Accelerating Finite-sum Convex Optimization and Highly-smooth Convex Optimization

ZHOU, Kaiwen

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Master of Philosophy
in
Computer Science and Engineering

The Chinese University of Hong Kong
July 2019

Thesis Assessment Committee

Professor TAO Yufei (Chair)
Professor CHENG James (Thesis Supervisor)
Professor YANG Ming Chang (Committee Member)
Professor WONG Raymond Chi Wing (External Examiner)

Abstract of thesis entitled:
Accelerating Finite-sum Convex Optimization and Highly-smooth Convex Optimization
Submitted by ZHOU, Kaiwen
for the degree of Master of Philosophy in Computer Science and Engineering
at The Chinese University of Hong Kong in July 2019

Acceleration in convex optimization is, for a long time, a vivid research topic in both machine learning and optimization communities. The basic motivation of acceleration is to derive algorithms with faster worst-case convergence rates given certain oracles (e.g., gradient oracle, hessian oracle).

In recent years, due to the increasing dimensionality of optimization problems in the machine learning community, gradient descent (and its variants) become a popular choice of optimizer. The reason is that these methods are known to be dimension-free (i.e., oracle complexity independent of dimensionality). Although the optimal accelerated gradient descent has been discovered decades ago, researchers are investigating certain problem types that have potentials for designing even faster algorithms. One problem type that attracts a lot of attention recently is the finite-sum convex problem, which is commonly seen in many machine learning tasks (e.g., empirical risk minimization). Based on this problem structure, very efficient stochastic variants of gradient descent have been proposed in recent years. In this thesis, we try to derive simple and practical accelerated variants for these stochastic methods. While achieving the optimal convergence rates in this problem type, our proposed methods are as implementable as these popular stochastic methods.

On the other hand, in the optimization community, accelerations for highly-smooth (i.e., with Lipschitz continuous high-order derivative) convex problems are gathering attention recently. Although many accelerated methods under this setting were proposed, there is still a lack of interpretation for these methods. In the work of Allen-Zhu and Hazan, Nesterov's accelerated method is interpreted as coupling gradient descent and mirror descent, which is called Linear Coupling. Their work provides an intuitive understanding of the source of first-order acceleration. However, Linear Coupling only works in the first-order setting. In this thesis, we consider extending Linear Coupling into a broader scope. We show that many accelerated high-order methods can also be formulated as coupling two sequences of steps and be cast into a unified analysis framework, which we call High-order Linear Coupling. Based on this framework, we provide similar intuition towards high-order acceleration and also identify potential improvements and open problems.

摘要：

凸優化中的加速一直是機器學習和優化中的熱門研究課題。加速的基本目的是在給定某些信息的情況下（例如，梯度信息，海森矩陣信息），推導出具有更快的最壞情況收斂速度的算法。

近年來，由於機器學習中優化問題的維數的劇烈增加，梯度下降（及其變體）已經成爲了優化器的首選。其原因是這些方法的迭代複雜度與維數無關。儘管最優的加速梯度下降算法早在幾十年前已被提出，研究人員仍在探索一些具有設計更快算法的潛力的問題類型。有限和凸優化問題是最近非常熱門的一類問題。這類問題結構在機器學習的任務中（例如，經驗風險最小化）很常見。基於該問題結構，近年來有許多非常高效的梯度下降的隨機變體被提出。在本論文中，我們嘗試加速這些熱門的隨機梯度下降方法。我們提出的方法在實現了最優收斂速度的同時，維持了與這些熱門隨機方法同樣的可實施性。

另一方面，在優化領域中，高度平滑的凸優化問題（即擁有Lipschitz連續的高階導數）的加速算法受到了研究者們的關注。雖然有許多這種場景下的加速方法已被提出，我們仍然缺乏一個對這些方法的直觀解釋。在Allen-Zhu和Hazan的工作中，Nesterov的加速方法被解釋爲梯度下降和鏡像下降的一種耦合。他們稱之爲線性耦合。他們的工作提供了對一階加速的一種直觀理解。然而，線性耦合僅僅適用於一階場景中。在本論文中，我們考慮將線性耦合推廣到高階場景中。我們證明了許多加速的高階方法也可以表示爲兩個迭代序列的耦合。因此，我們提出了一個統一的分析框架，叫做高階線性耦合。基於這個框架，我們提供了類似的對高階加速算法的直觀理解，並指出了一些高階優化中的潛在改進和開放問題。

Acknowledgments

I would like to thank my supervisor, Prof. James Cheng, for giving me the chance of pursuing MPhil degree in his lab. Without this chance, I would never be able to promote myself to be a researcher. I would also like to specially thank Prof. Fanhua Shang for helping me at the early stage of my research and involving me in some of his great works.

I also learned a lot from my teammates in the machine learning group. I would like to express my gratitude to (in alphabetical order) Xinyan Dai, Qinghua Ding, Jinfeng Li, Jie Liu, Kelvin Ng, Xiao Yan, Han Yang for their very useful discussions. In addition, it is my pleasure to have the teammates in the system group to be my close friends.

Last but not the least, I give my special thanks to my parents, Guizi Zhou and Meiqin Zhou, for their financial support and encouragement during my MPhil study. No words could express my gratitude. I can only try my best to be a good son, and love you back in the same way you love me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the last half-century, a great amount of literature has been devoted to convex optimization. If no specific problem structure is assumed other than convexity, very strong results are known for first-order methods on both the upper-bound (i.e., accelerated algorithms) and the lower-bound side (e.g, [29, 31]). Due to these results, a widely held attitude is that convex problems are well-solved.

However, if we can utilize certain problem structures, the lower bounds no longer hold. Thus, it is still a popular direction in convex optimization to design faster algorithms by exploiting the problem structure. We would always expect the assumptions on the problem structure to be general, and thus applicable for a wide range of practical problems. One typical assumption is the finite-sum structure, which has attracted a lot of attention in recent years:

$$F(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x).$$

This structure is prevalent in machine learning and statistics such as the regularized empirical risk minimization (ERM) (e.g., $\ell 2$-logistic regression, ridge regression, LASSO, $\ell 2$-SVM). By carefully exploiting this structure, striking results were shown based on the technique called variance reduction (the first work is [41]). The significance is that the stochastic variance reduced methods achieve the convergence rates of deterministic methods (i.e., gradient descent (GD)) while enjoying the lower per-iteration complexity as that of stochastic gradient descent (SGD). This huge improvement motivates researchers to establish upper bounds and lower bounds for convex problems with finite-sum structure.

Another potential in convex optimization is to study higher-order methods. While the algorithms derived in this case may not be as implementable as first-order meth-

ods due to high memory complexities and restricted problem types, they are important in the development of optimization theory. Perhaps surprisingly, unlike the first-order case, the optimal second-order (or higher-order) method is still unknown. We have near optimal algorithms (i.e., optimal up to log factors) but they are not perfect from many perspectives (e.g., high per-iteration complexity and extremely complicated algorithm structure). Also, higher-order methods seem to be more restricted than first-order methods (e.g., restrictions on the problem domain). Thus, there are still many interesting open problems in this line of research.

This thesis discusses both directions mentioned above. Specifically, in the direction of exploiting finite-sum structure, we propose two accelerated methods based on two popular stochastic variance reduced methods, respectively. In another direction on highly-smooth convex optimization, we provide a general framework for many accelerated methods in this case and discuss some potential improvements.

Thesis organization.

The rest of the thesis is organized as follows. Chapter 2 discusses finite-sum convex optimization, in which we first review some basics and the classic algorithms and then introduce accelerated methods based on them. Chapter 3 focuses on highly-smooth convex optimization, where we first propose a general acceleration framework and then by casting many accelerated algorithms as instances, we provide a unified analysis and an intuitive interpretation to them. Chapter 4 concludes this thesis.

1.1 Publications related to this thesis

Some content in this thesis has been published in conference proceedings. Chapter 2 is based on my publications [50, 51] during the Master of Philosophy period. Precisely, the MiG method is published in [51] and the SSNM method is in [50]. Chapter 3 is based on my unpublished scripts written during the Master of Philosophy period and the key contribution is a structured interpretation to high-order accelerations.

Chapter 2

Finite-sum Convex Optimization

Problems with finite-sum structure can be written as (informal)

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

Assuming smoothness and strong convexity for each $f_i(\cdot)$, traditional analysis shows that GD yields a fast linear convergence rate (i.e., iteration number required to achieve ϵ sub-optimality $\propto \log(1/\epsilon)$) but with a high per-iteration cost (i.e., n calculations of $\nabla f_i(\cdot)$ per iteration), and thus may not be suitable for problems with a very large n. As an alternative for large-scale problems, SGD [40] uses only one or a mini-batch of gradients in each iteration, and thus enjoys a significantly lower per-iteration complexity than GD. However, due to the undiminished variance of the gradient estimator, vanilla SGD is shown to yield only a sub-linear convergence rate. Recently, stochastic variance reduced methods (notably SAG [41], SVRG [16], SAGA [11], and their proximal variants, such as [43], [47] and [17]) were proposed to solve this type of problems. All these methods are equipped with various variance reduction techniques, which help them achieve low per-iteration complexities comparable with SGD and at the same time maintain the same linear convergence rate as GD.

In this chapter, we first review some popular stochastic variance reduced methods and then introduce the acceleration techniques for these classic methods.

2.1 Preliminaries

In this chapter, we consider problems in standard Euclidean space with the Euclidean norm denoted by $\|\cdot\|$. We use $\mathbb{E}_i$ to denote that the expectation is taken

with respect to sample i conditioned on all previous randomness and $\mathbb{E}$ is to all randomness. We consider optimizing the following composite finite-sum problem, which arises frequently in machine learning and statistics such as supervised learning and regularized empirical risk minimization (ERM):

$$\min_{x\in\mathbb{R}^d}\Big\{F(x)\triangleq f(x)+h(x)\Big\}, \tag{2.1}$$

where $f(x)=\frac{1}{n}\sum_{i=1}^n f_i(x)$ is an average of n continuously differentiable and convex function $f_i(x)$, and $h(x)$ is a simple and convex (but possibly non-differentiable) function. We assume the proximal operator of $h(x)$ is easy to compute, which is

$$\text{prox}_h^\eta(v)\triangleq\operatorname*{argmin}_{x\in\mathbb{R}^d}\left\{h(x)+\frac{1}{2\eta}\|x-v\|^2\right\},\forall v\in\mathbb{R}^d.$$

Detailed interpretation and examples of proximal operator can be found in [36]. $x^\star$ denotes one solution of Problem (2.1). Here, we also define $F_i(x)=f_i(x)+h(x)$ with $\nabla F_i(x)=\nabla f_i(x)+\partial h(x)$ and $\partial h(x)$ denotes a sub-gradient of $h(\cdot)$ at x, which will be used in this chapter.

In order to further categorize the objective functions, we define that a convex function $f:\mathbb{R}^d\to\mathbb{R}$ is said to be L-smooth if for all $x,y\in\mathbb{R}^d$, it holds that

$$f(x)\le f(y)+\langle\nabla f(y),x-y\rangle+\frac{L}{2}\|x-y\|^2, \tag{2.2}$$

and μ-strongly convex if for all $x,y\in\mathbb{R}^d$,

$$f(x)\ge f(y)+\langle\mathcal{G},x-y\rangle+\frac{\mu}{2}\|x-y\|^2, \tag{2.3}$$

where $\mathcal{G}\in\partial f(y)$, the set of sub-gradient of $f(\cdot)$ at y for non-differentiable $f(\cdot)$. If $f(\cdot)$ is differentiable, we can simply replace $\mathcal{G}\in\partial f(y)$ with $\mathcal{G}=\nabla f(y)$. Then we make the following assumptions to identify the main objective condition (strongly convex) that is the focus of this chapter:

Assumption 1. *In Problem* (2.1)*, each* $f_i(\cdot)$[1] *is* L*-smooth and convex,* $h(\cdot)$ *is* μ*-strongly convex.*

Assumption 2. *In Problem* (2.1)*, each* $f_i(\cdot)$ *is* L*-smooth and convex,* $h(\cdot)$ *is convex and* $F(\cdot)$ *is* μ*-strongly convex.*

[1]In fact, if each $f_i(\cdot)$ is L-smooth, the averaged function $f(\cdot)$ is itself L-smooth — but probably with a smaller L. We keep using L as the smoothness constant for a consistent analysis.

Assumption 3. *In Problem* (2.1)*, each* $f_i(\cdot)$ *is* L*-smooth and* μ*-strongly convex,* $h(\cdot)$ *is convex.*

The strong convexity assumptions, which are used for different algorithms, have some subtle differences due to certain analytical requirements. We will discuss this subtlety for the proposed methods at 2.3.2 and 2.3.4.

We denote $\kappa \triangleq \frac{L}{\mu}$ throughout this chapter, which is known as the condition number of an L-smooth and μ-strongly convex function.

Oracle complexity in this chapter, denoted by $\mathcal{O}(\cdot)$, is the number of calls to Incremental First-order Oracle (IFO) (i.e., $\nabla f_i(\cdot)$) + Proximal operator Oracle (PO).

We focus on achieving a highly accurate solution (very small ϵ) for Problem (2.1), although for practical optimization tasks, such as supervised learning, low empirical risk may result in a high generalization error. In this chapter, we treat Problem (2.1) as a pure optimization problem.

2.2 Variance reduced methods

Among the recently proposed stochastic variance reduced methods (e.g., SAG [41], SVRG [16], SAGA [11], SDCA [44], S2GD [18], SARAH [34]), SVRG and SAGA are the two most recognized and widely used algorithms. This is due to their simplicity in both algorithm structure and theoretical analysis, and also since their gradient estimators are unbiased, SVRG and SAGA are more intuitive, in the sense that they can be understood as the control variate, a well-known technique in Monte-Carlo simulation [42].

In this section, we review some basic constructions and convergence results of SVRG and SAGA. First, recall that if we want to estimate an unknown expectation of a random variable X, suppose that we have another random variable Y, whose expectation $\mathbb{E}\big[Y\big]$ is known, we can construct an unbiased estimation of $\mathbb{E}\big[X\big]$ as

$$X_Y = X - Y + \mathbb{E}\big[Y\big]. \tag{2.4}$$

This is known as the control variate technique [42]. The variance of X_Y has the form $\mathbb{V}[X_Y] = \mathbb{V}[X] + \mathbb{V}[Y] - 2 \cdot Cov(X, Y)$, where $Cov(\cdot, \cdot)$ is the covariance. Thus, it is clear that $\mathbb{V}[X_Y]$ is smaller when the control variate Y is more positively correlated to X.

Based on this technique, suppose that now X is a stochastic gradient $\nabla f_i(x)$, we want to find some Y that can sufficiently control the variance of $\nabla f_i(x)_Y$. A natural idea is that we can choose some previously calculated gradient estimators as Y, since we want them to be highly correlated. This idea forms the basic constructions of

SVRG and SAGA. They choose Y in different ways and carefully ensure that $\mathbb{E}\big[Y\big]$ is accessible. In the next two subsections, we review them individually in details.

2.2.1 SVRG

Algorithm 1 Prox-SVRG [47]

Input: Initial guess $x_0 \in \mathbb{R}^d$, learning rate η, epoch number $\mathcal{S}$, epoch length m.
Initialize: $\tilde{x}_0 = x_0^0 = x_0$.
1: **for** $s = 0, \dots, \mathcal{S} - 1$ **do**
2: Compute and store $\nabla f(\tilde{x}_s)$.
3: **for** $k = 0, \dots, m - 1$ **do**
4: Sample i_k uniformly from $\{1, \dots, n\}$.
5: $\widetilde{\nabla}_{x_k}^{(1)} = \nabla f_{i_k}(x_k^s) - \nabla f_{i_k}(\tilde{x}_s) + \nabla f(\tilde{x}_s)$.
6: $x_{k+1}^s = \operatorname{argmin}_{x \in \mathbb{R}^d} \Big\{ h(x) + \langle \widetilde{\nabla}_{x_k}^{(1)}, x \rangle + \frac{1}{2\eta} \|x - x_k^s\|^2 \Big\}$.
7: **end for**
8: $x_0^{s+1} = \tilde{x}_{s+1} = \frac{1}{m} \sum_{i=1}^m x_i^s$.
9: **end for**
Output: $\tilde{x}_{\mathcal{S}}$.

Regarding the control variate construction (2.4), SVRG chooses Y as the stochastic gradient $\nabla f_i(\tilde{x})$ evaluated at a randomly previously calculated point $\tilde{x}$ [16], and thus $\mathbb{E}\big[Y\big] = \nabla f(\tilde{x})$. Suppose at the current iterate, our X in (2.4) is a stochastic gradient $\nabla f_{i_k}(x)$ calculated using sample i_k. Recall that we need to make X and Y as positively correlated as possible, Y should be chosen as $\nabla f_{i_k}(\tilde{x})$ for this iterate. Thus, we obtain the following gradient estimator of SVRG (denoted as $\widetilde{\nabla}_{(\cdot)}^{(1)}$):

$$\widetilde{\nabla}_x^{(1)} = \nabla f_{i_k}(x) - \nabla f_{i_k}(\tilde{x}) + \nabla f(\tilde{x}).$$

However, we cannot use the same $\tilde{x}$ for the whole optimization process, since as x moves "far away" from $\tilde{x}$, $\nabla f_{i_k}(x)$ becomes less correlated to $\nabla f_{i_k}(\tilde{x})$, and thus the variance of $\widetilde{\nabla}^{(1)}$ will be very large. SVRG resolves this problem by regularly updating $\tilde{x}$ [16], and thus it has a two-loop algorithm structure as shown in Algorithm 1 (here we bring the proximal variant of SVRG [47] to tackle the potentially non-smooth $h(\cdot)$ in Problem (2.1) and we choose a uniform sampling scheme for simplicity).

Analysis in [47] shows that Algorithm 1 satisfies the following theorem at the output point:

Theorem 2.2.1 (Theorem 1 in [47]). *Suppose Assumption 2 holds, if* $0 < \eta < \frac{1}{4L}$ *and* m *is sufficiently large so that*

$$\rho = \frac{1}{\mu\eta(1-4L\eta)m} + \frac{4L\eta(m+1)}{(1-4L\eta)m} < 1. \tag{2.5}$$

Then the Prox-SVRG method in Algorithm 1 has geometric convergence in expectation:

$$\mathbb{E}\big[F(\tilde{x}_{\mathcal{S}}) - F(x^\star)\big] \leq \rho^{\mathcal{S}} \left[F(x_0) - F(x^\star)\right].$$

For (2.5), we can simply choose $\eta = 1/10L$ and $m = \lceil 100\kappa \rceil + 4$ to get $\rho \leq 5/6$. Based on this choice, we can estimate the oracle complexity of Algorithm 1: in order to achieve ϵ sub-optimality (i.e., $\mathbb{E}\big[F(\tilde{x}_{\mathcal{S}}) - F(x^\star)\big] \leq \epsilon$), we need totally $\mathcal{S} = O\left(\log((F(x_0) - F(x^\star))/\epsilon)\right)$ epochs. Note that for each epoch, we need to calculate $n + m = O(n + \kappa)$ stochastic gradients. Thus, the overall oracle complexity for Prox-SVRG is

$$\mathcal{O}\left((n+\kappa)\log\left(\frac{F(x_0) - F(x^\star)}{\epsilon}\right)\right).$$

2.2.2 SAGA

Algorithm 2 SAGA [11]

Input: Initial guess x_0, learning rate η, iterations number K.

Initialize: "Gradients" table with $\nabla f_i(\phi_i^0) = \nabla f_i(x_0)$ for each $i \in \{1 \dots n\}$ and a running average for the "gradients" table.

1: **for** $k = 0, 1, \dots, K-1$ **do**

2: 1. Sample i_k uniformly in $\{1, \dots, n\}$ and compute the gradient estimator using the running average.

3: $\widetilde{\nabla}_{x_k}^{(2)} = \nabla f_{i_k}(x_k) - \nabla f_{i_k}(\phi_{i_k}^k) + \frac{1}{n}\sum_{i=1}^n \nabla f_i(\phi_i^k)$.

4: 2. Perform a proximal gradient step.

5: $x_{k+1} = \mathrm{argmin}_{x\in\mathbb{R}^d} \left\{ h(x) + \langle \widetilde{\nabla}_{x_k}^{(2)}, x\rangle + \frac{1}{2\eta}\|x_k - x\|^2 \right\}$.

6: 3. Take $\nabla f_{i_k}(\phi_{i_k}^{k+1}) = \nabla f_{i_k}(x_k)$. All other entries in the "gradients" table remain unchanged. Update the running average.

7: **end for**

Output: x_K

Unlike SVRG, SAGA constructs its control variates (2.4) in a different way. It chooses Y as a stochastic gradient that is previously calculated. As in SVRG, X

is a stochastic gradient $\nabla f_{i_k}(x)$ calculated using sample i_k. For SAGA, if we want to choose Y that is as positively correlated to X as possible, we need to pick a previously calculated stochastic gradient $\nabla f_{i_k}(y)$, which is (1) evaluated at sample i_k and, (2) the point y should not be too "far away" from the current iterate. Thus, in SAGA, Y is chosen as the $\nabla f_{i_k}(\cdot)$ that is lastly evaluated in previous iterates. In this case, SAGA maintains a table of gradients $\nabla f_i(\phi_i^k)$ for each $i \in \{1 \dots n\}$, where ϕ_i^k represents the latest position $\nabla f_i(\cdot)$ is evaluated at. Note that $\mathbb{E}\big[Y\big]$ is the average of all the gradients in the table for each iterate, which can be updated on the fly. The stochastic gradient estimator of SAGA (denoted as $\widetilde{\nabla}^{(2)}_{(\cdot)}$):

$$\widetilde{\nabla}^{(2)}_x = \nabla f_{i_k}(x) - \nabla f_{i_k}(\phi_{i_k}^k) + \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\phi_i^k).$$

Theorem 2.2.2 (Theorem 1 & Corollary 1 in [11])**.** *Suppose Assumption 3 holds, by choosing $\eta = 1/(2(\mu n + L))$, the following inequality holds in expectation:*

$$\begin{aligned}&\mathbb{E}\big[\|x_K - x^\star\|^2\big] \\ &\leq \left(1 - \frac{\mu}{2(\mu n + L)}\right)^K \left(\|x_0 - x^\star\|^2 + \frac{n}{\mu n + L}\left(f(x_0) - f(x^\star) - \langle \nabla f(x^\star), x_0 - x^\star\rangle\right)\right).\end{aligned}$$

The above theorem depicts the convergence rate of SAGA. In terms of oracle complexity to achieve ϵ sub-optimality, SAGA also converges at the rate of

$$\mathcal{O}\left((n + \kappa)\log\frac{D_0}{\epsilon}\right),$$

where $D_0 = \|x_0 - x^\star\|^2 + \frac{n}{\mu n + L}\left(f(x_0) - f(x^\star) - \langle \nabla f(x^\star), x_0 - x^\star\rangle\right)$. In comparison with Theorem 2.2.1, the guarantees hold on different objectives, but there is only constants difference since we have $F(x) - F(x^\star) \geq \frac{\mu}{2}\|x - x^\star\|^2$ based on strong convexity.

2.3 Accelerations

Inspired by the acceleration technique proposed in Nesterov's accelerated gradient descent [33], accelerated variants of stochastic variance reduced methods have been proposed in recent years, such as Acc-Prox-SVRG [35], APCG [24], APPA [13], Catalyst [23], SPDC [49] and Katyusha [2]. Among these algorithms, APPA and Catalyst achieve acceleration by using some carefully designed reduction techniques, which,

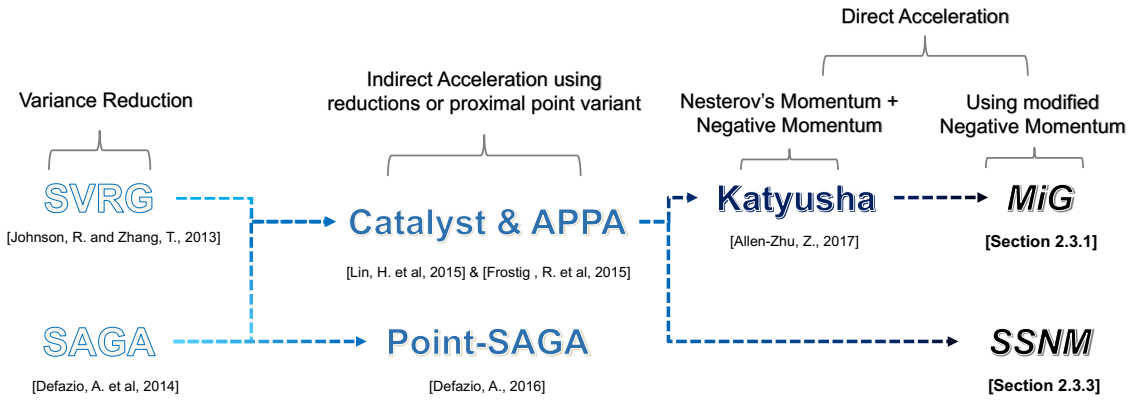

Figure 2.1: The roadmap of accelerated stochastic variance reduced methods.

however, result in additional log factors in their overall oracle complexities (near optimal). Katyusha, as the first directly accelerated variant of SVRG, introduced the idea of negative momentum (or Katyusha momentum): regarding the gradient estimator of SVRG

$$\widetilde{\nabla}_x^{(1)} = \nabla f_i(x) - \nabla f_i(\tilde{x}) + \nabla f(\tilde{x}),$$

the negative momentum is a $(\tilde{x} - x)$ offset added (with decay) to each update in this epoch. One can interpreted it as the momentum provided by a previously randomly computed point. Then, by combining it with Nesterov's momentum, Katyusha yields the best known[2] oracle complexity $\mathcal{O}((n + \sqrt{\kappa n}) \log(1/\epsilon))$ for strongly convex problems. Note that a lower bound $\Omega((n + \sqrt{\kappa n}) \log(1/\epsilon))$ of gradient evaluations for this type of problems has been proved in [20], and thus Katyusha is optimal up to a constant factor.

In this chapter, we consider a more refined usage of the negative momentum proposed in Katyusha. In Katyusha, the acceleration is interpreted as using a hybrid of negative momentum and Nesterov' momentum [2]. Thus, Katyusha has the following 3-points coupling scheme in each iteration (in the original notations):

$$x_{k+1} = \tau_1 z_k + \tau_2 \tilde{x}^s + (1 - \tau_1 - \tau_2) y_k.$$

In the next subsection, we show that the coupling steps $\{y_k\}$, which are understood as the Nesterov's momentum part of Katyusha, are not necessary to achieve the

[2]According to [5], this rate can only be attained when μ is known. Without knowing μ, the best known rate is $\mathcal{O}\left((n + \kappa) \log(1/\epsilon)\right)$ achieved by [22] and [48]. We assume μ is known throughout this chapter.

acceleration. After eliminating the sequence $\{y_k\}$, we derived a much simpler and elegant accelerated method, which we call MiG.

Inspired by the simple acceleration techniques used in MiG, we further extended the techniques to SAGA and proposed the first directly accelerated variant of SAGA, which we call SSNM. We depict the development of accelerated stochastic variance reduced methods in Figure 2.3.

