{aligned}
0 \le \frac{\alpha(1-\tau_1-\tau_2)}{\tau_1}(F(y_k) - F(x^*)) - \frac{\alpha}{\tau_1}\big(\mathbb{E}\big[F(y_{k+1})\big] - F(x^*)\big) + \frac{\alpha\tau_2}{\tau_1}\big(F(\widetilde{x}) - F(x^*)\big) \\
+ \frac{1}{2}\|z_k - x^*\|^2 - \frac{1+\alpha\sigma}{2}\mathbb{E}\big[\|z_{k+1} - x^*\|^2\big] \ .
\end{aligned}
$$

Proof. We first compute that

$$
\begin{aligned}
&\alpha\big(f(x_{k+1}) - f(u)\big) \stackrel{①}{\le} \alpha\langle \nabla f(x_{k+1}), x_{k+1} - u\rangle \\
&= \alpha\langle \nabla f(x_{k+1}), x_{k+1} - z_k\rangle + \alpha\langle \nabla f(x_{k+1}), z_k - u\rangle \\
&\stackrel{②}{=} \frac{\alpha\tau_2}{\tau_1}\langle \nabla f(x_{k+1}), \widetilde{x} - x_{k+1}\rangle + \frac{\alpha(1-\tau_1-\tau_2)}{\tau_1}\langle \nabla f(x_{k+1}), y_k - x_{k+1}\rangle + \alpha\langle \nabla f(x_{k+1}), z_k - u\rangle \\
&\stackrel{③}{\le} \frac{\alpha\tau_2}{\tau_1}\langle \nabla f(x_{k+1}), \widetilde{x} - x_{k+1}\rangle + \frac{\alpha(1-\tau_1-\tau_2)}{\tau_1}(f(y_k) - f(x_{k+1})) + \alpha\langle \nabla f(x_{k+1}), z_k - u\rangle \ .
\end{aligned}
$$

Above, ① uses the convexity of $f(\cdot)$, ② uses the choice that $x_{k+1} = \tau_1 z_k + \tau_2\widetilde{x} + (1-\tau_1-\tau_2)y_k$, and ③ uses the convexity of $f(\cdot)$ again. By applying Lemma 3.6 to the above inequality, we have

$$
\begin{aligned}
&\alpha\big(f(x_{k+1}) - F(u)\big) \le \frac{\alpha(1-\tau_1-\tau_2)}{\tau_1}(F(y_k) - f(x_{k+1})) \\
&+\frac{\alpha}{\tau_1}\Big(F(x_{k+1}) - \mathbb{E}\big[F(y_{k+1})\big] + \tau_2 F(\widetilde{x}) - \tau_2 f(x_{k+1})\Big) + \frac{1}{2}\|z_k - u\|^2 - \frac{1+\alpha\sigma}{2}\mathbb{E}\big[\|z_{k+1} - u\|^2\big] - \frac{\alpha}{\tau_1}\psi(x_{k+1})
\end{aligned}
$$

which implies

$$
\begin{aligned}
&\alpha\big(F(x_{k+1}) - F(u)\big) \le \frac{\alpha(1-\tau_1-\tau_2)}{\tau_1}(F(y_k) - F(x_{k+1})) \\
&+ \frac{\alpha}{\tau_1}\Big(F(x_{k+1}) - \mathbb{E}\big[F(y_{k+1})\big] + \tau_2 F(\widetilde{x}) - \tau_2 F(x_{k+1})\Big) + \frac{1}{2}\|z_k - u\|^2 - \frac{1+\alpha\sigma}{2}\mathbb{E}\big[\|z_{k+1} - u\|^2\big] \ .
\end{aligned}
$$

After rearranging and setting $u = x^*$, the above inequality yields

$$0 \le \frac{\alpha(1-\tau_1-\tau_2)}{\tau_1}(F(y_k)-F(x^*)) - \frac{\alpha}{\tau_1}\big(\mathbb{E}\big[F(y_{k+1})-F(x^*)\big]\big) + \frac{\alpha\tau_2}{\tau_1}\big(F(\widetilde{x})-F(x^*)\big) + \frac{1}{2}\|z_k-x^*\|^2 - \frac{1+\alpha\sigma}{2}\mathbb{E}\big[\|z_{k+1}-x^*\|^2\big] \ . \qquad \square$$

3.2 Proof of Theorem 3.1

We are now ready to combine the analyses across iterations, and derive our final Theorem 3.1. Our proof next requires a careful telescoping of Lemma 3.7 together with our specific parameter choices.

Proof of Theorem 3.1. Define $D_k \stackrel{\text{def}}{=} F(y_k) - F(x^*)$, $\widetilde{D}^s \stackrel{\text{def}}{=} F(\widetilde{x}^s) - F(x^*)$, and rewrite Lemma 3.7:

$$0 \le \frac{(1-\tau_1-\tau_2)}{\tau_1}D_k - \frac{1}{\tau_1}D_{k+1} + \frac{\tau_2}{\tau_1}\mathbb{E}\big[\widetilde{D}^s\big] + \frac{1}{2\alpha}\|z_k-x^*\|^2 - \frac{1+\alpha\sigma}{2\alpha}\mathbb{E}\big[\|z_{k+1}-x^*\|^2\big] \ .$$

At this point, let us define $\theta = 1+\alpha\sigma$ and multiply the above inequality by θ^j for each $k = sm+j$. Then, we sum up the resulting m inequalities for all $j = 0, 1, \ldots, m-1$:

$$0 \le \mathbb{E}\Big[\frac{(1-\tau_1-\tau_2)}{\tau_1}\sum_{j=0}^{m-1} D_{sm+j}\cdot\theta^j - \frac{1}{\tau_1}\sum_{j=0}^{m-1} D_{sm+j+1}\cdot\theta^j\Big] + \frac{\tau_2}{\tau_1}\widetilde{D}^s\cdot\sum_{j=0}^{m-1}\theta^j + \frac{1}{2\alpha}\|z_{sm}-x^*\|^2 - \frac{\theta^m}{2\alpha}\big[\|z_{(s+1)m}-x^*\|^2\big] \ .$$

Note that in the above inequality we have assumed all the randomness in the first $s-1$ epochs are fixed and the only source of randomness comes from epoch s. We can rearrange the terms in the above inequality and get

$$\mathbb{E}\Big[\frac{\tau_1+\tau_2-(1-1/\theta)}{\tau_1}\sum_{j=1}^{m} D_{sm+j}\cdot\theta^j\Big] \le \frac{(1-\tau_1-\tau_2)}{\tau_1}\Big(D_{sm} - \theta^m\mathbb{E}\big[D_{(s+1)m}\big]\Big) + \frac{\tau_2}{\tau_1}\widetilde{D}^s\cdot\sum_{j=0}^{m-1}\theta^j + \frac{1}{2\alpha}\|z_{sm}-x^*\|^2 - \frac{\theta^m}{2\alpha}\mathbb{E}\big[\|z_{(s+1)m}-x^*\|^2\big] \ .$$

Using the special choice that $\widetilde{x}^{s+1} = \big(\sum_{j=0}^{m-1}\theta^j\big)^{-1}\cdot\sum_{j=0}^{m-1} y_{sm+j+1}\cdot\theta^j$ and the convexity of $F(\cdot)$, we derive that $\widetilde{D}^{s+1} \le \big(\sum_{j=0}^{m-1}\theta^j\big)^{-1}\cdot\sum_{j=0}^{m-1} D_{sm+j+1}\cdot\theta^j$. Substituting this into the above inequality, we get

$$\frac{\tau_1+\tau_2-(1-1/\theta)}{\tau_1}\theta\mathbb{E}\big[\widetilde{D}^{s+1}\big]\cdot\sum_{j=0}^{m-1}\theta^j \le \frac{(1-\tau_1-\tau_2)}{\tau_1}\Big(D_{sm} - \theta^m\mathbb{E}\big[D_{(s+1)m}\big]\Big) + \frac{\tau_2}{\tau_1}\widetilde{D}^s\cdot\sum_{j=0}^{m-1}\theta^j + \frac{1}{2\alpha}\|z_{sm}-x^*\|^2 - \frac{\theta^m}{2\alpha}\mathbb{E}\big[\|z_{(s+1)m}-x^*\|^2\big] \ . \quad (3.3)$$

We consider two cases next.

