Can Gaussian Sketching Converge Faster on a Preconditioned Landscape?

Yilong Wang [1 2] **Haishan Ye** [1 3] **Guang Dai** [3] **Ivor W. Tsang** [4 5]

Abstract

This paper focuses on the large-scale optimization which is very popular in the big data era. The gradient sketching is an important technique in the large-scale optimization. Specifically, the random coordinate descent algorithm is a kind of gradient sketching method with the random sampling matrix as the sketching matrix. In this paper, we propose a novel gradient sketching called GSGD (Gaussian Sketched Gradient Descent). Compared with the classical gradient sketching methods such as the random coordinate descent and SEGA (Hanzely et al., 2018), our GSGD does not require the importance sampling but can achieve a fast convergence rate matching the ones of these methods with importance sampling. Furthermore, if the objective function has a non-smooth regularization term, our GSGD can also exploit the implicit structure information of the regularization term to achieve a fast convergence rate. Finally, our experimental results substantiate the effectiveness and efficiency of our algorithm.

1. Introduction

Optimization is an important pillar of modern machine learning because it needs optimization algorithms to train machine learning models. In this paper, we consider the following composite optimization problem:

$$\min_{x\in\mathbb{R}^d} F(x) \overset{def}{=} f(x) + \varphi(x), \tag{1}$$

where function $f : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex and differentiable, while the regularization term φ is convex but may be non-differentiable. There is a large class of machine learning models that can be presented as Eq. (1) such as the elastic net (Zou & Hastie, 2005) and sparse logistic regression (Xiao & Zhang, 2014).

Optimization problems related machine learning models often are of high complexity and dimension because of the rapid development of data science and the accelerated accumulation of industry data. Thus, the large-scale optimization has become an important research topic in machine learning and has attracted great research interest. Because of the high dimension, that is, d is very large, the gradient sketching method has become an important technique to conquer the dilemma of large-scale optimization. Specifically, the classical random coordinate gradient descent is a kind of gradient sketching method.

In order to facilitate the following description, we first explain the following definition. The meaning of the separability of φ is that the regularization term can be written as a finite sum of all coordinates or block coordinates. In most cases, the inseparable regularization term is a generalized indicator function.

When the regularization term φ is separable, the classical random coordinate gradient descent methods can effectively and efficiently solve the problem (1) (Nesterov, 2012; Wright, 2015). These methods can achieve linear convergence rates. Because of fast convergence rates and low computation costs for each iteration, random coordinate gradient descent methods have been used in practical applications for many years, and their popularity continues to grow because they are useful in data analysis and machine learning.

However, if the regularization term φ is not separable, the corresponding random coordinate gradient will incur an inherent non-zero variance at the optimum. Thus, in this case, the linear convergence rate of random coordinate descent is not achievable (Richtárik & Takáč, 2014). To conquer this dilemma, Hanzely et al. (2018) propose SEGA algorithm and SVRCD algorithm (Hanzely & Richtárik, 2019b) which base on the variance reduction. These two algorithms modify the random coordinate descent algorithms to reduce the variance incurred by the random coordinate gradient. Accordingly, both SEGA and SVRCD can achieve linear convergence rates. If the regularization term $\varphi = 0$, then

[1]Center for Intelligent Decision-Making and Machine Learning, School of Management, Xi'an Jiaotong University, China. [2]This work was completed during the internship at SGIT AI Lab, State Grid Corporation of China. [3]SGIT AI Lab, State Grid Corporation of China. [4]CFAR and IHPC, Agency for Science, Technology and Research (A*STAR), Singapore [5]College of Computing and Data Science, NTU, Singapore.. Correspondence to: Haishan Ye <yehaishan@xjtu.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

SEGA requires the importance sampling to achieve a rate $1-\mathcal{O}\left(\frac{\mathrm{tr}(\mathbf{M})}{\mu}\right)$ if $f(x)$ is $\mathbf{M}$-smooth and μ-strongly convex where $\mathbf{M}$ is a positive definite matrix. However, the importance sampling requires to access the diagonal entries of $\mathbf{M}$ which is impossible or very computationally expensive. For example, if one can only access to the sketched gradient, then the diagonal entries of $\mathbf{M}$ can not be obtained.

Therefore, an interesting question arises: can we design an algorithm that can achieve the same convergence rate to SEGA both for non-separable φ and for $\varphi = 0$ with importance sampling? In this paper, we endeavor to address this problem.

1.1. Literature Review

In this part, we briefly review the algorithms for solving large-scale high-dimensional optimization problems.

Some stochastic coordinate descent algorithms and stochastic subspace descent algorithms (Nesterov, 2012; Gower & Richtárik, 2015b; Wright, 2015) have been proposed and received extensive attention. At the same time, some stochastic coordinate descent methods for parallel computing (Liu et al., 2014; Fercoq & Richtárik, 2015; Liu & Wright, 2015) have been proposed, but they can only solve the problem with separable regularization term. Among them, the stochastic coordinate descent algorithm has been developed many variants. Let us first give the definitions of v_i and p_i: these parameters come from the ESO method (Qu & Richtárik, 2016), in case of single coordinate sketches, parameters v_i are equal to coordinate-wise smoothness constants of f and p_i represents the probability of the i-th coordinate being sampled. Richtárik & Takáč (2016) propose a non-accelerated with arbitrary sampling, and proof that the iteration complexity of the algorithm is $(\max_i \frac{v_i}{p_i\mu})\log\frac{1}{\varepsilon}$. Nesterov (2012) proposes a non-accelerated with importance sampling, and proofs that the iteration complexity of the algorithm is $\frac{\mathrm{tr}(\mathbf{M})}{\mu}\log\frac{1}{\varepsilon}$. Hanzely & Richtárik (2019a) propose an accelerated with arbitrary sampling, and proof that the iteration complexity of the algorithm is $(\max_i \frac{v_i}{p_i^2\mu})\log\frac{1}{\varepsilon}$. Allen-Zhu et al. (2016) propose an accelerated with importance sampling, and proof that the iteration complexity of the algorithm is $\frac{\sum_i\sqrt{\mathbf{M}_{ii}}}{\sqrt{\mu}}\log\frac{1}{\varepsilon}$. To our best knowledge, the methods with importance sampling are state-of-the-art variants of stochastic coordinate descent algorithms. But Hanzely et al. (2018) once again indicate above algorithms does not work with inseparable φ and create a new algorithm to overcome this dilemma. Since then, the limitations of the traditional stochastic coordinate descent algorithms have gradually been concerned.

In recent years, scholars pay more attention to using next two techniques to create new algorithms in this field. A technique called variance reduction which is used by SVRG (Johnson & Zhang, 2013) and SAGA (Defazio et al., 2014) to make SGD (Robbins & Monro, 1951; Nemirovski et al., 2009) converge linearly and is summarized in Gower et al. (2020) finally. The other technique called sketch-and-project (Gower & Richtárik, 2015a; Gower, 2016) which is first used to reduce the dimension and solve complex linear systems begins to be applied in this field. Hanzely et al. (2018) use above techniques to create the SEGA algorithm and proof that the iteration complexity of the algorithm is $(\max_i \frac{4v_i+\mu}{p_i\mu})\log\frac{1}{\varepsilon}$, which well solves the limitations of the traditional stochastic coordinate descent algorithms in dealing with the problem with an inseparable regularization term. Its convergence rate is the same as traditional stochastic coordinate descent algorithm. However, in the special case of setting the regularization term $\varphi = 0$, SEGA algorithm requires importance sampling to achieve the best convergence rate of state-of-the-art variant. In the next year, Kozak et al. (2019) propose VRSSD with controlling variates, but this method only work when $\varphi = 0$. Then, Hanzely & Richtárik (2019b) propose SVRCD algorithm which is a variant of SEGA and prove that the iteration complexity of the algorithm is $(\max_i \frac{4v_i}{p_i\mu} + \frac{1}{\rho})\log\frac{1}{\varepsilon}$. SVRCD updates h_k to the current true gradient with a fixed probability ρ in order to obtain the scheme with smaller constant factor of convergence rate, rather than updating part of the coordinates like SEGA. But SVRCD has no obvious advantages over SEGA, both theoretically and practically. SVRCD even performs worse than SEGA in most experiments. The main purpose of studying SVRCD is to facilitate the introduction of momentum terms to obtain an accelerated algorithm. Hanzely et al. (2020) first combine Nesterov's momentum with SVRCD algorithm to obtain an accelerated version of SVRCD called ASVRCD and accelerated the SVRCD and ASVRCD by introducing a projection matrix further. Chorobura & Necoara (2023) uses a new perspective to obtain a linearly convergent coordinate proximal gradient by introducing KL property. Recently, the application range of SEGA algorithm is extended by proposing a stochastic coordinate algorithm called SEGA-SGDA to deal with variational inequality problems (Beznosikov et al., 2023).

Our main works include designing an algorithm that converges faster under the premise of setting the regularization term $\varphi = 0$ without introducing importance sampling method, and can also achieve the same convergence rate to SEGA and even faster convergence rate by introducing the projection matrix when the regularization term is inseparable. It should be pointed out that importance sampling has certain limitations in practice. The importance sampling needs to calculate every L_i-smooth constant about coordinates under coordinate-wise Lipschitz continuous assumption. If we study the optimization problem in d-dimensional space, the cost of the above calculation is at least $\mathcal{O}(d)$. Even in some zero-order cases, we cannot

calculate these constants at all. It is worth noting that (Hanzely et al., 2018) only focus on the theoretical analysis of the Coordinate sketching version of the SEGA, but we are more concerned about Gaussian sketching. Therefore, our works can be regarded as a further extension of (Hanzely et al., 2018). It is worth pointing out that some papers also consider Gaussian sketching under federated learning. We need to explain the essential differences between our work and these works from two aspects. From the perspective of results, (Rothchild et al., 2020) do not propose an algorithm with linear convergence rate. Although (Song et al., 2023) only propose a linear convergence algorithm without the regularization term, the complexity of their algorithm is $\mathcal{O}((LN/\mu)\max\{d, \sqrt{\sigma^2/(\mu\varepsilon)}\}\log(L\mathbb{E}[\|\omega^0-\omega^*\|_2^2]/\varepsilon))$ indicates that their algorithm does not converge as fast as our algorithm GSGD whose complexity is described detailedly in Corollary 4.4. Regardless of whether there is a regularization term, our algorithm GSGD gives linear convergence rate conclusions. So, our paper has a more comprehensive analysis. Especially for GSGD without regularization term, the complexity is $\mathcal{O}((\mathrm{tr}(\mathbf{M})/\mu)\log(1/\varepsilon))$, this shows that our algorithm is better than the algorithms involved in the above two papers. From the perspective of applications, these works are dedicated to reducing the communication cost in federated learning by using gradient methods. Our algorithm GSGD can not only be used in situations where gradients need to be calculated, but can also be used in situations where gradients cannot be obtained. As we mentioned in Remark 3.1, the use of Gaussian vectors makes our algorithm can be naturally applied to the field of zero-order optimization when the gradient is difficult or impossible to compute. All in all, our paper is compared with the two papers mentioned above in the field of federated learning has a more comprehensive analysis and our algorithm GSGD has wider application prospects.