2.3.1 MiG: Simple and scalable accelerated SVRG

Algorithm 3 MiG

Input: Learning rate $\eta = \begin{cases} \sqrt{\frac{1}{3\mu m L}} & \text{if } \frac{m}{\kappa} \leq \frac{3}{4}, \\ \frac{2}{3L} & \text{if } \frac{m}{\kappa} > \frac{3}{4}. \end{cases}$, parameter $\theta = \begin{cases} \sqrt{\frac{m}{3\kappa}} & \text{if } \frac{m}{\kappa} \leq \frac{3}{4}, \\ \frac{1}{2} & \text{if } \frac{m}{\kappa} > \frac{3}{4}. \end{cases}$,
epoch number $\mathcal{S}$, epoch length $m = \Theta(n)$, initial guess x_0.
Initialize: $\tilde{x}_0 = x_0^0 = x_0$, $\omega = 1 + \eta\mu$;
for $s = 0, \ldots, \mathcal{S} - 1$ **do**
  Compute and store $\nabla f(\tilde{x}_s)$.
  for $k = 0, \ldots, m - 1$ **do**
    Sample i_k uniformly in $\{1 \ldots n\}$.
    $y_k = \theta x_k^s + (1 - \theta)\tilde{x}_s$. //temp variable y
    $\widetilde{\nabla}_{y_k}^{(1)} = \nabla f_{i_k}(y_k) - \nabla f_{i_k}(\tilde{x}_s) + \nabla f(\tilde{x}_s)$.
    $x_{k+1}^s = \text{argmin}_{x \in \mathbb{R}^d} \left\{ \frac{1}{2\eta} \|x - x_k^s\|^2 + \langle \widetilde{\nabla}_{y_k}^{(1)}, x \rangle + h(x) \right\}$.
  end for
  $\tilde{x}_{s+1} = \theta \big(\sum_{k=0}^{m-1} \omega^k \big)^{-1} \sum_{k=0}^{m-1} \omega^k x_{k+1}^s + (1 - \theta)\tilde{x}_s$.
  $x_0^{s+1} = x_m^s$.
end for
Output: $\tilde{x}_{\mathcal{S}}$.

We formally introduce MiG in Algorithm 3. In order to further illustrate some ideas behind the algorithm structure, we make the following remarks:

- *Temp variable y.* As we can see in Algorithm 3, y is a convex combination of x and $\tilde{x}$ with the parameter θ. So for implementation, we do not need to keep track of y in the whole inner loop. For the purpose of giving a clean proof, we mark y with iteration number k.

- *Fancy update for $\tilde{x}_s$.* One can easily verify that this update for $\tilde{x}_s$ is equivalent to using ω^k weighted averaged y_{k+1} to update $\tilde{x}_s$, which is written as: $\tilde{x}_s =$

$\left(\sum_{k=0}^{m-1}\omega^k\right)^{-1}\sum_{k=0}^{m-1}\omega^k y_{k+1}$. Since we only keep track of x, we adopt this expended fancy update for $\tilde{x}_s$ — but it is still quite simple in implementation.

- *Choice of* x_0^{s+1}. In recent years, some existing stochastic algorithms such as [16, 47] choose to use $\tilde{x}_s$ as the initial vector for new epoch. For MiG, when using $\tilde{x}_s$, the overall oracle complexity will degenerate to a non-accelerated one for some ill-conditioned problems, which is $\mathcal{O}\big((n+\kappa)\log(1/\epsilon)\big)$. It is reported that even in practice, using the last iterate yields a better performance as discussed in [3].

Scalability of MiG: An sparse and asynchronous variant.

Algorithm	Complexity	Memory	S&A
SVRG	$\mathcal{O}\big((n+\kappa)\log\frac{1}{\epsilon}\big)$	1 Vector	$\surd$
SAGA	$\mathcal{O}\big((n+\kappa)\log\frac{1}{\epsilon}\big)$	1 Vector, 1∇ Table	$\surd$
Katyusha	$\mathcal{O}\big((n+\sqrt{\kappa n})\log\frac{1}{\epsilon}\big)$	2 Vectors	$\times$
MiG	$\mathcal{O}\big((n+\sqrt{\kappa n})\log\frac{1}{\epsilon}\big)$	1 Vector	$\surd$

Table 2.1: Comparison of different stochastic variance reduced algorithms. ("Complexity" is for strongly-convex problems. "Memory" is those used to store variables."S&A" refers to efficient (lock-free) Sparse & Asynchronous variant.)

Inspired by emerging multi-core computer architectures, asynchronous variants of the above stochastic gradient methods have been proposed in recent years, e.g., Hogwild! [38], Lock-Free SVRG [39], KroMagnon [26] and ASAGA [21]. Among them, KroMagnon and ASAGA (as the sparse and asynchronous variants of SVRG and SAGA) enjoy a fast linear convergence rate for strongly convex objectives. However, there still lacks a variant of accelerated algorithms in these settings.

The main issue for those accelerated algorithms is that most of their algorithm designs (e.g., [2] and [23]) involve tracking at least two highly correlated coupling vectors[3] (in the inner loop). This kind of algorithm structure prevents us from deriving efficient (lock-free) asynchronous sparse variants for those algorithms. More critically, when the number of concurrent threads is large (e.g., 20 threads), the high

[3]Here we refer to the number of variable vectors involved in one update.

perturbation (i.e., updates on shared variables from concurrent threads) may even destroy their convergence guarantees. Thanks to the simplicity of MiG, we are able to derive efficient sparse and asynchronous variant for it. The variant and its analysis is formally given in A. We summarize some stochastic variance reduced algorithms in Table 2.1.

2.3.2 Convergence analysis of MiG

To start the theoretical analysis of MiG, we first give the variance bound of its stochastic gradient estimator, which is identical to the one in Katyusha.

Lemma 2.3.1. (Variance Bound for MiG) *Suppose Assumption 1 holds, we can upper bound the variance of* $\widetilde{\nabla}_{y_k}^{(1)}$ *as*

$$\mathbb{E}_{i_k}\big[\|\nabla f(y_k) - \widetilde{\nabla}_{y_k}^{(1)}\|^2\big] \leq 2L\big(f(\tilde{x}_s) - f(y_k) - \langle \nabla f(y_k), \tilde{x}_s - y_k\rangle\big).$$

Proof. This lemma is identical to Lemma 3.4 in [2], which provides a tighter upper bound on the gradient estimator variance than those in [16, 47]. □

In order to prove the convergence of both MiG and SSNM (in subsection 2.3.3), we also need the following useful lemma, which can be regarded as using the 3-point equality of Bregman divergence in the Euclidean norm setting:

Lemma 2.3.2. *If two vectors* x_{k+1}, $x_k \in \mathbb{R}^d$ *satisfy* $x_{k+1} = \operatorname{argmin}_{x\in\mathbb{R}^d}\{h(x) + \langle\widetilde{\nabla}, x\rangle + \frac{1}{2\eta}\|x_k - x\|^2\}$ *with a vector* $\widetilde{\nabla}$ *and a* μ*-strongly convex function* $h(\cdot)$*, then for all* $u \in \mathbb{R}^d$*, we have*

$$\langle\widetilde{\nabla}, x_{k+1}-u\rangle \leq -\frac{1}{2\eta}\|x_{k+1}-x_k\|^2+\frac{1}{2\eta}\|x_k-u\|^2-\frac{1+\eta\mu}{2\eta}\|x_{k+1}-u\|^2+h(u)-h(x_{k+1}).$$

Proof. This Lemma is identical to Lemma 3.5 in [2], and hence the proof is omitted. □

Then we formally give the convergence rate of MiG in terms of oracle complexity as follows.

Theorem 2.3.3. *Suppose Assumption 1 holds,* MiG *achieves an* ϵ*-additive error with the following oracle complexities in expectation:*

$$\begin{cases} \mathcal{O}\big(\sqrt{\kappa n}\log\frac{F(x_0)-F(x^\star)}{\epsilon}\big), & \text{if } \ \frac{m}{\kappa} \leq \frac{3}{4}, \\ \mathcal{O}\big(n\log\frac{F(x_0)-F(x^\star)}{\epsilon}\big), & \text{if } \ \frac{m}{\kappa} > \frac{3}{4}. \end{cases}$$

In other words, the overall oracle complexity of MiG *is* $\mathcal{O}\big((n+\sqrt{\kappa n})\log\frac{F(x_0)-F(x^\star)}{\epsilon}\big)$.

Proof. In order to give a clean proof, we omit the superscripts for iterates in the same epoch s as x_k instead of x_k^s unless otherwise specified.

First, we add the following constraint on the parameters η and θ, which is crucial in the proof of Theorem 2.3.3:

$$L\theta + \frac{L\theta}{1-\theta} \leq \frac{1}{\eta} \text{ , or equivalently } \eta \leq \frac{1-\theta}{L\theta(2-\theta)}. \tag{2.6}$$

We start with convexity of $f(\cdot)$ at y_k. By definition, for any vector $u \in \mathbb{R}^d$, we have

$$\begin{aligned} f(y_k) - f(u) &\leq \langle \nabla f(y_k),\ y_k - u \rangle \\ &= \langle \nabla f(y_k),\ y_k - x_k \rangle + \langle \nabla f(y_k),\ x_k - u \rangle \\ &\overset{(\star)}{=} \frac{1-\theta}{\theta} \langle \nabla f(y_k),\ \tilde{x}_s - y_k \rangle + \langle \nabla f(y_k),\ x_k - u \rangle, \end{aligned} \tag{2.7}$$

where $(\star)$ follows from the fact that $y_k = \theta x_k + (1-\theta)\tilde{x}_s$.

Then we further expand $\langle \nabla f(y_k),\ x_k - u \rangle$ as

$$\langle \nabla f(y_k),\ x_k - u \rangle = \langle \nabla f(y_k) - \widetilde{\nabla}_{y_k}^{(1)},\ x_k - u \rangle + \langle \widetilde{\nabla}_{y_k}^{(1)},\ x_k - x_{k+1} \rangle + \langle \widetilde{\nabla}_{y_k}^{(1)},\ x_{k+1} - u \rangle. \tag{2.8}$$

Using L-smooth (2.2) of $f(\cdot)$ at (y_{k+1}, y_k), we get

$$\begin{aligned} & f(y_{k+1}) - f(y_k) \\ \leq\ & \langle \nabla f(y_k),\ y_{k+1} - y_k \rangle + \frac{L}{2}\|y_{k+1} - y_k\|^2 \\ \overset{(\star)}{=}\ & \theta \langle \nabla f(y_k),\ x_{k+1} - x_k \rangle + \frac{L\theta^2}{2}\|x_{k+1} - x_k\|^2 \\ =\ & \theta\big[\langle \nabla f(y_k) - \widetilde{\nabla}_{y_k}^{(1)},\ x_{k+1} - x_k \rangle + \langle \widetilde{\nabla}_{y_k}^{(1)},\ x_{k+1} - x_k \rangle\big] + \frac{L\theta^2}{2}\|x_{k+1} - x_k\|^2, \\ & \langle \widetilde{\nabla}_{y_k}^{(1)}, x_k - x_{k+1} \rangle \\ \leq\ & \frac{1}{\theta}\big(f(y_k) - f(y_{k+1})\big) + \langle \nabla f(y_k) - \widetilde{\nabla}_{y_k}^{(1)}, x_{k+1} - x_k \rangle + \frac{L\theta}{2}\|x_{k+1} - x_k\|^2, \end{aligned}$$

where $(\star)$ uses the definition of y_k.

After plugging in the constraint (2.6), we have

$$\begin{aligned} & \langle \widetilde{\nabla}_{y_k}^{(1)},\ x_k - x_{k+1} \rangle \\ \leq & \frac{1}{\theta}\big(f(y_k) - f(y_{k+1})\big) + \langle \nabla f(y_k) - \widetilde{\nabla}_{y_k}^{(1)},\ x_{k+1} - x_k \rangle \\ & + \frac{1}{2\eta}\|x_{k+1} - x_k\|^2 - \frac{L\theta}{2(1-\theta)}\|x_{k+1} - x_k\|^2. \end{aligned} \tag{2.9}$$

Then we are ready to combine (2.7), (2.8), (2.9), as well as Lemma 2.3.2, which gives

$$\begin{aligned}
&f(y_k) - f(u) \\
\leq\ & \frac{1-\theta}{\theta}\langle \nabla f(y_k), \tilde{x}_s - y_k\rangle + \langle \nabla f(y_k) - \widetilde{\nabla}^{(1)}_{y_k}, x_{k+1} - u\rangle \\
&+ \frac{1}{\theta}(f(y_k) - f(y_{k+1})) - \frac{L\theta}{2(1-\theta)}\|x_{k+1} - x_k\|^2 + \frac{1}{2\eta}\|x_k - u\|^2 \\
&- \frac{1+\eta\mu}{2\eta}\|x_{k+1} - u\|^2 + h(u) - h(x_{k+1}).
\end{aligned}$$

After taking expectation with respect to the sample i_k, we obtain

$$\begin{aligned}
&f(y_k) - f(u) \\
\overset{(a)}{\leq}\ & \frac{1-\theta}{\theta}\langle \nabla f(y_k), \tilde{x}_s - y_k\rangle + \mathbb{E}_{i_k}\big[\langle \nabla f(y_k) - \widetilde{\nabla}^{(1)}_{y_k}, x_{k+1} - x_k\rangle\big] \\
&+ \frac{1}{\theta}(f(y_k) - \mathbb{E}_{i_k}\big[f(y_{k+1})\big]) - \frac{L\theta}{2(1-\theta)}\mathbb{E}_{i_k}\big[\|x_{k+1} - x_k\|^2\big] + \frac{1}{2\eta}\|x_k - u\|^2 \\
&- \frac{1+\eta\mu}{2\eta}\mathbb{E}_{i_k}\big[\|x_{k+1} - u\|^2\big] + h(u) - \mathbb{E}_{i_k}\big[h(x_{k+1})\big] \\
\overset{(b)}{\leq}\ & \frac{1-\theta}{\theta}\langle \nabla f(y_k), \tilde{x}_s - y_k\rangle + \frac{1}{2\beta}\mathbb{E}_{i_k}\big[\|\nabla f(y_k) - \widetilde{\nabla}^{(1)}_{y_k}\|^2\big] + \frac{\beta}{2}\mathbb{E}_{i_k}\big[\|x_{k+1} - x_k\|^2\big] \\
&+ \frac{1}{\theta}(f(y_k) - \mathbb{E}_{i_k}\big[f(y_{k+1})\big]) - \frac{L\theta}{2(1-\theta)}\mathbb{E}_{i_k}\big[\|x_{k+1} - x_k\|^2\big] + \frac{1}{2\eta}\|x_k - u\|^2 \\
&- \frac{1+\eta\mu}{2\eta}\mathbb{E}_{i_k}\big[\|x_{k+1} - u\|^2\big] + h(u) - \mathbb{E}_{i_k}\big[h(x_{k+1})\big],
\end{aligned}$$

where (a) holds due to the unbiasedness of the gradient estimator $\mathbb{E}_{i_k}\big[\nabla f(y_k) - \widetilde{\nabla}^{(1)}_{y_k}\big] = \mathbf{0}$, and (b) uses the Young's inequality to expand $\mathbb{E}_{i_k}\big[\langle \nabla f(y_k) - \widetilde{\nabla}^{(1)}_{y_k}, x_{k+1} - x_k\rangle\big]$ with the parameter $\beta > 0$.

Applying Lemma 2.3.1 to bound the variance term $\mathbb{E}_{i_k}\big[\|\nabla f(y_k) - \widetilde{\nabla}^{(1)}_{y_k}\|^2\big]$, we get

$$\begin{aligned}
&f(y_k) - f(u) \\
\leq\ & \frac{1-\theta}{\theta}\langle \nabla f(y_k), \tilde{x}_s - y_k\rangle + \frac{L}{\beta}\big(f(\tilde{x}_s) - f(y_k) - \langle \nabla f(y_k), \tilde{x}_s - y_k\rangle\big) \\
&+ \frac{\beta}{2}\mathbb{E}_{i_k}\big[\|x_{k+1} - x_k\|^2\big] + \frac{1}{\theta}(f(y_k) - \mathbb{E}_{i_k}\big[f(y_{k+1})\big]) - \frac{L\theta}{2(1-\theta)}\mathbb{E}_{i_k}\big[\|x_{k+1} - x_k\|^2\big] \\
&+ \frac{1}{2\eta}\|x_k - u\|^2 - \frac{1+\eta\mu}{2\eta}\mathbb{E}_{i_k}\big[\|x_{k+1} - u\|^2\big] + h(u) - \mathbb{E}_{i_k}\big[h(x_{k+1})\big].
\end{aligned}$$

Let $\beta = \frac{L\theta}{1-\theta} > 0$, by rearranging the above inequality, we obtain

$$\begin{aligned}
0 \leq & \frac{1-\theta}{\theta} f(\tilde{x}_s) - \frac{1}{\theta}\mathbb{E}_{i_k}\big[f(y_{k+1})\big] + F(u) - \mathbb{E}_{i_k}\big[h(x_{k+1})\big] \\
& + \frac{1}{2\eta}\|x_k - u\|^2 - \frac{1+\eta\mu}{2\eta}\mathbb{E}_{i_k}\big[\|x_{k+1} - u\|^2\big] \\
\overset{(\star)}{\leq} & \frac{1-\theta}{\theta} F(\tilde{x}_s) - \frac{1}{\theta}\mathbb{E}_{i_k}\big[F(y_{k+1})\big] + F(u) \\
& + \frac{1}{2\eta}\|x_k - u\|^2 - \frac{1+\eta\mu}{2\eta}\mathbb{E}_{i_k}\big[\|x_{k+1} - u\|^2\big]
\end{aligned}$$

$$\begin{aligned}
\frac{1}{\theta}\big(\mathbb{E}_{i_k}\big[F(y_{k+1})\big] - F(u)\big) \leq & \frac{1-\theta}{\theta}\big(F(\tilde{x}_s) - F(u)\big) + \frac{1}{2\eta}\|x_k - u\|^2 \\
& - \frac{1+\eta\mu}{2\eta}\mathbb{E}_{i_k}\big[\|x_{k+1} - u\|^2\big],
\end{aligned} \tag{2.10}$$

where $(\star)$ follows from the Jensen's inequality and the definition of y_{k+1}, which leads to $-h(x_{k+1}) \leq \frac{1-\theta}{\theta} h(\tilde{x}_s) - \frac{1}{\theta} h(y_{k+1})$.

Let $u = x^\star$, using our choice of $\omega = 1 + \eta\mu$ to sum (2.10) over $k = 0 \dots m-1$ with increasing weight ω^k. After taking expectation with respect to all randomness in this epoch, we have

$$\begin{aligned}
& \frac{1}{\theta}\sum_{k=0}^{m-1} \omega^k \big(\mathbb{E}\big[F(y_{k+1})\big] - F(x^\star)\big) + \frac{\omega^m}{2\eta}\mathbb{E}\big[\|x_m - x^\star\|^2\big] \\
\leq & \frac{1-\theta}{\theta}\sum_{k=0}^{m-1} \omega^k \big(F(\tilde{x}_s) - F(x^\star)\big) + \frac{1}{2\eta}\|x_0 - x^\star\|^2.
\end{aligned}$$

Using the Jensen's inequality and $\tilde{x}_{s+1} = \theta\big(\sum_{k=0}^{m-1}\omega^k\big)^{-1}\sum_{k=0}^{m-1}\omega^k x_{k+1} + (1-\theta)\tilde{x}_s = (\sum_{k=0}^{m-1}\omega^k)^{-1}\sum_{k=0}^{m-1}\omega^k y_{k+1}$, we have

$$\begin{aligned}
& \frac{1}{\theta}\sum_{k=0}^{m-1} \omega^k \big(\mathbb{E}\big[F(\tilde{x}_{s+1})\big] - F(x^\star)\big) + \frac{\omega^m}{2\eta}\mathbb{E}\big[\|x_m - x^\star\|^2\big] \\
\leq & \frac{1-\theta}{\theta}\sum_{k=0}^{m-1} \omega^k \big(F(\tilde{x}_s) - F(x^\star)\big) + \frac{1}{2\eta}\|x_0 - x^\star\|^2.
\end{aligned} \tag{2.11}$$

Case I: Consider the first case in Theorem 2.3.3 with $\frac{m}{\kappa} \leq \frac{3}{4}$, we set $\eta = \sqrt{\frac{1}{3\mu m L}}$, $\theta = \sqrt{\frac{m}{3\kappa}} \leq \frac{1}{2}$, and $m = \Theta(n)$.

First, we evaluate the crucial constraint (2.6). By substituting in our parameter settings, the constraint becomes

$$L\theta + \frac{L\theta}{1-\theta} \leq \frac{1}{\eta} \to \sqrt{\frac{m}{\kappa}} \leq \frac{\sqrt{3}}{2}.$$

Thus the constraint is satisfied by meeting the case assumption.

Then we focus on $(1-\theta)\omega^m$, observed that

$$(1-\theta)\omega^m = \left(1 - \sqrt{\frac{m}{3\kappa}}\right) \cdot \left(1 + \sqrt{\frac{1}{3m\kappa}}\right)^m.$$

Let $\zeta = \sqrt{\frac{m}{\kappa}}$, $\zeta \in (0, \frac{\sqrt{3}}{2}]$, we can denote

$$\phi(\zeta) = \left(1 - \frac{\sqrt{3}}{3}\zeta\right) \cdot \left(1 + \frac{\sqrt{3}}{3} \cdot \frac{\zeta}{m}\right)^m$$

as a function of ζ.

By taking derivative with respect to ζ, we find that $\phi(\zeta)$ is monotonically decreasing on $[0, \frac{\sqrt{3}}{2}]$ for any $m > 0$, which means

$$(1-\theta)\omega^m \leq \max_{\zeta \in (0, \frac{\sqrt{3}}{2}]} \phi(\zeta) \leq \phi(0) = 1.$$

Thus we have $\frac{1}{\theta} \geq \frac{1-\theta}{\theta}\omega^m$. By using this inequality in (2.11), we get

$$\begin{aligned} &\frac{1-\theta}{\theta}\sum_{k=0}^{m-1} \omega^k \big(\mathbb{E}\big[F(\tilde{x}_{s+1})\big] - F(x^\star)\big) + \frac{1}{2\eta}\mathbb{E}\big[\|x_m - x^\star\|^2\big] \\ \leq\ &\omega^{-m} \cdot \Big(\frac{1-\theta}{\theta}\sum_{k=0}^{m-1} \omega^k \big(F(\tilde{x}_s) - F(x^\star)\big) + \frac{1}{2\eta}\|x_0 - x^\star\|^2\Big). \end{aligned}$$

Dividing both sides of the above inequality by $\frac{1-\theta}{\theta}\sum_{k=0}^{m-1}\omega^k$, we get

$$\begin{aligned} &\big(\mathbb{E}\big[F(\tilde{x}_{s+1})\big] - F(x^\star)\big) + \frac{\theta}{2\eta(1-\theta)\sum_{k=0}^{m-1}\omega^k}\mathbb{E}\big[\|x_m - x^\star\|^2\big] \\ \leq\ &\omega^{-m} \cdot \Big(\big(F(\tilde{x}_s) - F(x^\star)\big) + \frac{\theta}{2\eta(1-\theta)\sum_{k=0}^{m-1}\omega^k}\|x_0 - x^\star\|^2\Big). \end{aligned}$$

Summing the above inequality over $s = 0 \dots \mathcal{S} - 1$, we get

$$\begin{aligned}&\big(\mathbb{E}\big[F(\tilde{x}_{\mathcal{S}})\big] - F(x^\star)\big) + \frac{\theta}{2\eta(1-\theta)\sum_{k=0}^{m-1}\omega^k}\mathbb{E}\big[\|x_m^{\mathcal{S}-1} - x^\star\|^2\big] \\ &\leq \omega^{-\mathcal{S}m}\cdot\Big(\big(F(\tilde{x}_0) - F(x^\star)\big) + \frac{\theta}{2\eta(1-\theta)\sum_{k=0}^{m-1}\omega^k}\|x_0^0 - x^\star\|^2\Big).\end{aligned}$$

Notice that in order to prevent confusion, we mark iterates with epoch number, such as $x_m^{\mathcal{S}-1}$ represent the last iterate in epoch $\mathcal{S} - 1$.

Using the fact that $\sum_{k=0}^{m-1}\omega^k \geq m$, we have

$$\big(\mathbb{E}\big[F(\tilde{x}_{\mathcal{S}})\big] - F(x^\star)\big) \leq \omega^{-\mathcal{S}m}\cdot\Big(\big(F(\tilde{x}_0) - F(x^\star)\big) + \frac{\theta}{2\eta(1-\theta)m}\|x_0^0 - x^\star\|^2\Big).$$

Using the μ-strongly convexity of $F(\cdot)$ to bound $\|x_0^0 - x^\star\|^2$, which is $\|x_0^0 - x^\star\|^2 \leq \frac{2}{\mu}\big(F(x_0^0) - F(x^\star)\big)$, we obtain

$$\mathbb{E}\big[F(\tilde{x}_{\mathcal{S}}) - F(x^\star)\big] \leq (1+\eta\mu)^{-\mathcal{S}m}\cdot\Big(1 + \frac{\theta}{\eta(1-\theta)m\mu}\Big)\cdot\big(F(\tilde{x}_0) - F(x^\star)\big).$$

Note that $\tilde{x}_0 = x_0^1 = x_0$.

By substituting with our parameters setting, we get

$$\begin{aligned}\mathbb{E}\big[F(\tilde{x}_{\mathcal{S}}) - F(x^\star)\big] &\overset{(\star)}{\leq} \left(O\left(1 + \sqrt{\frac{1}{3n\kappa}}\right)\right)^{-\mathcal{S}m}\cdot O\left(1 + 2\theta\sqrt{\frac{\kappa}{n}}\right)\cdot\big(F(\tilde{x}_0) - F(x^\star)\big) \\ &\leq \left(O\left(1 + \sqrt{\frac{1}{3n\kappa}}\right)\right)^{-\mathcal{S}m}\cdot O\big(F(\tilde{x}_0) - F(x^\star)\big),\end{aligned}$$

where $(\star)$ holds due to the fact that $\theta \leq \frac{1}{2}$.

The above result implies that the oracle complexity in the case $\frac{m}{\kappa} \leq \frac{3}{4}$ to achieve an ϵ-additive error is $\mathcal{O}\Big(\sqrt{\kappa n}\log\frac{F(\tilde{x}_0)-F(x^\star)}{\epsilon}\Big)$.

Case II: For another case with $\frac{m}{\kappa} > \frac{3}{4}$, we set $\eta = \frac{2}{3L}$, $\theta = \frac{1}{2}$, and $m = \Theta(n)$.

Again, we evaluate the constraint (2.6) first. By substituting the parameter setting, the constraint becomes

$$L\theta + \frac{L\theta}{1-\theta} \leq \frac{1}{\eta} \to \eta \leq \frac{2}{3L}.$$

Thus the constraint is satisfied by our parameter choice.