Case 1. Suppose $\frac{m\sigma}{L} \le \frac{3}{4}$. In this case, we choose $\alpha = \frac{1}{\sqrt{3m\sigma L}}$ and $\tau_1 = \frac{1}{3\alpha L} = m\alpha\sigma = \frac{\sqrt{m\sigma}}{\sqrt{3L}} \in [0, \frac{1}{2}]$ for `Katyusha`. It implies $\alpha\sigma \le 1/2m$ and therefore the following inequality holds:

$$\tau_2(\theta^{m-1}-1) + (1-1/\theta) = \frac{1}{2}((1+\alpha\sigma)^{m-1}-1) + (1-\frac{1}{1+\alpha\sigma}) \le (m-1)\alpha\sigma + \alpha\sigma = m\alpha\sigma = \tau_1 \ .$$

In other words, we have $\tau_1+\tau_2-(1-1/\theta)\geq\tau_2\theta^{m-1}$ and thus (3.3) implies that

$$\mathbb{E}\Big[\frac{\tau_2}{\tau_1}\widetilde{D}^{s+1}\cdot\sum_{j=0}^{m-1}\theta^j+\frac{1-\tau_1-\tau_2}{\tau_1}D_{(s+1)m}+\frac{1}{2\alpha}\|z_{(s+1)m}-x^*\|^2\Big]$$
$$\leq\theta^{-m}\cdot\Big(\frac{\tau_2}{\tau_1}\widetilde{D}^{s}\cdot\sum_{j=0}^{m-1}\theta^j+\frac{1-\tau_1-\tau_2}{\tau_1}D_{sm}+\frac{1}{2\alpha}\|z_{sm}-x^*\|^2\Big)\ .$$

If we telescope the above inequality over all epochs $s=0,1,\ldots,S-1$, we obtain

$$\begin{aligned}\mathbb{E}\big[F(\widetilde{x}^S)-F(x^*)\big]=\mathbb{E}\big[\widetilde{D}^S\big]&\overset{①}{\leq}\theta^{-Sm}\cdot O\Big(\widetilde{D}^0+D_0+\frac{\tau_1}{\alpha m}\|x_0-x^*\|^2\Big)\\&\overset{②}{\leq}\theta^{-Sm}\cdot O\Big(1+\frac{\tau_1}{\alpha m\sigma}\Big)\cdot\big(F(x_0)-F(x^*)\big)\\&\overset{③}{=}O((1+\alpha\sigma)^{-Sm})\cdot\big(F(x_0)-F(x^*)\big)\ .\end{aligned}\tag{3.4}$$

Above, ① uses the fact that $\sum_{j=0}^{m-1}\theta^j\geq m$ and $\tau_2=\frac{1}{2}$; ② uses the strong convexity of $F(\cdot)$ which implies $F(x_0)-F(x^*)\geq\frac{\sigma}{2}\|x_0-x^*\|^2$; and ③ uses our choice of τ_1.

Case 2. Suppose $\frac{m\sigma}{L}>\frac{3}{4}$. In this case, we choose $\tau_1=\frac{1}{2}$ and $\alpha=\frac{1}{3\tau_1L}=\frac{2}{3L}$ as in `Katyusha`. Our parameter choices help us simplify (3.3) as

$$2\mathbb{E}\big[\widetilde{D}^{s+1}\big]\cdot\sum_{j=0}^{m-1}\theta^j\leq\widetilde{D}^s\cdot\sum_{j=0}^{m-1}\theta^j+\frac{1}{2\alpha}\|z_{sm}-x^*\|^2-\frac{\theta^m}{2\alpha}\mathbb{E}\big[\|z_{(s+1)m}-x^*\|^2\big]\ .$$

Since $\theta^m=(1+\alpha\sigma)^m\geq1+\alpha\sigma m=1+\frac{2\sigma m}{3L}\geq\frac{3}{2}$, the above inequality implies

$$\frac{3}{2}\mathbb{E}\big[\widetilde{D}^{s+1}\big]\cdot\sum_{j=0}^{m-1}\theta^j+\frac{9L}{8}\mathbb{E}\big[\|z_{(s+1)m}-x^*\|^2\big]\leq\widetilde{D}^s\cdot\sum_{j=0}^{m-1}\theta^j+\frac{3L}{4}\|z_{sm}-x^*\|^2\ .$$

If we telescope this inequality over all the epochs $s=0,1,\ldots,S-1$, we immediately have

$$\mathbb{E}\Big[\widetilde{D}^S\cdot\sum_{j=0}^{m-1}\theta^j+\frac{3L}{4}\|z_{Sm}-x^*\|^2\Big]\leq\big(\frac{2}{3}\big)^S\cdot\Big(\widetilde{D}^0\cdot\sum_{j=0}^{m-1}\theta^j+\frac{3L}{4}\|z_0-x^*\|^2\Big)\ .$$

Finally, since $\sum_{j=0}^{m-1}\theta^j\geq m$ and $\frac{\sigma}{2}\|z_0-x^*\|^2\leq F(x_0)-F(x^*)$ owing to the strong convexity of $F(\cdot)$, we conclude that

$$\mathbb{E}\big[F(\widetilde{x}^S)-F(x^*)\big]\leq O\big(1.5^{-S}\big)\cdot\big(F(x_0)-F(x^*)\big)\ .\tag{3.5}$$

Combining (3.4) and (3.5) we finish the proof of Theorem 3.1. □

4 Applications on ERM Problems

In this section we apply reductions from Theorem 2.3 to translate our Theorem 3.1 into optimal algorithms also for non-strongly convex objectives and/or non-smooth objectives. To begin with, it is an immediate corollary of Theorem 3.1 that `Katyusha` satisfies the HOOD property:

Corollary 4.1. `Katyusha` *satisfies the HOOD property with* $T(L,\sigma)=O\big(n+\frac{\sqrt{nL}}{\sqrt{\sigma}}\big)$ *iterations.*

Remark 4.2. Existing accelerated stochastic methods (even for the simpler Problem (1.3)) either do not satisfy HOOD property or satisfy HOOD with an additional factor $\log(L/\sigma)$ in the number of iterations. This is why they do not yield optimal convergence rates even if Theorem 2.3 is used.

Combining Corollary 4.1 with Theorem 2.3, we have the following corollaries:

Corollary 4.3. *If each $f_i(x)$ is convex, L-smooth and $\psi(\cdot)$ is not necessarily strongly convex in Problem (1.1), then by applying* AdaptReg *on* Katyusha *with a starting vector x_0, we obtain an output x satisfying $\mathbb{E}[F(x)] - F(x^*) \le \varepsilon$ in at most*

$$O\Big(n \log \tfrac{F(x_0)-F(x^*)}{\varepsilon} + \tfrac{\sqrt{nL}\cdot\|x_0-x^*\|}{\sqrt{\varepsilon}}\Big) \propto \tfrac{1}{\sqrt{\varepsilon}} \text{ iterations.}$$

In contrast, the best known convergence rate was

$$\text{Catalyst:} \quad O\Big(\Big(n + \tfrac{\sqrt{nL}\cdot\|x_0-x^*\|}{\sqrt{\varepsilon}}\Big) \log \tfrac{F(x_0)-F(x^*)}{\varepsilon} \log \tfrac{L\|x_0-x^*\|^2}{\varepsilon}\Big) \propto \tfrac{\log^2(1/\varepsilon)}{\sqrt{\varepsilon}} \text{ iterations .}$$

Corollary 4.4. *If each $f_i(x)$ is G-Lipschitz continuous and $\psi(x)$ is σ-SC in Problem (1.3), then by applying* AdaptSmooth *on* Katyusha *with a starting vector x_0, we obtain an output x satisfying $\mathbb{E}[F(x)] - F(x^*) \le \varepsilon$ in at most*

$$O\Big(n \log \tfrac{F(x_0)-F(x^*)}{\varepsilon} + \tfrac{\sqrt{n}G}{\sqrt{\sigma\varepsilon}}\Big) \propto \tfrac{1}{\sqrt{\varepsilon}} \text{ iterations.}$$

In contrast, the best known convergence rate was

$$\text{APCG/SPDC:} \quad O\Big(\Big(n + \tfrac{\sqrt{n}G}{\sqrt{\sigma\varepsilon}}\Big) \log \tfrac{nG(F(x_0)-F(x^*))}{\sigma\varepsilon}\Big) \propto \tfrac{\log(1/\varepsilon)}{\sqrt{\varepsilon}} \text{ iterations .}$$

Corollary 4.5. *If each $f_i(x)$ is G-Lipschitz continuous and $\psi(x)$ is not necessarily strongly convex in Problem (1.3), then by applying* JointAdaptRegSmooth *on* Katyusha *with a starting vector x_0, we obtain an output x satisfying $\mathbb{E}[F(x)] - F(x^*) \le \varepsilon$ in at most*

$$O\Big(n \log \tfrac{F(x_0)-F(x^*)}{\varepsilon} + \tfrac{\sqrt{n}G\|x_0-x^*\|}{\varepsilon}\Big) \propto \tfrac{1}{\varepsilon} \text{ iterations.}$$

In contrast, the best known convergence rate was

$$\text{APCG/SPDC:} \quad O\Big(\Big(n + \tfrac{\sqrt{n}G\|x_0-x^*\|}{\varepsilon}\Big) \log \tfrac{nG\|x_0-x^*\|^2(F(x_0)-F(x^*))}{\varepsilon^2}\Big) \propto \tfrac{\log(1/\varepsilon)}{\varepsilon} \text{ iterations .}$$

5 Katyusha in the Non-Strongly Convex Setting

Due to the increasing popularity of *non-strongly convex* minimization tasks (most notably ℓ_1-regularized problems), researchers often make additional efforts to design separate methods for minimizing the non-strongly convex variant of Problem (1.1) that are *direct*, meaning without restarting and in particular without using any reductions such as Theorem 2.3 [12, 15].

In this section, we also develop our *direct and accelerated* method for the non-strongly convex variant of Problem (1.1). We call it $\texttt{Katyusha}^{\texttt{ns}}$ and state it in Algorithm 2.