The main contributions of this paper are summarized as follows:

- We propose a novel gradient sketching method called GSGD. When $\varphi = 0$, compared with the SEGA algorithm with the importance sampling, our algorithm can achieves the same convergence rate but without the importance sampling. SEGA uses importance sampling technology will increase the computational overhead and may even be inapplicable in the field of many zero-order cases, but our algorithm GSGD can be naturally expanded to the field of zero-order optimization. Thus, GSGD has a wider application range, especially suitable for the cases that only sketched gradient can be accessed.
- For the problems that regularization term φ is non-separable, we prove that it can achieve a linear convergence rate which is also comparable to the one of SEGA.
- If the regularization term φ has some special structure, our GSGD can inherently exploit the special structure in the regularization term to improve the convergence rate.
- Extensive experiments confirm our algorithm's superiority in terms of computational efficiency when compared to existing state-of-the art algorithms.

2. Notation and Assumptions

Throughout this paper, we denote the proximal operator of regularization term φ

$$\mathrm{Prox}_{\alpha\varphi}(x) \overset{def}{=} \arg\min_{y\in\mathbb{R}^d}\{\varphi(y) + \frac{1}{2\alpha}\|y-x\|^2\}.$$

Let us define the weighted Euclidean norm and weighted inner product associated with a positive weight matrix $\mathbf{M} \succ 0$

$$\|x\|_{\mathbf{M}} \overset{def}{=} \langle x, x\rangle_{\mathbf{M}}^{\frac{1}{2}},$$
$$\langle x, y\rangle_{\mathbf{M}} \overset{def}{=} \langle \mathbf{M}x, y\rangle.$$

In addition, we make the following assumptions for objective function f and regularization term φ in Eq. (1). Firstly, we will use the following general version of smoothness and standard version of strong convexity.

Assumption 2.1. A differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is $\mathbf{M}$-smooth for some positive definite matrix $\mathbf{M} \succ 0$. That is, for all $x, y \in \mathbb{R}^d$, the following inequality is satisfied

$$f(x) \le f(y) + \langle \nabla f(y), x-y\rangle + \frac{1}{2}\|x-y\|_{\mathbf{M}}^2. \quad (2)$$

If f is L-smooth, then for all $x, y \in \mathbb{R}^d$, the following inequality is satisfied

$$f(x) \le f(y) + \langle \nabla f(y), x-y\rangle + \frac{L}{2}\|x-y\|^2. \quad (3)$$

Assumption 2.2. A differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex. That is, for all $x, y \in \mathbb{R}^d$, the following inequality is satisfied

$$f(x) \ge f(y) + \langle \nabla f(y), x-y\rangle + \frac{\mu}{2}\|x-y\|^2. \quad (4)$$

For the regularization term φ, we let it include an indicator function of some affine subspace of $\mathbb{R}^d$ associated with the started point and the range of projection matrix $\mathbf{W}$, indicating the dimension of affine subspaces. The regularization term φ satisfies the following assumption.

Algorithm 1 GSGD:Gaussian Sketched Gradient Descent

Initialize: $x_0, h_0 \sim N(0, \mathbf{I}_d)$, stepsize $\eta > 0$
for $k = 0, 1, \cdots$ **do**
  Sample $u \sim N(0, \mathbf{I}_d)$
  $g_k = h_k + uu^\top(\nabla f(x_k) - h_k)$
  $h_{k+1} = h_k + \frac{uu^\top}{d+2}(\nabla f(x_k) - h_k)$
  $x_{k+1} = \mathrm{Prox}_{\alpha\varphi}(x_k - \eta g_k)$
end for

Assumption 2.3 ((Hanzely et al., 2020)). Assume that $\mathbf{W}$ is an orthogonal projection matrix, then we obtain the composite regularization term

$$\varphi(x) = \begin{cases} \psi(x) & \text{if } x \in \{x_0 + \mathrm{Range}(\mathbf{W})\} \\ \infty & \text{if } x \notin \{x_0 + \mathrm{Range}(\mathbf{W})\} \end{cases},$$

where $\psi(x)$ is a general convex function. Furthermore, it is necessary to assume that the proximal operator of φ is easy to calculate.

This assumption will be used to accelerate our algorithm under certain circumstances.

Proposition 2.4 ((Hanzely et al., 2020)). *If $f : \mathbb{R}^d \to \mathbb{R}$ is $\mathbf{M}$-smooth, for all $x, y \in \mathbb{R}^d$, the following inequality is satisfied:*

$$f(x) \geq f(y) + \langle \nabla f(y), x - y \rangle + \frac{1}{2} \|\nabla f(x_k) - \nabla f(x^*)\|^2_{\mathbf{M}^{-1}}. \tag{5}$$

This proposition will be utilized in the convergence analysis of our algorithm 1 with regularization term in Assumption 2.3.

3. The Gaussian Sketched Gradient Descent Algorithm

This section commences with a detailed description of the algorithm. Then, we introduce the concept of Gaussian sketch and the structure of the inseparable regularization term. Finally, we present the main theoretical results.

3.1. Algorithm Description

To more efficiently address the composite optimization problem as delineated in Eq. (1), we introduce a principal algorithm named Gaussian Sketched Gradient Descent whose main algorithmic procedure is listed as follows

$$g_k = h_k + uu^\top(\nabla f(x_k) - h_k), \tag{6}$$

$$h_{k+1} = h_k + \frac{uu^\top}{d+2}(\nabla f(x_k) - h_k), \tag{7}$$

$$x_{k+1} = \mathrm{Prox}_{\alpha\varphi}(x_k - \eta g_k), \tag{8}$$

where $u \sim N(0, \mathbf{I}_d)$. When the regularization term $\varphi = 0$, then the update rule of x_{k+1} is reduced to

$$x_{k+1} = x_k - \eta g_k. \tag{9}$$

The detailed description of GSGD is listed in Algorithm (1).

Classical gradient sketching methods commonly directly use the sketched gradient $uu^\top \nabla f(x_k)$ to update x_k (Gower et al., 2019; Gower & Richtárik, 2015b; Nesterov, 2012). That is, it conducts the following update:

$$x_{k+1} = x_k - \eta_t uu^\top \nabla f(x_k).$$

However, once $\varphi \neq 0$, then it holds that $\|\nabla f(x^*)\| \neq 0$. At the same time, it holds that $\mathbb{E}[\|uu^\top \nabla f(x)\|^2] = \mathcal{O}\left(\|\nabla f(x)\|^2\right)$ (Nesterov & Spokoiny, 2017; Hanzely et al., 2018). Thus, classical gradient sketching methods can not achieve a linear convergence rate for the composite problems. Thus, our method introduces an extra variance reduction variate h_k tracks the exact gradient. Eq. (14) shows that for each iteration, the variance related to h_k reduces with a rate $1 - \frac{1}{d+2}$. However, h_k is a biased estimation of the gradient. Thus, we introduce g_k which is an unbiased estimation of the gradient and full exploit the information of h_k. Therefore, GSGD can achieve a linear convergence rate when the objective function has a regularization term.

Compared with the Coordinate sketching version of SEGA, our GSGD replaces the coordinate sampling vector e_i to the Gaussian sampling vector u, where k is the iteration number of the algorithm, and d is the dimension of x. The Gaussian sampling vector u of our method is the key to achieve a faster convergence rate matching SEGA with the importance sampling.

3.2. Gaussian Sketching View

Let $u \sim N(0, \mathbf{I}_d)$ be a random vector and $x \in \mathbb{R}^d$ be an arbitrary point. Then, we can compress the real gradient and obtain the Gaussian sketch of real gradient

$$\delta(u, x) \overset{def}{=} u^T \nabla f(x), \quad u \sim N(0, \mathbf{I}_d). \tag{10}$$

If we replace the Gaussian distribution $N(0, \mathbf{I}_d)$ with the uniform distribution over standard basis vectors. Then $\delta(e_i, x)$ becomes the i-th partial derivative of function f at x, and it is actually used by the Coordinate sketching version of the SEGA algorithm.

GSGD is closely related to the sketch-and-project technique which is first used to solve the approximate solution of linear systems iteratively. Let x_k be the current iteration, h_k be the current middle estimate of the real gradient of f, and let u_k be the k-th Gaussian sampling vector during the iterative process. The sketch-and-project tries to find a vector h_{k+1}

that minimizes the following optimization problem:

$$h_{k+1} = \arg\min_{h\in\mathbb{R}^d} \|h - h_k\|^2, \quad s.t. \quad u_k^T h = \delta(u_k, x_k). \tag{11}$$

The idea of the form of objective function is to try to preserve as much of the information learned so far as possible, as condensed in the current middle gradient estimate h_k. Eq. (11) has the the following closed-form solution,

$$h_{k+1} = h_k + u_k(u_k^T u_k)^{-1} u_k^T(\nabla f(x_k) - h_k). \tag{12}$$

The above equation is the general form of SEGA for the Gaussian sketching. However, this is different from the update of h_{k+1} in Eq. (7). Note that $\frac{u_k}{\|u_k\|}$ follows the uniform spherical distribution. Thus, even u_k is of standard Gaussian distribution, SEGA only admits a gradient sketching with respect to the uniform spherical distribution. In contrast, our GSGD bases on the standard Gaussian distribution. Although the differences between Eq. (7) and Eq. (12) are indeed not significant from numerical and experimental perspectives, the update of h_{k+1} we proposed is more in line with the concept of Gaussian sketching than the update formula in SEGA.

The reason why we set the denominator to $d+2$ instead of some other constants like $d+\mathcal{O}(1)$ is to help us make convergence analysis easier. For example, in the proof of Lemma 4.2, if we do not specify the denominator as $d+2$, eventually we will not be able to eliminate terms $\langle h_k, \nabla f(x_k)\rangle$ that is detrimental to our proof of the convergence theorem. At the same time, it is also impossible to make the coefficients before $\|h_k\|^2$ and $\|\nabla f(x_k)\|^2$ all less than 1 and this does not imply that $\|h_{k+1}\|^2$ is decreasing.

Remark 3.1. It is worth mentioning that our algorithm has a zero-order version when calculating the gradient or storing the real full gradient is very difficult. In fact, $\delta(u, x)$ is the directional derivative of function f at x in direction u. We all know that the directional derivative can be approximated by the finite difference of the function value. In other words, we can replace $u^T\nabla f(x)$ with $\frac{1}{\alpha}(f(x+\alpha u) - f(x))$, where $\alpha > 0$ is sufficiently small. The specific theoretical guarantee of this operation can refer to (Nesterov & Spokoiny, 2017; Berahas et al., 2022) .

4. Main Theoretical Results

This section offers an in-depth examination of the iteration complexity of our algorithm under different assumptions.

We give several important lemmas that helps to derive the main theorems in this part. First, we give two key lemmas which describe the bounds of $\|g_k\|_{\mathbf{M}}^2$ and $\|h_{k+1}\|^2$. By these Lemmas, we can easily find that the norms of g_k and h_{k+1} decrease with the increase of iteration steps.