Substituting the parameter setting into (2.11), we get

$$\begin{aligned}
&2\sum_{k=0}^{m-1}\omega^k\big(\mathbb{E}\big[F(\tilde{x}_{s+1})\big]-F(x^\star)\big)+\frac{3L\omega^m}{4}\mathbb{E}\big[\|x_m-x^\star\|^2\big]\\
\leq&\sum_{k=0}^{m-1}\omega^k\big(F(\tilde{x}_s)-F(x^\star)\big)+\frac{3L}{4}\|x_0-x^\star\|^2.
\end{aligned}$$

Notice that based on the Bernoulli's inequality, $\omega^m=(1+\frac{2}{3\kappa})^m\geq 1+\frac{2m}{3\kappa}\geq\frac{3}{2}$, which leads to

$$\begin{aligned}
&\frac{3}{2}\sum_{k=0}^{m-1}\omega^k\big(\mathbb{E}\big[F(\tilde{x}_{s+1})\big]-F(x^\star)\big)+\frac{9L}{8}\mathbb{E}\big[\|x_m-x^\star\|^2\big]\\
\leq&\sum_{k=0}^{m-1}\omega^k\big(F(\tilde{x}_s)-F(x^\star)\big)+\frac{3L}{4}\|x_0-x^\star\|^2,\\
&\sum_{k=0}^{m-1}\omega^k\big(\mathbb{E}\big[F(\tilde{x}_{s+1})\big]-F(x^\star)\big)+\frac{3L}{4}\mathbb{E}\big[\|x_m-x^\star\|^2\big]\\
\leq&\frac{2}{3}\cdot\big(\sum_{k=0}^{m-1}\omega^k\big(F(\tilde{x}_s)-F(x^\star)\big)+\frac{3L}{4}\|x_0-x^\star\|^2\big).
\end{aligned}$$

Again, by telescoping the above inequality from $s=0\dots\mathcal{S}-1$, we get

$$\begin{aligned}
&\sum_{k=0}^{m-1}\omega^k\big(\mathbb{E}\big[F(\tilde{x}_{\mathcal{S}})\big]-F(x^\star)\big)+\frac{3L}{4}\mathbb{E}\big[\|x_m^{\mathcal{S}-1}-x^\star\|^2\big]\\
\leq&\left(\frac{2}{3}\right)^{\mathcal{S}}\cdot\left(\sum_{k=0}^{m-1}\omega^k\big(F(\tilde{x}_0)-F(x^\star)\big)+\frac{3L}{4}\|x_0^0-x^\star\|^2\right).
\end{aligned}$$

Since $\sum_{k=0}^{m-1}\omega^k\geq m$, the above inequality can be rewritten as follows:

$$\begin{aligned}
\big(\mathbb{E}\big[F(\tilde{x}_{\mathcal{S}})\big]-F(x^\star)\big)&\overset{(\star)}{\leq}\left(\frac{2}{3}\right)^{\mathcal{S}}\cdot\left(1+\frac{3\kappa}{2m}\right)\cdot\big(F(\tilde{x}_0)-F(x^\star)\big)\\
&\leq\left(\frac{2}{3}\right)^{\mathcal{S}}\cdot O\big(F(\tilde{x}_0)-F(x^\star)\big),
\end{aligned}$$

where $(\star)$ uses the μ-strongly convexity of $F(\cdot)$, that is, $\|x_0^0-x^\star\|^2\leq\frac{2}{\mu}\big(F(x_0^0)-F(x^\star)\big)$.

This result implies that the oracle complexity in this case is $\mathcal{O}\big(n\log\frac{F(\tilde{x}_0)-F(x^\star)}{\epsilon}\big)$. □

A variant of MiG using Assumption 2.

Algorithm 4 MiG2

Input: Learning rate $\eta = \begin{cases} \sqrt{\frac{1}{3\mu m L}} & \text{if } \frac{m}{\kappa} \leq \frac{3}{4}, \\ \frac{1}{2m\mu} & \text{if } \frac{m}{\kappa} > \frac{3}{4}. \end{cases}$, parameter $\theta = \begin{cases} \sqrt{\frac{m}{3\kappa}} & \text{if } \frac{m}{\kappa} \leq \frac{3}{4}, \\ \frac{1}{2} & \text{if } \frac{m}{\kappa} > \frac{3}{4}. \end{cases}$,
outer iteration number $\mathcal{R}$, epoch length $m = \Theta(n)$, initial guess x_0.
Initialize: $\tilde{x}_0 = x_0^0 = x_{restart}^0 = x_0$;
for $r = 0, \ldots, \mathcal{R} - 1$ **do**
  // Restart every $\mathcal{S}$ epochs.
  $\mathcal{S} = \left\lceil 2 \cdot \left(\frac{1-\theta}{\theta} + \frac{1}{\eta m \mu} \right) \right\rceil$.
  Initialize $\tilde{x}_0 = x_0^0 = x_{restart}^r$.
  for $s = 0, \ldots, \mathcal{S} - 1$ **do**
    Compute and store $\nabla f(\tilde{x}_s)$.
    for $k = 0, \ldots, m - 1$ **do**
      Sample i_k uniformly in $\{1 \ldots n\}$.
      $y_k = \theta x_k^s + (1 - \theta)\tilde{x}_s$.
      $\widetilde{\nabla}_{y_k}^{(1)} = \nabla f_{i_k}(y_k) - \nabla f_{i_k}(\tilde{x}_s) + \nabla f(\tilde{x}_s)$.
      $x_{k+1}^s = \text{argmin}_{x \in \mathbb{R}^d} \left\{ \frac{1}{2\eta} \|x - x_k^s\|^2 + \langle \widetilde{\nabla}_{y_k}^{(1)}, x \rangle + h(x) \right\}$.
    end for
    $\tilde{x}_{s+1} = \frac{\theta}{m} \sum_{k=0}^{m-1} x_{k+1}^s + (1 - \theta)\tilde{x}_s$.
    $x_0^{s+1} = x_m^s$.
  end for
  $x_{restart}^{r+1} = \frac{1}{\mathcal{S}} \sum_{s=0}^{\mathcal{S}-1} \tilde{x}_{s+1}$.
end for
Output: $\tilde{x}_{restart}^{\mathcal{R}}$.

As shown in Theorem 2.2.1, SVRG uses the strong convexity Assumption 2 instead of Assumption 1. In comparison, Assumption 2 is slightly more general than Assumption 1. Here we give a variant of MiG which uses Assumption 2 (MiG2). However, as we see in Algorithm 4, the variant requires regular restarts and is quite complicated, which make it less elegant than MiG (Algorithm 3).

Theorem 2.3.4. *Suppose Assumption 2 holds,* MiG2 *achieves an ϵ-additive error with the following oracle complexities in expectation:*

$$\begin{cases} \mathcal{O}\big(\sqrt{\kappa n} \log \frac{F(x_0) - F(x^\star)}{\epsilon}\big), & \text{if } \frac{m}{\kappa} \leq \frac{3}{4}, \\ \mathcal{O}\big(n \log \frac{F(x_0) - F(x^\star)}{\epsilon}\big), & \text{if } \frac{m}{\kappa} > \frac{3}{4}. \end{cases}$$

In other words, the overall oracle complexity of MiG2 *is* $\mathcal{O}\big((n+\sqrt{\kappa n})\log\frac{F(x_0)-F(x^\star)}{\epsilon}\big)$.

Proof. In order to prevent redundancies in the proof, we adapt the analysis of Theorem 2.3.3 here. Under Assumption 2, we can regard the μ for $h(\cdot)$ is 0 in Lemma 2.3.2. In this case, by following the analysis of Theorem 2.3.3 until inequality (2.10), we obtain (set $u = x^\star$)

$$\begin{aligned}\frac{1}{\theta}\big(\mathbb{E}_{i_k}\big[F(y_{k+1})\big]-F(x^\star)\big) \le \frac{1-\theta}{\theta}\big(F(\tilde{x}_s)-F(x^\star)\big)+\frac{1}{2\eta}\|x_k-x^\star\|^2 \\ -\frac{1}{2\eta}\mathbb{E}_{i_k}\big[\|x_{k+1}-x^\star\|^2\big].\end{aligned}$$

Summing the above inequality from $k = 0,\dots,m-1$ and using Jensen's inequality (note that $\tilde{x}_{s+1} = \frac{\theta}{m}\sum_{k=0}^{m-1}x_{k+1}+(1-\theta)\tilde{x}_s = \frac{1}{m}\sum_{k=0}^{m-1}y_{k+1}$), we obtain

$$\begin{aligned}\frac{1}{\theta}\big(\mathbb{E}\big[F(\tilde{x}_{s+1})\big]-F(x^\star)\big) &\le \frac{1-\theta}{\theta}\big(F(\tilde{x}_s)-F(x^\star)\big)+\frac{1}{2\eta m}\|x_0^s-x^\star\|^2 \\ &\quad -\frac{1}{2\eta m}\mathbb{E}\big[\|x_m^s-x^\star\|^2\big], \\ \big(\mathbb{E}\big[F(\tilde{x}_{s+1})\big]-F(x^\star)\big) &\le \frac{1-\theta}{\theta}\Big(\big(F(\tilde{x}_s)-F(x^\star)\big)-\big(\mathbb{E}\big[F(\tilde{x}_{s+1})\big]-F(x^\star)\big)\Big) \\ &\quad +\frac{1}{2\eta m}\|x_0^s-x^\star\|^2-\frac{1}{2\eta m}\mathbb{E}\big[\|x_0^{s+1}-x^\star\|^2\big].\end{aligned}$$

Summing the above inequality from $s = 0,\dots,\mathcal{S}-1$ and using Jensen's inequality (note that $x_{restart}^{r+1} = \frac{1}{\mathcal{S}}\sum_{s=0}^{\mathcal{S}-1}\tilde{x}_{s+1}$), we have

$$\begin{aligned}\big(\mathbb{E}\big[F(x_{restart}^{r+1})\big]-F(x^\star)\big) &\le \frac{1-\theta}{\theta\mathcal{S}}\Big(\big(F(\tilde{x}_0)-F(x^\star)\big)-\big(\mathbb{E}\big[F(\tilde{x}_{\mathcal{S}})\big]-F(x^\star)\big)\Big) \\ &\quad +\frac{1}{2\eta m\mathcal{S}}\left(\|x_0^0-x^\star\|^2-\mathbb{E}\big[\|x_m^{\mathcal{S}}-x^\star\|^2\big]\right) \\ &\le \frac{1-\theta}{\theta\mathcal{S}}\big(F(\tilde{x}_0)-F(x^\star)\big)+\frac{1}{2\eta m\mathcal{S}}\|x_0^0-x^\star\|^2 \\ &\overset{(a)}{\le} \frac{1}{\mathcal{S}}\left(\frac{1-\theta}{\theta}+\frac{1}{\eta m\mu}\right)\big(F(\tilde{x}_0)-F(x^\star)\big) \\ &\overset{(b)}{\le} \frac{1}{2}\big(F(\tilde{x}_0)-F(x^\star)\big),\end{aligned}$$

where (a) follows from the μ-strong convexity of $F(\cdot)$ in Assumption 2 and (b) uses the choice $\mathcal{S} = \left\lceil 2\cdot\left(\frac{1-\theta}{\theta}+\frac{1}{\eta m\mu}\right)\right\rceil$.

Now it is clear that, for every restart, in expectation we reduce the error by $\frac{1}{2}$. Thus, in order to achieve an ϵ-additive error, we need totally $O\left(\log\left(\frac{F(\tilde{x}_0)-F(x^\star)}{\epsilon}\right)\right)$ restarts.

Case I: Consider the first case with $\frac{m}{\kappa} \leq \frac{3}{4}$. By choosing identical parameters settings $\eta = \sqrt{\frac{1}{3\mu m L}}$, $\theta = \sqrt{\frac{m}{3\kappa}} \leq \frac{1}{2}$ and $m = \Theta(n)$ as in Algorithm 3 (so the constraint (2.6) is satisfied), we have $\mathcal{S} = O(\sqrt{\frac{\kappa}{n}})$, which means that the total oracle complexity is

$$\mathcal{O}\bigg(\mathcal{S}(m+n)\cdot O\left(\log\frac{F(\tilde{x}_0)-F(x^\star)}{\epsilon}\right)\bigg) = \mathcal{O}\bigg(\sqrt{\kappa n}\log\frac{F(\tilde{x}_0)-F(x^\star)}{\epsilon}\bigg).$$

Case II: Consider another case with $\frac{m}{\kappa} > \frac{3}{4}$. By choosing $\theta = \frac{1}{2}$, $\eta = \frac{1}{2m\mu} \leq \frac{1-\theta}{L\theta(2-\theta)}$ (the constraint (2.6) is satisfied) and $m = \Theta(n)$, we have $\mathcal{S} = O(1)$, the total oracle complexity $\mathcal{O}\big(n\log\frac{F(\tilde{x}_0)-F(x^\star)}{\epsilon}\big)$. □

2.3.3 SSNM: The first directly accelerated SAGA

Although a considerable amount of work has been done for accelerating SVRG, another popular stochastic variance reduced method, SAGA, does not have a directly accelerated variant until recently. Accelerating frameworks such as APPA or Catalyst can be used to accelerate SAGA, but the reduction techniques proposed in these works are always difficult to implement and may also result in additional log factors in the overall oracle complexity. A notable variant of SAGA is Point-SAGA [10]. Point-SAGA requires the proximal operator oracle of each $F_i(\cdot)$ and with the help of that, it can adopt a much larger learning rate than SAGA, which results in the accelerated complexity $\mathcal{O}((n+\sqrt{\kappa n})\log(1/\epsilon))$. However, the proximal operator of each $F_i(\cdot)$ may not be efficiently computed in practice. Even for logistic regression, we need to run an individual loop (Newton's method) for its proximal operator oracle. Therefore, a directly accelerated variant of SAGA is of real interests.

Following the idea of adding only negative momentum to SVRG in MiG, we consider adding negative momentum to SAGA. However, unlike SVRG, which keeps a constant snapshot in each inner loop, the "snapshot" of SAGA is a table of points, each corresponding to the position that the component function gradient $\nabla f_i(\cdot)$ was lastly evaluated. Thus, it is non-trivial to directly accelerate SAGA. In this section, we propose a novel *Sampled Negative Momentum* for SAGA. We further show that adding such a momentum has the same acceleration effect as adding negative momentum to SVRG.

The proposed algorithm SSNM (SAGA with Sampled Negative Momentum) is formally given in Algorithm 5. As we can see, there are some unusual tricks used

Algorithm 5 SAGA with Sampled Negative Momentum (SSNM)

Input: Iterations number K, initial point x_0, learning rate $\eta = \begin{cases} \sqrt{\frac{1}{3\mu n L}} & \text{if } \frac{n}{\kappa} \leq \frac{3}{4}, \\ \frac{1}{2\mu n} & \text{if } \frac{n}{\kappa} > \frac{3}{4}. \end{cases}$, parameter $\tau = \frac{n\eta\mu}{1+\eta\mu}$.

Initialize: "Points" table ϕ with $\phi_1^0 = \phi_2^0 = \ldots = \phi_n^0 = x_0$ and a running average for the gradients of "points" table.

1: **for** $k = 0, 1, \ldots, K-1$ **do**

2: 1. Sample i_k uniformly in $\{1, \ldots, n\}$ and compute the gradient estimator using the running average.

3: $y_{i_k}^k = \tau x_k + (1-\tau)\phi_{i_k}^k$;

4: $\widetilde{\nabla}_{y^k}^{(2)} = \nabla f_{i_k}(y_{i_k}^k) - \nabla f_{i_k}(\phi_{i_k}^k) + \frac{1}{n}\sum_{i=1}^n \nabla f_i(\phi_i^k)$;

5: 2. Perform a proximal step.

6: $x_{k+1} = \arg\min_x \left\{ h(x) + \langle \widetilde{\nabla}_{y^k}^{(2)}, x \rangle + \frac{1}{2\eta}\|x_k - x\|^2 \right\}$;

7: 3. Sample I_k uniformly in $\{1, \ldots, n\}$, take $\phi_{I_k}^{k+1} = \tau x_{k+1} + (1-\tau)\phi_{I_k}^k$. All other entries in the "points" table remain unchanged. Update the running average corresponding to the change in the "points" table.

8: **end for**

Output: x_K

in Algorithm 5. Thus we elaborate some ideas behind Algorithm 5 by making the following remarks:

- *Coupled point $y_{i_k}^k$ correlates to the randomness of i_k.* Unlike the negative momentum used for Katyusha, which comes from a fixed snapshot $\tilde{x}$, the negative momentum of SAGA can only be found on a "points" table that changes over time. Thus, in SSNM, we choose to use the i_kth entry of the "points" table to provide the negative momentum, which makes the coupled point correlate to the randomness of sample i_k. In fact, all the possible coupled points y_i^k form a "coupled table". Although the table is never explicitly computed, we shall see that the concept of "coupled table" is critical in the proof of SSNM. The 3rd step in Algorithm 5 can thus be regarded as sampling a point in such a table.

- *"Biased" gradient estimator $\widetilde{\nabla}_{y^k}^{(2)}$.* The expectation of the semi-stochastic gradient estimator $\widetilde{\nabla}_{y^k}^{(2)}$ defined in Algorithm 5 is the average of the gradients computed in the "coupled table", $\mathbb{E}_{i_k}\left[\widetilde{\nabla}_{y^k}^{(2)}\right] = \frac{1}{n}\sum_{i=1}^n \nabla f_i(y_i^k)$, which seems to be surprising as this expectation (except $\widetilde{\nabla}_{x_0}^{(2)}$) does not correspond to any

gradient of $f(\cdot)$, but can be used to show convergence to the optimal solution of $F(\cdot)$. In some sense, $\widetilde{\nabla}^{(2)}_{y^k}$ is a "biased" gradient estimator.

- *Independent samples I_k and i_k.* The additional sample I_k is crucial for the convergence analysis of Algorithm 5, which chooses an index to store the updated point in the "points" table. The insight of this choice is that it separates the randomness of x_{k+1} and the update index in the "points" table so as to make certain inequalities valid.

- *Two learning rates for two cases.* Using different parameter settings for different objective conditions (ill-condition and well-condition) is common for accelerated methods [45, 2]. If some parameters such as L, μ are unknown, SSNM is still a practical algorithm with tuning only η and τ, as compared with Katyusha which has 4 parameters that need to be tuned. Note that we have tried to make the parameter settings in SSNM similar to Katyusha and MiG. We believe that it can help conduct some fair experimental comparisons with these methods.

- *Only one variable vector with a simple algorithm structure.* Same as MiG, SSNM only has one variable vector in the main loop. Coupled point $y^k_{i_k}$ can be computed whenever used and does not need to be explicitly stored. Moreover, SSNM has a one loop structure compared to those variants of SVRG. Such a structure is good for asynchronous implementation since algorithms with two loops in this setting always require a synchronization after each inner loop [26]. Moreover, the algorithm structure of SSNM is more elegant than Katyusha and MiG, both of which require a tricky weighted averaged scheme at the end of each inner loop[4].

Implementation.

We discuss the following implementation issues about SSNM:

- **Memory.** For many problems associated with loss minimization of linear predictors (i.e., logistic regression and least squares), we can write each $f_i(x)$ in Problem (2.1) as $\psi_i(\langle a_i, x\rangle)$, where $a_1, \ldots, a_n$ are data vectors. In this case, $\nabla f_i(\phi_i) = \nabla\psi_i(\langle a_i, \phi_i\rangle) \cdot a_i$ and thus we can reduce the memory consumption of SAGA by storing the scalar $\nabla\psi_i(\langle a_i, \phi_i\rangle)$ instead of the gradient vector. For Point-SAGA, similar trick can be used for objectives with square loss or

[4]These two algorithms can adopt an uniformly average scheme, but in this case, both algorithms require certain restarting tricks, which make them less implementable.

hinge loss [10]. However, when an $\ell 2$-regularizer is included in each $F_i(\cdot)$, the "gradient" of Point-SAGA is correlated with current iterate, and thus the memory overhead of Point-SAGA will always be $O(nd)$ in this case. For SSNM, we can reduce the memory complexity by storing the inner product $\langle a_i, \phi_i \rangle$, and thus SSNM enjoys the same $O(n)$ memory consumption as that of SAGA. We provide the key steps of Algorithm 5 using this trick here.

Stored: "Inner products" table Φ^k with $\Phi_i^k = \langle a_i, \phi_i^k \rangle$ and a running average Ψ^k.
At iteration k:

1. Sample i_k uniformly in $\{1, \ldots, n\}$ and compute the gradient estimator.
   $\langle a_{i_k}, y_{i_k}^k \rangle = \tau \langle a_{i_k}, x_k \rangle + (1-\tau)\Phi_{i_k}^k$;
   $\widetilde{\nabla}_k = \left(\nabla \psi_{i_k}(\langle a_{i_k}, y_{i_k}^k \rangle) - \nabla \psi_{i_k}(\Phi_{i_k}^k)\right) \cdot a_{i_k} + \Psi^k$;
2. Perform a proximal update for x_{k+1}.
3. Sample I_k uniformly in $\{1, \ldots, n\}$, take $\Phi_{I_k}^{k+1} = \tau \langle a_{I_k}, x_{k+1} \rangle + (1-\tau)\Phi_{I_k}^k$.
4. Update the running average.
   $\Psi^{k+1} = \Psi^k + \frac{1}{n}\left(\nabla \psi_{I_k}(\Phi_{I_k}^{k+1}) - \nabla \psi_{I_k}(\Phi_{I_k}^k)\right) \cdot a_{I_k}$;

- **Per-iteration complexity.** In general, each iteration of SSNM requires computing 4 stochastic gradients, i.e., 2 for calculating the gradient estimator and 2 for updating the running average. In the above case where we use linear predictors, we may consider storing additional n scalars $\nabla \psi_i(\Phi_i^k)$ to reduce the per-iteration IFO calls to 2. In comparison, SAGA only computes 1 stochastic gradient in an iteration.

- **Sparse data vector.** We can use the "just in time" update [41] or "lazy/delayed update" [17] technique for SSNM. The only difference is that in each iteration, we need to consider the coordinates that belong to $\mathit{support}(a_{i_k}) \cup \mathit{support}(a_{I_k})$. We may also use the sparse proximal technique in [37], which results in a cleaner implementation, but at the expense of potentially losing the accelerated rate as is the case for MiG mentioned in Appendix A.

Since Point-SAGA and SAGA are closely related to SSNM, we compare them in details in Table 2.2. SSNM yields the same fast $\mathcal{O}((n + \sqrt{\kappa n}) \log(1/\epsilon))$ convergence rate as Point-SAGA without requiring additional assumptions, demonstrating the advantage of direct acceleration. Note that even for logistic regression, the proximal operator oracle required by Point-SAGA does not have a closed form solution. We

	Complexity	Requirements	Memory
SAGA	$\mathcal{O}((n+\kappa)\log(1/\epsilon))$	IFO of $f(\cdot)$, PO of $h(\cdot)$	$O(nd)$ or $O(n)$ for L.M.
Point-SAGA	$\mathcal{O}((n+\sqrt{\kappa n})\log(1/\epsilon))$	PO of each $F_i(\cdot)$	$O(nd)$ or $O(n)$ for L.M.
SSNM	$\mathcal{O}((n+\sqrt{\kappa n})\log(1/\epsilon))$	IFO of $f(\cdot)$, PO of $h(\cdot)$	$O(nd)$ or $O(n)$ for L.M.

Table 2.2: Comparison of variants of SAGA (All complexities are for strongly convex objectives, L.M. stands for models using linear predictors).

may need to run several Newton steps for an inexact oracle as in [10]. In comparison, the gradient oracle required by SSNM and SAGA is much easier to access. For the memory complexity, as we will discuss in the next subsection, if the objective is some linear models (e.g., loss function with linear predictors), all three methods enjoy an efficient $O(n)$ memory overhead. These aspects demonstrate that SSNM is clearly superior to both SAGA and Point-SAGA.

2.3.4 Convergence analysis of SSNM

In this subsection, we theoretically analyze the performance of SSNM. First, we give a variance bound of the stochastic gradient estimator of SSNM shown in Lemma 2.3.5. Since the stochastic gradient estimator of SSNM is computed at a coupled point that contains randomness, the variance bound for SSNM, unlike most of the variance bounds in previous work, is built with respect to the expectation of the "biased" gradient estimator[5].

Lemma 2.3.5 (Variance Bound for SSNM)**.** *Using the same notations as in Algorithm 5, we can bound the variance of stochastic gradient estimator* $\widetilde{\nabla}^{(2)}_{y^k}$ *as*

$$\begin{aligned}&\mathbb{E}_{i_k}\left[\left\|\widetilde{\nabla}^{(2)}_{y^k}-\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(y_i^k)\right\|^2\right]\\ \leq& 2L\left(\frac{1}{n}\sum_{i=1}^{n}\left(f_i(\phi_i^k)-f(y_i^k)\right)-\frac{1}{n}\sum_{i=1}^{n}\left\langle\nabla f_i(y_i^k),\phi_i^k-y_i^k\right\rangle\right).\end{aligned}$$

[5]Other methods using biased gradient estimators include SARAH [34], JacSketch [14]

Proof.

$$
\begin{aligned}
&\mathbb{E}_{i_k}\Big[\Big\|\widetilde{\nabla}^{(2)}_{y^k} - \frac{1}{n}\sum_{i=1}^n \nabla f_i(y_i^k)\Big\|^2\Big] \\
&= \mathbb{E}_{i_k}\Big[\Big\|\Big(\nabla f_{i_k}(y_{i_k}^k) - \nabla f_{i_k}(\phi_{i_k}^k)\Big) - \frac{1}{n}\sum_{i=1}^n \big(\nabla f_i(y_i^k) - \nabla f_i(\phi_i^k)\big)\Big)\Big\|^2\Big] \\
&\overset{(a)}{\leq} \mathbb{E}_{i_k}\Big[\Big\|\nabla f_{i_k}(y_{i_k}^k) - \nabla f_{i_k}(\phi_{i_k}^k)\Big\|^2\Big] \\
&\overset{(b)}{\leq} 2L\cdot \mathbb{E}_{i_k}\Big[f_{i_k}(\phi_{i_k}^k) - f_{i_k}(y_{i_k}^k) - \Big\langle \nabla f_{i_k}(y_{i_k}^k), \phi_{i_k}^k - y_{i_k}^k\Big\rangle\Big] \\
&= 2L\left(\frac{1}{n}\sum_{i=1}^n \big(f_i(\phi_i^k) - f(y_i^k)\big) - \frac{1}{n}\sum_{i=1}^n \Big\langle \nabla f_i(y_i^k), \phi_i^k - y_i^k\Big\rangle\right),
\end{aligned}
$$

where (a) follows from $\mathbb{E}\big[\|\zeta - \mathbb{E}\zeta\|^2\big] \leq \mathbb{E}\|\zeta\|^2$ and (b) uses Theorem 2.1.5 in [31]. □

Now we can formally present the main theorem of SSNM below. As stated in [2], the major task of the negative momentum is to cancel the additional inner product term shown in the variance bound so as to keep a close connection in each iteration. As we shall see shortly, our proposed sampled negative momentum effectively cancels the inner product term, which is where the acceleration comes from.