The only difference between $\texttt{Katyusha}^{\texttt{ns}}$ and Katyusha is that we choose $\tau_1 = \tau_{1,s} = \frac{2}{s+4}$ to be a parameter that depends on the epoch index s, and accordingly $\alpha = \alpha_s = \frac{1}{3L\tau_{1,s}}$. This should not be a big surprise because in accelerated full-gradient methods, the values τ_1 and α also decrease (although with respect to k rather than s) when there is no strong convexity [8].

We state the following convergence theorem for $\texttt{Katyusha}^{\texttt{ns}}$ and defer its proof to Appendix B. The proof also relies on the one-iteration inequality in Lemma 3.7, but requires telescoping such inequalities in a different manner as compared with Theorem 3.1.

Algorithm 2 $\texttt{Katyusha}^{\texttt{ns}}(x_0, S, \sigma, L)$

$m \leftarrow 2n$; ◇ *epoch length*
$\tau_2 \leftarrow \frac{1}{2}$;
$y_0 = z_0 = \widetilde{x}^0 \leftarrow x_0$; ◇ *initial vectors*
for $s \leftarrow 0$ **to** $S-1$ **do**
  $\tau_{1,s} \leftarrow \frac{2}{s+4}$, $\alpha_s \leftarrow \frac{1}{3\tau_{1,s}L}$ ◇ *different parameter choices comparing to* Katyusha
  $\mu^s \leftarrow \nabla f(\widetilde{x}^s)$; ◇ *compute the full gradient only once every m iterations*
  for $j \leftarrow 0$ **to** $m-1$ **do**
    $k \leftarrow (sm)+j$;
    $x_{k+1} \leftarrow \tau_{1,s} z_k + \tau_2 \widetilde{x}^s + (1-\tau_{1,s}-\tau_2) y_k$;
    $\widetilde{\nabla}_{k+1} \leftarrow \mu^s + \nabla f_i(x_{k+1}) - \nabla f_i(\widetilde{x}^s)$ where i is randomly chosen from $\{1, 2, \ldots, n\}$;
    $z_{k+1} = \arg\min_z \left\{ \frac{1}{2\alpha_s}\|z - z_k\|^2 + \langle \widetilde{\nabla}_{k+1}, z\rangle + \psi(z) \right\}$;
    Option I: $y_{k+1} \leftarrow \arg\min_y \left\{ \frac{3L}{2}\|y - x_{k+1}\|^2 + \langle \widetilde{\nabla}_{k+1}, y\rangle + \psi(y) \right\}$;
    Option II: $y_{k+1} \leftarrow x_{k+1} + \tau_{1,s}(z_{k+1} - z_k)$ ◇ *we analyze only I but II also works*
  end for
  $\widetilde{x}^{s+1} \leftarrow \frac{1}{m}\sum_{j=1}^{m} y_{sm+j}$; ◇ *compute snapshot* $\widetilde{x}$
end for
return $\widetilde{x}^S$.

Theorem 5.1. *If each $f_i(x)$ is convex, L-smooth in Problem (1.1) and $\psi(\cdot)$ is not necessarily strongly convex, then* $\texttt{Katyusha}^{\texttt{ns}}(x_0, S, L)$ *satisfies*

$$\mathbb{E}\big[F(\widetilde{x}^S)\big] - F(x^*) \le O\Big(\frac{F(x_0) - F(x^*)}{S^2} + \frac{L\|x_0 - x^*\|^2}{mS^2}\Big)$$

In other words, choosing $m = \Theta(n)$, $\texttt{Katyusha}^{\texttt{ns}}$ *achieves an ε-additive error (i.e., $\mathbb{E}\big[F(\widetilde{x}^S)\big] - F(x^*) \le \varepsilon$) using at most $O\Big(\frac{n\sqrt{F(x_0)-F(x^*)}}{\sqrt{\varepsilon}} + \frac{\sqrt{nL}\|x_0-x^*\|}{\sqrt{\varepsilon}}\Big)$ iterations.*

Remark 5.2. $\texttt{Katyusha}^{\texttt{ns}}$ is a *direct, accelerated* solver for the non-SC case of Problem (1.1). It is illustrative to compare it with the convergence theorem of a *direct, non-accelerated* solver of the same setting. Below is the convergence theorem of SAGA after translating to our notations:

$$\text{SAGA:}\quad \mathbb{E}\big[F(x)\big] - F(x^*) \le O\Big(\frac{F(x_0) - F(x^*)}{S} + \frac{L\|x_0 - x^*\|^2}{nS}\Big)\ .$$

It is clear from this comparison that $\texttt{Katyusha}^{\texttt{ns}}$ is a factor S faster than non-accelerated methods such as SAGA, where $S = T/n$ if T is the total number of stochastic iterations. This convergence can also be written in terms of the number of iterations which is $O\big(\frac{n(F(x_0)-F(x^*))}{\varepsilon} + \frac{L\|x_0-x^*\|^2}{\varepsilon}\big)$.

Remark 5.3. Theorem 5.1 is slightly worse than the reduction-based complexity in Corollary 4.5. This can be fixed by making epoch lengths to grow rather than stay as a constant $2n$. Since it complicates the proofs and the notations we refrain from doing so in this version of the paper.[15]

[15] More precisely, recall that a similar issue has also happened in the non-accelerated world: the iteration complexity $O(\frac{n+L}{\varepsilon})$ in SAGA can be improved to $O(n\log\frac{1}{\varepsilon} + \frac{L}{\varepsilon})$ by doubling the epoch length across epochs [12]. Similar techniques can also be used to improve our result above.

6 Extensions to Other Smoothness Regimes

We mentioned since the first version of this paper that `Katyusha` and `Katyusha`$^{\mathsf{ns}}$ naturally extend
- to settings where each $f_i(x)$ can have a different smoothness parameter, and/or
- to settings where the smoothness can be with respect to a non-Euclidean norm,

respectively using techniques [11] and [8]. In this section, we explain how this can be done in Section 6.1 and 6.2, and state our extended theorems in Section 6.3.

6.1 Non-Uniform Smoothness Parameters

Suppose each $f_i(x)$ in Problem (1.1) is L_i-smooth for possibly different parameters L_i, and suppose $f(x) = \frac{1}{n}\sum_{i=1}^n f_i(x)$ is L-smooth. Denoting by $\overline{L} = \frac{1}{n}\sum_{i=1}^n L_i$, it is easy to verify $L \le \overline{L}$ using triangle inequality and the definition of smoothness (see Definition 2.1).

Algorithmic Changes. In this refined case, perform the following changes to the algorithms:
- Redefine the gradient estimator $\widetilde{\nabla}_{k+1}$ to be $\nabla f(x) + \frac{1}{np_i}\big(\nabla f_i(x_{k+1}) - \nabla f_i(\widetilde{x})\big)$, where each $i \in [n]$ is chosen with probability $p_i \stackrel{\text{def}}{=} L_i/n\overline{L}$. It is easy to verify $\mathbb{E}_i[\widetilde{\nabla}_{k+1}] = \nabla f(x_{k+1})$.
- Change all the occurrences of L in the pseudocode of `Katyusha` or `Katyusha`$^{\mathsf{ns}}$ to $\overline{L}$.

Proof Changes. At a high level, all statements of the lemmas and theorems can be revised by simply replacing L with $\overline{L}$, and the proofs are subject to minor changes. More specifically, since our final theorems Theorem 3.1 and Theorem 5.1 both serve as corollaries to our one-iteration analysis lemmas in Section 3.1, it suffices to revise Section 3.1 to adapt to non-uniform parameters L_i. In fact, all lemma statements in Section 3.1 can be revised by replacing L with $\overline{L}$, and the proofs are nearly the same so left to be simple exercises for the readers. For instance, Lemma 3.5 and Lemma 3.7 stay exactly the same as before, but Lemma 3.3, Lemma 3.4, and Lemma 3.6 need to be revised by replacing L with $\overline{L}$. We remark that due to the revision of Lemma 3.6, we now require $\tau_1 \le \frac{3}{\alpha\overline{L}}$ rather than $\tau_1 \le \frac{3}{\alpha L}$.

6.2 Non-Euclidean Norm Smoothness

Suppose we consider smoothness (and strongly convexity) with respect to an arbitrary norm $\|\cdot\|$ in some domain $Q \stackrel{\text{def}}{=} \{x \in \mathbb{R}^d : \psi(x) < +\infty\}$. Symbolically, we say
- f is σ-strongly convex if $\forall x, y \in Q$, it satisfies $f(y) \ge f(x) + \langle \nabla f(x), y - x\rangle + \frac{\sigma}{2}\|x - y\|^2$;
- f is L-smooth if $\forall x, y \in Q$, it satisfies $\|\nabla f(x) - \nabla f(y)\|_* \le L\|x - y\|$.[16]

Above, $\|\cdot\|_* \stackrel{\text{def}}{=} \max\{\langle \xi, x\rangle : \|x\| \le 1\}$ is the dual norm of $\|\cdot\|$. For instance, ℓ_p norm is dual to ℓ_q norm if $\frac{1}{p} + \frac{1}{q} = 1$. Some famous problems have better smoothness parameters when non-Euclidean norms are adopted, see the discussions in [8].

Preassumption Changes. In a non-Euclidean setting, following the traditions of first-order methods [8], one has to first select a *distance generating function* $w(\cdot)$ that is 1-strongly convex.[17] Accordingly, the *Bregman divergence function* is $V_x(y) \stackrel{\text{def}}{=} w(y) - w(x) - \langle \nabla w(x), y - x\rangle$, and the final algorithms and proofs will be described using $V_x(y)$.