Lemma 4.1. *For all k>0, the variance of g_k (defined in Eq. (6)) as an estimator of $\nabla f(x_k)$ can be bounded as follows*

$$\mathbb{E}_u\left[\|g_k\|_{\mathbf{M}}^2\right] \le 4\mathrm{tr}(\mathbf{M})\|h_k\|^2 + 5\mathrm{tr}(\mathbf{M})\|\nabla f(x_k)\|^2. \tag{13}$$

Lemma 4.2. *For all k>0, the following equality between h_{k+1} and h_k holds*

$$\mathbb{E}_u\left[\|h_{k+1}\|^2\right] = \left(1 - \frac{1}{d+2}\right)\|h_k\|^2 + \frac{1}{d+2}\|\nabla f(x_k)\|^2. \tag{14}$$

Because Lyapunov analysis is well applied in article (Wilson et al., 2016) and fully summarized in article (Wilson, 2018; Sanz Serna & Zygalakis, 2021), we try to combine these techniques to demonstrate the convergence later. Using Lemma 4.1 and 4.2, we can obtain the following theorem.

Theorem 4.3. *Let the objective function $F(x)$ be of the form (1) with $\varphi = 0$. Assume that f is $\mathbf{M}$-smooth and μ-strongly convex, that is, Assumption 2.1 and 2.2 hold. Define the following Lyapunov function*

$$\Phi^k \overset{def}{=} f(x_k) - f(x^*) + \eta\alpha_1(d+2)\cdot\mathrm{tr}(\mathbf{M})\|h_k\|^2,$$

and choose

$$\eta = \frac{1}{20\mathrm{tr}(\mathbf{M})}, \quad \alpha_1 = \frac{1}{2\mathrm{tr}(\mathbf{M})}, \tag{15}$$

$$c_1 = \max\left\{1 - \frac{\mu}{40\mathrm{tr}(\mathbf{M})}, 1 - \frac{3}{5(d+2)}\right\}. \tag{16}$$

Then, the number of iterations of the algorithm 1 satisfy

$$\mathbb{E}_u\left[\Phi^k\right] \le c_1^k\Phi^0.$$

Next, by the convergence rate of our algorithm shown in Theorem 4.3, we will give computational cost of algorithm 1 in the following corollary.

Corollary 4.4. *Let the objective function satisfy the properties described in Theorem 4.3 and select the parameters in Theorem 4.3. To find an ε-suboptimal solution, the iteration complexity of Algorithm 1 is*

$$T = \mathcal{O}\left(\left(\frac{\mathrm{tr}(\mathbf{M})}{\mu} + d\right)\log\frac{1}{\varepsilon}\right). \tag{17}$$

Remark 4.5. It is easy to check that $\mathrm{tr}(\mathbf{M})/\mu$ is no less than d. Thus, Eq. (17) demonstrates that our algorithm has an iteration complexity of $K = \mathcal{O}\left(\frac{\mathrm{tr}(\mathbf{M})}{\mu}\log\frac{1}{\varepsilon}\right)$, aligning with the one of the Coordinate sketching version of the SEGA algorithm with importance sampling (Hanzely et al., 2018). However, our GSGD does not need access to the diagonal entries of matrix $\mathbf{M}$ which may be unaccessible such as in

the zeroth-order optimization setting just as discussed in Remark 3.1. A similar iteration complexity is also obtained by the recent work (Yue et al., 2023). However, because of lacking of variance reduction variate, the method of Yue et al. (2023) is only suitable for the smooth optimization. In contrast, our GSGD can be used for the composite optimization, that is, there is a regularization term contained in the objective function $F(x)$.

Next, we will analyze the iteration complexity of algorithm 1 when the objective function has a non-smooth regularization term. First, we give three important lemmas which are needed to derive Theorem 4.9. The first two Lemmas also show that the variances of g_k and h_{k+1} all decrease in the iterative process. This indicates that our algorithm 1 with an inseparable regularization term is also a variance reduction algorithm. Lemma 4.8 describes the phenomenon that the iterative sequences finally tend to be stable in the convergence process of the proximal gradient method.

Lemma 4.6. *For all $k > 0$, we define distances $v_1(k) = \|h_k - \nabla f(x^*)\|$ and $v_2(k) = \|\nabla f(x_k) - \nabla f(x^*)\|$, then, the variance of $g_k - \nabla f(x^*)$ can be bounded as follows*

$$\mathbb{E}_u\left[\|g_k - \nabla f(x^*)\|^2\right] \leq 2(d+2)v_1(k)^2 + 2(d+3)v_2(k)^2. \tag{18}$$

Lemma 4.7. *For all $k > 0$, we also denote distances $v_1 = \|h_k - \nabla f(x^*)\|$ and $v_2 = \|\nabla f(x_k) - \nabla f(x^*)\|$, then, the following equality is satisfied*

$$\mathbb{E}_u\left[\|h_{k+1} - \nabla f(x^*)\|^2\right] = \left(1 - \frac{1}{d+2}\right)v_1^2 + \frac{1}{d+2}v_2^2. \tag{19}$$

Lemma 4.8 ((Parikh et al., 2014))**.** *The proximal gradient algorithm can also be interpreted as a fixed point iteration. A point x^* is a solution of 1, if and only if*

$$0 \in \nabla f(x^*) + \partial\varphi(x^*).$$

Then, because the proximal operator is single-valued, we can obtain the following equality by the optimality condition

$$x^* = \mathrm{Prox}_{\alpha\varphi}\left(x^* - \eta\nabla f(x^*)\right). \tag{20}$$

Using Lemma 4.6, 4.7 and 4.8, we can obtain the following theorem.

Theorem 4.9. *Let the objective function $F(x)$ contain the general inseparable regularization term $\varphi(x)$, and assume that f is L-smooth and μ-strongly convex. That is, Assumption 2.1 and 2.2 hold. Define the following Lyapunov function*

$$\Psi^k \overset{def}{=} \|x_k - x^*\|^2 + \alpha_2\|h_k - \nabla f(x^*)\|^2,$$

and choose

$$\eta_2 = \frac{1}{2(3d+7)L}, \quad \alpha_2 = \frac{(d+2)^2}{(3d+7)^2L^2}, \tag{21}$$

$$c_2 = 1 - \frac{\mu}{2(3d+7)L}. \tag{22}$$

Then, the number of iterations of the algorithm satisfy

$$\mathbb{E}_u\left[\Psi^k\right] \leq c_2^k\Psi^0.$$

So, above theorem shows that our Algorithm 1 converges with a rate $1 - \mathcal{O}\left(\frac{\mu}{dL}\right)$. Accordingly, we will give the iteration complexity of Algorithm 1 in the following corollary.

Corollary 4.10. *Let the objective function satisfy the properties described in Theorem 4.9 and select the parameters in Theorem 4.9. To find an ε-suboptimal solution, the iteration complexity of Algorithm 1 is*

$$K = \mathcal{O}\left(\frac{dL}{\mu}\log\frac{1}{\varepsilon}\right). \tag{23}$$

Remark 4.11. Our algorithm can also achieve a linear convergence rate even the objective function has an inseparable regularization term. Eq. (23) demonstrates that our Algorithm 1 attains an iteration complexity of $K = \mathcal{O}\left(\frac{dL}{\mu}\log\frac{1}{\varepsilon}\right)$.

Next, we will show that when the regularization term has the special structure that $\varphi(x)$ satisfies Assumption 2.3, our algorithm can exploit this special structure to improve the convergence rate. We first give two more important lemmas which help to derive our theorem.

Lemma 4.12 ((Hanzely et al., 2020))**.** *Let $\{x_k\}_{k\geq 0}$ be a sequence of iterates of algorithm and let x^* be optimal solution for (1). Then*

$$x^k \in \{x_0 + \mathrm{Range}(\mathbf{W})\}, x^* \in \{x_0 + \mathrm{Range}(\mathbf{W})\},$$

for all k. Furthermore, for any $x, y \in \mathbb{R}^d$ we have

$$\|\mathrm{Prox}_{\alpha\varphi}(x) - \mathrm{Prox}_{\alpha\varphi}(y)\|^2 \leq \|x - y\|_{\mathbf{W}}^2. \tag{24}$$

Lemma 4.13. *For all $k > 0$, we define distances $v_1(k) = \|h_k - \nabla f(x^*)\|$ and $v_2(k) = \|\nabla f(x_k) - \nabla f(x^*)\|$, then, $\|g_k - \nabla f(x^*)\|_{\mathbf{W}}^2$ can be bounded as follows*

$$\mathbb{E}_u\left[\|g_k - \nabla f(x^*)\|_{\mathbf{W}}^2\right] \leq 6\mathrm{tr}(\mathbf{W})v_1(k)^2 + 7\mathrm{tr}(\mathbf{W})v_2(k)^2. \tag{25}$$

Using Lemma 4.7, 4.8, 4.12 and 4.13, we can obtain the following theorem.

Theorem 4.14. *Let the objective function $F(x)$ contain an inseparable regularization term which includes affine subspaces of $\mathbb{R}^d$, and assume that f is $\mathbf{M}$-smooth and μ-strongly convex. That is, Assumption (2.1), (2.2), (2.3) hold. Define the following Lyapunov function*

$$\Upsilon^k \overset{def}{=} \|x_k - x^*\|^2 + \alpha_3\|h_k - \nabla f(x^*)\|^2,$$

Table 1. Setting of diagonal matrix Σ used in Eq. (30) to construct $\mathbf{M}$.

Type	Σ
1	Matrix with first 481 components equal to 1 and the rest equal to 500
2	Matrix with first 499 components equal to 1 and the remaining one equal to 500
3	Starting from 401–th component is the value from 1 to 100 and the rest equal to 1
4	Matrix with components coming from uniform distribution U(0,1)

and choose

$$\eta_3 = \frac{1}{19L\mathrm{tr}(\mathbf{W})}, \quad \alpha_3 = 12(d+2)\eta^2\mathrm{tr}(\mathbf{W}), \tag{26}$$

$$c_3 = 1 - \frac{\mu}{19L\mathrm{tr}(\mathbf{W})}. \tag{27}$$

Then, the number of iterations of the algorithm satisfy

$$\mathbb{E}_u\left[\Upsilon^k\right] \leq c_3^k \Upsilon^0.$$

So, theorem 4.14 shows that our algorithm 1 converges with a rate $1 - \mathcal{O}\left(\frac{\mu}{L\mathrm{tr}(\mathbf{W})}\right)$. Next, by the convergence rate of our algorithm shown in Theorem 4.14, we will give iteration complexity of Algorithm 1 in the following corollary.