Theorem 2.3.6. *Let $x^\star$ be the solution of Problem* (2.1)*, define the following Lyapunov function T, which is the same as the one in SAGA [11]:*

$$
\begin{aligned}
T^k &\triangleq T(x_k, \phi^k) \\
&\triangleq \frac{1}{n\eta\mu}\left(\frac{1}{n}\sum_{i=1}^n F_i(\phi_i^k) - F(x^\star) - \frac{1}{n}\sum_{i=1}^n \langle \nabla F_i(x^\star), \phi_i^k - x^\star\rangle\right) + \frac{1}{2\eta n}\|x_k - x^\star\|^2.
\end{aligned}
$$

If Assumption 1 holds, then by choosing $\tau = \frac{n\eta\mu}{1+\eta\mu}$, steps of Algorithm 5 satisfy the following contraction for the Lyapunov function in expectation (conditional on T^k):

$$
\mathbb{E}_{i_k, I_k}\big[T^{k+1}\big] \leq (1+\eta\mu)^{-1} T^k.
$$

Thus, by carefully choosing η, we have the following inequalities in two cases:

(I) (For ill-conditioned problems). *If $\frac{n}{\kappa} \leq \frac{3}{4}$, with $\eta = \sqrt{\frac{1}{3\mu n L}}$ it holds that*

$$
\mathbb{E}\big[\|x_K - x^\star\|^2\big] \leq \left(1 + \sqrt{\frac{1}{3n\kappa}}\right)^{-K}\left(\frac{2}{\mu}\left(F(x_0) - F(x^\star)\right) + \|x_0 - x^\star\|^2\right).
$$

The above inequality implies that in order to reduce the squared norm distance to ϵ, we have an $\mathcal{O}(\sqrt{\kappa n}\log(1/\epsilon))$ oracle complexity as $\epsilon \to 0$ in expectation.

(II) (For well-conditioned problems). *If $\frac{n}{\kappa} > \frac{3}{4}$, by choosing $\eta = \frac{1}{2\mu n}$, we have*

$$\mathbb{E}\big[\|x_K - x^\star\|^2\big] \le \left(1 + \frac{1}{2n}\right)^{-K} \left(\frac{2}{\mu}\big(F(x_0) - F(x^\star)\big) + \|x_0 - x^\star\|^2\right).$$

This inequality implies that in this case we have an $\mathcal{O}(n\log(1/\epsilon))$ oracle complexity as $\epsilon \to 0$ in expectation.

Proof. First, we analyze Algorithm 5 at the kth iteration, given that the randomness from previous iterations are fixed.

We start with the convexity of $f_{i_k}(\cdot)$ at $(y_{i_k}^k, x^\star)$. By definition, we have

$$\begin{aligned}
&f_{i_k}(y_{i_k}^k) - f_{i_k}(x^\star) \\
&\le \langle \nabla f_{i_k}(y_{i_k}^k), y_{i_k}^k - x^\star\rangle \\
&\overset{(\star)}{=} \frac{1-\tau}{\tau}\langle \nabla f_{i_k}(y_{i_k}^k), \phi_{i_k}^k - y_{i_k}^k\rangle + \langle \nabla f_{i_k}(y_{i_k}^k) - \widetilde{\nabla}_{y^k}^{(2)}, x_k - x^\star\rangle + \langle \widetilde{\nabla}_{y^k}^{(2)}, x_k - x_{k+1}\rangle \\
&\quad + \langle \widetilde{\nabla}_{y^k}^{(2)}, x_{k+1} - x^\star\rangle,
\end{aligned}$$

where $(\star)$ uses the definition of the i_kth entry of "coupled table" that $y_{i_k}^k = \tau x_k + (1-\tau)\phi_{i_k}^k$.

As we will see, the first term on the right side is used to cancel the unwanted inner product term in the variance bound.

By taking expectation with respect to sample i_k and using the unbiasedness that $\mathbb{E}_{i_k}\big[\nabla f_{i_k}(y_{i_k}^k) - \widetilde{\nabla}_{y^k}^{(2)}\big] = \mathbf{0}$, we obtain

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^n f_i(y_i^k) - f(x^\star) \\
\le &\frac{1-\tau}{\tau n}\sum_{i=1}^n \langle \nabla f_i(y_i^k), \phi_i^k - y_i^k\rangle + \mathbb{E}_{i_k}\big[\langle \widetilde{\nabla}_{y^k}^{(2)}, x_k - x_{k+1}\rangle\big] + \mathbb{E}_{i_k}\big[\langle \widetilde{\nabla}_{y^k}^{(2)}, x_{k+1} - x^\star\rangle\big].
\end{aligned} \tag{2.12}$$

In order to bound $\mathbb{E}_{i_k}\big[\langle \widetilde{\nabla}_{y^k}^{(2)}, x_k - x_{k+1}\rangle\big]$, we use the L-smoothness of $f_{I_k}(\cdot)$ at $(\phi_{I_k}^{k+1}, y_{I_k}^k)$, which is

$$f_{I_k}(\phi_{I_k}^{k+1}) - f_{I_k}(y_{I_k}^k) \le \langle \nabla f_{I_k}(y_{I_k}^k), \phi_{I_k}^{k+1} - y_{I_k}^k\rangle + \frac{L}{2}\|\phi_{I_k}^{k+1} - y_{I_k}^k\|^2.$$

Taking expectation with respect to sample I_k and using our choice of $\phi_{I_k}^{k+1} = \tau x_{k+1} + (1-\tau)\phi_{I_k}^k$ as well as the definition of "coupled table", we conclude that

$$\mathbb{E}_{I_k}\big[f_{I_k}(\phi_{I_k}^{k+1})\big] - \frac{1}{n}\sum_{i=1}^n f_i(y_i^k) \le \tau\Big\langle \frac{1}{n}\sum_{i=1}^n \nabla f_i(y_i^k), x_{k+1} - x_k\Big\rangle + \frac{L\tau^2}{2}\|x_{k+1} - x_k\|^2,$$

$$\begin{aligned}
&\langle \widetilde{\nabla}_{y^k}^{(2)}, x_k - x_{k+1}\rangle \\
\le &\frac{1}{\tau n}\sum_{i=1}^n f_i(y_i^k) - \frac{1}{\tau}\mathbb{E}_{I_k}\big[f_{I_k}(\phi_{I_k}^{k+1})\big] + \Big\langle \frac{1}{n}\sum_{i=1}^n \nabla f_i(y_i^k) - \widetilde{\nabla}_{y^k}^{(2)}, x_{k+1} - x_k\Big\rangle \\
&+ \frac{L\tau}{2}\|x_{k+1} - x_k\|^2.
\end{aligned}$$

Here we see the effect of the independent sample I_k. It decouples the randomness of x_{k+1} and the update position so as to make the above inequalities valid.

Taking expectation with respect to sample i_k, we obtain

$$\begin{aligned}
&\mathbb{E}_{i_k}\big[\langle \widetilde{\nabla}_{y^k}^{(2)}, x_k - x_{k+1}\rangle\big] \\
\le &\frac{1}{\tau n}\sum_{i=1}^n f_i(y_i^k) - \frac{1}{\tau}\mathbb{E}_{i_k, I_k}\big[f_{I_k}(\phi_{I_k}^{k+1})\big] + \mathbb{E}_{i_k}\Big[\Big\langle \frac{1}{n}\sum_{i=1}^n \nabla f_i(y_i^k) - \widetilde{\nabla}_{y^k}^{(2)}, x_{k+1} - x_k\Big\rangle\Big] \\
&+ \frac{L\tau}{2}\mathbb{E}_{i_k}\big[\|x_{k+1} - x_k\|^2\big]. \qquad (2.13)
\end{aligned}$$

By upper bounding (2.12) using (2.13) and Lemma 2.3.2 (with $h(\cdot)$ μ-strongly convex and $u = x^\star$), we obtain

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^n f_i(y_i^k) - f(x^\star) \\
\le &\frac{1-\tau}{\tau n}\sum_{i=1}^n \langle \nabla f_i(y_i^k), \phi_i^k - y_i^k\rangle + \frac{1}{\tau n}\sum_{i=1}^n f_i(y_i^k) - \frac{1}{\tau}\mathbb{E}_{i_k, I_k}\big[f_{I_k}(\phi_{I_k}^{k+1})\big] \\
&+ \mathbb{E}_{i_k}\Big[\Big\langle \frac{1}{n}\sum_{i=1}^n \nabla f_i(y_i^k) - \widetilde{\nabla}_{y^k}^{(2)}, x_{k+1} - x_k\Big\rangle\Big] + \frac{L\tau}{2}\mathbb{E}_{i_k}\big[\|x_{k+1} - x_k\|^2\big] \\
&- \frac{1}{2\eta}\mathbb{E}_{i_k}\big[\|x_{k+1} - x_k\|^2\big] + \frac{1}{2\eta}\|x_k - x^\star\|^2 - \frac{1+\eta\mu}{2\eta}\mathbb{E}_{i_k}\big[\|x_{k+1} - x^\star\|^2\big] \\
&+ h(x^\star) - \mathbb{E}_{i_k}\big[h(x_{k+1})\big].
\end{aligned}$$

Here we add a constraint that $L\tau \leq \frac{1}{\eta} - \frac{L\tau}{1-\tau}$, which is identical to the one used in MiG. Using Young's inequality $\langle a, b\rangle \leq \frac{1}{2\beta}\|a\|^2 + \frac{\beta}{2}\|b\|^2$ to upper bound $\mathbb{E}_{i_k}\Big[\langle \frac{1}{n}\sum_{i=1}^n \nabla f_i(y_i^k) - \widetilde{\nabla}^{(2)}_{y^k}, x_{k+1} - x_k\rangle\Big]$ with $\beta = \frac{L\tau}{1-\tau} > 0$, we can simplify the above inequality as

$$
\begin{aligned}
&\frac{1}{n}\sum_{i=1}^n f_i(y_i^k) - f(x^\star) \\
\leq\ & \frac{1-\tau}{\tau n}\sum_{i=1}^n \langle \nabla f_i(y_i^k), \phi_i^k - y_i^k\rangle + \frac{1}{\tau n}\sum_{i=1}^n f_i(y_i^k) - \frac{1}{\tau}\mathbb{E}_{i_k, I_k}\big[f_{I_k}(\phi_{I_k}^{k+1})\big] \\
&+ \frac{1-\tau}{2L\tau}\mathbb{E}_{i_k}\Big[\Big\|\frac{1}{n}\sum_{i=1}^n \nabla f_i(y_i^k) - \widetilde{\nabla}^{(2)}_{y^k}\Big\|^2\Big] + \frac{1}{2\eta}\|x_k - x^\star\|^2 \\
&- \frac{1+\eta\mu}{2\eta}\mathbb{E}_{i_k}\big[\|x_{k+1} - x^\star\|^2\big] + h(x^\star) - \mathbb{E}_{i_k}\big[h(x_{k+1})\big].
\end{aligned}
$$

By applying Lemma 2.3.5 to upper bound the variance term, we see that the additional variance term in the variance bound is canceled by the sampled momentum, which gives

$$
\begin{aligned}
&\frac{1}{n}\sum_{i=1}^n f_i(y_i^k) - f(x^\star) \\
\leq\ & \frac{1}{\tau n}\sum_{i=1}^n f_i(y_i^k) - \frac{1}{\tau}\mathbb{E}_{i_k, I_k}\big[f_{I_k}(\phi_{I_k}^{k+1})\big] + \frac{1-\tau}{\tau n}\sum_{i=1}^n \big(f_i(\phi_i^k) - f(y_i^k)\big) \\
&+ \frac{1}{2\eta}\|x_k - x^\star\|^2 - \frac{1+\eta\mu}{2\eta}\mathbb{E}_{i_k}\big[\|x_{k+1} - x^\star\|^2\big] + h(x^\star) - \mathbb{E}_{i_k}\big[h(x_{k+1})\big], \\
&\frac{1}{\tau}\mathbb{E}_{i_k, I_k}\big[f_{I_k}(\phi_{I_k}^{k+1})\big] - F(x^\star) \\
\leq\ & \frac{1-\tau}{\tau n}\sum_{i=1}^n f_i(\phi_i^k) + \frac{1}{2\eta}\|x_k - x^\star\|^2 - \frac{1+\eta\mu}{2\eta}\mathbb{E}_{i_k}\big[\|x_{k+1} - x^\star\|^2\big] - \mathbb{E}_{i_k}\big[h(x_{k+1})\big].
\end{aligned}
\tag{2.14}
$$

Using the convexity of $h(\cdot)$ and that $\phi_{I_k}^{k+1} = \tau x_{k+1} + (1-\tau)\phi_{I_k}^k$, we have

$$
h(\phi_{I_k}^{k+1}) \leq \tau h(x_{k+1}) + (1-\tau)h(\phi_{I_k}^k).
$$

After taking expectation with respect to sample I_k and sample i_k, we obtain

$$
-\mathbb{E}_{i_k}\big[h(x_{k+1})\big] \leq \frac{1-\tau}{\tau n}\sum_{i=1}^n h(\phi_i^k) - \frac{1}{\tau}\mathbb{E}_{i_k, I_k}\big[h(\phi_{I_k}^{k+1})\big].
$$

Combining the above inequality with (2.14) and using the definition that $F_i(\cdot) = f_i(\cdot) + h(\cdot)$, we can write (2.14) as

$$\begin{aligned}
&\frac{1}{\tau}\mathbb{E}_{i_k,I_k}\big[F_{I_k}(\phi_{I_k}^{k+1}) - F_{I_k}(x^\star)\big] \\
\leq&\frac{1-\tau}{\tau}\Big(\frac{1}{n}\sum_{i=1}^n F_i(\phi_i^k) - F(x^\star)\Big) + \frac{1}{2\eta}\|x_k - x^\star\|^2 - \frac{1+\eta\mu}{2\eta}\mathbb{E}_{i_k}\big[\|x_{k+1} - x^\star\|^2\big].
\end{aligned}$$

By dividing the above inequality by n and then adding both sides by $\frac{1}{\tau n}\mathbb{E}_{I_k}\big[\sum_{i\neq I_k}^n \big(F_i(\phi_i^k) - F_i(x^\star)\big)\big]$, we obtain

$$\begin{aligned}
&\frac{1}{\tau}\mathbb{E}_{i_k,I_k}\Big[\frac{1}{n}\sum_{i=1}^n F_i(\phi_i^{k+1}) - F(x^\star)\Big] \\
\leq&\frac{1-\tau}{\tau n}\Big(\frac{1}{n}\sum_{i=1}^n \big(F_i(\phi_i^k) - F_i(x^\star)\big)\Big) + \frac{1}{\tau n}\mathbb{E}_{I_k}\Big[\sum_{i\neq I_k}^n \big(F_i(\phi_i^k) - F_i(x^\star)\big)\Big] \\
&+ \frac{1}{2\eta n}\|x_k - x^\star\|^2 - \frac{1+\eta\mu}{2\eta n}\mathbb{E}_{i_k}\big[\|x_{k+1} - x^\star\|^2\big] \\
=&\frac{1-\tau}{\tau n}\Big(\frac{1}{n}\sum_{i=1}^n \big(F_i(\phi_i^k) - F_i(x^\star)\big)\Big) + \frac{1}{\tau n^2}\sum_{j=1}^n\sum_{i\neq j}^n \big(F_i(\phi_i^k) - F_i(x^\star)\big) \\
&+ \frac{1}{2\eta n}\|x_k - x^\star\|^2 - \frac{1+\eta\mu}{2\eta n}\mathbb{E}_{i_k}\big[\|x_{k+1} - x^\star\|^2\big] \\
=&\frac{1-\frac{\tau}{n}}{\tau}\Big(\frac{1}{n}\sum_{i=1}^n F_i(\phi_i^k) - F(x^\star)\Big) + \frac{1}{2\eta n}\|x_k - x^\star\|^2 \qquad (2.15)\\
&- \frac{1+\eta\mu}{2\eta n}\mathbb{E}_{i_k}\big[\|x_{k+1} - x^\star\|^2\big].
\end{aligned}$$

Since $\frac{1}{n}\sum_{i=1}^n F_i(\phi_i^k) - F(x^\star)$ may not be positive, we need to involve the following term in our Lyapunov function:

$$\begin{aligned}
&-\frac{1}{n}\sum_{i=1}^n \langle \nabla F_i(x^\star), \phi_i^{k+1} - x^\star\rangle \\
=&-\frac{1}{n}\langle \nabla F_{I_k}(x^\star), \phi_{I_k}^{k+1} - x^\star\rangle - \frac{1}{n}\sum_{i\neq I_k}^n \langle \nabla F_i(x^\star), \phi_i^k - x^\star\rangle \\
=&-\frac{\tau}{n}\langle \nabla F_{I_k}(x^\star), x_{k+1} - x^\star\rangle + \frac{\tau}{n}\langle \nabla F_{I_k}(x^\star), \phi_{I_k}^k - x^\star\rangle \\
&-\frac{1}{n}\sum_{i=1}^n \langle \nabla F_i(x^\star), \phi_i^k - x^\star\rangle.
\end{aligned}$$

After taking expectation with respect to sample I_k and i_k, we obtain

$$\mathbb{E}_{i_k,I_k}\Big[-\frac{1}{n}\sum_{i=1}^{n}\langle\nabla F_i(x^\star),\phi_i^{k+1}-x^\star\rangle\Big]=-\Big(1-\frac{\tau}{n}\Big)\Big(\frac{1}{n}\sum_{i=1}^{n}\langle\nabla F_i(x^\star),\phi_i^{k}-x^\star\rangle\Big). \tag{2.16}$$

In order to give a clean proof, we denote

$$D_k\triangleq\frac{1}{n}\sum_{i=1}^{n}F_i(\phi_i^k)-F(x^\star)-\frac{1}{n}\sum_{i=1}^{n}\langle\nabla F_i(x^\star),\phi_i^k-x^\star\rangle,$$

and $P_k\triangleq\|x_k-x^\star\|^2$, then by combining (2.15), (2.16), we can write the contraction as

$$\frac{1}{\tau}\mathbb{E}_{i_k,I_k}\big[D_{k+1}\big]+\frac{1+\eta\mu}{2\eta n}\mathbb{E}_{i_k}\big[P_{k+1}\big]\leq\frac{1-\frac{\tau}{n}}{\tau}D_k+\frac{1}{2\eta n}P_k. \tag{2.17}$$

Case I: Consider the first case with $\frac{n}{\kappa}\leq\frac{3}{4}$, choosing $\eta=\sqrt{\frac{1}{3\mu nL}}$ and $\tau=\frac{n\eta\mu}{1+\eta\mu}=\frac{\sqrt{\frac{n}{3\kappa}}}{1+\sqrt{\frac{1}{3n\kappa}}}<\frac{1}{2}$, we first evaluate the parameter constraint:

$$L\tau\leq\frac{1}{\eta}-\frac{L\tau}{1-\tau}\Rightarrow\underbrace{\frac{2-\tau}{1-\tau}}_{<3}\cdot\underbrace{\frac{\sqrt{\frac{n}{3\kappa}}}{1+\sqrt{\frac{1}{3n\kappa}}}}_{\leq\sqrt{\frac{n}{3\kappa}}}\leq\sqrt{\frac{3n}{\kappa}},$$

which means that the constraint is satisfied by our parameter choices.

Moreover, with this choice of τ, we have

$$\frac{1}{\tau(1+\eta\mu)}=\frac{1-\frac{\tau}{n}}{\tau}=\frac{1}{n\eta\mu}.$$

Thus, the contraction (2.17) can be written as

$$\frac{1}{n\eta\mu}\mathbb{E}_{i_k,I_k}\big[D_{k+1}\big]+\frac{1}{2\eta n}\mathbb{E}_{i_k}\big[P_{k+1}\big]\leq(1+\eta\mu)^{-1}\cdot\Big(\frac{1}{n\eta\mu}D_k+\frac{1}{2\eta n}P_k\Big).$$

After telescoping the above contraction from $k=0\dots K-1$ and taking expectation with respect to all randomness, we have

$$\frac{1}{n\eta\mu}\mathbb{E}\big[D_K\big]+\frac{1}{2\eta n}\mathbb{E}\big[P_K\big]\leq(1+\eta\mu)^{-K}\cdot\Big(\frac{1}{n\eta\mu}D_0+\frac{1}{2\eta n}P_0\Big).$$

Note that $D_0 = F(x_0) - F(x^\star)$ and $\mathbb{E}\big[D_K\big] \geq 0$ based on convexity. After substituting the parameter choices, we have

$$\mathbb{E}\big[\|x_K - x^\star\|^2\big] \leq \Big(1 + \sqrt{\frac{1}{3n\kappa}}\Big)^{-K} \cdot \Big(\frac{2}{\mu}\big(F(x_0) - F(x^\star)\big) + \|x_0 - x^\star\|^2\Big).$$

Case II: Consider another case with $\frac{n}{\kappa} > \frac{3}{4}$, choosing $\eta = \frac{1}{2\mu n}$, $\tau = \frac{n\eta\mu}{1+\eta\mu} = \frac{\frac{1}{2}}{1+\frac{1}{2n}} < \frac{1}{2}$. Again, we first evaluate the constraint:

$$L\tau \leq \frac{1}{\eta} - \frac{L\tau}{1-\tau} \Rightarrow \tau \cdot \underbrace{\frac{2-\tau}{1-\tau}}_{<3} < \frac{3}{2} < \frac{2n}{\kappa}.$$

Then by rewriting the contraction (2.17), telescoping from $k = 0 \dots K-1$ and taking expectation with respect to all randomness, we obtain

$$2\mathbb{E}\big[D_K\big] + \frac{1}{2\eta n}\mathbb{E}\big[P_K\big] \leq (1+\eta\mu)^{-K} \cdot \Big(2D_0 + \frac{1}{2\eta n}P_0\Big).$$

By substituting the parameter choices, we have

$$\mathbb{E}\big[\|x_K - x^\star\|^2\big] \leq \Big(1 + \frac{1}{2n}\Big)^{-K} \cdot \Big(\frac{2}{\mu}\big(F(x_0) - F(x^\star)\big) + \|x_0 - x^\star\|^2\Big).$$

□

Transforming Assumption 3 to Assumption 1.

Recall that in Theorem 2.2.2, the strongly convex assumption for SAGA (Assumption 3) is imposed on each $f_i(\cdot)$ (or the average $f(\cdot)$ as an extension) [11]. In comparison, SSNM requires the strong convexity of $h(\cdot)$ (Assumption 1), which seems to be critical in the proof. Below we show that the strong convexity assumption of each $f_i(\cdot)$ can be efficiently transformed into Assumption 1.

Suppose we have an objective in the form (2.1) with each $f_i(\cdot)$ L-smooth and μ-strongly convex, $h(\cdot)$ convex and proper (Assumption 3 of SAGA). By defining $f_i'(\cdot) = f_i(\cdot) - \frac{\mu}{2}\|\cdot\|^2$ for each $f_i(\cdot)$ and $h'(\cdot) = h(\cdot) + \frac{\mu}{2}\|\cdot\|^2$, the optimal solution of minimizing $F'(\cdot) = \frac{1}{n}\sum_{i=1}^n f_i'(\cdot) + h'(\cdot)$ is equivalent to that of (2.1) and it can be verified that each $f_i'(\cdot)$ is $(L-\mu)$-smooth and convex, $h'(\cdot)$ is μ-strongly convex. Moreover, the proximal operator $\mathrm{prox}_{h'}^{\eta}(v) \triangleq \arg\min_x\{h'(x) + \frac{1}{2\eta}\|x - v\|^2\}, \forall v \in \mathbb{R}^d$ can be efficiently computed as

$$\mathrm{prox}_{h'}^{\eta}(v) = \mathrm{prox}_{h}^{\eta/(1+\eta\mu)}\left(\frac{v}{1+\eta\mu}\right).$$

Conversely, Assumption 1 may not be reducible to Assumption 3 using the above trick, since the modified regularizer $h(\cdot) - \frac{\mu}{2}\|\cdot\|^2$ may not be as "simple" as $h(\cdot)$.

2.3.5 Understanding the acceleration trick

In [2], the negative momentum (or Katyusha momentum) is described as a "magnet" that reduces the error of the semi-stochastic gradient estimator for variance reduced algorithms. Thus, the author combined this idea with Nesterov's momentum (or "positive" momentum) to achieve acceleration. However, as shown in Section 2.3.1 (MiG) as well as SSNM, it seems that merely using the negative momentum trick is enough to obtain the same accelerated convergence rate, which makes this acceleration somewhat "counter-intuitive". In theory, it is clear that with the help of negative momentum, we can adopt a much tighter variance bound. However, this theoretical effect does not explain the source of acceleration. In this section, we try to build a connection between the negative momentum and the standard Nesterov's momentum in [31].

For simplicity, we mainly focus on the objective (2.1) with $h(\cdot) \equiv 0$ in this section. First, consider the deterministic case with $n = 1$, SSNM degenerates into an algorithm with the following key steps[6] (with $z \in \mathbb{R}^d$ denoting the one item "points" table ϕ):

$$\begin{aligned} y_k &= \tau x_k + (1-\tau) z_k; \\ x_{k+1} &= x_k - \eta \cdot \nabla f(y_k); \\ z_{k+1} &= \tau x_{k+1} + (1-\tau) z_k. \end{aligned}$$

This is exactly the scheme of IGA [7] in the Euclidean setting. Note that we can completely eliminate the sequence $\{x_k\}$, which results in a simple scheme below.

$$\begin{aligned} z_{k+1} &= y_k - \eta\tau \cdot \nabla f(y_k); \\ y_{k+1} &= z_{k+1} + (1-\tau)(z_{k+1} - z_k). \end{aligned}$$

By carefully choosing parameters η and τ, we recover the original Nesterov's accelerated gradient method with constant stepsize [31]. This observation motivates us to formulate the key steps in SSNM (Algorithm 5) and MiG (Algorithm 3)[7] into the following schemes (outer loops are omitted for simplicity):

[6] Actually, in the deterministic case, MiG also degenerates to this scheme.

[7] We informally adopt a uniform averaged scheme for MiG for simplicity.

SSNM

$$\widetilde{\nabla}^{(2)}_{y^k} = \nabla f_{i_k}(y^k_{i_k}) - \nabla f_{i_k}(\phi^k_{i_k}) + \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\phi^k_i);$$

$$\phi^{k+1}_{I_k} = y^k_{I_k} - \eta\tau \cdot \widetilde{\nabla}^{(2)}_{y^k};$$

$$y^{k+1}_{i_{k+1}} = \phi^{k+1}_{I_k} + \underline{(1-\tau)(\phi^{k+1}_{i_{k+1}} - \phi^k_{I_k})};$$

MiG

for $k = 1 \dots m$:

$$\widetilde{\nabla}^{(1)}_{y_k} = \nabla f_{i_k}(y^s_k) - \nabla f_{i_k}(\tilde{x}_s) + \nabla f(\tilde{x}_s);$$

$$y^s_{k+1} = y^s_k - \eta\tau \cdot \widetilde{\nabla}^{(1)}_{y_k};$$

$$\tilde{x}_{s+1} = \frac{1}{m}\sum_{k=1}^{m} y^s_{k+1};$$

$$y^{s+1}_1 = y^s_{m+1} + \underline{(1-\tau)(\tilde{x}_{s+1} - \tilde{x}_s)};$$

The underlined parts of both algorithms can be regarded as the source of acceleration, since setting $\tau = 1$ makes both algorithms degenerate into SAGA or Prox-SVRG[8]. A more careful analysis shows that: For MiG, the momentum $\tilde{x}_{s+1} - \tilde{x}_s$ is provided every m stochastic steps, where $m = \Theta(n)$ as suggested in Theorem 2.3.3; for SSNM, although a little bit messy in randomness, we can observe that in expectation, every n steps, the momentum is provided by the newly computed iterate. In comparison, the momentum in Acc-Prox-SVRG [35] is added in every stochastic step. However, as analyzed in [35], in pure stochastic setting (mini-batch size is 1)[9], no acceleration can be guaranteed for Acc-Prox-SVRG in theory. The intuition here is that we may not trust the momentum provided in every stochastic step; instead, we trust the momentum provided by the average information of n stochastic steps.