In addition, as far as the linear-convergence result Theorem 3.1 is concerned, one has to require $\psi(\cdot)$ to be σ-strongly convexity with respect to function $V_x(y)$ rather than the $\|\cdot\|$ norm; or

[16] This definition has another equivalent form: $\forall x, y \in Q$, it satisfies $f(y) \le f(x) + \langle \nabla f(x), y - x\rangle + \frac{L}{2}\|y - x\|^2$.

[17] For instance, if $Q = \mathbb{R}^d$ and $\|\cdot\|_p$ is the ℓ_p norm for some $p \in (1, 2]$, one can choose $w(x) = \frac{1}{2(p-1)}\|x\|_p^2$; if $Q = \{x \in \mathbb{R}^d : \sum_i x_i = 1\}$ is the probability space and $\|\cdot\|_1$ is the ℓ_1 norm, one can choose $w(x) = \sum_i x_i \log x_i$.

symbolically, $\psi(y) \geq \psi(x) + \langle \nabla\psi(x), y - x\rangle + \sigma V_x(y)$.[18] This is known as the "generalized strong convexity" [37] and is necessary for any linear-convergence result. In contrast, this is not necessary for the non-SC result Theorem 5.1 because $\psi(\cdot)$ need not be strongly convex in that theorem.

Algorithmic Changes. Suppose each $f_i(x)$ is L-smooth with respect to norm $\|\cdot\|$ and a Bregman divergence function $V_x(y)$ is given. We perform the following changes to the algorithms:

- In Line 10 of `Katyusha` (resp. Line 11 of $\texttt{Katyusha}^{\mathsf{ns}}$), change the arg min to be the so-called mirror descent update [8]: $z_{k+1} = \arg\min_z \left\{ \frac{1}{\alpha_s} V_{z_k}(z) + \langle \widetilde{\nabla}_{k+1}, z\rangle + \psi(z) \right\}$
- Forbidden Option II and use Option I only (but *without* replacing $\|y - x_{k+1}\|^2$ with $V_{x_{k+1}}(y)$).

Interested readers can find discussions regarding why such changes are natural in [8].

Proof Changes. We only discuss the necessary changes to Section 3.1 in order to capture non-Euclidean norm smoothness. At a high level, all occurrences of $\|\cdot\|$ applied on *gradients* need to be replaced with $\|\cdot\|_*$, and some of the $\|\cdot\|^2$ quantities such as $\|z_k - u\|^2$ need to be replaced with $V_{z_k}(u)$.[19] More specifically,

- Redefine $\sigma_{k+1}^2 = \|\nabla f(x_{k+1}) - \widetilde{\nabla}_{k+1}\|_*^2$.
- Lemma 3.3 is not changed.
- Lemma 3.4 is not changed, except on the left hand side we put $\|\widetilde{\nabla}_{k+1} - \nabla f(x_{k+1})\|_*^2$.
- Lemma 3.5 becomes:

$$\alpha\langle \widetilde{\nabla}_{k+1}, z_{k+1} - u\rangle + \alpha\psi(z_{k+1}) - \alpha\psi(u) \leq -\frac{1}{2}\|z_k - z_{k+1}\|^2 + \frac{1}{2}V_{z_k}(u) - (1+\alpha\sigma)V_{z_{k+1}}(u) \ .$$

- Lemma 3.6 becomes

$$\begin{aligned} &\alpha\langle \nabla f(x_{k+1}), z_k - u\rangle - \alpha\psi(u) \\ \leq\ & \frac{\alpha}{\tau_1}\Big(F(x_{k+1}) - \mathbb{E}\big[F(y_{k+1})\big] + \tau_2 F(\widetilde{x}) - \tau_2\mathbb{E}\big[F(x_{k+1})\big] - \tau_2\langle \nabla f(x_{k+1}), \widetilde{x} - x_{k+1}\rangle\Big) \\ &+ V_{z_k}(u) - (1+\alpha\sigma)\mathbb{E}\big[V_{z_{k+1}}(u)\big] + \frac{\alpha(1-\tau_1-\tau_2)}{\tau_1}\psi(y_k) - \frac{\alpha}{\tau_1}\psi(x_{k+1}) \ . \end{aligned}$$

- Lemma 3.7 becomes

$$\begin{aligned} 0 \leq \frac{\alpha(1-\tau_1-\tau_2)}{\tau_1}(F(y_k) - F(x^*)) - \frac{\alpha}{\tau_1}\big(\mathbb{E}\big[F(y_{k+1})\big] - F(x^*)\big) + \frac{\alpha\tau_2}{\tau_1}\big(F(\widetilde{x}) - \tau_2 F(x^*)\big) \\ + V_{z_k}(x^*) - (1+\alpha\sigma)\mathbb{E}\big[V_{z_{k+1}}(x^*)\big] \ . \end{aligned}$$

Above, only the proof of Lemma 3.5 needs some moderate changes but that is classically known as the convergence lemma for mirror descent [8]. The rest of the proofs are almost identical as before, except that one has to use the non-Euclidean variant of Young's inequality $\langle a, b\rangle \leq \frac{\|a\|^2}{2} + \frac{\|b\|_*^2}{2}$.

6.3 Theorem Restatements

We now state our final theorems for the extended smoothness settings. The corresponding algorithms $\texttt{Katyusha}^{\mathsf{ext}}$ and $\texttt{Katyusha}^{\mathsf{ns,ext}}$ are included in Appendix C, and the proofs follow from our discussions in Section 6.1 and 6.2.

Suppose each $f_i(x)$ is convex and L_i-smooth with respect to some norm $\|\cdot\|$. Suppose in addition that the norm $\|\cdot\|$ has a Bregman divergence function $V_x(y)$ defined in Section 6.2. Then,

[18] For instance, this is usually satisfied by choosing $\omega(y) \stackrel{\text{def}}{=} \frac{1}{\sigma}\psi(y)$.

[19] Such changes were not necessary in Section 3.1 because $\|\cdot\|_* = \|\cdot\|$ and $V_{z_k}(u) = \|z_k - u\|^2$ if the underlying norm is Euclidean.

Theorem 6.1 (extension of Theorem 3.1)**.** *If $\psi(x)$ is σ-strongly convex with respect to $V_x(y)$, then* $\texttt{Katyusha}^{\texttt{ext}}(x_0, S, \sigma, (L_1, \dots, L_n))$ *satisfies*

$$\mathbb{E}\big[F(\widetilde{x}^S)\big] - F(x^*) \leq \begin{cases} O\Big(\Big(1+\sqrt{\sigma/(3\overline{L}m)}\Big)^{-Sm}\Big) \cdot \big(F(x_0) - F(x^*)\big), & \text{if } m\sigma/\overline{L} \leq \frac{3}{4}; \\ O\big(1.5^{-S}\big) \cdot \big(F(x_0) - F(x^*)\big), & \text{if } m\sigma/\overline{L} > \frac{3}{4}. \end{cases}$$

In other words, choosing $m = \Theta(n)$, $\texttt{Katyusha}^{\texttt{ext}}$ *achieves an ε-additive error (i.e., $\mathbb{E}\big[F(\widetilde{x}^S)\big] - F(x^*) \leq \varepsilon$) using at most $O\big(\big(n + \sqrt{n\overline{L}/\sigma}\big) \cdot \log \frac{F(x_0)-F(x^*)}{\varepsilon}\big)$ iterations.*

Theorem 6.2 (extension of Theorem 5.1)**.** *If $\psi(\cdot)$ is not necessarily strongly convex, then algorithm* $\texttt{Katyusha}^{\texttt{ns,ext}}(x_0, S, (L_1, \dots, L_n))$ *satisfies*

$$\mathbb{E}\big[F(\widetilde{x}^S)\big] - F(x^*) \leq O\Big(\frac{F(x_0) - F(x^*)}{S^2} + \frac{LV_{x_0}(x^*)}{nS^2}\Big) \ .$$

In other words, $\texttt{Katyusha}^{\texttt{ns,ext}}$ *achieves an ε-additive error (i.e., $\mathbb{E}\big[F(\widetilde{x}^S)\big] - F(x^*) \leq \varepsilon$) using at most $O\Big(\frac{n\sqrt{F(x_0)-F(x^*)}}{\sqrt{\varepsilon}} + \frac{\sqrt{nLV_{x_0}(x^*)}}{\sqrt{\varepsilon}}\Big)$ iterations.*

7 Empirical Evaluations

We conclude this paper with empirical evaluations to our theoretical speed-ups. We work on Lasso and ridge regressions (with regularizer $\frac{\lambda}{2}\|x\|^2$ for ridge and regularizer $\lambda\|x\|_1$ for Lasso) on the following six datasets: adult, web, mnist, rcv1, covtype, sensit. We defer dataset and implementation details to Appendix A.