Corollary 4.15. *Let the objective function satisfy the properties described in Theorem 4.14 and select the parameters in Theorem 4.14. To find an ε-suboptimal solution, the iteration complexity of Algorithm 1 is*

$$K = \mathcal{O}\left(\frac{L\mathrm{tr}(\mathbf{W})}{\mu}\log\frac{1}{\varepsilon}\right). \tag{28}$$

Remark 4.16. In the case of a special setting of the inseparable regularization term structure, that is, Assumption 2.3 holds, our algorithm can also achieve a linear convergence rate. Eq. (28) demonstrates that Algorithm 1 attains an iteration complexity of $K = \mathcal{O}\left(\frac{L\mathrm{tr}(\mathbf{W})}{\mu}\log\frac{1}{\varepsilon}\right)$. This indicates that the convergence rate of our Algorithm 1 is determined by the rank of the projection matrix $\mathbf{W}$. The smaller the rank of the projection matrix is, the faster the convergence rate of our Algorithm 1 will be. Thus, our Algorithm 1 can exploit the special structure of the regularization term to obtain a faster convergence rate.

5. Experiments

We have provided a comprehensive theoretical analysis of our Algorithm 1 in the preceding sections. This section is dedicated to the empirical validation of our algorithm's effectiveness and superiority. Our experiments will focus on the quadratic minimization problem, whose objective function adheres to the form delineated in Eq. (1), characterized by

$$\min_{x\in\mathbb{R}^d} F(x) = \frac{1}{2}x^T\mathbf{M}x - b^Tx + \varphi(x), \tag{29}$$

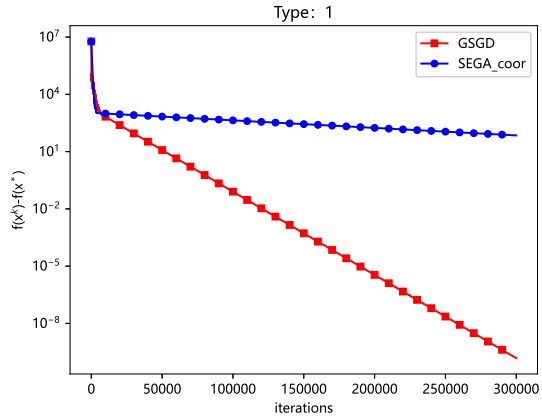
where $\varphi(x)$ is divided into two cases, the first situation is $\varphi(x) = 0$, the other situation is $\varphi(x)$ is an indicator function of the unit ball intersected with $\mathrm{Range}(\mathbf{W})$. In the second case, we will reveal that the special structure of the inseparable regularization term can accelerate our Algorithm 1 through experimental results. It can be observed that the quadratic function part in Eq. (29) exhibits both μ-strong convexity and smoothness if we properly choose $\mathbf{M}$. Simultaneously, $\varphi(x)$ is convex but may be non-differentiable. Therefore, the experimental setup is consistent with our theoretical analysis. At the same time, the quadratic minimization problem fulfills all assumptions required for our Algorithm 1.

The parameters of the quadratic function which we construct as follows: the dimension of feature vector x is 500. We set

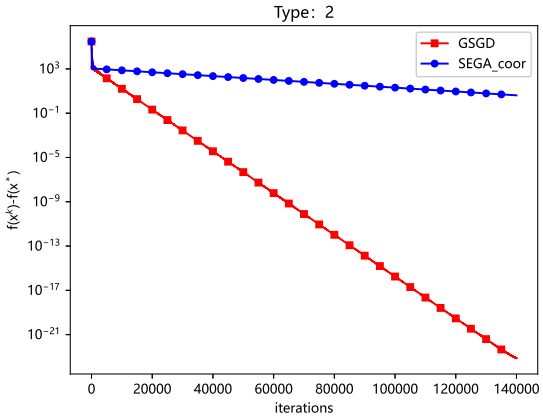
$$\mathbf{M} \overset{def}{=} \mathbf{U}\Sigma\mathbf{U}^T, \tag{30}$$

where $\mathbf{U}$ obtained from QR decomposition of random matrix with independent entries from $N(0,1)$ and Σ is set as Table 1 and b is a random vector with independent entries drawn from $N(0,1)$. For each problem, the starting point was chosen to be a vector with independent entries from $N(0,1)$.

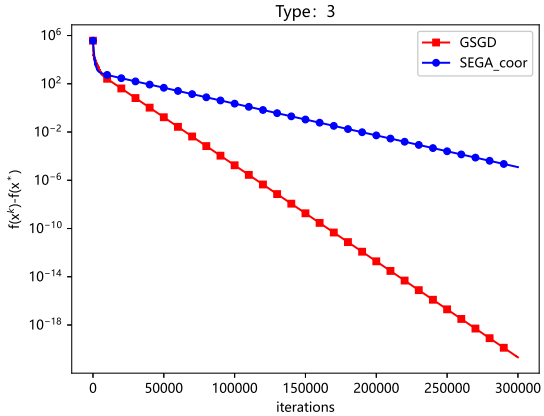
In the first experiment, we compare our GSGD with the Coordinate sketching version of SEGA algorithm (Hanzely et al., 2018) for problems with $\varphi(x) = 0$. In the experiment, we properly choose the step sizes of these two algorithms. According to the theoretical results of our algorithm and SEGA, step sizes of these two algorithms should be proportional to $\mathcal{O}(1/\mathrm{tr}(\mathbf{M}))$ and $\mathcal{O}(1/(d\lambda(\mathbf{M})))$, respectively. We report the experiment results in Figure 1. We can observe that in the first three experiments, our algorithm is significantly faster than the Coordinate sketching version of the SEGA algorithm. This is because the iteration complexity of our algorithm is linear to $\mathrm{tr}(\mathbf{M})/\mu$, while the iteration complexity of the Coordinate sketching version of the SEGA algorithm is linear to dL/μ. At the same time, the first three experimental settings has the properties that the Σ matrices guarantee $\mathrm{tr}(\mathbf{M})/\mu$ is much less than dL/μ. This result matches our theoretical analysis in Corollary 4.4. In the forth experiment, our algorithm has a similar performance to SEGA. This is because it holds that $\mathrm{tr}(\mathbf{M})/\mu \approx dL/\mu$ in the case that the diagonal elements of $\mathbf{M}$ matrix obey uniform distribution.



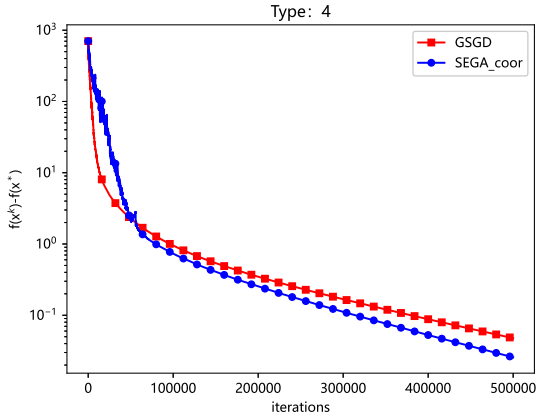
(a) The comparison on the first type diagonal matrix of Table 1



(b) The comparison on the second type diagonal matrix of Table 1



(c) The comparison on the third type diagonal matrix of Table 1



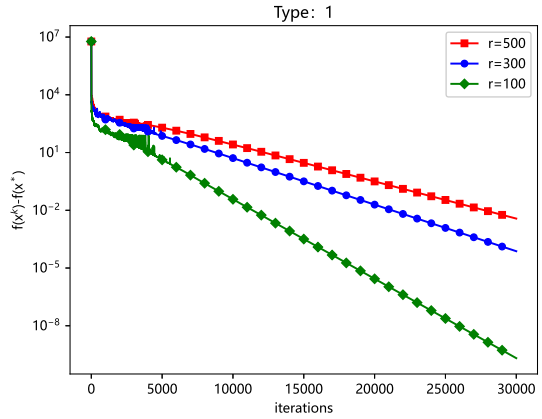
(d) The comparison on the fourth type diagonal matrix of Table 1

Figure 1. Comparison of both our algorithm and the Coordinate sketching version of the SEGA algorithm with uniform sampling.

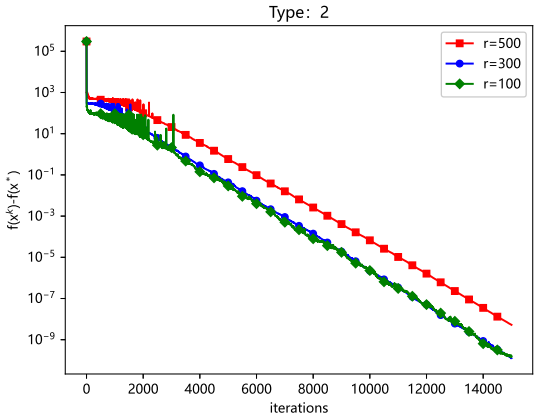
The second experiment compares the performance of our algorithm for various $\mathbf{W}$ when $\varphi(x)$ is an indicator function of the unit ball intersected with $\mathrm{Range}(\mathbf{W})$. We conduct experiments under the condition that the rank of the projection matrix is 500, 300, 100. According to the theoretical result of our algorithm, the step size should be proportional to $1/(L\mathrm{tr}(\mathbf{W}))$, respectively. Figure 2 shows the result. We can find that the smaller $\mathrm{Rank}(\mathbf{W})$ is, the faster the convergence of our algorithm is in the great majority of cases. This experimental result is consistent with our theoretical proof. Specifically, in the first experiment and the third experiment, with the decrease of the rank of the projection matrix $\mathbf{W}$, the acceleration effect of our algorithm is more obvious, and this matches our theoretical analysis. The acceleration effect of the second experiment is more obvious when the rank of $\mathbf{W}$ decreases from 500 to 300. However, this acceleration is not obvious when the rank of $\mathbf{W}$ decreases from 300 to 100. And the acceleration effect of the fourth experiment is not obvious at all. Combining the common characteristics of the parameter settings of the second experiment and the fourth experiment, it can be concluded that $\mathrm{tr}(\mathbf{M})$ is very small and the characteristic coefficients are dispersed very uniformly may cause this poor result.

6. Conclusion and Future Work

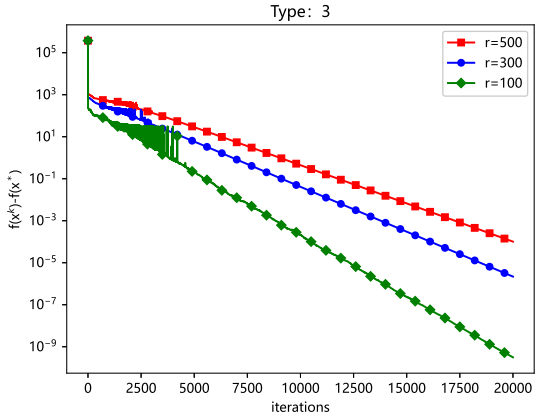
In this paper, we first propose a novel algorithm called GSGD. When the objective function without any regularization term that is $\varphi(x) = 0$, the iteration complexity of GSGD is $\mathcal{O}\left(\frac{\mathrm{tr}(\mathbf{M})}{\mu}\log\frac{1}{\varepsilon}\right)$. Compared with the Coordinate sketching version of SEGA, it can achieve a faster convergence rate without importance sampling. Furthermore, we prove that our algorithm can still achieve a linear convergence rate with an inseparable regularization term, and its iteration complexity is $\mathcal{O}\left(\frac{dL}{\mu}\log\frac{1}{\varepsilon}\right)$. Moreover, if the regulariza-



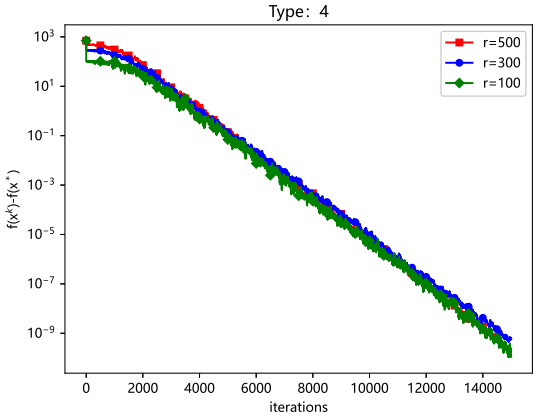
(a) The comparison on the first type diagonal matrix of Table 1



(b) The comparison on the second type diagonal matrix of Table 1



(c) The comparison on the third type diagonal matrix of Table 1



(d) The comparison on the fourth type diagonal matrix of Table 1

Figure 2. Comparison of our algorithm performance for $\mathbf{W}$ with different rank

tion term also satisfies some special structure, specifically, satisfying Assumption 2.3, our GSGD can also exploit the special structure of the the regularization term to obtain faster convergence rate. Finally, the effectiveness and efficiency of our algorithm for different settings have been validated by our experiments.