Based on the above observation, we may understand the negative momentum in SSNM and MiG as the Nesterov's momentum based on average information, in addition to attaining tighter variance bounds.

[8]In fact, setting $\tau = 1$ does not make SSNM and MiG exactly the same as SAGA and Prox-SVRG. For SSNM, the update index for the "points" table is different; for MiG, the initial point y^{s+1}_1 for the new epoch is different.

[9]Pure stochastic setting is important since it is proven that in order to achieve the optimal convergence rate per data access, we should always choose a mini-batch size of 1 for a family of variance reduction methods [25].

2.4 Empirical justifications

In this section, we conducted experiments to justify our theoretical results (Theorem 2.3.3 and 2.3.6). All the algorithms were implemented in C++ and executed through a MATLAB interface for a fair comparison. We ran experiments on an HP Z440 machine with a single Intel Xeon E5-1630v4 with 3.70GHz cores, 16GB RAM, Ubuntu 16.04 LTS with GCC 4.9.0, MATLAB R2017b.

We are optimizing the following binary problem with $a_i \in \mathbb{R}^d$, $b_i \in \{-1, +1\}$, $i = 1 \dots m$:

$$\text{Logistic Regression:} \quad \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + \exp\left(-b_i a_i^T x\right)\right) + \frac{\lambda}{2}\|x\|^2,$$

where λ is the regularization parameter and all the datasets used were normalized before running the experiments.

The experiments were designed as some ill-conditioned problems (with very small λ), since ill-condition is where all the accelerated first-order methods take effect. We tested the following algorithms with their corresponding parameter settings:

- SAGA. We set the learning rate as $\frac{1}{2(\mu n + L)}$, which is analyzed theoretically in [11].
- SSNM. We used the same settings as suggested in Algorithm 5, which are $\eta = \sqrt{\frac{1}{3\mu n L}}$ and $\tau = \frac{n\eta\mu}{1+\eta\mu}$.
- Katyusha. As suggested by the author, we fixed $\tau_2 = \frac{1}{2}$, set $\eta = \frac{1}{3\tau_1 L}$ and chose $\tau_1 = \sqrt{\frac{m}{3\kappa}}$ [2] (In the notations of the original work).
- MiG. We set $\eta = \frac{1}{3\theta L}$ and chose $\theta = \sqrt{\frac{m}{3\kappa}}$ as suggested in Algorithm 3.

We report the results in Figure 2.2. From the results, we can make the following observations to justify the accelerated convergence rates stated in Theorem 2.3.3 and 2.3.6:

- *Similar convergence results for all the accelerated methods.* In fact, we are surprised by the excellent performance of SSNM on the covtype dataset. For this dataset, SSNM is even faster than Katyusha and MiG in terms of the number of epochs (though in theory, Katyusha and MiG yield the same convergence rate as SSNM). The fast convergence of SSNM in practice imply that the algorithm could potentially benefit many applications.

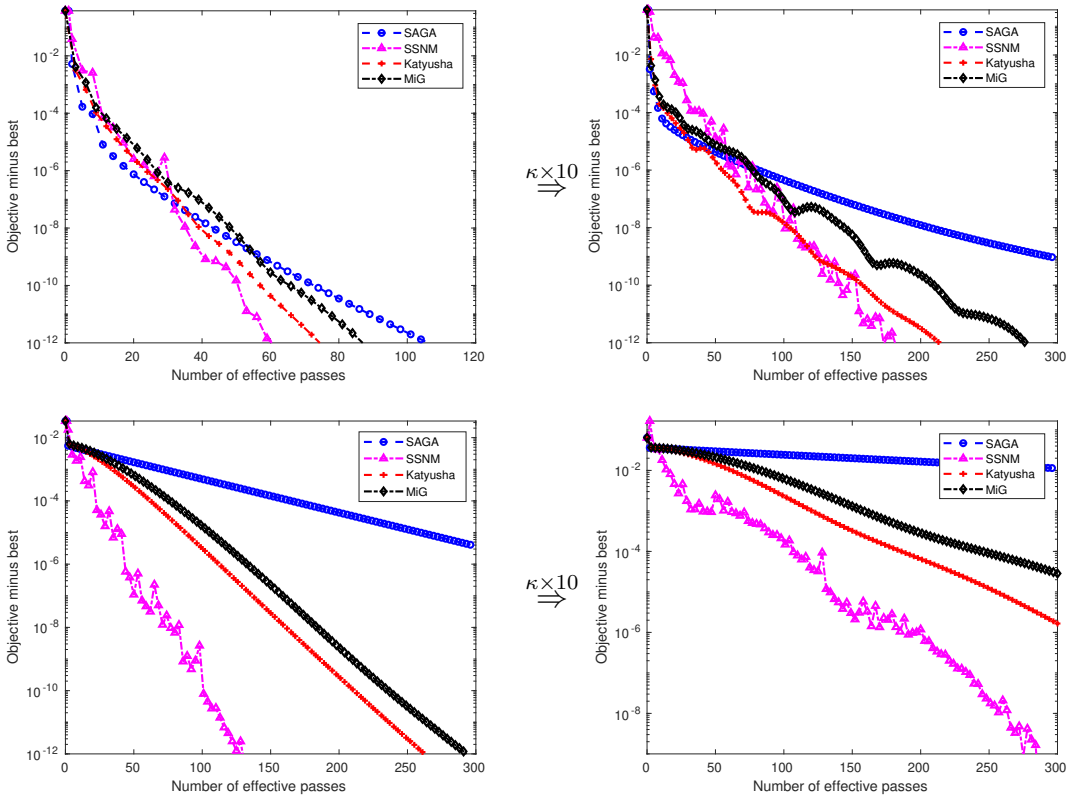

Figure 2.2: Evaluations of SAGA, SSNM, Katyusha and MiG on the a9a dataset with $\lambda = 10^{-6}$ and 10^{-7} (the first two figures) and the covtype dataset with $\lambda = 10^{-8}$ and 10^{-9} (the last two figures).

- *MiG being slightly slower than Katyusha in terms of oracle complexity.* We notice that Katyusha is slightly faster than MiG in terms of the number of oracle calls, which is reasonable since Katyusha has one more Nesterov's Momentum. From the result of Acc-Prox-SVRG in [35], we see that Nesterov's Momentum is effective in this case, but without significant improvement. As analyzed in [35], using large enough mini-batch is a requirement to make Acc-Prox-SVRG improve its convergence rate in theory (see Table 1 in [35]), which also explains the limited difference between MiG and Katyusha.

- *Around 3 times slow-down when κ is 10 times larger.* It can be observed that

using the same dataset, when we divide λ by 10 (the same as multiply κ by 10), approximately $\sqrt{10}$ times slow-down ($\sqrt{10}$ times more oracle calls required to achieve the same accuracy) is recorded for all the accelerated methods. In comparison, SAGA shows significant slow-down when κ is increased in both experiments. This observation justifies the $\sqrt{\kappa}$ dependency for accelerated methods.

Another observation is that accelerated methods seem to perform worse in the experiments on the `a9a` dataset at first several passes. We conjecture that this is because the objective is locally well-conditioned around the initial point. For well-conditioned problem, accelerated methods do not yield a faster rate in theory. In practice, we always found that a smaller amount of momentum yields a better performance. Non-accelerated methods (SVRG, SAGA) always perform better in this case, since they are the accelerated methods without momentum. In the parameter schemes of SSNM, MiG, and Katyusha, the amounts of negative momentum are all set to be $\geq 1/2$ for simplicity in the proofs. To achieve more consistent performance, we can derive parameter schemes that have a smaller amount of momentum.

Chapter 3

Highly-smooth Convex Optimization

Acceleration in convex optimization is always achieved in an intriguing way. Take the famous Nesterov's accelerated gradient [33] as an example. Nesterov's method is proven to yield an optimal convergence rate $\mathcal{O}(1/T^2)$[1] [31] for a class of smooth and convex problems and has a surprisingly simple algorithm structure. Since the original work of Nesterov's method only contains pure algebraic tricks, to intuitively explain this acceleration becomes an arduous work. Allen-Zhu and Hazan[4] interpret this acceleration as forming a linear coupling between two slow algorithms, namely gradient descent and mirror descent. Then, the intuition is explained as that Nesterov's method finds a perfect balance between these two steps [4]. In the unconstrained case, we can regard the gradient step as finding an upper bound for the gradient norm and the mirror step as constructing the corresponding lower bound. Then, using the correct parameter setting, these two steps are bridged by the gradient norm and achieve the acceleration.

Beyond the first-order scenario, Nesterov [30] first extended the acceleration technique (i.e., *estimate sequence*) into second-order methods, which is called Accelerated Cubic Regularized Newton's method (Acc-Cubic). The analysis in [30] shows an $\mathcal{O}(1/T^3)$ worst-case complexity for problems with Lipschitz continuous Hessian. Later in the unpublished work [8], the acceleration technique is extended to solve the problems with mth-order ($m \geq 1$) Lipschitz continuous derivative and yields an $\mathcal{O}(1/T^{(m+1)})$ convergence rate. Monteiro and Svaiter [28] proposed an accelerated

[1]Different from previous chapter, where the complexity is based on achieving ϵ sub-optimality, we mainly use the accuracy attained by running T iterations to depict the convergence rate, denoted using $\mathcal{O}(\cdot)$.

second-order method A-NPE based on the *Large Step* A-HPE framework, which converges at the rate of $\mathcal{O}(1/T^{3.5})$[2]. Recently, Arjevani et al. [6] and Agarwal and Hazan [1] proved the lower oracle complexity bounds for second-order methods. Specially, the lower bound given in [6] justifies that A-NPE is near optimal. Nesterov [32] provided refined analyses to the above topics. While writing this thesis, we noticed that two works [15, 9] independently proposed generalized versions of A-HPE and all of them converges at the rate of $\mathcal{O}(1/T^{\frac{3m+1}{2}})$.

In this chapter, we provide an acceleration framework that has a coupling structure, which covers many accelerated high-order methods as its instances. Based on this framework, we show that the intuition of upper bounding and lower bounding the gradient norm is generalized to the high-order case. Moreover, we provide insights that lead to the potential of constructing new high-order methods.

3.1 Preliminaries

In this chapter, we consider the problem in a finite dimensional inner product real vector space[3], denoted by $\mathbb{E}$ and we use $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$ to denote the inner product and the induced norm, respectively. We are interested in solving the following problem in this section:

$$\min_{x \in \mathbb{E}} f(x), \tag{3.1}$$

where $f : \mathbb{E} \to \mathbb{R}$ is convex and m times differentiable ($m \geq 1$) with its directional derivatives (with $1 \leq p \leq m$) along the directions $v_i \in \mathbb{E}, i = 1, \ldots, p$, denoted as

$$\nabla^p f(\cdot)[v_1, \ldots, v_p].$$

For any $x \in \mathbb{E}$, the norm of $\nabla^p f(x)$ is defined as

$$\|\nabla^p f(x)\| = \max_{\|v_1\|=1} \cdots \max_{\|v_p\|=1} \nabla^p f(x)[v_1, \ldots, v_p].$$

[2]Since each iteration of A-NPE involves a line search procedure, which requires $O\left(\log(\frac{1}{\epsilon})\right)$ calls to the second-order oracle $(f(\cdot), \nabla f(\cdot), \nabla^2 f(\cdot))$ with ϵ denoting the required accuracy, A-NPE requires at most $O\left(\epsilon^{-2/7} \log(\frac{1}{\epsilon})\right)$ oracle calls to reduce the error to ϵ. Then, we can make a fair comparison between A-NPE and Acc-Cubic, whose convergence rates are $O(\log^{3.5}(K)/K^{3.5})$ and $O(1/K^3)$ with K denoting the number of oracle calls, respectively.

[3]The most general problem setting for accelerated high-order methods in the literature, to the best of our knowledge, is a finite-dimensional real vector space together with a *conjugate Euclidean norm* [30]. We consider a simpler problem setting in this section.

We use $f_x^{(m)}(\cdot)$ to denote the mth-order Taylor approximation of $f(\cdot)$ at x, which is for any $y \in \mathbb{E}$,

$$f_x^{(m)}(y) = f(x) + \nabla f(x)[y-x] + \cdots + \frac{1}{m!}\nabla^m f(x)[y-x, \ldots, y-x].$$

Then, we assume the following regularity on f:

Assumption 3.1.1. *The mth-derivative of $f(\cdot)$ is L_m-Lipschitz continuous, which is for any $x, y \in \mathbb{E}$,*

$$\|\nabla^m f(x) - \nabla^m f(y)\| \leq L_m \|x - y\|.$$

Based on this assumption and by the standard integration arguments, we can conclude the following useful results: for any $x, y \in \mathbb{E}$,

$$|f(y) - f_x^{(m)}(y)| \leq \frac{L_m}{(m+1)!}\|y - x\|^{m+1}, \tag{3.2}$$

$$\|\nabla f(y) - \nabla f_x^{(m)}(y)\| \leq \frac{L_m}{m!}\|y - x\|^m. \tag{3.3}$$

Moreover, the following definitions are also used in this chapter and thus we state here for completeness.

Definition 3.1.2 (Bregman divergence)**.** *Given a strictly convex and continuously differentiable function $d(\cdot)$ on $\mathbb{E}$, for any $x, y \in \mathbb{E}$, the Bregman divergence is defined as*

$$V_d(x, y) \triangleq d(x) - d(y) - \langle \nabla d(y), x - y \rangle.$$

Definition 3.1.3 (Uniformly convex [46])**.** *We say a differentiable function $g(\cdot)$ on $\mathbb{E}$ is m-uniformly convex ($m \geq 2$) with parameter μ_m if for any $x, y \in \mathbb{E}$,*

$$g(y) - g(x) - \langle \nabla g(x), y - x \rangle \geq \frac{\mu_m}{m}\|x - y\|^m.$$

As a concrete example of uniformly convex functions, we give the following lemma, which is identical to Lemma 4 (2.7) in [30],

Lemma 3.1.4. *Let $d_m(x) \triangleq \frac{1}{m}\|x - x_0\|^m$ for some $x_0 \in \mathbb{E}$ and $m \geq 2$, then $d_m(\cdot)$ is m-uniformly convex with $\mu_m = \frac{1}{2^{m-2}}$.*

Specially, we fix a point $x_0 \in \mathbb{E}$ as the initial state and denotes $x^\star$ as one solution of problem (3.1). T is also given as the iteration number to be executed.

Algorithm 6 High-order Linear Coupling (HLC) at step k

Input: $y_k, z_k \in \mathbb{E}$, the mth-order Taylor approximation $f_x^{(m)}(\cdot)$, a regularizer $R(\cdot,\cdot)$, a Bregman divergence $V_d(\cdot,\cdot)$.

1: $x_{k+1} = \tau_k z_k + (1-\tau_k)\, y_k$.

2: $y_{k+1} \in$ first-order stationary points of $\left\{\gamma_k f_{x_{k+1}}^{(m)}(x) + R(x, x_{k+1}) \mid x \in \mathbb{E}\right\}$.

3: $z_{k+1} = \arg\min_{x\in\mathbb{E}} \left\{\langle \eta_k \nabla f(y_{k+1}), x - z_k\rangle + V_d(x, z_k)\right\}$.

3.2 A General Acceleration Framework

Our proposed High-order Linear Coupling (HLC) is formally given in Algorithm 6 (with $0 < \tau_k < 1$, $\gamma_k > 0$, $\eta_k > 0$ undetermined). In order to clearly illustrate the underlying ideas, we make the following remarks:

- *A "microscopic" perspective.* In [31], Nesterov described that the construction of the accelerated method based on taking a "global" viewpoint, which is first deciding certain relation formulas and then trying to ensure them in the algorithm design. In this chapter, we instead take a "microscopic" perspective, which means that we inspect the accelerated methods by first building a contraction in one step and then deciding the parameter choices or initial state when telescoping. We believe that such a perspective directly reflects the influence of algorithmic components and sometimes is easier to understand intuitively.

- *Coupling structure.* Following [4], we formulate Algorithm 6 as the result of coupling a regularized high-order step and a mirror step[4]. Although differences such as the choice of hyperplane in Step 3 can be observed, we will show that the intuition of HLC, similar to that of Linear Coupling [4], is still balancing the upper bound and lower bound of gradient norm.

- *A potentially non-convex auxiliary problem in Step 2.* When $m > 2$, sometimes we cannot ensure that the auxiliary problem is convex (e.g., the corresponding auxiliary problem in Algorithm 4.2 [8]). Fortunately, it seems that obtaining a stationary point is enough to achieve the acceleration in some cases.

[4]As specified in Preliminaries 3.1 and to be concluded in Section 3.7, at the current point, high-order methods do not work for non-Euclidean norms and thus the choice of Bregman divergence in Step 3 is limited to $V_{d_p}(\cdot,\cdot)$ for some positive integer p. However, to stress on this potential improvement, we keep using $V_d(\cdot,\cdot)$ here.

As we can see, the mirror step (Step 3) does not correlate to mth-order Taylor model. Recall that the mirror steps are trying to construct lower bound to the optimal value $f^\star$. Thus, we may conclude the following intuition to explain this choice: (1) The hyperplane $\langle \nabla f(y_{k+1}), x\rangle$ is a reasonably good lower bound in high-order case and can be regarded as a natural adaption from the first-order case[5]. (2) The mirror step could be easy to solve in practice since we can write it as

$$\arg\min_{x\in\mathbb{E}} \Big\{ \langle \eta_k \nabla f(y_{k+1}) - \nabla d(z_k), x\rangle + d(x) \Big\},$$

which is to minimize the summation of a linear function and $d(\cdot)$.

Since in HLC, the mirror step (Step 3) and the convex combination (Step 1) are relatively fixed, let us see what we can conclude from these two steps.

Theorem 3.2.1. *Using the settings and based on Step 1 and Step 3 in Algorithm 6, defining the following residual term at step k:*

$$S^k_{HLC} = \langle \nabla f(y_{k+1}), z_{k+1} - z_k\rangle + \frac{1}{\eta_k} V_d(z_{k+1}, z_k) + \frac{1}{\tau_k}\langle \nabla f(y_{k+1}), x_{k+1} - y_{k+1}\rangle, \quad (3.4)$$

we can conclude that

$$\frac{1}{\tau_k}\big(f(y_{k+1}) - f(x^\star)\big) \le \frac{1-\tau_k}{\tau_k}\big(f(y_k) - f(x^\star)\big) + \frac{1}{\eta_k}\big(V_d(x^\star, z_k) - V_d(x^\star, z_{k+1})\big) - S^k_{HLC}. \quad (3.5)$$

Proof. Using the optimality condition for z_{k+1} and the triangle equality of Bregman divergence, we have

$$\begin{aligned}\langle \eta_k \nabla f(y_{k+1}), z_{k+1} - x^\star\rangle &= \langle \nabla d(z_{k+1}) - \nabla d(z_k), x^\star - z_{k+1}\rangle \\ &= V_d(x^\star, z_k) - V_d(x^\star, z_{k+1}) - V_d(z_{k+1}, z_k).\end{aligned}$$

Thus,

$$\begin{aligned}\langle \eta_k \nabla f(y_{k+1}), z_k - x^\star\rangle = &-\big(\langle \eta_k \nabla f(y_{k+1}), z_{k+1} - z_k\rangle + V_d(z_{k+1}, z_k)\big) \\ &+ V_d(x^\star, z_k) - V_d(x^\star, z_{k+1}).\end{aligned}$$

For the left side of above equality, by the definition of x_{k+1}, we have

$$\begin{aligned}\langle \nabla f(y_{k+1}), z_k - x^\star\rangle &= \Big\langle \nabla f(y_{k+1}), \frac{1}{\tau_k}(x_{k+1} - y_{k+1}) + \frac{1-\tau_k}{\tau_k}(y_{k+1} - y_k) + y_{k+1} - x^\star \Big\rangle \quad (3.6)\\ &\ge \frac{1}{\tau_k}\langle \nabla f(y_{k+1}), x_{k+1} - y_{k+1}\rangle + \frac{1}{\tau_k} f(y_{k+1}) - \frac{1-\tau_k}{\tau_k} f(y_k) - f(x^\star).\end{aligned}$$

[5]The choice of "normal vector" is different, in our notation, Linear Coupling chooses $\nabla f(x_{k+1})$ [4].

Re-arranging the above inequalities completes the proof. □

From (3.5), a direct observation is that if we can ensure $S^k_{HLC} \geq 0$ for $k = 0, \ldots, T-1$ and somehow determine the relation between η_k and τ_k, the inequality (3.5) potentially telescopes. Note that the goal of building contraction in one step is to form a relation that telescopes. Since then, we can depict the difference between the error at the output point and the initial point. A simple case is that $S^k_{HLC} \geq 0$ is satisfied and $\eta_k = \frac{1}{C\tau_k^\beta}$ with two constants $C > 0$ and $\beta > 0$ given. In this case, all the possible parameter settings are concentrated into the choice of $\{\tau_k\}_{k=0}^{T-1}$ and we can conclude the following result:

Proposition 3.2.2. *Suppose that we have sequences of vectors $\{y_k\}_{k=0}^T$, $\{z_k\}_{k=0}^T$ that satisfy the following contraction for any $k \in \{0, \ldots, T-1\}$:*

$$\frac{1}{\tau_k^\rho}\big(f(y_{k+1}) - f(x^\star)\big) \leq \frac{1-\tau_k}{\tau_k^\rho}\big(f(y_k) - f(x^\star)\big) + C_0\big(V_d(x^\star, z_k) - V_d(x^\star, z_{k+1})\big), \quad (3.7)$$

where $\rho \geq 1$ and $C_0 > 0$ are constants and $\{\tau_k\}_{k=0}^{T-1}$ are variables with $0 < \tau_k \leq \frac{\rho}{\rho+1} < 1$, we can conclude that

$$f(y_T) - f(x^\star) \leq \frac{(\rho+1)^{\rho-1}\big(f(y_0) - f(x^\star)\big) + C_0\rho^\rho V_d(x^\star, z_0)}{(T+\rho)^\rho},$$

which implies an $\mathcal{O}(1/T^\rho)$ convergence rate.

Proof. First, for any $k \in \{0, \ldots, T-1\}$, by setting $\tau_k = \frac{\rho}{k+\rho+1}$, we have

$$\left(1 - \frac{1}{k+\rho+2}\right)^\rho \geq \frac{k+2}{k+\rho+2} \implies \frac{1}{\tau_k^\rho} \geq \frac{1-\tau_{k+1}}{\tau_{k+1}^\rho}.$$

Then, by telescoping (3.7) from $k = 0, \ldots, T-1$, we obtain

$$\frac{1}{\tau_{T-1}^\rho}\big(f(y_T) - f(x^\star)\big) \leq \frac{1-\tau_0}{\tau_0^\rho}\big(f(y_0) - f(x^\star)\big) + C_0 V_d(x^\star, z_0).$$

Substituting the parameter setting for τ_{T-1} completes the proof. □

After clearly identifying the target in analysis, we then focus on how to ensure the crucial requirement: $S^k_{HLC} \geq 0$ for $k = 0, \ldots, T-1$. Note that since S^k_{HLC}

contains η_k and τ_k (see (3.4)), the relation between η_k and τ_k is always determined when attempting to ensure this requirement. First, let us identify the terms in S^k_{HLC}:

$$S^k_{HLC} = \underbrace{\langle \nabla f(y_{k+1}), z_{k+1} - z_k \rangle + \frac{1}{\eta_k} V_d(z_{k+1}, z_k)}_{\text{Mirror Terms (Step 3)}} + \underbrace{\frac{1}{\tau_k} \langle \nabla f(y_{k+1}), x_{k+1} - y_{k+1} \rangle}_{\text{High-order Term (Step 2)}}. \quad (3.8)$$

For the mirror terms, it is standard to lower bound them assuming that $V_d(\cdot, \cdot)$ is uniformly convex at some degree[6].

Lemma 3.2.3. *If $d(\cdot)$ is m-uniformly convex with parameter μ_m, then the mirror terms in S^k_{HLC} satisfy*

$$\langle \nabla f(y_{k+1}), z_{k+1} - z_k \rangle + \frac{1}{\eta_k} V_d(z_{k+1}, z_k) \geq -\frac{m-1}{m} \left(\frac{\eta_k}{\mu_m} \right)^{\frac{1}{m-1}} \|\nabla f(y_{k+1})\|^{\frac{m}{m-1}}.$$

Proof. Using Young's inequality, we have

$$\begin{aligned}
&\langle \nabla f(y_{k+1}), z_{k+1} - z_k \rangle + \frac{1}{\eta_k} V_d(z_{k+1}, z_k) \\
&\geq -\|\nabla f(y_{k+1})\| \cdot \|z_k - z_{k+1}\| + \frac{\mu_m}{\eta_k m} \|z_k - z_{k+1}\|^m \\
&\geq -\left(\frac{m-1}{m} \left(\frac{\eta_k}{\mu_m} \right)^{\frac{1}{m-1}} \|\nabla f(y_{k+1})\|^{\frac{m}{m-1}} + \frac{\mu_m}{\eta_k m} \|z_k - z_{k+1}\|^m \right) + \frac{\mu_m}{\eta_k m} \|z_k - z_{k+1}\|^m \\
&= -\frac{m-1}{m} \left(\frac{\eta_k}{\mu_m} \right)^{\frac{1}{m-1}} \|\nabla f(y_{k+1})\|^{\frac{m}{m-1}}.
\end{aligned}$$

□

Then, based on the above lower bound and regarding S^k_{HLC} in the one-step contraction (3.5), we can now interpret the function of mirror step in Algorithm 6 as lower-bounding the gradient norm $\|\nabla f(y_{k+1})\|^{\frac{m}{m-1}}$, which is consistent with the interpretation in Linear Coupling [4].

Finally, in order to ensure $S^k_{HLC} \geq 0$, the only problem left is how to upper bound the gradient norm by the high-order term in S^k_{HLC}. The intuition here is also generalized naturally from the first-order case where we use gradient steps to provide such an upper bound [4]. In the following subsections, we discuss several methods that can be regarded as instances of HLC.

[6]For example, standard mirror descent requires $V_d(\cdot, \cdot)$ to be strongly convex (2-uniformly convex).

3.3 Accelerated Second-order Methods

In the second-order case, which is also the first high-order case where acceleration was introduced, we consider problem (2.1) where $f(\cdot)$ has L_2-Lipschitz continuous Hessian (Assumption 3.1.1 with $m = 2$). Known results include that Acc-Cubic achieves an $\mathcal{O}(1/T^3)$ convergence rate [30] and A-NPE achieves an $\mathcal{O}(1/T^{3.5})$ rate [28]. It seems that they are equipped with different analysis and acceleration techniques. To clearly understand this diversity, we cast both of them into instances of HLC (Algorithm 6) and then follow the analysis framework of HLC to provide unified proofs for them.