Algorithms and Parameter Tuning. We have implemented the following algorithms:

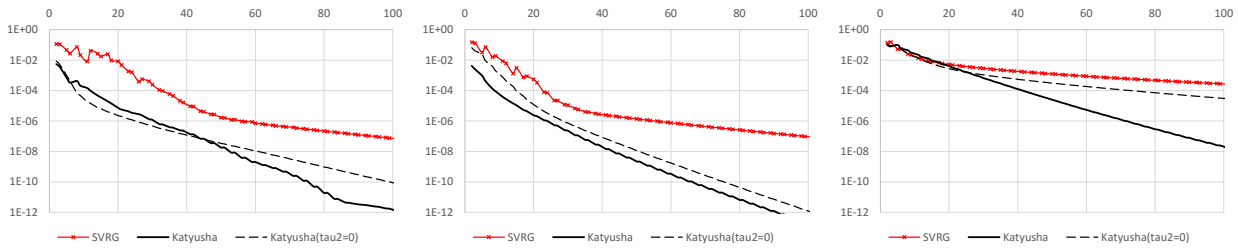
- SVRG [21] with default epoch length $m = 2n$. We tune only *one parameter*: the learning rate.
- Katyusha for ridge and $\texttt{Katyusha}^{\texttt{ns}}$ for Lasso. We tune only *one parameter*: the learning rate.
- SAGA [15]. We tune only *one parameter*: the learning rate.
- Catalyst [26] on top of SVRG. We tune *three parameters*: SVRG's learning rate, Catalyst's learning rate, as well as the regularizer weight in the Catalyst reduction.
- APCG [27]. We tune the learning rate. For Lasso, we also tune the ℓ_2 regularizer weight.
- APCG+AdaptReg (Lasso only). Since APCG intrinsically require an ℓ_2 regularizer to be added on Lasso, we apply AdaptReg from [1] to adaptively learn this regularizer and improve APCG's performance. Two parameters to be tuned: APCG's learning rate and σ_0 in AdaptReg.

All of the parameters were equally, fairly, and automatically tuned by our code base. For interested readers, we discuss more details in Appendix A.

We emphasize that Katyusha is *as simple as SAGA or SVRG in terms of parameter tuning.* In contrast, APCG for Lasso requires two parameters to be tuned, and Catalyst requires three. [25]

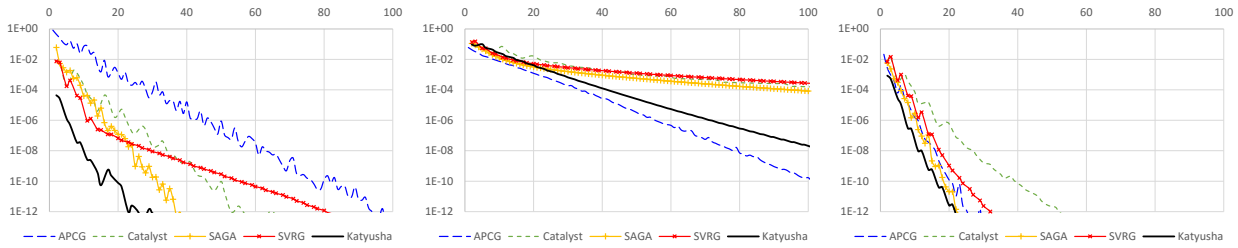
Performance Plots. Following the tradition of ERM experiments, we use the number of "passes" of the dataset as the x-axis. Letting n be the number of feature vectors, each new stochastic gradient computation $\nabla f_i(\cdot)$ counts as $1/n$ pass, and a full gradient computation $\nabla f(\cdot)$ counts as 1 pass.

The y-axis in all of our plots represent the objective distance to the minimum. We emphasize that it is practically also crucial to study high-accuracy regimes (such as objective distance $\leq 10^{-7}$). This is because nowadays there is an increasing number of methods that reduce large-scale machine

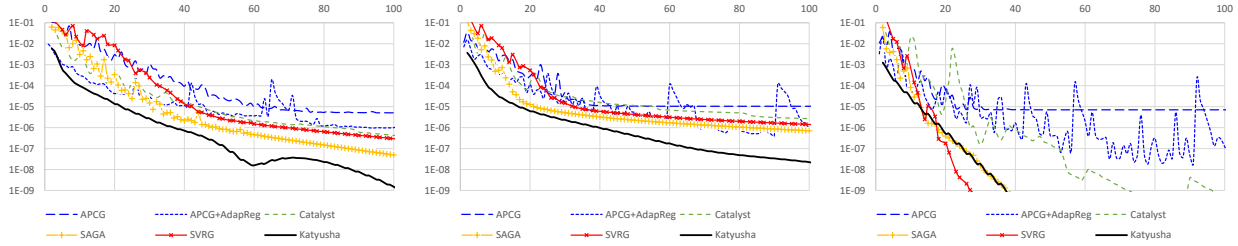


(a) web, ridge $\lambda = 10^{-6}$ (b) mnist, ridge $\lambda = 10^{-6}$ (c) rcv1, ridge $\lambda = 10^{-6}$

Figure 1: Comparing SVRG vs. `Katyusha` vs. `Katyusha` with $\tau_2 = 0$.



(a) covtype, ridge $\lambda = 10^{-6}$ (small) (b) rcv1, ridge $\lambda = 10^{-7}$ (small) (c) adult, ridge $\lambda = 10^{-4}$ (large)



(d) web, ridge $\lambda = 10^{-6}$ (small) (e) mnist, ridge $\lambda = 10^{-6}$ (small) (f) sensit, ridge $\lambda = 10^{-4}$ (large)

Figure 2: Some representative performance charts where λ is the regularizer weight.

learning tasks to multiple black-box calls to ERM solvers [4, 6]. In all such applications, due to error blowups between oracle calls, the ERM solver is required to be *very accurate in training error*.

7.1 Effectiveness of Katyusha Momentum

In our `Katyusha` method, τ_1 controls to the classical Nesterov's momentum and τ_2 controls our newly introduced Katyusha momentum. We find in our theory that setting $\tau_2 = 1/2$ is a good choice so we universally set it to be $1/2$ without tuning in all our experiments. (Of course, if time permits, tuning τ_2 could only help in performance.)

Before this paper, researchers have tried heuristics that is to add Nesterov's momentums directly to stochastic gradient methods [35], and this corresponds to setting $\tau_2 = 0$ in `Katyusha`. In Figure 1, we compare `Katyusha` with $\tau_2 = 1/2$ and $\tau_2 = 0$ in order to illustrate the importance and effectiveness of our Katyusha momentum.

We conclude that the old heuristics (i.e., $\tau_2 = 0$) sometimes indeed make the method faster after careful parameter tuning. However, for certain tasks such as Figure 1(c), without Katyusha momentum the algorithm does not even enjoy an accelerated convergence rate.

7.2 Performance Comparison Across Algorithms

For each of the six datasets and each objective (ridge or lasso), we experiment on three different magnitudes of regularizer weights.[20] This totals 36 performance charts, and we include them in full at the end of this paper. For the sake of cleanness, in Figure 2 we select 6 representative charts for ridge regression and make the following observations.

- Accelerated methods are more powerful when the regularizer weights are small (cf. [11, 27, 41]). For instance, Figure 2(c) and 2(f) are for large values of λ and `Katyusha` performs relatively the same as compared with SVRG / SAGA; however, `Katyusha` significantly outperforms SVRG / SAGA for small values of λ, see for instance Figure 2(b) and 2(e).
- `Katyusha` almost always either outperform or equal-perform its competitors. The only notable place it gets outperformed is by SVRG (see Figure 2(f)); however, this performance gap cannot be large because `Katyusha` is capable of recovering SVRG if $\tau_1 = \tau_2 = 0$.[21]
- Catalyst does not work as beautiful as its theory in high-accuracy regimes, even though we have carefully tuned parameters α_0 and κ in Catalyst in addition to its learning rate. Indeed, in Figure 2(a), 2(c) and 2(f) Catalyst (which is a reduction on SVRG) is outperformed by SVRG.
- APCG performs poorly on all Lasso tasks (cf. Figure 2(d), 2(e), 2(f)) because it is not designed for non-SC objectives. The reduction in [1] helps to fix this issue, but not by a lot.
- APCG can sometimes be largely dominated by SVRG or SAGA (cf. Figure 2(f)): this is because for datasets such as sensit, dual-based methods (such as APCG) cannot make use of the implicity local strong convexity in the objective. In such cases, `Katyusha` is not lost to SVRG or SAGA.

Acknowledgements

This paper is partially supported by a Microsoft Research Grant, no. 0518584. We thank Shai Shalev-Shwartz for useful feedbacks and suggestions on this paper, thank Blake Woodworth and Nati Srebro for pointer to their paper [42], and thank Guanghui Lan for correcting our citation of [14]. We also acknowledge Xu Chen from Peking University and Zhe Li from the University of Iowa for verifying the proofs and correcting typos.

Appendix

A Experiment Details

The datasets we used in this paper are downloaded from the LibSVM website [16]:

- the adult (a9a) dataset (32,561 samples and 123 features).
- the web (w8a) dataset (49,749 samples and 300 features).
- the covtype (binary.scale) dataset (581,012 samples and 54 features).
- the mnist (class 1) dataset (60,000 samples and 780 features).
- the rcv1 (train.binary) dataset (20,242 samples and 47,236 features).

[20] We choose three values λ that are powers of 10 and around $10/n, 1/n, 1/10n$. This range can be verified to contain the best regularization weights using cross validation.