In the future work, there are the following aspects can be studied. The first is the relationship between our acceleration scheme and the sparsity of the solution. It is undoubtedly exciting when we want to obtain a sparse solution while also obtaining a good acceleration effect. Secondly, it is significant to try to apply our acceleration scheme to the finite difference representation of the real gradient when computing and storing gradient is difficult. Finally, further exploring the influence of $\mathrm{tr}(\mathbf{M})$ on the acceleration effect of our algorithm is also meaningful.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant 12101491 and A*star Centre for Frontier AI Research.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Allen-Zhu, Z., Qu, Z., Richtárik, P., and Yuan, Y. Even faster accelerated coordinate descent using non-uniform sampling. In *International Conference on Machine Learning*, pp. 1110–1119. PMLR, 2016.

Berahas, A. S., Cao, L., Choromanski, K., and Scheinberg, K. A theoretical and empirical comparison of gradient approximations in derivative-free optimization. *Foundations of Computational Mathematics*, 22(2):507–560, 2022.

Beznosikov, A., Gorbunov, E., Berard, H., and Loizou, N. Stochastic gradient descent-ascent: Unified theory and new efficient methods. In *International Conference on Artificial Intelligence and Statistics*, pp. 172–235. PMLR, 2023.

Chorobura, F. and Necoara, I. Random coordinate descent methods for nonseparable composite optimization. *SIAM Journal on Optimization*, 33(3):2160–2190, 2023.

Defazio, A., Bach, F., and Lacoste-Julien, S. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in neural information processing systems*, 27, 2014.

Fercoq, O. and Richtárik, P. Accelerated, parallel, and proximal coordinate descent. *SIAM Journal on Optimization*, 25(4):1997–2023, 2015.

Gower, R., Kovalev, D., Lieder, F., and Richtárik, P. Rsn: randomized subspace newton. *Advances in Neural Information Processing Systems*, 32, 2019.

Gower, R. M. Sketch and project: Randomized iterative methods for linear systems and inverting matrices. *arXiv preprint arXiv:1612.06013*, 2016.

Gower, R. M. and Richtárik, P. Randomized iterative methods for linear systems. *SIAM Journal on Matrix Analysis and Applications*, 36(4):1660–1690, 2015a.

Gower, R. M. and Richtárik, P. Stochastic dual ascent for solving linear systems. *arXiv preprint arXiv:1512.06890*, 2015b.

Gower, R. M., Schmidt, M., Bach, F., and Richtárik, P. Variance-reduced methods for machine learning. *Proceedings of the IEEE*, 108(11):1968–1983, 2020.

Hanzely, F. and Richtárik, P. Accelerated coordinate descent with arbitrary sampling and best rates for minibatches. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 304–312. PMLR, 2019a.

Hanzely, F. and Richtárik, P. One method to rule them all: Variance reduction for data, parameters and many new methods. *arXiv preprint arXiv:1905.11266*, 2019b.

Hanzely, F., Mishchenko, K., and Richtárik, P. Sega: Variance reduction via gradient sketching. *Advances in Neural Information Processing Systems*, 31, 2018.

Hanzely, F., Kovalev, D., and Richtarik, P. Variance reduced coordinate descent with acceleration: New method with a surprising application to finite-sum problems. In *International Conference on Machine Learning*, pp. 4039–4048. PMLR, 2020.

Johnson, R. and Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26, 2013.

Kozak, D., Becker, S., Doostan, A., and Tenorio, L. Stochastic subspace descent. *arXiv preprint arXiv:1904.01145*, 2019.

Liu, J. and Wright, S. J. Asynchronous stochastic coordinate descent: Parallelism and convergence properties. *SIAM Journal on Optimization*, 25(1):351–376, 2015.

Liu, J., Wright, S., Ré, C., Bittorf, V., and Sridhar, S. An asynchronous parallel stochastic coordinate descent algorithm. In *International Conference on Machine Learning*, pp. 469–477. PMLR, 2014.

Magnus, J. R. et al. *The moments of products of quadratic forms in normal variables*. Univ., Instituut voor Actuariaat en Econometrie, 1978.

Nemirovski, A., Juditsky, A., Lan, G., and Shapiro, A. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009.

Nesterov, Y. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

Nesterov, Y. and Spokoiny, V. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17:527–566, 2017.

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

Qu, Z. and Richtárik, P. Coordinate descent with arbitrary sampling ii: Expected separable overapproximation. *Optimization Methods and Software*, 31(5):858–884, 2016.

Richtárik, P. and Takáč, M. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144 (1-2):1–38, 2014.

Richtárik, P. and Takáč, M. On optimal probabilities in stochastic coordinate descent methods. *Optimization Letters*, 10:1233–1243, 2016.

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

Rothchild, D., Panda, A., Ullah, E., Ivkin, N., Stoica, I., Braverman, V., Gonzalez, J., and Arora, R. Fetchsgd: Communication-efficient federated learning with sketching. In *International Conference on Machine Learning*, pp. 8253–8265. PMLR, 2020.

Sanz Serna, J. M. and Zygalakis, K. C. The connections between lyapunov functions for some optimization algorithms and differential equations. *SIAM Journal on Numerical Analysis*, 59(3):1542–1565, 2021.

Song, Z., Wang, Y., Yu, Z., and Zhang, L. Sketching for first order method: efficient algorithm for low-bandwidth channel and vulnerability. In *International Conference on Machine Learning*, pp. 32365–32417. PMLR, 2023.

Wilson, A. *Lyapunov arguments in optimization*. University of California, Berkeley, 2018.

Wilson, A. C., Recht, B., and Jordan, M. I. A lyapunov analysis of momentum methods in optimization. *arXiv preprint arXiv:1611.02635*, 2016.

Wright, S. J. Coordinate descent algorithms. *Mathematical programming*, 151(1):3–34, 2015.

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

Yue, P., Yang, L., Fang, C., and Lin, Z. Zeroth-order optimization with weak dimension dependency. In *The Thirty Sixth Annual Conference on Learning Theory*, pp. 4429–4472. PMLR, 2023.

Zou, H. and Hastie, T. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67(2):301–320, 2005.

A. Several Useful Lemmas

The following lemma shows that the expectation of the product of two quadratic forms of the random Gaussian vector is related to the trace of the corresponding matrix.

Lemma A.1 ((Magnus et al., 1978)). *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be two symmetric matrices, and* u *obeys the Gaussian distribution, that is,* $u \sim N(0, \mathbf{I}_d)$. *Define* $z = u^\top \mathbf{A}u \cdot u^\top \mathbf{B}u$. *The expectation of* z *is*

$$\mathbb{E}_u[z] = (\mathrm{tr}\mathbf{A})(\mathrm{tr}\mathbf{B}) + 2(\mathrm{tr}\mathbf{A}\boldsymbol{B}). \tag{31}$$

Lemma A.2. *If we have a positive definite matrix* $\mathbf{B}$ *defined as weighted inner product, for all* $x \in \mathbb{R}^d$, *we can obtain the following inequality*

$$\|x\|_{\mathbf{B}}^2 \leq \mathrm{tr}(\mathbf{B}) \|x\|^2. \tag{32}$$

Proof. For a positive definite matrix $\mathbf{B}$, there must exist an orthogonal matrix $\mathbf{T}$ such that $\mathbf{B}$ is similar to a diagonal matrix whose elements are eigenvalues of matrix $\mathbf{B}$. We denote λ_i be the i-th eigenvalue of matrix $\mathbf{B}$, then, we can obtain an equation as follows

$$\mathbf{B} = \mathbf{T}\mathrm{diag}\left\{\lambda_1, \lambda_2, \cdots, \lambda_d\right\}\mathbf{T}^{-1}. \tag{33}$$

Then, we can easily prove this Lemma,

$$\begin{aligned}\|x\|_{\mathbf{B}}^2 =& \langle \mathbf{B}x, x\rangle = x^T\mathbf{B}x \overset{(33)}{=} x^T\mathbf{T}\mathrm{diag}\left\{\lambda_1, \lambda_2, \cdots, \lambda_d\right\}\mathbf{T}^{-1}x \\ \leq& x^T\mathbf{T}\sum_{i=1}^{d}\lambda_i \cdot \mathbf{I}_d\mathbf{T}^{-1}x = \mathrm{tr}(\mathbf{B})\|x\|^2.\end{aligned}$$

□

We can learn that the regularization term includes two parts under Assumption 2.3, one is the ordinary regularization term, and the other is the indicator function of the $\mathrm{Range}(\mathbf{W})$. The second part of the inseparable regularization term implies that we need to project the iteration point into the column space of the matrix $\mathbf{W}$ in the process of iteration. Since $\mathbf{W}$ is a projection matrix, we have the following hold.

Lemma A.3. *Let the current iteration point be* x_k. *Since* $x_k \in \mathrm{Range}(\mathbf{W})$ *under Assumption 2.3, we can obtain the following relation*

$$x_k = \mathbf{W}x_k. \tag{34}$$

Proof. We can use the idempotence and symmetry of the projection matrix to prove this Lemma easily,

$$\begin{aligned}x_k &= \mathbf{W}(\mathbf{W}^T\mathbf{W})^\dagger\mathbf{W}^T x_k \\ &= \mathbf{W}(\mathbf{W}^2)^\dagger\mathbf{W}x_k \\ &= \mathbf{W}(\mathbf{W})^\dagger\mathbf{W}x_k \\ &= \mathbf{W}x_k.\end{aligned}$$

□

B. Proof of Important Lemmas

In this section, we give some details of proof about some important Lemmas.