3.3.1 Acc-Cubic

Observing the auxiliary problems of the accelerated scheme in [30], we may cast Acc-Cubic as the following instance of HLC:

Instance 1. *Acc-Cubic corresponds to a case of Algorithm 6 where* $m = 2$, $\forall x, y \in \mathbb{E}, R(x, y) = \frac{1}{6}\|x-y\|^3$ *and* $d(\cdot)$ *is* 3*-uniformly convex with parameter* μ_3 *(e.g., choose* $d(\cdot) = d_3(\cdot)$*).*

As specified at (3.8), the major task is to upper bound the gradient norm at y_{k+1} by the high-order term. The following lemma provides this critical relation, which is identical to Lemma 6 in [30].

Lemma 3.3.1. *In Instance 1, for* $k = 0, \ldots, T-1$*, if* $\gamma_k = \frac{1}{2L_2}$*, then*

$$\langle \nabla f(y_{k+1}), x_{k+1} - y_{k+1} \rangle \geq \sqrt{\frac{2}{3L_2}} \|\nabla f(y_{k+1})\|^{\frac{3}{2}}.$$

Then, we obtained the upper bound (Lemma (3.3.1)) and lower bound (Lemma (3.2.3) with $m = 3$) for the gradient norm. The only task left is to carefully balance them to ensure $S^k_{HLC} \geq 0$.

Proposition 3.3.2. *In Instance 1, for* $k = 0, \ldots, T-1$*, by setting* $\eta_k = \frac{3\mu_3}{2\tau_k^2 L_2}$*, we have* $S^k_{HLC} \geq 0$.

Proof. Based on the choices in Instance 1 and using Lemma 3.3.1, Lemma 3.2.3 with $m = 3$, we have

$$\begin{aligned} S^k_{HLC} &\geq \langle \nabla f(y_{k+1}), z_{k+1} - z_k \rangle + \frac{\mu_3}{3\eta_k}\|z_{k+1} - z_k\|^3 + \sqrt{\frac{2}{3\tau_k^2 L_2}}\|\nabla f(y_{k+1})\|^{\frac{3}{2}} \\ &\geq \left(\sqrt{\frac{2}{3\tau_k^2 L_2}} - \frac{2}{3}\sqrt{\frac{\eta_k}{\mu_3}} \right) \|\nabla f(y_{k+1})\|^{\frac{3}{2}}. \end{aligned}$$

Substituting the choice of η_k completes the proof. □

Thus, by substituting the choice of η_k in Theorem 3.2.1 and since no limitation is imposed on the choice of $\{\tau_k\}_{k=0}^{T-1}$, we can directly apply Proposition 3.2.2 to estimate the convergence rate of Instance 1:

Corollary 3.3.3. *For Instance 1, using the result in Proposition 3.3.2 and based on Theorem 3.2.1, we can apply Proposition 3.2.2 with $\rho = 3$ and $C_0 = \frac{2L_2}{3\mu_3}$, which concludes an $\mathcal{O}(1/T^3)$ convergence rate. Concisely, with $y_0 = z_0 = x_0$ chosen, the following inequality holds at step $T-1$[7]:*

$$f(y_T) - f(x^\star) \leq \frac{16\big(f(x_0) - f(x^\star)\big) + \frac{18}{\mu_3} L_2 V_d(x^\star, x_0)}{(T+3)^3}.$$

As we can see in Proposition 3.3.2, the intuition of coupling is generalized naturally from the first-order case [4].

3.3.2 A-NPE

In this section, we consider a more complicated second-order method A-NPE [28], an accelerated variant of the NPE method analyzed in [27], which achieves the near optimal convergence rate in this case [6]. For simplicity, we do not consider a composite proximal function and the inexactness in the Newton solution of A-NPE, that is we fix $h(\cdot) \equiv 0$ and $\epsilon_{k+1} = 0$ (in the original notation). Similarly, to clearly understand the intuition behind the superior rate, we cast A-NPE as an instance of HLC (Algorithm 6):

Instance 2. *A-NPE corresponds to a case of Algorithm 6 where $m = 2$, $\forall x, y \in \mathbb{E}, R(x,y) = \frac{1}{2}\|x-y\|^2$ and $d(\cdot)$ is strongly convex with parameter μ_2 (e.g., choose $d(\cdot) = d_2(\cdot)$).*

Perhaps the most tricky part of A-NPE is a line search procedure required in each step, which ensures the conditions in the following proposition (here we do not specify the dependency between τ_k and γ_k for ease of reading. As will be specified at (3.10), τ_k depends on γ_k and thus x_{k+1} depends on γ_k.):

Proposition 3.3.4 (Line search, Section 7, [28])**.** *In Instance 2, defining $\sigma = \frac{L_2\gamma_k}{2}\|y_{k+1} - x_{k+1}\|$ with $0 < \sigma_l < \sigma_r < 1$ and $\bar{\rho} > 0$ given as constants, we can use the line search procedure in [28] to either obtain a γ_k that satisfies $\sigma_l \leq \sigma \leq \sigma_r$ or justify that $\|\nabla f(y_{k+1})\| \leq \bar{\rho}$.*

[7]Note that we do not optimize the constants in this chapter for consistency.

In order to analyze the worst-case performance of A-NPE, we assume that A-NPE does not terminate[8] before step T (i.e., we can always find a γ_k in the above proposition for $k = 0, \ldots, T-1$). At this point, the intuition of the line search procedure is still not clear. Let us start with constructing the per-iteration contraction for A-NPE following the analysis of HLC.

Proposition 3.3.5. *In Instance 2, for $k = 0, \ldots, T-1$, if $\eta_k = \frac{\gamma_k \mu_2}{\tau_k}$, then $S^k_{HLC} \geq \frac{1-\sigma_r^2}{2\gamma_k \tau_k}\|y_{k+1} - x_{k+1}\|^2 > 0$.*

Proof. First, based on the optimal condition of y_{k+1}, we can conclude that

$$-\nabla f^{(2)}_{x_{k+1}}(y_{k+1}) = \frac{1}{\gamma_k}(y_{k+1} - x_{k+1}).$$

Using the L_2-Lipschitz continuity of Hessian (3.3) and the definitions in Proposition 3.3.4, we have

$$\begin{aligned}
\frac{L_2^2}{4}\|y_{k+1} - x_{k+1}\|^4 &\geq \|\nabla f(y_{k+1}) - \nabla f^{(2)}_{x_{k+1}}(y_{k+1})\|^2 \\
&= \|\nabla f(y_{k+1}) + \frac{1}{\gamma_k}(y_{k+1} - x_{k+1})\|^2 \\
&= \|\nabla f(y_{k+1})\|^2 + \frac{2}{\gamma_k}\langle \nabla f(y_{k+1}), y_{k+1} - x_{k+1}\rangle \\
&\quad + \frac{1}{\gamma_k^2}\|y_{k+1} - x_{k+1}\|^2, \\
\langle \nabla f(y_{k+1}), x_{k+1} - y_{k+1}\rangle &\geq \frac{\gamma_k}{2}\|\nabla f(y_{k+1})\|^2 + \frac{1-\sigma^2}{2\gamma_k}\|y_{k+1} - x_{k+1}\|^2.
\end{aligned}$$

Then, we can lower bound S^k_{HLC} using the above inequality and Lemma 3.2.3 with $m = 2$,

$$\begin{aligned}
S^k_{HLC} &\geq \left(\frac{\gamma_k}{2\tau_k} - \frac{\eta_k}{2\mu_2}\right)\|\nabla f(y_{k+1})\|^2 + \frac{1-\sigma^2}{2\gamma_k\tau_k}\|y_{k+1} - x_{k+1}\|^2 \\
&\overset{(\eta_k = \frac{\gamma_k\mu_2}{\tau_k})}{=} \frac{1-\sigma^2}{2\gamma_k\tau_k}\|y_{k+1} - x_{k+1}\|^2.
\end{aligned}$$

Based on the property of γ_k in Proposition 3.3.4, $S^k_{HLC} \geq \frac{1-\sigma_r^2}{2\gamma_k\tau_k}\|y_{k+1} - x_{k+1}\|^2 > 0$. □

[8] If the line search fails to find a γ_k in Proposition 3.3.4, the algorithm terminates since y_{k+1} is close enough to the optimal solution set.

Now it is clear that the line search procedure ensures that S^k_{HLC} is bounded away from 0 for $k = 0, \dots, T-1$, which, as we will see, boosts the convergence rate of A-NPE.

By applying Theorem 3.2.1 with S^k_{HLC} bounded as in the above proposition, we obtain the following per-iteration contraction for A-NPE:

$$\frac{\gamma_k}{\tau_k^2}\big(f(y_{k+1}) - f(x^\star)\big) \le \frac{\gamma_k(1-\tau_k)}{\tau_k^2}\big(f(y_k) - f(x^\star)\big) + \frac{1}{\mu_2}\big(V_d(x^\star, z_k) - V_d(x^\star, z_{k+1})\big) - \frac{1-\sigma_r^2}{2\tau_k^2}\|y_{k+1} - x_{k+1}\|^2. \tag{3.9}$$

Another tricky part of A-NPE is that, all the parameter choices should be concentrated into the choice of $\{\gamma_k\}_{k=0}^{T-1}$ since that, based on Proposition 3.3.4, we cannot determine anything "concrete" for the sequence $\{\gamma_k\}_{k=0}^{T-1}$. The only property we can figure out is that $\lim_{k\to\infty}\gamma_k = \infty$, but it does not contribute to formulating a per-iteration contraction that telescopes. Thus, in order to telescope (3.9), for $k = 0, \dots, T-1$, we may adopt the following parameter scheme:

$$\frac{\gamma_k}{\tau_k^2} = \frac{\gamma_{k+1}(1-\tau_{k+1})}{\tau_{k+1}^2} \Rightarrow \frac{\gamma_k}{\tau_k^2}\tau_{k+1}^2 + \gamma_{k+1}\tau_{k+1} - \gamma_{k+1} = 0. \tag{3.10}$$

Then, the dependencies are that given γ_k and τ_k, τ_{k+1} is the positive solution (≤ 1) of the above equation, which correlates to γ_{k+1}. Note that this parameter scheme leaves τ_0 undecided and is indeed identical to the original parameter choice[9] in [28].

Proposition 3.3.6. *In Instance 2, using the parameter scheme* (3.10)*, at iteration* $T-1$*, we have*

$$\begin{aligned}&\frac{\gamma_{T-1}}{\tau_{T-1}^2}\big(f(y_T) - f(x^\star)\big) + \frac{1-\sigma_r^2}{2}\sum_{k=0}^{T-1}\frac{1}{\tau_k^2}\|y_{k+1} - x_{k+1}\|^2 \\ \le &\frac{\gamma_0(1-\tau_0)}{\tau_0^2}\big(f(y_0) - f(x^\star)\big) + \frac{1}{\mu_2}V_d(x^\star, z_0).\end{aligned}$$

Proof. This proposition is a direct result of telescoping (3.9) from $k = 0, \dots, T-1$. □

[9] One can verify that by re-assigning the notation ($\gamma_k \to \lambda_{k+1}$, $\frac{\gamma_k}{\tau_k^2} \to A_{k+1}$, $\frac{\gamma_k}{\tau_k} \to a_{k+1}$), the parameter scheme in this section is equivlent to the original work.

Denote $D_0 = \frac{\gamma_0(1-\tau_0)}{\tau_0^2}\big(f(y_0) - f(x^\star)\big) + \frac{1}{\mu_2}V_d(x^\star, z_0)$. Note that by setting $\tau_0 = 1$, $y_0 = z_0 = x_0$ and choosing $d(\cdot) = \frac{1}{2}\|x - x_0\|^2$, Proposition 3.3.6 is equivalent to Theorem 3.6 in [28].

In comparison with the final contraction of Acc-Cubic (Proposition 3.2.2), the extra term in Proposition 3.3.6 is a direct result of bounding S_{HLC}^k away from 0. This term plays a crucial role in estimating how fast $\frac{\gamma_{T-1}}{\tau_{T-1}^2}$ grows with respect to T. The proof to the next lemma is similar to that of Theorem 4.1 in [28] and is given in Appendix B.2.

Lemma 3.3.7. *Based on Proposition 3.3.4, 3.3.6 and the parameter scheme* (3.10)*, setting $\tau_0 = 1$, the following inequality holds:*

$$\frac{\gamma_{T-1}}{\tau_{T-1}^2} \geq \sqrt{\frac{2(1-\sigma_r^2)}{3^7 D_0}}\frac{\sigma_l}{L_2} \cdot T^{\frac{7}{2}}.$$

Now it is clear that using Lemma 3.3.7 and Proposition 3.3.6, we can obtain an $\mathcal{O}(1/T^{3.5})$ convergence rate for A-NPE.

3.3.3 Comments

Based on the analysis in this subsection, some interesting observations can be made:

1. The only source of cumulated information in Acc-Cubic is the previous iterates y_k and z_k. In comparison, A-NPE utilizes another cumulated information to build its convergence rate, which is the lower bound of $\sum_{k=0}^{T-1} S_{HLC}^k$. In some sense, the scope of A-NPE is more "global" than Acc-Cubic.

2. For Acc-Cubic, both the high-order steps and mirror steps themselves (i.e., setting $\tau_k = 0$ or 1, respectively) are convergent sequences whose convergence rates are $\mathcal{O}(1/T^2)$ [30] and $\mathcal{O}(1/T)$, respectively (assuming a bounded level set); For A-NPE, the high-order steps themselves (together with the line search procedure), which are exactly the steps of NPE [27], achieve an $\mathcal{O}(1/T^{1.5})$ rate on the ergodic mean of the iterates (Proposition 7.5 [27]). In fact, for the auxiliary problem in one high-order step of A-NPE, the line search procedure in Proposition 3.3.4 only ensures it to be an upper estimation at point y_{k+1}. Compared with solving a global upper estimation in each Cubic step, this choice complicates the analysis and also does not yield a better convergence rate. This observation suggests that we may not need to choose a good convergent sequences to be the high-order steps in HLC, since, as reflected in Theorem 3.2.1,

the only requirement for the high-order steps is to provide a good lower bound for $\langle \nabla f(y_{k+1}), x_{k+1} - y_{k+1} \rangle$.

Although A-NPE achieves a faster iteration rate than Acc-Cubic, from a practical viewpoint, considering the oracle efficiency[10] of these two algorithms, the superior iteration rate may be killed by the additional logarithm time oracle calls in each A-NPE step. To make this point clear, as mentioned in footnote 2, if the required accuracy is ϵ, A-NPE and Acc-Cubic require $O\left(\epsilon^{-2/7}\log(\frac{1}{\epsilon})\right)$ and $O(\epsilon^{-1/3})$ oracle calls to reduce the additive error to ϵ, respectively. If we ignore the constants and other variables in these bounds, we may need to optimize the problem to 10^{-32} accuracy in order to show the superiority of the improved rate. This problem is exacerbated when considering the tensor variants for these two algorithms in the following section.

3.4 Accelerated Tensor Methods

As a natural generalization of second-order methods, in this subsection, we consider accelerated tensor (or high-order) methods. Similarly, we focus on problem (2.1) under high-order setting, which is that $f(\cdot)$ has L_m-Lipschitz continuous m-th order derivative (Assumption 3.1.1). In [8], Baes generalized the idea in Acc-Cubic to the high-order setting and achieves an $\mathcal{O}(1/T^{m+1})$ rate. Recently, Nesterov [32] revisited this problem and discussed ways to improve the implementability of accelerated tensor methods. Two concurrent works [15, 9] propose generalized versions of A-HPE and they all achieve an $\mathcal{O}(1/T^{\frac{3m+1}{2}})$ rate.

3.4.1 Acc-Tensor

The accelerated tensor methods (Acc-Tensor) based on the *estimate sequence* technique have been discussed in [8, 32]. The algorithm structures of the two methods are almost identical with the only difference in the regularization parameter of the high-order steps. The parameter choice in [32] ensures the convexity of the auxiliary problem, which makes it easier to solve. Here we consider the Nesterov's version of Acc-Tensor and cast it as an instance of HLC.

Instance 3. *Acc-Tensor corresponds to a case of Algorithm 6 where* $m \geq 2$ *and* $\forall x, y \in \mathbb{E}, R(x,y) = \frac{m}{(m+1)!}\|x-y\|^{m+1}$ *and* $d(\cdot)$ *is* $(m+1)$*-uniformly convex with parameter* μ_{m+1} *(e.g., choose* $d(\cdot) = d_{m+1}(\cdot)$*).*

[10] Oracle calls are always considered as the most computationally intensive operation in the algorithm.

Similar to Acc-Cubic, the following lemma lower-bounds the high-order term in S_{HLC}^k to the gradient norm, which is identical to Corollary 1 [32].

Lemma 3.4.1. *In Instance 3, for $k = 0, \ldots, T-1$, if $\gamma_k = \frac{1}{2L_m}$, then*

$$\langle \nabla f(y_{k+1}), x_{k+1} - y_{k+1} \rangle \geq \frac{C_m}{L_m^{\frac{1}{m}}} \|\nabla f(y_{k+1})\|^{\frac{m+1}{m}},$$

where

$$C_m = \frac{m}{2(m-1)} \left(\frac{3(m+1)}{m-1} \right)^{\frac{m-1}{2m}} ((m-1)!)^{\frac{1}{m}}.$$

Then, following the same strategy in Acc-Cubic, η_k is chosen to ensure $S_{HLC}^k \geq 0$.

Proposition 3.4.2. *In Instance 3, for $k = 0, \ldots, T-1$, if*

$$\eta_k = \frac{(m+1)^m C_m^m \mu_{m+1}}{m^m \tau_k^m L_m},$$

then $S_{HLC}^k \geq 0$.

Proof. Based on the choices in Instance 3 and using Lemma 3.4.1 as well as Lemma 3.2.3 (with $d(\cdot)$ $(m+1)$-uniformly convex), we have

$$S_{HLC}^k \geq \left(\frac{C_m}{\tau_k L_m^{\frac{1}{m}}} - \frac{m}{m+1} \left(\frac{\eta_k}{\mu_{m+1}} \right)^{\frac{1}{m}} \right) \|\nabla f(y_{k+1})\|^{\frac{m+1}{m}}.$$

Substitute the choice of η_k completes the proof. □

Finally, we can substitute this η_k in Theorem 3.2.1 and apply Proposition 3.2.2 to estimate the convergence rate of Instance 3:

Corollary 3.4.3. *For Instance 3, using the result in Proposition 3.4.2 and based on Theorem 3.2.1, we can apply Proposition 3.2.2 with $\rho = m+1$ and*

$$C_0 = \frac{m^m L_m}{(m+1)^m C_m^m \mu_{m+1}},$$

which concludes an $\mathcal{O}(1/T^{m+1})$ convergence rate. Concisely, with $y_0 = z_0 = x_0$ chosen, the following inequality holds at step $T-1$:

$$f(y_T) - f(x^\star) \leq \frac{(m+2)^m \big(f(x_0) - f(x^\star)\big) + C_0 (m+1)^{m+1} V_d(x^\star, x_0)}{(T+m+1)^{m+1}}.$$

3.4.2 Generalized A-HPE

Here we choose the one in [9] (Accelerated Taylor Descent, ATD) as the generalized A-HPE we discuss in this section. Although there seems to be some differences in the versions [15] and [9], their convergence results are basically the same. It can be verified that the Accelerated Taylor Descent can be cast into the following instance of HLC (Algorithm 6):

Instance 4. *ATD corresponds to a case of Algorithm 6 where* $m \geq 2$ *and* $\forall x, y \in \mathbb{E}, R(x,y) = \frac{\gamma_k L_m}{m!}\|x-y\|^{m+1}$ *(γ_k is canceled in the high-order step) and* $d(\cdot)$ *is strongly convex with parameter* μ_2 *(e.g., choose* $d(\cdot) = d_2(\cdot)$*).*

First, similar to A-NPE, we specify the parameter scheme for ATD:

$$\frac{\gamma_k}{\tau_k^2} = \frac{\gamma_{k+1}(1-\tau_{k+1})}{\tau_{k+1}^2} \Rightarrow \frac{\gamma_k}{\tau_k^2}\tau_{k+1}^2 + \gamma_{k+1}\tau_{k+1} - \gamma_{k+1} = 0. \tag{3.11}$$

As we can see, the choice of γ_k affects τ_k and thus affects x_{k+1} in the high-order step. As is the case of A-NPE, we need the following binary search result:

Proposition 3.4.4 (Binary search, Theorem 4.6, [9])**.** *Let* $\epsilon > 0$ *be the required accuracy, defining* $\sigma_m = \frac{L_m\gamma_k}{(m-1)!}\|y_{k+1} - x_{k+1}\|^{m-1}$ *and using the parameter scheme* (3.11)*, we can either find a* γ_k *that satisfies* $\frac{1}{2} \leq \sigma_m \leq \frac{m}{m+1}$ *or we can conclude that* $f(y_{k+1}) - f(x^\star) \leq \epsilon$*.*

Actually, the choice $R(x,y)$ ensures that the sub-problem in the high-order step is strictly convex, which contributes to the result that the complexity of binary search is controlled at $O(\log(1/\epsilon))$. Then, we consider constructing a contraction for ATD.

Proposition 3.4.5. *In Instance 4, for* $k = 0, \ldots, T-1$*, if* $\eta_k = \frac{\gamma_k\mu_2}{\tau_k}$*, then* $S_{HLC}^k \geq \frac{1-\sigma_r^2}{2\gamma_k\tau_k}\|y_{k+1} - x_{k+1}\|^2 > 0$*.*

Proof. Since y_{k+1} is the optimal solution in Step 2, Algorithm 6 (the sub-problem is strictly convex), the following holds:

$$-\nabla f_{x_{k+1}}^{(m)}(y_{k+1}) = \frac{m+1}{m!}L_m\|y_{k+1} - x_{k+1}\|^{m-1}(y_{k+1} - x_{k+1}).$$

Denote $\mathcal{C}_m \triangleq \frac{m+1}{m!}L_m\|y_{k+1}-x_{k+1}\|^{m-1} = \frac{m+1}{m}\cdot\frac{\sigma_m}{\gamma_k} \leq \frac{1}{\gamma_k}$. Based on the L_m-Lipschitz continuity of the mth-order derivative (3.3), we have

$$\begin{aligned}\frac{L_m^2}{(m!)^2}\|y_{k+1}-x_{k+1}\|^{2m} &\geq \|\nabla f(y_{k+1})-\nabla f^{(m)}_{x_{k+1}}(y_{k+1})\|^2\\ &= \|\nabla f(y_{k+1})\|^2 + 2\mathcal{C}_m\langle\nabla f(y_{k+1}), y_{k+1}-x_{k+1}\rangle\\ &\quad + \mathcal{C}_m^2\|y_{k+1}-x_{k+1}\|^2,\\ \langle\nabla f(y_{k+1}), x_{k+1}-y_{k+1}\rangle &\geq \frac{\gamma_k}{2}\|\nabla f(y_{k+1})\|^2 + \frac{\sigma_m}{2\gamma_k}\cdot\frac{m+2}{m+1}\|y_{k+1}-x_{k+1}\|^2.\end{aligned}$$

Using Lemma 3.2.3 with $m=2$ and based on Proposition 3.4.4, we can conclude that

$$S^k_{HLC} \geq \left(\frac{\gamma_k}{2\tau_k}-\frac{\eta_k}{2\mu_2}\right)\|\nabla f(y_{k+1})\|^2 + \frac{1}{4\tau_k\gamma_k}\cdot\frac{m+2}{m+1}\|y_{k+1}-x_{k+1}\|^2.$$

Substitute the choice of η_k completes the proof. □

Proposition 3.4.6. *In Instance 4, using Theorem 3.2.1, Proposition 3.4.5 and the parameter scheme* (3.11)*, at iteration $T-1$, we have*

$$\frac{\gamma_{T-1}}{\tau^2_{T-1}}\big(f(y_T)-f(x^\star)\big) + \frac{m+2}{4(m+1)}\sum_{k=0}^{T-1}\frac{1}{\tau_k^2}\|y_{k+1}-x_{k+1}\|^2 \leq D_0,$$

where $D_0 \triangleq \frac{\gamma_0(1-\tau_0)}{\tau_0^2}\big(f(y_0)-f(x^\star)\big) + \frac{1}{\mu_2}V_d(x^\star, z_0)$.

Proof. Using Theorem 3.2.1, we get the per-iteration contraction,

$$\begin{aligned}\frac{\gamma_k}{\tau_k^2}\big(f(y_{k+1})-f(x^\star)\big) \leq \frac{\gamma_k(1-\tau_k)}{\tau_k^2}\big(f(y_k)-f(x^\star)\big) + \frac{1}{\mu_2}\big(V_d(x^\star,z_k)-V_d(x^\star,z_{k+1})\big)\\ - \frac{1}{4\tau_k^2}\cdot\frac{m+2}{m+1}\|y_{k+1}-x_{k+1}\|^2.\end{aligned}$$

Telescope the above inequality from $k=0,\ldots,T$ completes the proof. □

Finally, we can estimate how fast $\frac{\gamma_{T-1}}{\tau^2_{T-1}}$ grows with respect to T using similar techniques in proving Lemma 3.3.7. The proof to the following lemma is given in Appendix B.3.

Lemma 3.4.7. *Based on Proposition 3.4.4, 3.4.6 and the parameter scheme* (3.11)*, setting* $\tau_0 = 1$*, the following inequality holds:*

$$\frac{\gamma_{T-1}}{\tau_{T-1}^2} \geq \left(\frac{m+2}{4D_0(m+1)}\right)^{\frac{m-1}{2}} \left(\frac{1}{m+1}\right)^{\frac{3m+1}{2}} \frac{(m-1)!}{2L_m} \cdot T^{\frac{3m+1}{2}}.$$

From Lemma 3.4.7 and Proposition 3.4.6, it is clear that GA-HPE converges at the rate of $\mathcal{O}(1/T^{\frac{3m+1}{2}})$.

3.4.3 Comments

Similar to the discussions in Section 3.3.3, generalized A-HPE has the same practical issue when compared with Acc-Tensor, that we may need a ridiculously small ϵ to show the improvement of the rate $\mathcal{O}(1/T^{\frac{3m+1}{2}})$. Actually, this issue is exacerbated when m is larger.

An analytical observation is that no matter how we change the order of the regularization term $R(x, y)$, we always need a bounded term $\gamma_k\|y_{k+1} - x_{k+1}\|^{m-1}$ in order to achieve acceleration[11], which means a binary search is inevitable in this case. On the other hand, altering the order of $R(x, y)$ may cause the loss of strict convexity in the high-order step. This observation suggests that in order to design the optimal high-order method, simply change the the order of $R(x, y)$ is not enough.

3.5 HLC in the first-order case

Interesting though, in the first-order case (Assumption 3.1.1 with $m = 1$), by setting $m = 1$ in Algorithm 6, we get an accelerated method that is different from the standard Nesterov's accelerated method (AGD) [33]. Such a method is implied but not analyzed in [8]. A recently work [12] proposed an accelerated extra-gradient method (AXGD), whose algorithm structure is significantly different from AGD and works with a generic norm. In this subsection, we show that AXGD in the Euclidean norm setting (E-AXGD) is equivalent to this accelerated first-order method.