[21] The only reason `Katyusha` does not match the performance of SVRG in Figure 2(f) is because we have not tuned parameter τ_2. If we also tune τ_2 for the best performance, `Katyusha` shall no longer be outperformed by SVRG. In any case, it is not really necessary to tune τ_2 because the performance of `Katyusha` is already superb.

- the sensit (combined) dataset (78,823 samples and 100 features).

To make easier comparison across datasets, we scale every vector by the average Euclidean norm of all the vectors in the dataset. In other words, we ensure that the data vectors have an average Euclidean norm 1. This step is for comparison only and not necessary in practice.

Parameter-tuning details. We select learning rates from the set $\{10^{-k}, 2\times 10^{-k}, 5\times 10^{-k} : k\in\mathbb{Z}\}$, and select regularizer weights (for APCG) from the set $\{10^{-k} : k\in\mathbb{Z}\}$. We have fully automated the parameter tuning procedure to ensure a fair and strong comparison.

While the learning rates were explicitly defined for SVRG and SAGA, there were implicit for all accelerated methods. For Catalyst, the learning rate is in fact their α_0 in the paper [25]. Instead of choosing it to be the theory-predicted value, we multiply it with an extra factor to be tuned and call this factor the "learning rate". Similarly, for `Katyusha` and `Katyusha`$^{\mathsf{ns}}$, we multiply the theory-predicted τ_1 with an extra factor and this serves as a learning rate. For APCG, we use their Algorithm 1 in the paper and multiply their theory-predicted μ with an extra factor.

For Catalyst, in principle one also has to tune the stopping criterion. After communicating with an author of Catalyst, we learned that one can terminate the inner loop whenever the duality gap becomes no more than, say one fourth, of the last duality gap from the previous epoch [25]. This stopping criterion was also found by the authors of [1] to be a good choice for reduction-based methods.

B Proof of Theorem 5.1

Proof of Theorem 5.1. First of all, the parameter choices satisfy the presumptions in Lemma 3.6, so again by defining $D_k \stackrel{\text{def}}{=} F(y_k)-F(x^*)$ and $\widetilde{D}^s \stackrel{\text{def}}{=} F(\widetilde{x}^s)-F(x^*)$, we can rewrite Lemma 3.7 as follows:

$$0 \le \frac{\alpha_s(1-\tau_{1,s}-\tau_2)}{\tau_{1,s}}D_k - \frac{\alpha_s}{\tau_{1,s}}\mathbb{E}\big[D_{k+1}\big] + \frac{\alpha_s\tau_2}{\tau_{1,s}}m\widetilde{D}^s + \frac{1}{2}\|z_k-x^*\|^2 - \frac{1}{2}\mathbb{E}\big[\|z_{k+1}-x^*\|^2\big] \ .$$

Summing up the above inequality for all the iterations $k = sm, sm+1, \ldots, sm+m-1$, we have

$$\begin{aligned}&\mathbb{E}\Big[\alpha_s\frac{1-\tau_{1,s}-\tau_2}{\tau_{1,s}}D_{(s+1)m} + \alpha_s\frac{\tau_{1,s}+\tau_2}{\tau_{1,s}}\sum_{j=1}^{m}D_{sm+j}\Big]\\ &\le \alpha_s\frac{1-\tau_{1,s}-\tau_2}{\tau_{1,s}}D_{sm} + \alpha_s\frac{\tau_2}{\tau_{1,s}}m\widetilde{D}^s + \frac{1}{2}\|z_{sm}-x^*\|^2 - \frac{1}{2}\mathbb{E}\big[\|z_{(s+1)m}-x^*\|^2\big] \ . \qquad \text{(B.1)}\end{aligned}$$

Note that in the above inequality we have assumed all the randomness in the first $s-1$ epochs are fixed and the only source of randomness comes from epoch s.

If we define $\widetilde{x}^s = \frac{1}{m}\sum_{j=1}^{m} y_{(s-1)m+j}$, then by the convexity of function $F(\cdot)$ we have $m\widetilde{D}^s \le \sum_{j=1}^{n} D_{(s-1)m+j}$. Therefore, for every $s\ge 1$ we can derive from (B.1) that

$$\begin{aligned}&\mathbb{E}\Big[\frac{1}{\tau_{1,s}^2}D_{(s+1)m} + \frac{\tau_{1,s}+\tau_2}{\tau_{1,s}^2}\sum_{j=1}^{m-1}D_{sm+j}\Big]\\ &\le \frac{1-\tau_{1,s}}{\tau_{1,s}^2}D_{sm} + \frac{\tau_2}{\tau_{1,s}^2}\sum_{j=1}^{m-1}D_{(s-1)m+j} + \frac{3L}{2}\|z_{sm}-x^*\|^2 - \frac{3L}{2}\mathbb{E}\big[\|z_{(s+1)m}-x^*\|^2\big] \ . \qquad \text{(B.2)}\end{aligned}$$

For the base case $s = 0$, we can also rewrite (B.1) as

$$\mathbb{E}\Big[\frac{1}{\tau_{1,0}^2} D_m + \frac{\tau_{1,0}+\tau_2}{\tau_{1,0}^2}\sum_{j=1}^{m-1} D_j\Big]$$
$$\leq \frac{1-\tau_{1,0}-\tau_2}{\tau_{1,0}^2} D_0 + \frac{\tau_2 n}{\tau_{1,0}^2}\widetilde{D}^0 + \frac{3L}{2}\|z_0 - x^*\|^2 - \frac{3L}{2}\mathbb{E}\big[\|z_m - x^*\|^2\big] \ . \tag{B.3}$$

At this point, if we choose $\tau_{1,s} = \frac{2}{s+4} \leq \frac{1}{2}$, it satisfies

$$\frac{1}{\tau_{1,s}^2} \geq \frac{1-\tau_{1,s+1}}{\tau_{1,s+1}^2} \quad \text{and} \quad \frac{\tau_{1,s}+\tau_2}{\tau_{1,s}^2} \geq \frac{\tau_2}{\tau_{1,s+1}^2} \ .$$

Using these two inequalities, we can telescope (B.3) and (B.2) for all $s = 0, 1, \ldots, S-1$. We obtain in the end that

$$\mathbb{E}\Big[\frac{1}{\tau_{1,S-1}^2} D_{Sm} + \frac{\tau_{1,S-1}+\tau_2}{\tau_{1,S-1}^2}\sum_{j=1}^{m-1} D_{(S-1)m+j} + \frac{3L}{2}\|z_{Sm} - z^*\|^2\Big]$$
$$\leq \frac{1-\tau_{1,0}-\tau_2}{\tau_{1,0}^2} D_0 + \frac{\tau_2 n}{\tau_{1,0}^2}\widetilde{D}^0 + \frac{3L}{2}\|z_0 - x^*\|^2 \tag{B.4}$$

Since we have $\widetilde{D}^S \leq \frac{1}{m}\sum_{j=1}^m D_{(S-1)m+j}$ which is no greater than $\frac{2\tau_{1,S-1}^2}{m}$ times the left hand side of (B.4), we conclude that

$$\mathbb{E}\big[F(\widetilde{x}^S) - F(x^*)\big] = \mathbb{E}\big[\widetilde{D}^S\big] \leq O\big(\frac{\tau_{1,S}^2}{m}\big)\cdot\Big(\frac{1-\tau_{1,0}-\tau_2}{\tau_{1,0}^2} D_0 + \frac{\tau_2 n}{\tau_{1,0}^2}\widetilde{D}^0 + \frac{3L}{2}\|z_0 - x^*\|^2\Big)$$
$$= O\big(\frac{1}{mS^2}\big)\cdot\Big(m\big(F(x_0) - F(x^*)\big) + L\|x_0 - x^*\|^2\Big) \ . \qquad \square$$

C Katyusha Pseudo-Codes in the Extended Settings

Algorithm 3 $\texttt{Katyusha}^{\mathsf{ext}}(x_0, S, \sigma, (L_1, \dots, L_n))$

$m \leftarrow n$; $\overline{L} = (L_1 + \cdots + L_n)/n$;
$\tau_2 \leftarrow \frac{1}{2}$, $\tau_1 \leftarrow \min\{\sqrt{m\sigma/3\overline{L}}, \frac{1}{2}\}$, $\alpha \leftarrow \frac{1}{3\tau_1\overline{L}}$;
$y_0 = z_0 = \widetilde{x}^0 \leftarrow x_0$;
for $s \leftarrow 0$ **to** $S-1$ **do**
  $\mu^s \leftarrow \nabla f(\widetilde{x}^s)$;
  for $j \leftarrow 0$ **to** $m-1$ **do**
    $k \leftarrow (sm) + j$;
    $x_{k+1} \leftarrow \tau_1 z_k + \tau_2 \widetilde{x}^s + (1-\tau_1-\tau_2) y_k$;
    Pick i randomly from $\{1, 2, \dots, n\}$, each with probability $L_i/n\overline{L}$;
    $\widetilde{\nabla}_{k+1} \leftarrow \mu^s + \nabla f_i(x_{k+1}) - \nabla f_i(\widetilde{x}^s)$;
    $z_{k+1} = \arg\min_z \{\frac{1}{\alpha} V_{z_k}(z) + \langle \widetilde{\nabla}_{k+1}, z\rangle + \psi(z)\}$;
      ◇ *where $V_x(y)$ is the Bregman divergence function, see Section 6.2*
    $y_{k+1} \leftarrow \arg\min_y \{\frac{3\overline{L}}{2}\|y - x_{k+1}\|^2 + \langle \widetilde{\nabla}_{k+1}, y\rangle + \psi(y)\}$;
  end for
  $\widetilde{x}^{s+1} \leftarrow \big(\sum_{j=0}^{m-1}(1+\alpha\sigma)^j\big)^{-1} \cdot \big(\sum_{j=0}^{m-1}(1+\alpha\sigma)^j \cdot y_{sm+j+1}\big)$;
end for
return $\widetilde{x}^S$.