B.1. Proof of Lemma 4.1

Proof.

$$\begin{aligned}\mathbb{E}_u\left[\|g_k\|_{\mathbf{M}}^2\right] =& \mathbb{E}_u\left[\left\|h_k + uu^\top(\nabla f(x_k) - h_k)\right\|_{\mathbf{M}}^2\right] \\ =& \mathbb{E}_u\left[\|h_k\|_{\mathbf{M}}^2 + 2\langle h_k, uu^\top(\nabla f(x_k) - h_k)\rangle_{\mathbf{M}} + \left\|uu^\top(\nabla f(x_k) - h_k)\right\|_{\mathbf{M}}^2\right] \\ =& \|h_k\|_{\mathbf{M}}^2 + 2\langle h_k, \nabla f(x_k) - h_k\rangle_{\mathbf{M}} + \mathbb{E}_u\left[\left\|uu^\top(\nabla f(x_k) - h_k)\right\|_{\mathbf{M}}^2\right].\end{aligned}$$

Using the (31), we can compute the last term to get the following

$$\begin{aligned}\mathbb{E}_u\left[\left\|uu^\top(\nabla f(x_k)-h_k)\right\|_{\mathbf{M}}^2\right] =&\mathbb{E}_u\left[(\nabla f(x_k)-h_k)^\top uu^\top \mathbf{M}^\top uu^\top(\nabla f(x_k)-h_k)\right]\\ =&\mathbb{E}_u\left[\mathrm{tr}((\nabla f(x_k)-h_k)^\top uu^\top \mathbf{M}^\top uu^\top(\nabla f(x_k)-h_k))\right]\\ =&\mathbb{E}_u\left[\mathrm{tr}(u^\top \mathbf{M}^\top uu^\top(\nabla f(x_k)-h_k)(\nabla f(x_k)-h_k)^\top u)\right]\\ \overset{(31)}{=}&\mathrm{tr}(\mathbf{M})\mathrm{tr}((\nabla f(x_k)-h_k)(\nabla f(x_k)-h_k)^\top)\\ &+2\mathrm{tr}((\nabla f(x_k)-h_k)^\top)\mathbf{M}^\top(\nabla f(x_k)-h_k)\\ =&\mathrm{tr}(\mathbf{M})\left\|\nabla f(x_k)-h_k\right\|^2+2\left\|\nabla f(x_k)-h_k\right\|_{\mathbf{M}}^2.\end{aligned} \tag{35}$$

Plugging this into the equation with which we started the proof, we deduce

$$\begin{aligned}\mathbb{E}_u\left[\left\|g_k\right\|_{\mathbf{M}}^2\right] =&\left\|h_k\right\|_{\mathbf{M}}^2+2\langle h_k,\nabla f(x_k)-h_k\rangle_{\mathbf{M}}+\mathrm{tr}(\mathbf{M})\left\|\nabla f(x_k)-h_k\right\|^2+2\left\|\nabla f(x_k)-h_k\right\|_{\mathbf{M}}^2\\ \leq&\left\|h_k\right\|_{\mathbf{M}}^2+2\left\|\nabla f(x_k)\right\|_{\mathbf{M}}^2-2\langle\nabla f(x_k),h_k\rangle_{\mathbf{M}}+2\mathrm{tr}(\mathbf{M})(\left\|\nabla f(x_k)\right\|^2+\left\|h_k\right\|^2)\\ \leq&2\left\|h_k\right\|_{\mathbf{M}}^2+3\left\|\nabla f(x_k)\right\|_{\mathbf{M}}^2+2\mathrm{tr}(\mathbf{M})(\left\|\nabla f(x_k)\right\|^2+\left\|h_k\right\|^2)\\ \overset{(32)}{\leq}&4\mathrm{tr}(\mathbf{M})\left\|h_k\right\|^2+5\mathrm{tr}(\mathbf{M})\left\|\nabla f(x_k)\right\|^2.\end{aligned} \tag{36}$$

□

B.2. Proof of Lemma 4.2

Proof.

$$\begin{aligned}\mathbb{E}_u\left[\left\|h_{k+1}\right\|^2\right] =&\mathbb{E}_u\left[\left\|h_k+\frac{uu^\top}{d+2}(\nabla f(x_k)-h_k)\right\|^2\right]\\ =&\mathbb{E}_u\left[\left\|h_k\right\|^2+\frac{2}{d+2}\langle h_k,uu^\top(\nabla f(x_k)-h_k)\rangle+\frac{1}{(d+2)^2}\left\|uu^\top(\nabla f(x_k)-h_k)\right\|^2\right]\\ =&\left\|h_k\right\|^2+\frac{2}{d+2}\langle h_k,(\nabla f(x_k)-h_k)\rangle+\frac{1}{(d+2)^2}\mathbb{E}_u\left[\left\|uu^\top(\nabla f(x_k)-h_k)\right\|^2\right].\end{aligned}$$

Using the (31), we can also compute the last term

$$\begin{aligned}\mathbb{E}_u\left[\left\|uu^\top(\nabla f(x_k)-h_k)\right\|^2\right] =&\mathbb{E}_u\left[(\nabla f(x_k)-h_k)^\top uu^\top uu^\top(\nabla f(x_k)-h_k)\right]\\ =&\mathbb{E}_u\left[\mathrm{tr}((\nabla f(x_k)-h_k)^\top uu^\top uu^\top(\nabla f(x_k)-h_k))\right]\\ =&\mathbb{E}_u\left[\mathrm{tr}(u^\top uu^\top(\nabla f(x_k)-h_k)(\nabla f(x_k)-h_k)^\top u)\right]\\ \overset{(31)}{=}&d\cdot\mathrm{tr}((\nabla f(x_k)-h_k)(\nabla f(x_k)-h_k)^\top)\\ &+2\mathrm{tr}((\nabla f(x_k)-h_k)^\top)(\nabla f(x_k)-h_k)\\ =&(d+2)\left\|\nabla f(x_k)-h_k\right\|^2.\end{aligned}$$

Plugging this into the above equation, we can deduce

$$\begin{aligned}\mathbb{E}_u\left[\left\|h_{k+1}\right\|^2\right] =&\left\|h_k\right\|^2+\frac{2}{d+2}\langle h_k,(\nabla f(x_k)-h_k)\rangle+\frac{1}{d+2}\left\|\nabla f(x_k)-h_k\right\|^2\\ =&\left(1-\frac{1}{d+2}\right)\left\|h_k\right\|^2+\frac{1}{d+2}\left\|\nabla f(x_k)\right\|^2.\end{aligned} \tag{37}$$

□

B.3. Proof of Lemma 4.6

Proof.

$$\begin{aligned}\mathbb{E}_u\left[\|g_k-\nabla f(x^*)\|^2\right]=&\mathbb{E}_u\left[\left\|h_k-\nabla f(x^*)+uu^\top(\nabla f(x_k)-h_k)\right\|^2\right]\\&=\|h_k-\nabla f(x^*)\|^2+2\langle h_k-\nabla f(x^*),\nabla f(x_k)-h_k\rangle+\mathbb{E}_u\left[\left\|uu^\top(\nabla f(x_k)-h_k)\right\|\right].\end{aligned}$$

Let $\mathbf{M}=\mathbf{I}_d$, we can turn (35) into the following form

$$\mathbb{E}_u\left[\left\|uu^\top(\nabla f(x_k)-h_k)\right\|^2\right]=(d+2)\left\|\nabla f(x_k)-h_k\right\|^2. \tag{38}$$

Plugging this into the above equation, we can deduce

$$\begin{aligned}\mathbb{E}_u\left[\|g_k-\nabla f(x^*)\|^2\right]=&\|h_k-\nabla f(x^*)\|^2+2\langle h_k-\nabla f(x^*),\nabla f(x_k)-\nabla f(x^*)+\nabla f(x^*)-h_k\rangle\\&+(d+2)\left\|\nabla f(x_k)-\nabla f(x^*)+\nabla f(x^*)-h_k\right\|^2\\=&2\langle h_k-\nabla f(x^*),\nabla f(x_k)-\nabla f(x^*)\rangle-\|h_k-\nabla f(x^*)\|^2+(d+2)\left\|\nabla f(x_k)-\nabla f(x^*)\right\|^2\\&+2(d+2)\langle\nabla f(x_k)-\nabla f(x^*),\nabla f(x^*)-h_k\rangle+(d+2)\left\|\nabla f(x^*)-h_k\right\|^2\\\le&2(d+2)\left\|h_k-\nabla f(x^*)\right\|^2+\left\|\nabla f(x_k)-\nabla f(x^*)\right\|^2+2(d+2)\left\|\nabla f(x_k)-\nabla f(x^*)\right\|^2\\\le&2(d+2)\left\|h_k-\nabla f(x^*)\right\|^2+2(d+3)\left\|\nabla f(x_k)-\nabla f(x^*)\right\|^2.\end{aligned} \tag{39}$$

□

B.4. Proof of Lemma 4.7

Proof.

$$\begin{aligned}\mathbb{E}_u\left[\|h_{k+1}-\nabla f(x^*)\|^2\right]=&\mathbb{E}_u\left[\left\|h_k+\frac{uu^\top}{d+2}(\nabla f(x_k)-h_k)-\nabla f(x^*))\right\|^2\right]\\&=\|h_k-\nabla f(x^*)\|^2+\frac{2}{d+2}\langle\nabla f(x_k)-h_k,h_k-\nabla f(x^*)\rangle\\&\quad+\frac{1}{(d+2)^2}\mathbb{E}_u\left[\left\|uu^\top(\nabla f(x_k)-h_k)\right\|^2\right]\\&\overset{(38)}{=}\|h_k-\nabla f(x^*)\|^2+\frac{2}{d+2}\langle\nabla f(x_k)-\nabla f(x^*)+\nabla f(x^*)-h_k,h_k-\nabla f(x^*)\rangle\\&\quad+\frac{1}{d+2}\left\|\nabla f(x_k)-\nabla f(x^*)+\nabla f(x^*)-h_k\right\|^2\\&=\left(1-\frac{1}{d+2}\right)\|h_k-\nabla f(x^*)\|^2+\frac{1}{d+2}\left\|\nabla f(x_k)-\nabla f(x^*)\right\|^2.\end{aligned} \tag{40}$$

□

B.5. Proof of Lemma 4.13

Proof.

$$\begin{aligned}\mathbb{E}_u\left[\|g_k-\nabla f(x^*)\|_{\mathbf{W}}^2\right]=&\mathbb{E}_u\left[\left\|h_k-\nabla f(x^*)+uu^\top(\nabla f(x_k)-h_k)\right\|_{\mathbf{W}}^2\right]\\&=\|h_k-\nabla f(x^*)\|_{\mathbf{W}}^2+2\langle h_k-\nabla f(x^*),\nabla f(x_k)-h_k\rangle_{\mathbf{W}}+\mathbb{E}_u\left[\left\|uu^\top(\nabla f(x_k)-h_k)\right\|_{\mathbf{W}}\right].\end{aligned}$$

Let $\mathbf{M}=\mathbf{W}$, we can turn (35) into the following form

$$\mathbb{E}_u\left[\left\|uu^\top(\nabla f(x_k)-h_k)\right\|_{\mathbf{W}}^2\right]=\operatorname{tr}(\mathbf{W})\left\|\nabla f(x_k)-h_k\right\|^2+2\left\|\nabla f(x_k)-h_k\right\|_{\mathbf{W}}^2. \tag{41}$$