Instance 5. *It can be verified that E-AXGD corresponds to a case of Algorithm 6 where* $m = 1$*,* $\forall x, y \in \mathbb{E}, R(x, y) = \frac{1}{2}\|x - y\|^2$ *and* $d(x) = \frac{1}{2}\|x - x_0\|^2$*. Specially,* $\gamma_k = \eta_k\tau_k$ *is explicitly chosen based on original parameter choice.*

[11] This is because in the analysis, when $\gamma_k\|y_{k+1} - x_{k+1}\|^{m-1}$ is bounded, we can always fold some unwanted terms, which correlate to $\|y_{k+1} - x_{k+1}\|^{m-1}$, to be constants without affecting the proof.

Then, based on the analysis framework of HLC, we can conclude the following results for E-AXGD:

Proposition 3.5.1. *If* $\eta_k = \frac{1}{L_1\tau_k}$, *then* $S^k_{HLC} \geq 0$.

Proof. With L_1-Lipschitz continuous gradient assumption and the first-order optimality condition, we have

$$\begin{aligned} L_1^2\|y_{k+1} - x_{k+1}\|^2 &\leq \|\nabla f(y_{k+1}) - \nabla f(x_{k+1})\|^2 \\ &\leq \|\nabla f(y_{k+1}) + \frac{1}{\eta_k\tau_k}(y_{k+1} - x_{k+1})\|^2, \\ \langle \nabla f(y_{k+1}), x_{k+1} - y_{k+1}\rangle &\geq \frac{\eta_k\tau_k}{2}\|\nabla f(y_{k+1})\|^2 + \left(\frac{1}{2\eta_k\tau_k} - \frac{\eta_k\tau_k L_1^2}{2}\right)\|y_{k+1} - x_{k+1}\|^2 \\ &\overset{(\eta_k = \frac{1}{L_1\tau_k})}{\geq} \frac{1}{2L_1}\|\nabla f(y_{k+1})\|^2. \end{aligned}$$

Thus,

$$\begin{aligned} S^k_{HLC} &\geq \frac{1}{2L_1\tau_k}\|\nabla f(y_{k+1})\|^2 + \langle \nabla f(y_{k+1}), z_{k+1} - z_k\rangle + \frac{1}{2\eta_k}\|z_{k+1} - z_k\|^2 \\ &\geq \frac{1}{2L_1\tau_k}\|\nabla f(y_{k+1})\|^2 - \frac{\eta_k}{2}\|\nabla f(y_{k+1})\|^2 = 0. \end{aligned}$$

□

Then, it is clear that by using Theorem 3.2.2, E-AXGD yields an $\mathcal{O}(1/T^2)$ iteration complexity. In fact, we can derive E-AXGD by degenerating both Acc-Cubic and A-NPE (or A-HPE) into the first-order case.

3.6 Correlation to Linear Coupling

For $y \in \mathbb{E}$, the high-order step in HLC (Algorithm 6) uses the Taylor model $f^{(m)}_{x_{k+1}}(y)$ to approximate $f(y)$ and thus this step is close to the proximal point step. Based on this, the high-order step can be interpreted as a high-order approximate backward-Euler step. For Linear Coupling, this step becomes the classic forward-Euler step. Thus, using the ODE interpretation in [19], the correlation here is that HLC and Linear Coupling are applying different Euler discretization to the accelerated-mirror-descent dynamics. From an analytical viewpoint, the trick used in Linear Coupling is slightly more involved. Unlike the clear distinction of the contraction and the residual term in Theorem 3.2.1, the contraction of Linear Coupling is not obviously shown in its analysis.

3.7 Open problems

We identify the following open problems in highly-smooth convex optimization:

- *Based on the HLC framework, can we design an optimal second-order method that eliminates the log factor in the oracle complexity of A-NPE?* One possible solution is to design better auxiliary problems and thus the line search procedure is dropped. As we can see in this chapter, all these methods focus on modifying the high-order step (i.e., to upper bound the gradient norm) and use hyperplane to construct the lower bound. Can we improve the mirror step by taking high-order Lipschitz continuity (i.e., $f(y) \geq f_x^{(m)}(y) - \frac{L_m}{(m+1)!}\|y-x\|^{m+1}$) into consideration?

- *Extend high-order methods into non-Euclidean norm setting.* Currently, high-order methods does not work with non-Euclidean norm in the definition of Lipschitz continuity due to the proofs in Proposition 3.3.2, 3.3.5, 3.4.2, 3.4.5 and 3.5.1. Since AXGD works for non-Euclidean norm, can high-order methods adopt the techniques in AXGD?

- *Proximal accelerated high-order methods?* It seems that accelerated high-order methods cannot work with a proximal part in the objective and the current methods only work in the unconstrained setting. Deriving a proximal variant requires more delicate design of the mirror steps in Algorithm 6.

Chapter 4

Conclusion

In this thesis, we discussed accelerated methods in finite-sum convex optimization and high-order convex optimization. For the finite-sum case, based on SVRG and SAGA, which are the most popular stochastic variance reduced methods, we proposed two optimal algorithms, namely, MiG and SSNM. The proposed methods are all as implementable as SVRG and SAGA while enjoy a faster convergence rate, which implies that they could potentially benefit many large-scale real world problems. We also discussed how to make MiG and SSNM work under exactly the same objective assumptions of SVRG (Prox-SVRG) and SAGA, which shows that our acceleration techniques are direct and elegant. For the high-order case, we provided a structured understanding of accelerated methods under high-order smoothness assumption. Our proposed framework, i.e., High-order Linear Coupling, identifies the limitations of existing work and some potentials for future improvement. We also discussed some interesting connections between first-order and high-order acceleration.

Appendix A

Asynchronous and Sparse MiG

The asynchronous and sparse variant of MiG is formally given in Algorithm 7. As we can see, it is slightly different from MiG (Algorithm 3). We explain these differences by making the following remarks:

- *Sparse approximate gradient* $\tilde{\nabla}(\hat{y})$. In order to perform fully sparse updates, following [26], we use a diagonal matrix D to re-weigh the dense vector μ_s, whose entries are the inverse probabilities $\{p_k^{-1}\}$ of the corresponding coordinates $\{k \,|\, k=1,\ldots,d\}$ belonging to a uniformly sampled support T_{i_j} of sample i_j. P_{i_j} is the projection matrix for the support T_{i_j}. We define $D_{i_j} = P_{i_j} D$, which ensures the unbiasedness $\mathbb{E}_{i_j}\left[D_{i_j}\mu_s\right] = \mu_s$. Here we also define $D_m = \max_{k=1\ldots d} p_k^{-1}$ for future usage. Note that we only need to compute y on the support of sample i_j, and hence the entire inner loop updates sparsely.

- *Update $\tilde{x}$ with uniform average.* In the sparse and asynchronous setting, a weighted average in Algorithm 3 is not effective due to the perturbation both in theory and in practice. Thus, we choose a simple uniform average scheme for a better practical performance.

- *Two update options.* The difference between the two options is that Option II corresponds to averaging "fake" iterates defined at (A.2), while Option I is the average of inconsistent read[1] of x. Since the averaging scheme in Option II is not proposed before, we refer to it as "*fake average*". Option II is shown to be highly practical since the "*fake average*" scheme only requires updates on the support of samples and enjoys strong robustness when the actual number

[1]We could use "*fake average*" in Option I, but it leads to a complex proof and a worse convergence rate with factor ($\propto \kappa^{-2}$).

Algorithm 7 Asynchronous Sparse MiG

Input: Initial guess x_0, epoch length m, learning rate η, parameter θ.

$x :=$ shared variable, $\bar{x} :=$ average of x;
$\tilde{x}_0 = x = x_0$;
for $s = 1 \dots \mathcal{S}$ **do**
  Compute $\mu_s = \nabla F(\tilde{x}_{s-1})$ in parallel;
  Option I: $\bar{x} = \mathbf{0}$;
  Option II: $\bar{x} = x$;
  $j = 0$; {inner loop counter}
  while $j < m$ **do** {in parallel}
    $j = j + 1$; // atomic increase counter j
    Sample i_j uniformly in $\{1 \dots n\}$;
    $T_{i_j} :=$ support of sample i_j;
    $[\hat{x}]_{T_{i_j}} :=$ inconsistent read of $[x]_{T_{i_j}}$;
    $[\hat{y}]_{T_{i_j}} = \theta \cdot [\hat{x}]_{T_{i_j}} + (1-\theta) \cdot [\tilde{x}_{s-1}]_{T_{i_j}}$;
    $\tilde{\nabla}(\hat{y}) = \nabla f_{i_j}([\hat{y}]_{T_{i_j}}) - \nabla f_{i_j}([\tilde{x}_{s-1}]_{T_{i_j}}) + D_{i_j}\mu_s$;
    $[u]_{T_{i_j}} = -\eta \cdot \tilde{\nabla}(\hat{y})$;
    // atomic write x, $\bar{x}$ for each coordinate
    $[x]_{T_{i_j}} = [x]_{T_{i_j}} + [u]_{T_{i_j}}$;
    Option I: $\bar{x} = \bar{x} + \frac{1}{m} \cdot \hat{x}$;
    Option II: $[\bar{x}]_{T_{i_j}} = [\bar{x}]_{T_{i_j}} + [u]_{T_{i_j}} \cdot \frac{(m+1-j)_+}{m}$;
  end while
  $\tilde{x}_s = \theta\bar{x} + (1-\theta)\tilde{x}_{s-1}$;
  Option I: $x = \tilde{x}_s$;
  Option II: keep x unchanged;
end for

Output: $\tilde{x}_{\mathcal{S}}$.

of inner loops does not equal to m^2. Thus, Option II leads to a very practical implementation.

Then we consider analyzing the convergence of Asynchronous Sparse MiG. In order to give a clean proof, we first make a simpler assumption on the objective

[2]This phenomenon is prevalent in the asynchronous setting.

function, which is identical to those in [38, 26, 21]:

$$\min_{x\in\mathbb{R}^d} F(x) \triangleq \frac{1}{n}\sum_{i=1}^{n} f_i(x). \tag{A.1}$$

Assumption A.0.1 (Sparse and Asynchronous Settings)**.** *In Problem (A.1), each $f_i(\cdot)$ is L-smooth, and the averaged function $F(\cdot)$ is μ-strongly convex.*

Following [26], our analysis is based on the "fake" iterates x and y, which are defined as:

$$x_j = x_0 - \eta\sum_{i=0}^{j-1}\tilde{\nabla}(\hat{y}_i),\ y_j = \theta x_j + (1-\theta)\tilde{x}_{s-1}, \tag{A.2}$$

where the "perturbed" iterates $\hat{y}$, $\hat{x}$ with perturbation ξ are defined as

$$\hat{y}_j = \theta\hat{x}_j + (1-\theta)\tilde{x}_{s-1},\ \hat{x}_j = x_j + \xi_j. \tag{A.3}$$

Note that y is a temp variable, so the only source of perturbation comes from x. This is the benefit of keeping track of only one variable vector since it controls the perturbation and allows us to give a smooth analysis in asynchrony.

Next we give our convergence result as follows:

Theorem A.0.1. *If Assumption A.0.1 holds, then by choosing $m=60\kappa$, $\eta=1/(5L)$, $\theta=1/6$, suppose τ satisfies $\tau\leq\min\{\frac{5}{4\sqrt{\Delta}}, 2\kappa, \sqrt{\frac{2\kappa}{\sqrt{\Delta}}}\}$ (the linear speed-up condition), Algorithm 7 with Option I has the following oracle complexity:*

$$\mathcal{O}\left((n+\kappa)\log\frac{F(x_0)-F(x^\star)}{\epsilon}\right),$$

where τ represents the maximum number of overlaps between concurrent threads [26] and $\Delta=\max_{k=1\ldots d} p_k$, which is a measure of sparsity [21].

This result is better than that of KroMagnon, which correlates to κ^2 [26], and keeps up with ASAGA [21]. Although without significant improvement on theoretical bounds due to the existence of perturbation, the coupling step of MiG can still be regarded as a simple add-on boosting and stabilizing the performance of SVRG variants.

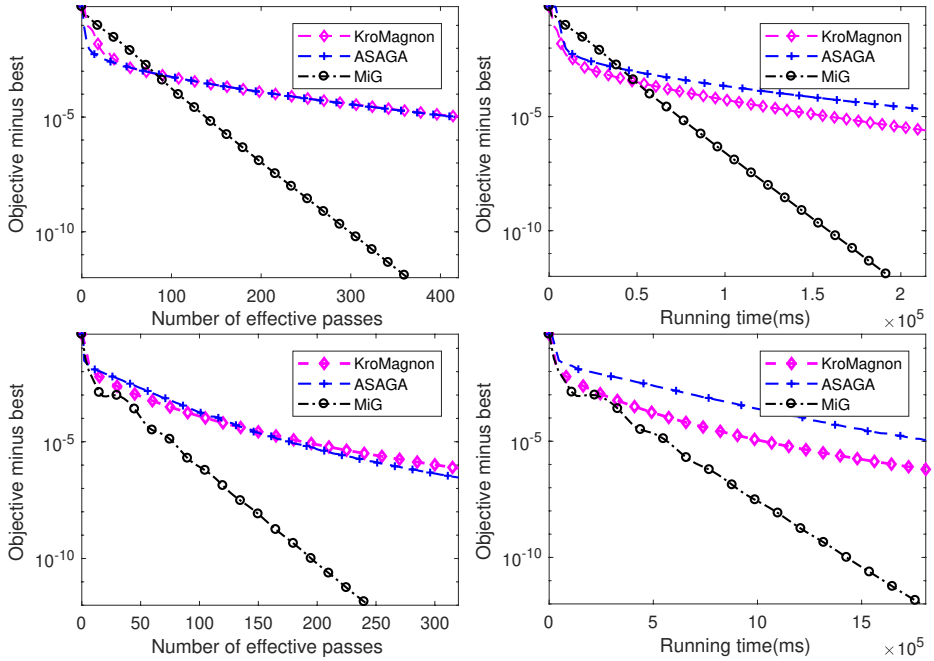

Figure A.1: Comparison of KroMagnon [26], ASAGA [21], and MiG (Algorithm 7 with Option II) with 16 threads. First row: RCV1, ℓ_2-logistic regression with $\lambda = 10^{-9}$. Second row: KDD2010, ℓ_2-logistic regression with $\lambda = 10^{-10}$.

A.1 Experimental results

Unlike in the serial dense case (Algorithm 3) where we have strong theoretical guarantees, in these settings, we mainly focus on practical performance and stability. So we carefully tuned the parameter(s) for each algorithm to achieve a best-tuned performance. We measure the performance on the two sparse datasets listed in Table A.1.

Dataset	# Data	# Features	Density
RCV1	697,641	47,236	1.5×10^{-3}
KDD2010	19,264,097	1,163,024	10^{-6}

Table A.1: Summary of the two sparse data sets.

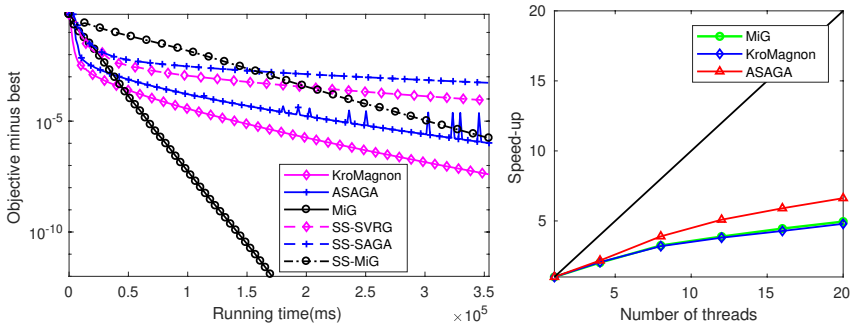

Figure A.2: Speed-up evaluation on RCV1. Left: Evaluation of sub-optimality in terms of running time for asynchronous versions (20 threads) and SS (Serial Sparse) versions. Right: Speed-up of achieving 10^{-5} sub-optimality in terms of the number of threads.

When comparing performance in terms of oracle calls, MiG significantly outperforms other algorithms, as shown in Figure A.1. When considering running time, the difference is narrowed due to the high simplicity of KroMagnon (which only uses one atomic vector) compared with ASAGA (which uses atomic gradient table and atomic gradient average vector) and MiG (which only uses atomic "*fake average*").

We then examine the speed-up gained from more parallel threads on RCV1. We evaluate the improvement of using asynchronous variants (20 threads) and the speed-up ratio as a function of the number of threads as shown in Figure A.2. For the latter evaluation, the running time is recorded when the algorithms achieve 10^{-5} sub-optimality. The speed-up ratio is calculated based on the running time of a single core.

A.2 Proof of Theorem A.0.1

Here we analyze Algorithm 7 based on the "perturbed iterate analysis" framework [26].

To begin with, we need to specify the iterates labeling order, which is crucial in our asynchronous analysis.

Choice of labeling order. There are "Before Read" [26] and "After Read" [21] labeling schemes proposed in recent years which are reasonable in asynchronous analysis. Among these two schemes, "Before Read" requires considering the updates from "fu-

ture", which leads to a complex analysis. "After Read" enjoys a simpler analysis but requires changing the order of sampling step to ensure uniform distributed samples[3]. In order to give a clean proof, we adopt the "After Read" labeling scheme and make the following assumptions:

Assumption A.2.1. *The labeling order increases after the step (14) in Algorithm 7 finished, so the future perturbation is not included in the effect of asynchrony in the current step.*

Assumption A.2.2. *We explicitly assume uniform distributed samples and the independence of the sample i_j with $\hat{x}_{j-1}$.*

In other words, we are analyzing the following procedure:

1. Inconsistent read the iterate $\hat{x}_{j-1}$.
2. Increase iterates counter j and sample a random index i_j.
3. Compute an update $-\eta \cdot \tilde{\nabla}(\hat{y}_{j-1})$.
4. Atomic write the update to shared memory coordinately.

Then, we give the sparse variance bound as follows, which is important in obtaining the linear convergence rate.

Lemma A.2.1 (Sparse Variance Bound)**.** *If Assumption A.0.1 holds, for any $y, \tilde{x} \in \mathbb{R}^d$ and the sample i_j, denote $\tilde{\nabla} = \nabla f_{i_j}(y) - \nabla f_{i_j}(\tilde{x}) + D_{i_j}\nabla F(\tilde{x})$, where D_{i_j} is defined in Section A, we can bound the variance $\mathbb{E}_{i_j}\big[\|\tilde{\nabla}\|^2\big]$ as*

$$\mathbb{E}_{i_j}\big[\|\tilde{\nabla}\|^2\big] \leq 4L\big(F(y) - F(x^\star)\big) + 4L\big(F(\tilde{x}) - F(x^\star)\big).$$

Proof. From Lemma 10 in [26], we have

$$\mathbb{E}_{i_j}\big[\|\tilde{\nabla}\|^2\big] \leq 2\mathbb{E}_{i_j}\big[\|\nabla f_{i_j}(y) - \nabla f_{i_j}(x^\star)\|^2\big] + 2\mathbb{E}_{i_j}\big[\|\nabla f_{i_j}(\tilde{x}) - \nabla f_{i_j}(x^\star)\|^2\big], \quad \text{(A.4)}$$

which provides an upper bound for the variance of the sparse stochastic variance reduced gradient estimator.

From Theorem 2.1.5 in [31], we have

$$\|\nabla f_{i_j}(y) - \nabla f_{i_j}(x^\star)\|^2 \leq 2L\big(f_{i_j}(y) - f_{i_j}(x^\star) - \langle \nabla f_{i_j}(x^\star), y - x^\star\rangle\big).$$

[3]So there are always two versions (analyzed, implemented) of algorithms in the works with "After Read" scheme [21, 37]

Taking expectation with the sample i_j, the above inequality becomes

$$\mathbb{E}_{i_j}\big[\|\nabla f_{i_j}(y) - \nabla f_{i_j}(x^\star)\|^2\big] \le 2L\big(F(y) - F(x^\star)\big).$$

Similarly, we have

$$\mathbb{E}_{i_j}\big[\|\nabla f_{i_j}(\tilde{x}) - \nabla f_{i_j}(x^\star)\|^2\big] \le 2L\big(F(\tilde{x}) - F(x^\star)\big).$$

Substituting the above inequalities into (A.4) yields the desired result. □

From [21], we can model the effect of asynchrony as follows:

$$\hat{x}_j - x_j = \eta \sum_{k=(j-1-\tau)_+}^{j-2} \mathcal{S}_k^j \tilde{\nabla}(\hat{y}_k), \tag{A.5}$$

where $\mathcal{S}_k^j$ is a diagonal matrix with entries in $\{0, +1\}$. This definition models the coordinate perturbation from the past updates. Here τ represents the maximum number of overlaps between concurrent threads [26]. We further denote $\Delta = \max_{k=1\ldots d} p_k$ following [21], which provides a measure of sparsity.

Then we start our analysis with the iterate difference between "fake" y_j and $x^\star$. By expanding the iterate difference and taking expectation with respect to the sample i_j, we get

$$\begin{aligned}
\mathbb{E}_{i_j}\big[\|y_j - x^\star\|^2\big] &= \mathbb{E}_{i_j}\big[\|\theta(x_{j-1} - \eta\cdot\tilde{\nabla}(\hat{y}_{j-1})) + (1-\theta)\tilde{x}_{s-1} - x^\star\|^2\big] \\
&= \mathbb{E}_{i_j}\big[\|y_{j-1} - \eta\theta\cdot\tilde{\nabla}(\hat{y}_{j-1}) - x^\star\|^2\big] \\
&\overset{(\star)}{=} \|y_{j-1} - x^\star\|^2 - 2\eta\theta\langle\nabla F(\hat{y}_{j-1}), \hat{y}_{j-1} - x^\star\rangle + \eta^2\theta^2\mathbb{E}_{i_j}\big[\|\tilde{\nabla}(\hat{y}_{j-1})\|^2\big] \\
&\quad + 2\eta\theta\mathbb{E}_{i_j}\big[\langle\tilde{\nabla}(\hat{y}_{j-1}), \hat{y}_{j-1} - y_{j-1}\rangle\big],
\end{aligned} \tag{A.6}$$

where $(\star)$ uses the unbiasedness $\mathbb{E}_{i_j}\big[\tilde{\nabla}(\hat{y}_{j-1})\big] = \nabla F(\hat{y}_{j-1})$ and the independence Assumption A.2.2.

Using Lemma A.2.1, we get the variance bound

$$\mathbb{E}_{i_j}\big[\|\tilde{\nabla}(\hat{y}_{j-1})\|^2\big] \le 4L\big(F(\hat{y}_{j-1}) - F(x^\star)\big) + 4L\big(F(\tilde{x}_{s-1}) - F(x^\star)\big). \tag{A.7}$$

Using the μ-strongly convex of $F(\cdot)$, we can bound $-\langle\nabla F(\hat{y}_{j-1}), \hat{y}_{j-1} - x^\star\rangle$ as follows:

$$\begin{aligned}
\langle\nabla F(\hat{y}_{j-1}), \hat{y}_{j-1} - x^\star\rangle &\ge F(\hat{y}_{j-1}) - F(x^\star) + \frac{\mu}{2}\|\hat{y}_{j-1} - x^\star\|^2 \\
&\overset{(\star)}{\ge} F(\hat{y}_{j-1}) - F(x^\star) + \frac{\mu}{4}\|y_{j-1} - x^\star\|^2 - \frac{\mu}{2}\|\hat{y}_{j-1} - y_{j-1}\|^2,
\end{aligned} \tag{A.8}$$

where $(\star)$ uses the fact that $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$.

Combining (A.6), (A.7) and (A.8), we get

$$\begin{aligned}\mathbb{E}_{i_j}\big[\|y_j - x^\star\|^2\big] \leq (1 - \frac{\eta\theta\mu}{2})\|y_{j-1} - x^\star\|^2 + \eta\theta\mu\|\hat{y}_{j-1} - y_{j-1}\|^2 \\ + 2\eta\theta\mathbb{E}_{i_j}\big[\langle\tilde{\nabla}(\hat{y}_{j-1}), \hat{y}_{j-1} - y_{j-1}\rangle\big] \\ + (4L\eta^2\theta^2 - 2\eta\theta)\big(F(\hat{y}_{j-1}) - F(x^\star)\big) \\ + 4L\eta^2\theta^2\big(F(\tilde{x}_{s-1}) - F(x^\star)\big). \end{aligned} \tag{A.9}$$

From Lemma 1 in [21], we borrow the notations $C_1 = 1+\sqrt{\Delta}\tau$, $C_2 = \sqrt{\Delta}+\eta\theta\mu C_1$ and bound the asynchronous variance terms $\|\hat{y}_{j-1} - y_{j-1}\|^2$, $\mathbb{E}_{i_j}\big[\langle\tilde{\nabla}(\hat{y}_{j-1}), \hat{y}_{j-1} - y_{j-1}\rangle\big]$ using (A.5) as

$$\mathbb{E}_{i_j}\big[\langle\tilde{\nabla}(\hat{y}_{j-1}), \hat{y}_{j-1} - y_{j-1}\rangle\big] \leq \frac{\eta\theta\sqrt{\Delta}}{2}\sum_{k=(j-1-\tau)_+}^{j-2}\|\tilde{\nabla}(\hat{y}_k)\|^2 + \frac{\eta\theta\sqrt{\Delta}\tau}{2}\mathbb{E}_{i_j}\big[\|\tilde{\nabla}(\hat{y}_{j-1})\|^2\big], \tag{A.10}$$

$$\|\hat{y}_{j-1} - y_{j-1}\|^2 \leq \eta^2\theta^2(1+\sqrt{\Delta}\tau)\sum_{k=(j-1-\tau)_+}^{j-2}\|\tilde{\nabla}(\hat{y}_k)\|^2. \tag{A.11}$$

Upper bounding the asynchronous terms in (A.9) using (A.10) and (A.11), we get

$$\begin{aligned}\mathbb{E}_{i_j}\big[\|y_j - x^\star\|^2\big] \leq (1 - \frac{\eta\theta\mu}{2})\|y_{j-1} - x^\star\|^2 \\ + \eta^2\theta^2(\sqrt{\Delta} + \eta\theta\mu(1+\sqrt{\Delta}\tau))\sum_{k=(j-1-\tau)_+}^{j-2}\|\tilde{\nabla}(\hat{y}_k)\|^2 \\ + (4L\eta^2\theta^2(1+\sqrt{\Delta}\tau) - 2\eta\theta)\big(F(\hat{y}_{j-1}) - F(x^\star)\big) \\ + 4L\eta^2\theta^2(1+\sqrt{\Delta}\tau)\big(F(\tilde{x}_{s-1}) - F(x^\star)\big).\end{aligned}$$

Defining $a_j \triangleq \|y_j - x^\star\|^2$, $\hat{D}_{j-1} = F(\hat{y}_{j-1}) - F(x^\star)$, $\tilde{D}_{s-1} = F(\tilde{x}_{s-1}) - F(x^\star)$ for a

clean proof and rearranging, we obtain

$$\begin{aligned}\mathbb{E}_{i_j}\big[a_j\big] &\le (1-\frac{\eta\theta\mu}{2})a_{j-1} + \eta^2\theta^2 C_2 \sum_{k=(j-1-\tau)_+}^{j-2} \|\tilde{\nabla}(\hat{y}_k)\|^2 + (4L\eta^2\theta^2 C_1 - 2\eta\theta)\hat{D}_{j-1} \\ &\quad + 4L\eta^2\theta^2 C_1 \tilde{D}_{s-1}, \\ (2\eta\theta - 4L\eta^2\theta^2 C_1)\hat{D}_{j-1} &\overset{(\star)}{\le} (a_{j-1} - \mathbb{E}_{i_j}\big[a_j\big]) + \eta^2\theta^2 C_2 \sum_{k=(j-1-\tau)_+}^{j-2} \|\tilde{\nabla}(\hat{y}_k)\|^2 \\ &\quad + 4L\eta^2\theta^2 C_1 \tilde{D}_{s-1}, \end{aligned} \tag{A.12}$$

where $(\star)$ uses the fact that $1 - \frac{\eta\theta\mu}{2} \le 1$.