Algorithm 4 $\texttt{Katyusha}^{\mathsf{ns,ext}}(x_0, S, \sigma, (L_1, \dots, L_n))$

$m \leftarrow n$; $\overline{L} = (L_1 + \cdots + L_n)/n$;
$\tau_2 \leftarrow \frac{1}{2}$;
$y_0 = z_0 = \widetilde{x}^0 \leftarrow x_0$;
for $s \leftarrow 0$ **to** $S-1$ **do**
  $\tau_{1,s} \leftarrow \frac{2}{s+4}$, $\alpha_s \leftarrow \frac{1}{3\tau_{1,s}\overline{L}}$
  $\mu^s \leftarrow \nabla f(\widetilde{x}^s)$;
  for $j \leftarrow 0$ **to** $m-1$ **do**
    $k \leftarrow (sm) + j$;
    $x_{k+1} \leftarrow \tau_{1,s} z_k + \tau_2 \widetilde{x}^s + (1-\tau_{1,s}-\tau_2) y_k$;
    Pick i randomly from $\{1, 2, \dots, n\}$, each with probability $L_i/n\overline{L}$;
    $\widetilde{\nabla}_{k+1} \leftarrow \mu^s + \nabla f_i(x_{k+1}) - \nabla f_i(\widetilde{x}^s)$;
    $z_{k+1} = \arg\min_z \{\frac{1}{\alpha_s} V_{z_k}(z) + \langle \widetilde{\nabla}_{k+1}, z\rangle + \psi(z)\}$;
      ◇ *where $V_x(y)$ is the Bregman divergence function, see Section 6.2*
    $y_{k+1} \leftarrow \arg\min_y \{\frac{3L}{2}\|y - x_{k+1}\|^2 + \langle \widetilde{\nabla}_{k+1}, y\rangle + \psi(y)\}$;
  end for
  $\widetilde{x}^{s+1} \leftarrow \frac{1}{m}\sum_{j=1}^{m} y_{sm+j}$;
end for
return $\widetilde{x}^S$.

References

[1] Zeyuan Allen-Zhu and Elad Hazan. Optimal Black-Box Reductions Between Optimization Objectives. In *NIPS*, 2016.

[2] Zeyuan Allen-Zhu and Elad Hazan. Variance Reduction for Faster Non-Convex Optimization. In *ICML*, 2016.

[3] Zeyuan Allen-Zhu, Yin Tat Lee, and Lorenzo Orecchia. Using optimization to obtain a width-independent, parallel, simpler, and faster positive SDP solver. In *SODA*, 2016.

[4] Zeyuan Allen-Zhu and Yuanzhi Li. Doubly Accelerated Methods for Faster CCA and Generalized Eigendecomposition. *ArXiv e-prints*, abs/1607.06017, July 2016.

[5] Zeyuan Allen-Zhu and Yuanzhi Li. Even Faster SVD Decomposition Yet Without Agonizing Pain. In *NIPS*, 2016.

[6] Zeyuan Allen-Zhu and Yuanzhi Li. Faster Principal Component Regression via Optimal Polynomial Approximation to sgn(x). *ArXiv e-prints*, abs/1608.04773, August 2016.

[7] Zeyuan Allen-Zhu, Zhenyu Liao, and Yang Yuan. Optimization Algorithms for Faster Computational Geometry. In *ICALP*, 2016.

[8] Zeyuan Allen-Zhu and Lorenzo Orecchia. Linear coupling: An ultimate unification of gradient and mirror descent. *ArXiv e-prints*, abs/1407.1537, July 2014.

[9] Zeyuan Allen-Zhu and Lorenzo Orecchia. Nearly-Linear Time Positive LP Solver with Faster Convergence Rate. In *STOC*, 2015. Newer version available at `http://arxiv.org/abs/1411.1124`.

[10] Zeyuan Allen-Zhu and Lorenzo Orecchia. Using optimization to break the epsilon barrier: A faster and simpler width-independent algorithm for solving positive linear programs in parallel. In *SODA*, 2015. Full version available at `http://arxiv.org/abs/1507.02259`.

[11] Zeyuan Allen-Zhu, Peter Richtárik, Zheng Qu, and Yang Yuan. Even Faster Accelerated Coordinate Descent Using Non-Uniform Sampling. In *ICML*, 2016.

[12] Zeyuan Allen-Zhu and Yang Yuan. Improved SVRG for Non-Strongly-Convex or Sum-of-Non-Convex Objectives. In *ICML*, 2016.

[13] Léon Bottou. Stochastic gradient descent. `http://leon.bottou.org/projects/sgd`.

[14] Cong Dang and Guanghui Lan. Randomized first-order methods for saddle point optimization. *ArXiv e-prints*, abs/1409.8625, October 2014.

[15] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives. In *NIPS*, 2014.

[16] Rong-En Fan and Chih-Jen Lin. LIBSVM Data: Classification, Regression and Multi-label. Accessed: 2015-06.

[17] 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.

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

[19] Elad Hazan and Satyen Kale. Beyond the regret minimization barrier: Optimal algorithms for stochastic strongly-convex optimization. *The Journal of Machine Learning Research*, 15(1):2489–2512, 2014.

[20] Chonghai Hu, Weike Pan, and James T Kwok. Accelerated gradient methods for stochastic optimization and online learning. In *Advances in Neural Information Processing Systems*, pages 781–789, 2009.

[21] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *NIPS*, pages 315–323, 2013.

[22] Jakub Konečnỳ, Jie Liu, Peter Richtárik, and Martin Takáč. Mini-batch semi-stochastic gradient descent in the proximal setting. *IEEE Journal of Selected Topics in Signal Processing*, 10(2):242–255, 2016.

[23] Guanghui Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1-2):365–397, January 2011.

[24] Guanghui Lan and Yi Zhou. An optimal randomized incremental gradient method. *ArXiv e-prints*, abs/1507.02000, October 2015.

[25] Hongzhou Lin. private communication, 2016.

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

[27] Qihang Lin, Zhaosong Lu, and Lin Xiao. An Accelerated Proximal Coordinate Gradient Method and its Application to Regularized Empirical Risk Minimization. In *NIPS*, pages 3059–3067, 2014.

[28] Zhaosong Lu and Lin Xiao. On the complexity analysis of randomized block-coordinate descent methods. *Mathematical Programming*, pages 1–28, 2013.

[29] Mehrdad Mahdavi, Lijun Zhang, and Rong Jin. Mixed optimization for smooth functions. In *Advances in Neural Information Processing Systems*, pages 674–682, 2013.

[30] 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.

[31] 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.

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

[33] Yurii Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, December 2005.

[34] Yurii Nesterov. Efficiency of Coordinate Descent Methods on Huge-Scale Optimization Problems. *SIAM Journal on Optimization*, 22(2):341–362, jan 2012.

[35] Atsushi Nitanda. Stochastic proximal gradient descent with acceleration techniques. In *NIPS*, pages 1574–1582, 2014.

[36] Alexander Rakhlin, Ohad Shamir, and Karthik Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. In *ICML*, 2012.

[37] Shai Shalev-Shwartz. *Online learning: Theory, algorithms, and applications*. PhD thesis, Hebrew University, 2007.

[38] Shai Shalev-Shwartz. SDCA without Duality. *arXiv preprint arXiv:1502.06177*, pages 1–7, 2015.

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

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

[41] Shai Shalev-Shwartz and Tong Zhang. Accelerated Proximal Stochastic Dual Coordinate Ascent for Regularized Loss Minimization. In *ICML*, pages 64–72, 2014.