Plugging this into the above equation, we can deduce

$$\begin{aligned}
\mathbb{E}_u\left[\|g_k-\nabla f(x^*)\|_{\mathbf{W}}^2\right] =& 2\langle h_k-\nabla f(x^*),\nabla f(x_k)-\nabla f(x^*)\rangle_{\mathbf{W}}-\|h_k-\nabla f(x^*)\|_{\mathbf{W}}^2 \\
&+\operatorname{tr}(\mathbf{W})\|\nabla f(x_k)-h_k\|^2+2\|\nabla f(x_k)-h_k\|_{\mathbf{W}}^2 \\
=& 2\langle h_k-\nabla f(x^*),\nabla f(x_k)-\nabla f(x^*)\rangle_{\mathbf{W}}-\|h_k-\nabla f(x^*)\|_{\mathbf{W}}^2 \\
&+\operatorname{tr}(\mathbf{W})\left[\|\nabla f(x_k)-\nabla f(x^*)\|^2+\|\nabla f(x^*)-h_k\|^2+2\langle\nabla f(x_k)-\nabla f(x^*),f(x^*)-h_k\rangle\right] \\
&+2\left[\|\nabla f(x_k)-\nabla f(x^*)\|_{\mathbf{W}}^2+\|\nabla f(x^*)-h_k\|_{\mathbf{W}}^2+2\langle\nabla f(x_k)-\nabla f(x^*),f(x^*)-h_k\rangle_{\mathbf{W}}\right] \\
\le& 4\|h_k-\nabla f(x^*)\|_{\mathbf{W}}^2+5\|\nabla f(x_k)-\nabla f(x^*)\|_{\mathbf{W}}^2 \\
&+2\operatorname{tr}(\mathbf{W})\left[\|\nabla f(x_k)-\nabla f(x^*)\|^2+\|h_k-\nabla f(x^*)\|^2\right] \\
\overset{(32)}{\le}& 6\operatorname{tr}(\mathbf{W})\|h_k-\nabla f(x^*)\|^2+7\operatorname{tr}(\mathbf{W})\|\nabla f(x_k)-\nabla f(x^*)\|^2. \qquad (42)
\end{aligned}$$

□

C. Proof of Main Theorems

In this section, we give some details of proof about our main Theorems.

C.1. Proof of Theorem 4.3

Proof. Firstly, we can deduce the expectation of $f(x_{k+1})$,

$$\begin{aligned}
\mathrm{E}_u\left[f(x_{k+1})\right] \overset{(2)}{\le}& f(x_k)+\mathrm{E}_u\left[\langle\nabla f(x_k),x_{k+1}-x_k\rangle+\frac{1}{2}\|x_{k+1}-x_k\|_{\mathbf{M}}^2\right] \\
\overset{(9)}{=}& f(x_k)-\eta_1\langle\nabla f(x_k),\mathrm{E}_u\left[g_k\right]\rangle+\frac{\eta_1^2}{2}\mathrm{E}_u\left[\|g_k\|_{\mathbf{M}}^2\right]. \\
=& f(x_k)-\eta_1\|\nabla f(x_k)\|^2+\frac{\eta_1^2}{2}\mathrm{E}_u\left[\|g_k\|_{\mathbf{M}}^2\right].
\end{aligned}$$

Combining it into the expectation of Lyapunov function Φ^{k+1},

$$\begin{aligned}
\mathrm{E}_u\left[\Phi^{k+1}\right] \le& f(x_k)-f(x^*)-\eta_1\|\nabla f(x_k)\|^2+\frac{\eta_1^2}{2}\mathrm{E}_u\left[\|g_k\|_{\mathbf{M}}^2\right]+\eta_1\alpha_1(d+2)\cdot\operatorname{tr}(\mathbf{M})\mathrm{E}_u\left[\|h_{k+1}\|^2\right] \\
\overset{(36)+(37)}{\le}& f(x_k)-f(x^*)-\eta_1\|\nabla f(x_k)\|^2+5\eta_1^2\operatorname{tr}(\mathbf{M})\|\nabla f(x_k)\|^2 \\
&+\eta_1\alpha_1\cdot\operatorname{tr}(\mathbf{M})\|\nabla f(x_k)\|^2+4\eta_1^2\operatorname{tr}(\mathbf{M})\|h_k\|^2+\eta_1\alpha_1(d+2)\left(1-\frac{1}{d+2}\right)\cdot\operatorname{tr}(\mathbf{M})\|h_k\|^2 \\
=& f(x_k)-f(x^*)-\eta_1\left(1-5\eta_1\operatorname{tr}(\mathbf{M})-\alpha_1\operatorname{tr}(\mathbf{M})\right)\|\nabla f(x_k)\|^2 \\
&+\eta_1\alpha_1(d+2)\left(1-\frac{1}{d+2}+\frac{4\eta_1}{\alpha_1(d+2)}\right)\cdot\operatorname{tr}(\mathbf{M})\|h_k\|^2 \\
\overset{(15)}{=}& f(x_k)-f(x^*)-\frac{1}{80\operatorname{tr}(\mathbf{M})}\|\nabla f(x_k)\|^2+\eta_1\alpha_1(d+2)\cdot\operatorname{tr}(\mathbf{M})\left(1-\frac{3}{5(d+2)}\right)\|h_k\|^2.
\end{aligned}$$

By the μ-strongly convexity, we can obtain that

$$-\|\nabla f(x_k)\|^2\le-2\mu(f(x_k)-f(x^*)).$$

Thus, we can obtain that

$$\begin{aligned}
\mathrm{E}_u\left[\Phi^{k+1}\right] &\leq \left(1-\frac{\mu}{40\mathrm{tr}(\mathbf{M})}\right)\left(f(x_k)-f(x^*)\right)+\eta_1\alpha_1(d+2)\cdot\mathrm{tr}(\mathbf{M})\left(1-\frac{3}{5(d+2)}\right)\|h_k\|^2 \\
&\leq \max\left\{1-\frac{\mu}{40\mathrm{tr}(\mathbf{M})},1-\frac{3}{5(d+2)}\right\}\left(f(x_k)-f(x^*)+\eta_1\alpha_1(d+2)\cdot\mathrm{tr}(\mathbf{M})\|h_k\|^2\right) \\
&\overset{(16)}{\leq} c^{k+1}\Phi^0.
\end{aligned}$$

□

C.2. Proof of Theorem 4.9

Proof. Firstly, we can deduce the expectation of $\|x_{k+1}-x^*\|^2$,

$$\begin{aligned}
\mathrm{E}_u\left[\|x_{k+1}-x^*\|^2\right] &\overset{(8)+(20)}{=} \mathrm{E}_u\left[\|\mathrm{Prox}_{\alpha\varphi}\left(x_k-\eta_2 g_k\right)-\mathrm{Prox}_{\alpha\varphi}\left(x^*-\eta_2\nabla f(x^*)\right)\|^2\right] \\
&\overset{(24)}{\leq} \mathrm{E}_u\left[\|x_k-x^*-\eta_2(g_k-\nabla f(x^*))\|^2\right] \\
&= \|x_k-x^*\|^2-2\eta_2\langle\nabla f(x_k)-\nabla f(x^*),x_k-x^*\rangle+\eta_2^2\mathrm{E}_u\left[\|g_k-\nabla f(x^*)\|^2\right] \\
&\overset{(4)}{\leq} \|x_k-x^*\|^2-2\eta_2\left(f(x_k)-f(x^*)-\langle\nabla f(x^*),x_k-x^*\rangle+\frac{\mu}{2}\|x_k-x^*\|^2\right) \\
&\quad+\eta_2^2\mathrm{E}_u\left[\|g_k-\nabla f(x^*)\|^2\right] \\
&= (1-\eta_2\mu)\|x_k-x^*\|^2-2\eta_2\left(f(x_k)-f(x^*)-\langle\nabla f(x^*),x_k-x^*\rangle\right)+\eta_2^2\mathrm{E}\left[\|g_k-\nabla f(x^*)\|^2\right].
\end{aligned}$$

Because f is L-smooth, we can obtain

$$f(x_k)-f(x^*)-\langle\nabla f(x^*),x_k-x^*\rangle\geq\frac{1}{2L}\|\nabla f(x_k)-\nabla f(x^*)\|^2. \tag{43}$$

Plugging this into above inequation, we can obtain

$$\mathrm{E}_u\left[\|x_{k+1}-x^*\|^2\right]\leq(1-\eta_2\mu)\|x_k-x^*\|^2-\frac{\eta_2}{L}\|\nabla f(x_k)-\nabla f(x^*)\|^2+\eta_2^2\mathrm{E}_u\left[\|g_k-\nabla f(x^*)\|^2\right]. \tag{44}$$

Then, we can deduce the expectation of Ψ^{k+1}, It is easy to check that the first element in c_2 is greater than the second one.

$$\begin{aligned}
\mathrm{E}_u\left[\Psi^{k+1}\right] =& \mathrm{E}_u\left[\|x_{k+1}-x^*\|^2+\alpha_2\|h_{k+1}-\nabla f(x^*)\|^2\right] \\
&\overset{(44)}{\leq} (1-\eta_2\mu)\|x_k-x^*\|^2-\frac{\eta_2}{L}\|\nabla f(x_k)-\nabla f(x^*)\|^2+\eta_2^2\mathrm{E}_u\left[\|g_k-\nabla f(x^*)\|^2\right] \\
&\quad+\alpha_2\mathrm{E}_u\left[\|h_{k+1}-\nabla f(x^*)\|^2\right] \\
&\overset{(39)+(40)}{\leq} (1-\eta_2\mu)\|x_k-x^*\|^2-\left(\frac{\eta_2}{L}-2(d+3)\eta_2^2-\frac{\alpha_2}{d+2}\right)\|\nabla f(x_k)-\nabla f(x^*)\|^2 \\
&\quad+\left(1-\frac{1}{d+2}+\frac{2\eta_2^2(d+2)}{\alpha_2}\right)\alpha_2\|h_k-\nabla f(x^*)\|^2 \\
&\overset{(21)}{\leq} \left(1-\frac{\mu}{2(3d+7)L}\right)\|x_k-x^*\|^2+\left(1-\frac{1}{2(d+2)}\right)\alpha_2\|h_k-\nabla f(x^*)\|^2 \\
&\leq \max\left\{1-\frac{\mu}{2(3d+7)L},1-\frac{1}{2(d+2)}\right\}\left(\|x_k-x^*\|^2+\alpha_2\|h_k-\nabla f(x^*)\|^2\right) \\
&\overset{(22)}{\leq} c_2^{k+1}\Psi^0.
\end{aligned}$$

Note that during the proof of the theorem, it is easy to check that the first element in $\max\left\{1-\frac{\mu}{2(3d+7)L},1-\frac{1}{2(d+2)}\right\}$ is greater than the second element. □