Summing (A.12) over $j = 1 \dots m$ and taking expectation with all randomness in this epoch, we get

$$\begin{aligned}(2\eta\theta - 4L\eta^2\theta^2 C_1)\sum_{j=1}^{m}\mathbb{E}\big[\hat{D}_{j-1}\big] &\le (a_0 - \mathbb{E}\big[a_m\big]) + \eta^2\theta^2 C_2 \sum_{j=1}^{m}\sum_{k=(j-1-\tau)_+}^{j-2} \mathbb{E}\big[\|\tilde{\nabla}(\hat{y}_k)\|^2\big] \\ &\quad + 4L\eta^2\theta^2 C_1 m \tilde{D}_{s-1}. \end{aligned} \tag{A.13}$$

Then we focus on upper bounding the second term on the right side of (A.13),

$$\begin{aligned}\sum_{j=1}^{m}\sum_{k=(j-1-\tau)_+}^{j-2} \mathbb{E}\big[\|\tilde{\nabla}(\hat{y}_k)\|^2\big] &\le \tau \sum_{j=1}^{m-1} \mathbb{E}\big[\|\tilde{\nabla}(\hat{y}_{j-1})\|^2\big] \\ &\le \tau \sum_{j=1}^{m} \mathbb{E}\big[\|\tilde{\nabla}(\hat{y}_{j-1})\|^2\big] \\ &\overset{(\star)}{\le} 4L\tau\big(\sum_{j=1}^{m}\mathbb{E}\big[\hat{D}_{j-1}\big] + m\tilde{D}_{s-1}\big), \end{aligned}$$

where $(\star)$ uses the variance bound (A.7).

Substituting the above inequality into (A.13), we get

$$\begin{aligned}(2\eta\theta - 4L\eta^2\theta^2 C_1 - 4L\eta^2\theta^2 C_2\tau)\sum_{j=1}^{m}\mathbb{E}\big[\hat{D}_{j-1}\big] &\le a_0 + (4L\eta^2\theta^2 C_1 m + 4L\eta^2\theta^2 C_2\tau m)\tilde{D}_{s-1}, \\ \tilde{D}_s &\overset{(\star)}{\le} \frac{\frac{2}{\mu} + 4L\eta^2\theta^2 C_1 m + 4L\eta^2\theta^2 C_2\tau m}{(2\eta\theta - 4L\eta^2\theta^2 C_1 - 4L\eta^2\theta^2 C_2\tau)m} \cdot \tilde{D}_{s-1}, \end{aligned}$$

where $(\star)$ uses the μ-strongly convex of $F(\cdot)$ and $\tilde{x}_0 = x_0 = y_0$, $\tilde{x}_s = \frac{1}{m}\sum_{j=0}^{m-1} \hat{y}_j$.

By choosing $m = 60\kappa$, $\eta = \frac{1}{5L}$, $\theta = \frac{1}{6}$, we get

$$\tilde{D}_s \leq \frac{2 + \frac{4}{15}(C_1 + C_2\tau)}{4 - \frac{4}{15}(C_1 + C_2\tau)} \cdot \tilde{D}_{s-1}.$$

In order to ensure linear speed up, τ needs to satisfy the following constraint:

$$\rho \triangleq \frac{2 + \frac{4}{15}(C_1 + C_2\tau)}{4 - \frac{4}{15}(C_1 + C_2\tau)} \leq 1.$$

By simply setting $\tau \leq \min\{\frac{5}{4\sqrt{\Delta}}, 2\kappa, \sqrt{\frac{2\kappa}{\sqrt{\Delta}}}\}$, the above constraint is satisfied with $\rho \leq 0.979$, which implies that the total oracle complexity is $\mathcal{O}\big((n+\kappa)\log\frac{F(\tilde{x}_0)-F(x^\star)}{\epsilon}\big)$.

Appendix B

Missing Proofs in Chapter 3

B.1 Technical Results

Lemma B.1.1. *Given positive parameters* $\{a_k\}_{k=0}^T$, $\{t_k\}_{k=0}^T$, n, C, *if*

$$\sum_{k=1}^{T} \frac{a_k}{t_k^n} \leq C,$$

then

$$\sum_{k=1}^{T} t_k \geq \frac{1}{C^{\frac{1}{n}}} \left(\sum_{k=1}^{T} a_k^{\frac{1}{n+1}} \right)^{\frac{n+1}{n}}.$$

Proof. Using Hölder's inequality, we have

$$\left(\sum_{k=1}^{T} \left(\frac{a_k^{\frac{1}{n+1}}}{t_k^{\frac{n}{n+1}}} \right)^{n+1} \right)^{\frac{1}{n+1}} \left(\sum_{k=1}^{T} \left(t_k^{\frac{n}{n+1}} \right)^{\frac{n+1}{n}} \right)^{\frac{n}{n+1}} \geq \sum_{k=1}^{T} a_k^{\frac{1}{n+1}},$$

$$C^{\frac{1}{n+1}} \left(\sum_{k=1}^{T} t_k \right)^{\frac{n}{n+1}} \geq \sum_{k=1}^{T} a_k^{\frac{1}{n+1}}.$$

Re-arranging the above inequality completes the proof. □

B.2 Proof of Lemma 3.3.7

The proof is based on the techniques in Lemma 4.2, [28] with some minor modifications that make the proof more natural in our parameter settings.

Recall the following two critical relations:

(I) $\frac{\gamma_k}{\tau_k^2} = \frac{\gamma_{k+1}}{\tau_{k+1}^2} - \frac{\gamma_{k+1}}{\tau_{k+1}}$ (Parameter scheme (3.10)), for $k = 0, \ldots, T-1$,

(II) $\frac{1-\sigma_r^2}{2} \sum_{k=0}^{T-1} \frac{1}{\tau_k^2} \|y_{k+1} - x_{k+1}\|^2 \leq D_0$ (Proposition 3.3.6).

From (I), we can conclude that

$$\frac{\gamma_{T-1}}{\tau_{T-1}^2} = \frac{\gamma_0}{\tau_0^2} + \sum_{k=1}^{T-1} \frac{\gamma_k}{\tau_k}. \tag{B.1}$$

Based on the result of the line search procedure (Proposition 3.3.4), we have

$$\|y_{k+1} - x_{k+1}\|^2 \geq \frac{4\sigma_l^2}{\gamma_k^2 L_2^2}.$$

Together with (II), we can conclude that

$$D_0 \geq \frac{1-\sigma_r^2}{2} \sum_{k=0}^{T-1} \frac{1}{\tau_k^2} \|y_{k+1} - x_{k+1}\|^2 \geq \frac{2(1-\sigma_r^2)\sigma_l^2}{L_2^2} \sum_{k=0}^{T-1} \frac{1}{\tau_k^2 \gamma_k^2}. \tag{B.2}$$

Then, we need to correlate the above inequality with (B.1). Note that

$$\frac{1}{\tau_k^2 \gamma_k^2} = \left(\frac{\gamma_k}{\tau_k^2}\right)^4 \cdot \left(\frac{\gamma_k}{\tau_k}\right)^{-6}.$$

By denoting $\mathcal{C} \triangleq \frac{D_0 L_2^2}{2(1-\sigma_r^2)\sigma_l^2}$, we can write (B.2) as

$$\sum_{k=0}^{T-1} \left(\frac{\gamma_k}{\tau_k^2}\right)^4 \cdot \left(\frac{\gamma_k}{\tau_k}\right)^{-6} \leq \mathcal{C}.$$

Now it is clear that we can apply Lemma B.1.1 with $a_{k+1} = \left(\frac{\gamma_k}{\tau_k^2}\right)^4, t_{k+1} = \frac{\gamma_k}{\tau_k}, n = 6, C = \mathcal{C}$, for $k = 0, \ldots, T-1$, which results in

$$\sum_{k=0}^{T-1} \frac{\gamma_k}{\tau_k} \geq \frac{1}{\mathcal{C}^{\frac{1}{6}}} \left(\sum_{k=0}^{T-1} \left(\frac{\gamma_k}{\tau_k^2}\right)^{\frac{4}{7}} \right)^{\frac{7}{6}}. \tag{B.3}$$

In comparison with (B.1), we see that it is crucial to ensure $\frac{\gamma_0}{\tau_0^2} = \frac{\gamma_0}{\tau_0} \Rightarrow \tau_0 = 1$.

Using the same technique as in [28], we construct two infinite non-negative sequences $\{b_i\}_{i=0}^{\infty}, \{\zeta_i\}_{i=0}^{\infty}$, which are required to satisfy

$$\frac{\gamma_k}{\tau_k^2} \geq b_i (k+1)^{\zeta_i}, \text{ for } k = 0, \ldots, T-1, \text{ and } i = 0, 1, \ldots. \tag{B.4}$$

First, to ensure (B.4) at $i = 0$, we can choose $b_0 = \gamma_0, \zeta_0 = 0$ and thus (B.4) holds due to (I) and $\tau_0 = 1$.

Then, suppose (B.4) holds at i, by (B.3) and $\tau_0 = 1$, we have

$$\begin{aligned}
\frac{\gamma_{T-1}}{\tau_{T-1}^2} = \sum_{k=0}^{T-1} \frac{\gamma_k}{\tau_k} &\geq \frac{1}{\mathcal{C}^{\frac{1}{6}}} \left(\sum_{k=0}^{T-1} \left(\frac{\gamma_k}{\tau_k^2} \right)^{\frac{4}{7}} \right)^{\frac{7}{6}} \\
&\geq \frac{b_i^{\frac{2}{3}}}{\mathcal{C}^{\frac{1}{6}}} \left(\sum_{k=1}^{T} \left(k^{4\zeta_i/7} \right) \right)^{\frac{7}{6}} \\
&\overset{(\star)}{\geq} \frac{b_i^{\frac{2}{3}}}{\mathcal{C}^{\frac{1}{6}}} \left(\frac{T^{4\zeta_i/7+1}}{4\zeta_i/7+1} \right)^{\frac{7}{6}} \\
&= \frac{b_i^{\frac{2}{3}}}{\mathcal{C}^{\frac{1}{6}} (4\zeta_i/7+1)^{\frac{7}{6}}} T^{2\zeta_i/3+7/6},
\end{aligned} \tag{B.5}$$

where $(\star)$ holds because $0 \leq x \mapsto x^{4\zeta_i/7}$ is non-decreasing and thus $\sum_{k=1}^{T} k^{4\zeta_i/7} \geq \int_0^T t^{4\zeta_i/7} dt$.

Thus, by requiring that $\zeta_{i+1} = \frac{2\zeta_i}{3} + \frac{7}{6}$, we can set $\{\zeta_i\}_{i=0}^{\infty}$ as

$$\zeta_i = \frac{7}{2} \left(1 - \left(\frac{2}{3} \right)^i \right), \text{ for } i = 0, 1, \ldots.$$

Note that based on this choice, $\zeta_i \leq \frac{7}{2}$ and thus

$$\frac{b_i^{\frac{2}{3}}}{\mathcal{C}^{\frac{1}{6}} (4\zeta_i/7+1)^{\frac{7}{6}}} \geq \frac{b_i^{\frac{2}{3}}}{\mathcal{C}^{\frac{1}{6}} 3^{\frac{7}{6}}}.$$

Then it is clear that by setting $b_{i+1} = \frac{b_i^{\frac{2}{3}}}{\mathcal{C}^{\frac{1}{6}} 3^{\frac{7}{6}}}$, the sequence $\{b_i\}_{i=0}^{\infty}$ can be chosen as

$$b_i = \frac{1}{\mathcal{C}^{1/2} 3^{7/2}} \left(\gamma_0 \cdot C^{1/2} 3^{7/2} \right)^{(2/3)^i}, \text{ for } i = 0, 1, \ldots.$$

Based on these choices of b_{i+1} and ζ_{i+1}, (B.5) can be written as $\frac{\gamma_{T-1}}{\tau_{T-1}^2} \geq b_{i+1} T^{\zeta_{i+1}}$. It remains to use the iterative nature of (II) (Proposition 3.3.6) to show that for $T' = 1, \dots T-2$, $\frac{\gamma_{T'}}{\tau_{T'}^2} \geq b_{i+1}(T'+1)^{\zeta_{i+1}}$ using exactly the above arguments. Then, by induction, $\{b_i\}_{i=0}^{\infty}, \{\zeta_i\}_{i=0}^{\infty}$ satisfy the target relation (B.4).

Finally, letting i goes to ∞, we obtain

$$\frac{\gamma_{T-1}}{\tau_{T-1}^2} \geq \sqrt{\frac{2(1-\sigma_r^2)}{3^7 D_0}} \frac{\sigma_l}{L_2} \cdot T^{\frac{7}{2}}.$$

B.3 Proof of Lemma 3.4.7

Similar to the proof of Lemma 3.3.7, we first state the following relations:

(I) $\frac{\gamma_k}{\tau_k^2} = \frac{\gamma_{k+1}}{\tau_{k+1}^2} - \frac{\gamma_{k+1}}{\tau_{k+1}}$ (Parameter scheme (3.11)), for $k = 0, \dots, T-1$,

(II) $\frac{m+2}{4(m+1)} \sum_{k=0}^{T-1} \frac{1}{\tau_k^2} \|y_{k+1} - x_{k+1}\|^2 \leq D_0$ (Proposition 3.4.6).

For (I), by choosing $\tau_0 = 1$, we have

$$\frac{\gamma_{T-1}}{\tau_{T-1}^2} = \sum_{k=0}^{T-1} \frac{\gamma_k}{\tau_k}. \tag{B.6}$$

For (II), using Proposition 3.4.4, which implies

$$\|y_{k+1} - x_{k+1}\|^2 \geq \left(\frac{(m-1)!}{2L_m \gamma_k} \right)^{\frac{2}{m-1}},$$

we can conclude that

$$\sum_{k=0}^{T-1} \frac{1}{\tau_k^2 \gamma_k^{\frac{2}{m-1}}} \leq \mathcal{C}_2, \tag{B.7}$$

where $\mathcal{C}_2 \triangleq \frac{4D_0(m+1)}{m+2} \left(\frac{2L_m}{(m-1)!} \right)^{\frac{2}{m-1}}$.

Reformulate (B.7),

$$\sum_{k=0}^{T-1} \left(\frac{\gamma_k}{\tau_k^2} \right)^{\frac{2m}{m-1}} \cdot \left(\frac{\gamma_k}{\tau_k} \right)^{-\frac{2m+2}{m-1}} \leq \mathcal{C}_2.$$

Applying Lemma B.1.1 with $a_{k+1} = \left(\frac{\gamma_k}{\tau_k^2}\right)^{\frac{2m}{m-1}}, t_{k+1} = \frac{\gamma_k}{\tau_k}, n = \frac{2m+2}{m-1}, C = \mathcal{C}_2$, we obtain

$$\sum_{k=0}^{T-1} \frac{\gamma_k}{\tau_k} \geq \frac{1}{\mathcal{C}_2^{\frac{m-1}{2m+2}}} \left(\sum_{k=0}^{T-1} \left(\frac{\gamma_k}{\tau_k^2}\right)^{\frac{2m}{3m+1}}\right)^{\frac{3m+1}{2m+2}}. \tag{B.8}$$

Here we construct two non-negative sequence $\{\zeta_i\}_{i=0}^{\infty}, \{b_i\}_{i=0}^{\infty}$ and the target relation is identical to relation (B.4).

$$\frac{\gamma_k}{\tau_k^2} \geq b_i (k+1)^{\zeta_i}, \text{ for } k = 0, \ldots, T-1, \text{ and } i = 0, 1, \ldots. \tag{B.9}$$

First, at $i = 0$, we choose $b_0 = \gamma_0, \zeta_0 = 0$ and then (B.9) holds due to (I).

Then, suppose (B.9) holds at i, by (B.8), (B.6), we have

$$\begin{aligned}
\frac{\gamma_{T-1}}{\tau_{T-1}^2} &\geq \frac{1}{\mathcal{C}_2^{\frac{m-1}{2m+2}}} \left(\sum_{k=0}^{T-1} \left(\frac{\gamma_k}{\tau_k^2}\right)^{\frac{2m}{3m+1}}\right)^{\frac{3m+1}{2m+2}} \\
&\geq \frac{b_i^{\frac{m}{m+1}}}{\mathcal{C}_2^{\frac{m-1}{2m+2}}} \left(\sum_{k=1}^{T} k^{\frac{2m\zeta_i}{3m+1}}\right)^{\frac{3m+1}{2m+2}} \\
&\overset{(\star)}{\geq} \frac{b_i^{\frac{m}{m+1}}}{\mathcal{C}_2^{\frac{m-1}{2m+2}}} \left(\frac{3m+1}{(2\zeta_i+3)m+1}\right)^{\frac{3m+1}{2m+2}} T^{\frac{(2\zeta_i+3)m+1}{2m+2}}
\end{aligned} \tag{B.10}$$

where $(\star)$ holds because $0 \leq x \mapsto x^{\frac{2m\zeta_i}{3m+1}}$ is non-decreasing, and thus

$$\sum_{k=1}^{T} k^{\frac{2m\zeta_i}{3m+1}} \geq \int_0^T t^{\frac{2m\zeta_i}{3m+1}} dt = \frac{3m+1}{(2\zeta_i+3)m+1} T^{\frac{2m\zeta_i}{3m+1}+1}.$$

Suppose we want to ensure $\zeta_{i+1} = \frac{(2\zeta_i+3)m+1}{2m+2}$, the sequence $\{\zeta_i\}_{i=0}^{\infty}$ can be defined as

$$\zeta_i = \frac{3m+1}{2}\left(1 - \left(\frac{m}{m+1}\right)^i\right), \text{ for } i = 0, 1, \ldots.$$

which implies that $\zeta_i \leq \frac{3m+1}{2}$.

For sequence $\{b_i\}_{i=0}^{\infty}$, observe that

$$\frac{b_i^{\frac{m}{m+1}}}{\mathcal{C}_2^{\frac{m-1}{2m+2}}} \left(\frac{3m+1}{(2\zeta_i+3)m+1}\right)^{\frac{3m+1}{2m+2}} \geq \frac{b_i^{\frac{m}{m+1}}}{\mathcal{C}_2^{\frac{m-1}{2m+2}}} \left(\frac{1}{m+1}\right)^{\frac{3m+1}{2m+2}},$$

we can set

$$b_{i+1} = \frac{b_i^{\frac{m}{m+1}}}{\mathcal{C}_2^{\frac{m-1}{2m+2}}} \left(\frac{1}{m+1}\right)^{\frac{3m+1}{2m+2}}$$

$$\implies b_i = \frac{1}{\mathcal{C}_2^{\frac{m-1}{2}}} \left(\frac{1}{m+1}\right)^{\frac{3m+1}{2}} \left(\frac{\gamma_0}{\mathcal{C}_2^{\frac{m-1}{2}}} \left(\frac{1}{m+1}\right)^{\frac{3m+1}{2}}\right)^{\left(\frac{m}{m+1}\right)^i}, \text{ for } i = 0, 1, \ldots.$$

Based on these choices, we have $\frac{\gamma_{T-1}}{\tau_{T-1}^2} \geq b_{i+1} T^{\zeta_{i+1}}$. It remains to use the iterative nature of (II) (Proposition 3.4.6) to show that for $T' = 1, \ldots T-2$, $\frac{\gamma_{T'}}{\tau_{T'}^2} \geq b_{i+1}(T'+1)^{\zeta_{i+1}}$ using exactly the above arguments. Then, by induction, $\{b_i\}_{i=0}^{\infty}, \{\zeta_i\}_{i=0}^{\infty}$ satisfy the target relation (B.9).

Finally, letting i goes to ∞, we obtain

$$\frac{\gamma_{T-1}}{\tau_{T-1}^2} \geq \left(\frac{m+2}{4D_0(m+1)}\right)^{\frac{m-1}{2}} \left(\frac{1}{m+1}\right)^{\frac{3m+1}{2}} \frac{(m-1)!}{2L_m} \cdot T^{\frac{3m+1}{2}}.$$

Bibliography

[1] N. Agarwal and E. Hazan. Lower bounds for higher-order convex optimization. In *COLT*, pages 774–792, 2018.

[2] Z. Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. In *STOC*, pages 1200–1205, 2017.

[3] Z. Allen-Zhu and E. Hazan. Variance reduction for faster non-convex optimization. In *ICML*, pages 699–707, 2016.

[4] Z. Allen-Zhu and L. Orecchia. Linear coupling: An ultimate unification of gradient and mirror descent. In *ITCS*, 2017.

[5] Y. Arjevani. Limitations on variance-reduction and acceleration schemes for finite sums optimization. In *NIPS*, pages 3540–3549, 2017.

[6] Y. Arjevani, O. Shamir, and R. Shiff. Oracle complexity of second-order methods for smooth convex optimization. *Mathematical Programming*, pages 1–34, 2017.

[7] A. Auslender and M. Teboulle. Interior gradient and proximal methods for convex and conic optimization. *SIAM Journal on Optimization*, 16(3):697–725, 2006.

[8] M. Baes. Estimate sequence methods: extensions and approximations. *Institute for Operations Research, ETH, Zürich, Switzerland*, 2009.

[9] S. Bubeck, Q. Jiang, Y. T. Lee, Y. Li, and A. Sidford. Near-optimal method for highly smooth convex optimization. *arXiv preprint arXiv:1812.08026*, 2018.

[10] A. Defazio. A simple practical accelerated method for finite sums. In *NIPS*, pages 676–684, 2016.

[11] A. Defazio, F. Bach, and S. Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *NIPS*, pages 1646–1654, 2014.

[12] J. Diakonikolas and L. Orecchia. Accelerated extra-gradient descent: A novel accelerated first-order method. *arXiv preprint arXiv:1706.04680*, 2017.

[13] R. Frostig, R. Ge, S. Kakade, and A. Sidford. Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In *ICML*, pages 2540–2548, 2015.

[14] R. M. Gower, P. Richtárik, and F. Bach. Stochastic quasi-gradient methods: Variance reduction via jacobian sketching. *arXiv:1805.02632*, 2018.

[15] B. Jiang, H. Wang, and S. Zhang. An optimal high-order tensor method for convex optimization. *arXiv preprint arXiv:1812.06557*, 2018.

[16] R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *NIPS*, pages 315–323, 2013.

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

[18] J. Konečnỳ and P. Richtárik. Semi-stochastic gradient descent methods. *arXiv preprint arXiv:1312.1666*, 2013.

[19] W. Krichene, A. Bayen, and P. L. Bartlett. Accelerated mirror descent in continuous and discrete time. In *NIPS*, pages 2845–2853, 2015.

[20] G. Lan and Y. Zhou. An optimal randomized incremental gradient method. *Mathematical programming*, pages 1–49, 2017.

[21] R. Leblond, F. Pedregosa, and S. Lacoste-Julien. ASAGA: Asynchronous parallel SAGA. In *AISTATS*, pages 46–54, 2017.

[22] L. Lei and M. Jordan. Less than a single pass: Stochastically controlled stochastic gradient. In *AISTATS*, pages 148–156, 2017.

[23] H. Lin, J. Mairal, and Z. Harchaoui. A universal catalyst for first-order optimization. In *NIPS*, pages 3366–3374, 2015.

[24] Q. Lin, Z. Lu, and L. Xiao. An accelerated proximal coordinate gradient method. In *NIPS*, pages 3059–3067, 2014.

[25] X. Liu and C.-J. Hsieh. Fast variance reduction method with stochastic batch size. In *ICML*, pages 3179–3188, 2018.

[26] H. Mania, X. Pan, D. Papailiopoulos, B. Recht, K. Ramchandran, and M. I. Jordan. Perturbed iterate analysis for asynchronous stochastic optimization. *SIAM Journal on Optimization*, 27(4):2202–2229, 2017.

[27] R. D. Monteiro and B. F. Svaiter. On the complexity of the hybrid proximal extragradient method for the iterates and the ergodic mean. *SIAM Journal on Optimization*, 20(6):2755–2787, 2010.

[28] R. D. Monteiro and B. F. Svaiter. An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods. *SIAM Journal on Optimization*, 23(2):1092–1125, 2013.

[29] A. S. Nemirovsky and D. B. Yudin. Problem complexity and method efficiency in optimization. 1983.

[30] Y. Nesterov. Accelerating the cubic regularization of newton's method on convex problems. *Mathematical Programming*, 112(1):159–181, 2008.

[31] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[32] Y. Nesterov. Implementable tensor methods in unconstrained convex optimization. Technical report, Université catholique de Louvain, Center for Operations Research and Econometrics (CORE), 2018.

[33] Y. E. Nesterov. A method for solving the convex programming problem with convergence rate $o(1/k^2)$. In *Dokl. Akad. Nauk SSSR*, volume 269, pages 543–547, 1983.

[34] L. M. Nguyen, J. Liu, K. Scheinberg, and M. Takáč. SARAH: A novel method for machine learning problems using stochastic recursive gradient. In *ICML*, pages 2613–2621, 2017.

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

[36] N. Parikh, S. Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.

[37] F. Pedregosa, R. Leblond, and S. Lacoste-Julien. Breaking the nonsmooth barrier: A scalable parallel method for composite optimization. In *NIPS*, pages 56–65, 2017.

[38] B. Recht, C. Re, S. Wright, and F. Niu. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *NIPS*, pages 693–701, 2011.

[39] S. J. Reddi, A. Hefny, S. Sra, B. Poczos, and A. J. Smola. On variance reduction in stochastic gradient descent and its asynchronous variants. In *NIPS*, pages 2629–2637, 2015.

[40] H. Robbins and S. Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[41] N. L. Roux, M. Schmidt, and F. R. Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. In *NIPS*, pages 2663–2671, 2012.

[42] R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo method*, volume 10. John Wiley & Sons, 2016.

[43] M. Schmidt, N. Le Roux, and F. Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112, 2017.

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

[45] S. Shalev-Shwartz and T. Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. In *ICML*, pages 64–72, 2014.

[46] A. Vladimirov, Y. E. Nesterov, and Y. N. Chekanov. On uniformly convex functionals. *Vestnik Moskov. Univ. Ser. XV Vychisl. Mat. Kibernet*, 3:12–23, 1978.

[47] L. Xiao and T. Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

[48] Y. Xu, Q. Lin, and T. Yang. Adaptive svrg methods under error bound conditions with unknown growth parameter. In *NIPS*, pages 3277–3287, 2017.

[49] Y. Zhang and L. Xiao. Stochastic primal-dual coordinate method for regularized empirical risk minimization. In *ICML*, pages 353–361, 2015.

[50] K. Zhou, Q. Ding, F. Shang, J. Cheng, D. Li, and Z.-Q. Luo. Direct Acceleration of SAGA using Sampled Negative Momentum. In *AISTATS*, pages 1602–1610, 2019.

[51] K. Zhou, F. Shang, and J. Cheng. A simple stochastic variance reduced algorithm with fast convergence rates. In *ICML*, pages 5980–5989, 2018.