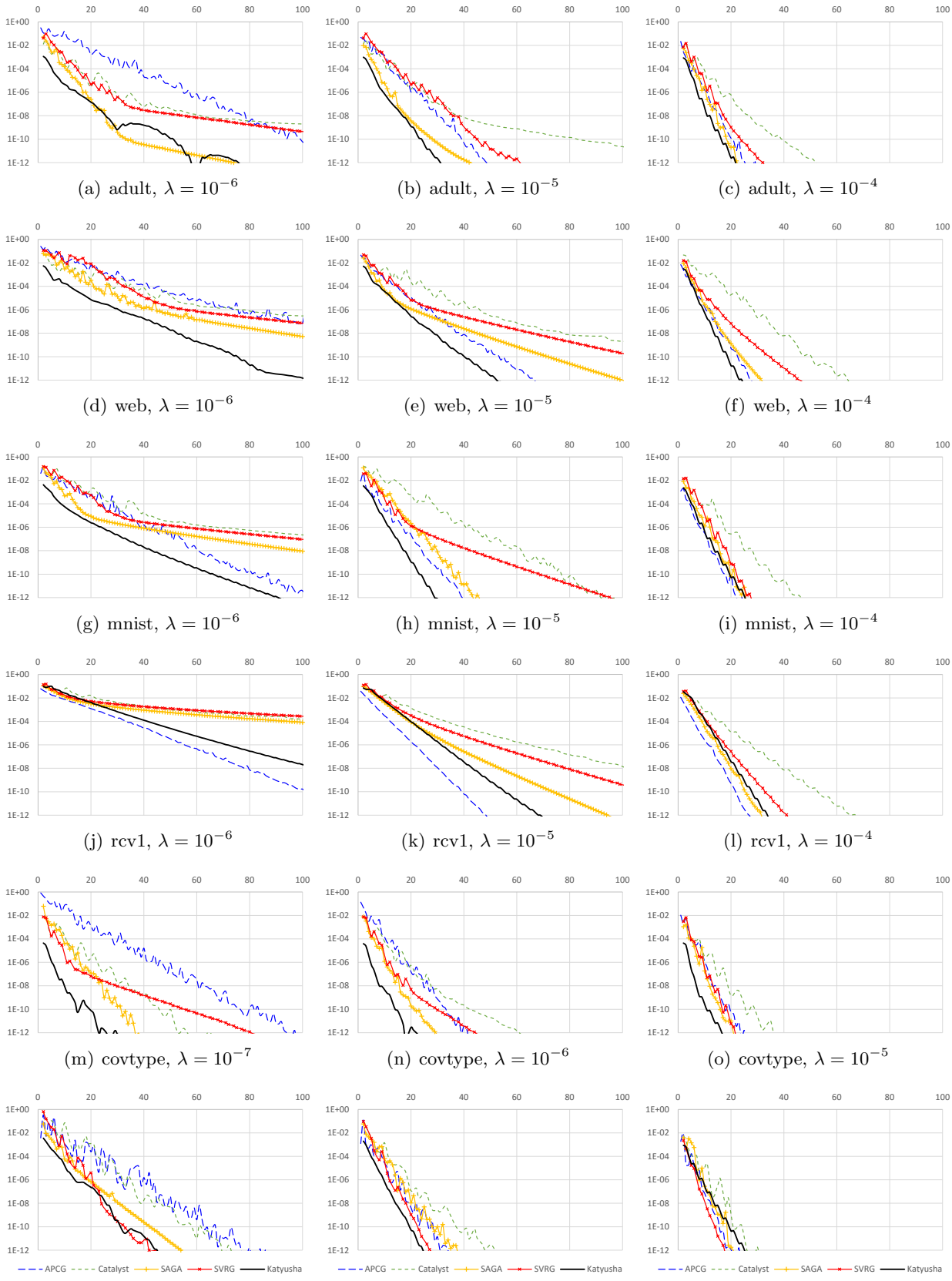
[42] Blake Woodworth and Nati Srebro. Tight Complexity Bounds for Optimizing Composite Objectives. *ArXiv e-prints*, abs/1605.08003, May 2016.

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

[44] Lijun Zhang, Mehrdad Mahdavi, and Rong Jin. Linear convergence with condition number independent access of full gradients. In *Advances in Neural Information Processing Systems*, pages 980–988, 2013.

[45] 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.

[46] Yuchen Zhang and Lin Xiao. Stochastic Primal-Dual Coordinate Method for Regularized Empirical Risk Minimization. In *ICML*, 2015.



(a) adult, $\lambda = 10^{-6}$ (b) adult, $\lambda = 10^{-5}$ (c) adult, $\lambda = 10^{-4}$

(d) web, $\lambda = 10^{-6}$ (e) web, $\lambda = 10^{-5}$ (f) web, $\lambda = 10^{-4}$

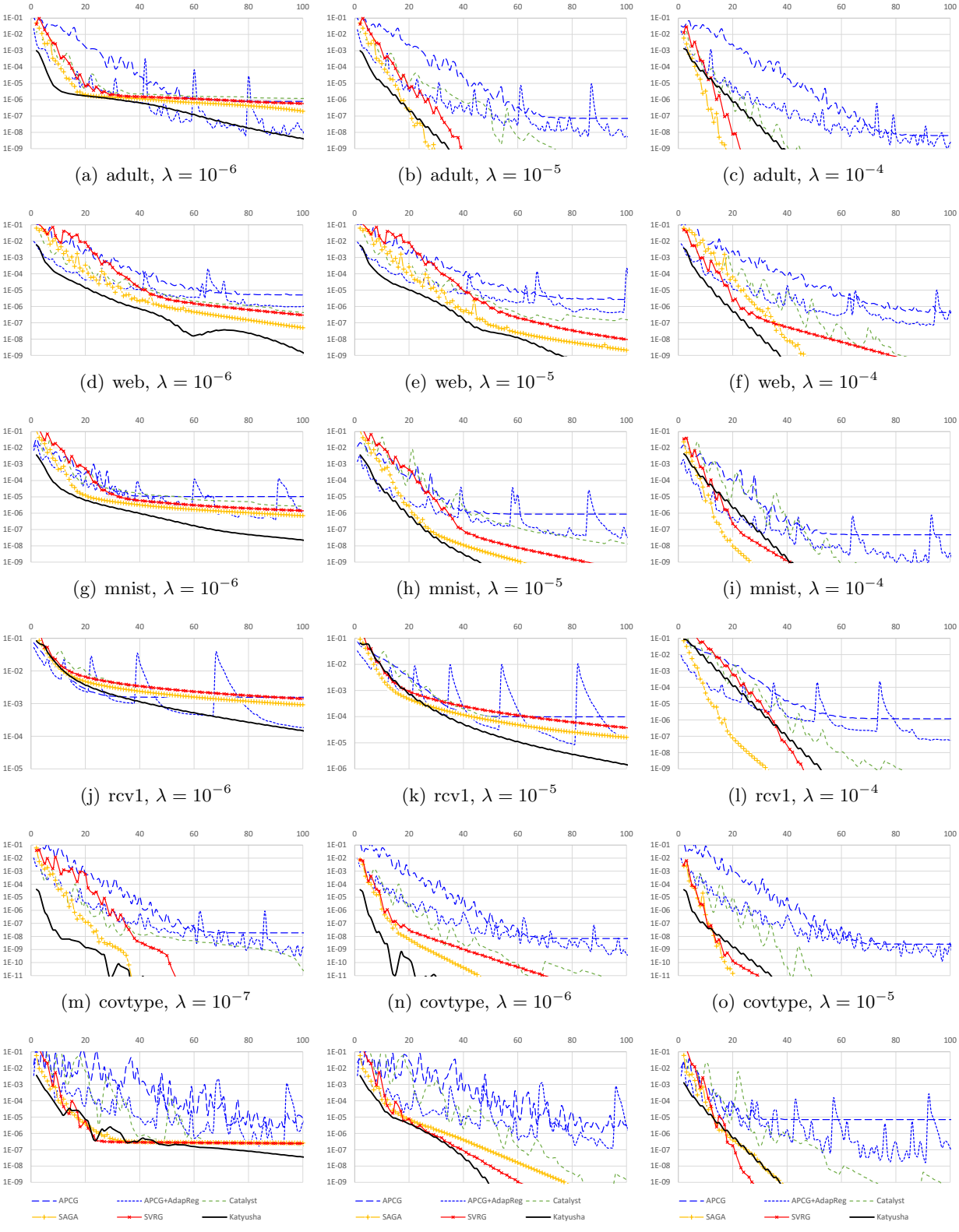
(g) mnist, $\lambda = 10^{-6}$ (h) mnist, $\lambda = 10^{-5}$ (i) mnist, $\lambda = 10^{-4}$

(j) rcv1, $\lambda = 10^{-6}$ (k) rcv1, $\lambda = 10^{-5}$ (l) rcv1, $\lambda = 10^{-4}$

(m) covtype, $\lambda = 10^{-7}$ (n) covtype, $\lambda = 10^{-6}$ (o) covtype, $\lambda = 10^{-5}$

(p) sensit, $\lambda = 10^{-6}$ (q) sensit, $\lambda = 10^{-5}$ (r) sensit, $\lambda = 10^{-4}$

Figure 3: Experiments on ridge regression with ℓ_2 regularizer weight λ.



(a) adult, $\lambda = 10^{-6}$ (b) adult, $\lambda = 10^{-5}$ (c) adult, $\lambda = 10^{-4}$

(d) web, $\lambda = 10^{-6}$ (e) web, $\lambda = 10^{-5}$ (f) web, $\lambda = 10^{-4}$

(g) mnist, $\lambda = 10^{-6}$ (h) mnist, $\lambda = 10^{-5}$ (i) mnist, $\lambda = 10^{-4}$

(j) rcv1, $\lambda = 10^{-6}$ (k) rcv1, $\lambda = 10^{-5}$ (l) rcv1, $\lambda = 10^{-4}$

(m) covtype, $\lambda = 10^{-7}$ (n) covtype, $\lambda = 10^{-6}$ (o) covtype, $\lambda = 10^{-5}$

(p) sensit, $\lambda = 10^{-6}$ (q) sensit, $\lambda = 10^{-5}$ (r) sensit, $\lambda = 10^{-4}$

Figure 4: Experiments on Lasso with ℓ_1 regularizer weight λ.