C.3. Proof of Theorem 4.14

Proof. Firstly, we can also deduce the expectation of $\|x_{k+1}-x^*\|^2$,

$$\begin{aligned}
\mathrm{E}_u\left[\|x_{k+1}-x^*\|^2\right] &\overset{(8)+(20)}{=} \mathrm{E}_u\left[\|\mathrm{Prox}_{\alpha\varphi}\left(x_k-\eta_3 g_k\right)-\mathrm{Prox}_{\alpha\varphi}\left(x^*-\eta_3\nabla f(x^*)\right)\|^2\right]\\
&\overset{(24)}{\leq} \mathrm{E}_u\left[\|x_k-x^*-\eta_3(g_k-\nabla f(x^*))\|_{\mathbf{W}}^2\right]\\
&= \|x_k-x^*\|_{\mathbf{W}}^2-2\eta_3\langle\nabla f(x_k)-\nabla f(x^*),x_k-x^*\rangle_{\mathbf{W}}+\eta_3^2\mathrm{E}_u\left[\|g_k-\nabla f(x^*)\|_{\mathbf{W}}^2\right]\\
&\overset{(4)+(34)}{\leq} \|x_k-x^*\|^2-2\eta_3\left(f(x_k)-f(x^*)-\langle\nabla f(x^*),x_k-x^*\rangle+\frac{\mu}{2}\|x_k-x^*\|^2\right)\\
&\quad+\eta_3^2\mathrm{E}_u\left[\|g_k-\nabla f(x^*)\|_{\mathbf{W}}^2\right]\\
&= (1-\eta_3\mu)\|x_k-x^*\|^2-2\eta_3\left(f(x_k)-f(x^*)-\langle\nabla f(x^*),x_k-x^*\rangle\right)+\eta_3^2\mathrm{E}\left[\|g_k-\nabla f(x^*)\|_{\mathbf{W}}^2\right].
\end{aligned}$$

Because f is $\mathbf{M}$-smooth, we can obtain the next equality by Proposition 2.4

$$f(x_k)-f(x^*)-\langle\nabla f(x^*),x_k-x^*\rangle\geq\frac{1}{2}\|\nabla f(x_k)-\nabla f(x^*)\|_{\mathbf{M}^{-1}}^2. \tag{45}$$

Plugging this into above inequation, we can deduce the expectation of Υ^{k+1},

$$\begin{aligned}
\mathrm{E}_u\left[\Upsilon^{k+1}\right] =& \mathrm{E}_u\left[\|x_{k+1}-x^*\|^2+\alpha_3\|h_{k+1}-\nabla f(x^*)\|^2\right]\\
&\overset{(45)}{\leq}(1-\eta_3\mu)\|x_k-x^*\|^2-\eta_3\|\nabla f(x_k)-\nabla f(x^*)\|_{\mathbf{M}^{-1}}^2+\eta_3^2\mathrm{E}_u\left[\|g_k-\nabla f(x^*)\|_{\mathbf{W}}^2\right]\\
&\quad+\alpha_3\mathrm{E}_u\left[\|h_{k+1}-\nabla f(x^*)\|^2\right]\\
&\overset{(40)+(42)}{\leq}(1-\eta_3\mu)\|x_k-x^*\|^2+\|\nabla f(x_k)-\nabla f(x^*)\|^2_{\left[\left(7\eta_3^2\mathrm{tr}(\mathbf{W})+\frac{\alpha_3}{d+2}\right)\cdot\mathbf{I}-\eta_3\mathbf{M}^{-1}\right]}\\
&\quad+\left(1-\frac{1}{d+2}+\frac{6\eta_3^2\mathrm{tr}(\mathbf{W})}{\alpha_3}\right)\alpha_3\|h_k-\nabla f(x^*)\|^2\\
&\overset{(26)}{\leq}\left(1-\frac{\mu}{19L\mathrm{tr}(\mathbf{W})}\right)\|x_k-x^*\|^2+\left(1-\frac{1}{2(d+2)}\right)\alpha_3\|h_k-\nabla f(x^*)\|^2\\
&\leq\max\left\{1-\frac{\mu}{19L\mathrm{tr}(\mathbf{W})},1-\frac{1}{2(d+2)}\right\}\left(\|x_k-x^*\|^2+\alpha_3\|h_k-\nabla f(x^*)\|^2\right)\\
&\overset{(27)}{\leq}c_3^{k+1}\Upsilon^0.
\end{aligned}$$

Note that during the proof of the theorem, it is easy to check that the first element in $\max\left\{1-\frac{\mu}{19L\mathrm{tr}(\mathbf{W})},1-\frac{1}{2(d+2)}\right\}$ is greater than the second element. □

D. Proof of Main Corollaries

We prove the iteration complexity in the main case in this part. Before we prove several important Corollaries, we introduce an important inequality first.

Lemma D.1. *In general, the variable K represents the number of iterations, the variable x represents a small positive number (0<x<1) . When K is large, we can get the following inequality*

$$(1-x)^K\leq e^{-xK}. \tag{46}$$

Proof. We first construct an auxiliary function,

$$m(x)=1-x-e^{-x}. \tag{47}$$

Then, we can easily calculate the derivative function and get the range of the derivative function

$$m^{'}(x) = e^{-x} - 1 \leq 0. \tag{48}$$

Next, we can obtain an inequality from the derivative function

$$1 - x - e^{-x} = m(x) \leq m(0) = 0. \tag{49}$$

Therefore, the original inequality is proved. □

D.1. Proof of Corollary 4.4

Proof. By Theorem 4.3, we can obtain that

$$\begin{aligned}
&f(x_K) - f(x^*) + \eta_1\alpha_1(d+2)\cdot \mathrm{tr}(\mathbf{M})\left\|h_K\right\|^2 \\
\leq&\Big(1 - \frac{\mu}{40\mathrm{tr}(\mathbf{M})}\Big)^K \Big(f(x_0) - f(x^*) + \eta_1\alpha_1(d+2)\cdot \mathrm{tr}(\mathbf{M})\left\|h_0\right\|^2\Big) \\
\overset{(46)}{\leq}& \exp\Big(-\frac{\mu}{40\mathrm{tr}(\mathbf{M})}K\Big)\Big(f(x_0) - f(x^*) + \eta_1\alpha_1(d+2)\cdot \mathrm{tr}(\mathbf{M})\left\|h_0\right\|^2\Big).
\end{aligned}$$

Thus, in order to achieve ε-suboptimal solution, K is required to be

$$\begin{aligned}
K =& \frac{40\mathrm{tr}(\mathbf{M})}{\mu}\Big(\log\frac{1}{\varepsilon} + \log\Big(f(x_0) - f(x^*) + \eta_1\alpha_1(d+2)\cdot \mathrm{tr}(\mathbf{M})\left\|h_0\right\|^2\Big)\Big) \\
=& \mathcal{O}\Big(\frac{\mathrm{tr}(\mathbf{M})}{\mu}\log\frac{1}{\varepsilon}\Big).
\end{aligned}$$

□

D.2. Proof of Corollary 4.10

Proof. By Theorem 4.9, we can obtain that

$$\begin{aligned}
&\left\|x_K - x^*\right\|^2 + \alpha_2\left\|h_K - \nabla f(x^*)\right\|^2 \\
\leq&\Big(1 - \frac{\mu}{2(3d+7)L}\Big)^K \Big(\left\|x_0 - x^*\right\|^2 + \alpha_2\left\|h_0 - \nabla f(x^*)\right\|^2\Big) \\
\overset{(46)}{\leq}& \exp\Big(-\frac{\mu}{2(3d+7)L}K\Big)\Big(\left\|x_0 - x^*\right\|^2 + \alpha_2\left\|h_0 - \nabla f(x^*)\right\|^2\Big).
\end{aligned}$$

Thus, in order to achieve ε-suboptimal solution, K is required to be

$$\begin{aligned}
K =& \frac{2(3d+7)L}{\mu}\Big(\log\frac{1}{\varepsilon} + \log\Big(\left\|x_0 - x^*\right\|^2 + \alpha_2\left\|h_0 - \nabla f(x^*)\right\|^2\Big)\Big) \\
=& \mathcal{O}\Big(\frac{dL}{\mu}\log\frac{1}{\varepsilon}\Big).
\end{aligned}$$

□

D.3. Proof of Corollary 4.15

Proof. The similar with D.2, in order to achieve ε-suboptimal solution, K is required to be

$$\begin{aligned}
K =& \frac{19L\mathrm{tr}(\mathbf{W})}{\mu}\Big(\log\frac{1}{\varepsilon} + \log\Big(\left\|x_0 - x^*\right\|^2 + \alpha_3\left\|h_0 - \nabla f(x^*)\right\|^2\Big)\Big) \\
=& \mathcal{O}\Big(\frac{L\mathrm{tr}(\mathbf{W})}{\mu}\log\frac{1}{\varepsilon}\Big).
\end{aligned}$$

□

E. Additional experiment

In this section we will test the advantages of our algorithm over the Coordinate sketching version of the SEGA on a real-world data set. We present performance comparison results for a linear regression task with $\varphi(x) = 0$ on the dataset called Appliances Energy Prediction. This data set has 29 features and 19735 records. In the experiment, we properly choose the step sizes of these two algorithms. According to the theoretical results of our algorithm and SEGA, step sizes of these two algorithms should be proportional to $\mathcal{O}(1/\mathrm{tr}(\mathbf{M}))$ and $\mathcal{O}(1/(d\lambda(\mathbf{M})))$, respectively.

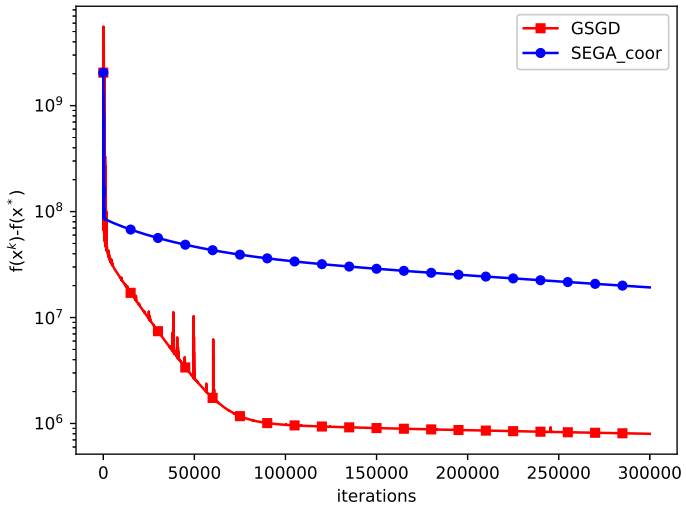


Figure 3. Comparison of both our algorithm and the Coordinate sketching version of the SEGA algorithm with uniform sampling on linear regression task.

We report the experiment result in Figure 3. We can observe that, our algorithm is significantly faster than the Coordinate sketching version of the SEGA. This is because $\mathrm{tr}(\mathbf{M}) \ll dL$ on this task. But the overall convergence speed is slow because the condition number of this task is large. We would like to remind again here that GSGD is better than SEGA when the eigenvalues of the Hessian matrix are very different ($\mathrm{tr}(\mathbf{M}) \ll dL$).