

Accelerated Stochastic ExtraGradient: Mixing Hessian and gradient similarity to reduce communication in distributed and federated learning

D. A. Bylinkin, K. D. Degtyarev, and A. N. Beznosikov

Abstract. Modern realities and trends in learning require more and more generalization ability of models, which leads to an increase in both models and training sample size. It is already difficult to solve such tasks in a single device mode. This is the reason why distributed and federated learning approaches are becoming more popular every day. Distributed computing involves communication between devices, which requires solving two key problems: efficiency and privacy. One of the most well-known approaches to combat communication costs is to exploit the similarity of local data. Both Hessian similarity and homogeneous gradients have been studied in the literature, but separately. In this paper we combine both of these assumptions in analyzing a new method that incorporates the ideas of using data similarity and clients sampling. Moreover, to address privacy concerns, we apply the technique of additional noise and analyze its impact on the convergence of the proposed method. The theory is confirmed by training on real datasets.

Bibliography: 45 titles.

Keywords: ASEG, distributed learning, federated learning, communication costs, Hessian similarity, homogeneous gradients, technique of additional noise.

Contents

1. Introduction	940
2. Related works	941
2.1. Distributed methods under similarity	941
2.2. Privacy preserving	942
2.3. Stochastic optimization	942
3. Our contribution	943
4. Stochastic version of Accelerated ExtraGradient	943

The work was done in the Laboratory of Federated Learning Problems of the Ivannikov Institute for System Programming of the Russian Academy of Sciences and supported by the Ministry of Science and Higher Education of the Russian Federation under appendix no. 2 to agreement no. 075-03-2024-214.

AMS 2020 *Mathematics Subject Classification.* Primary 68T05, 68T07.

5. Theoretical analysis	944
6. Analysis of the subproblem	948
6.1. The SGD approach	949
6.2. The SVRG approach	949
6.3. The loopless approach	950
7. Numerical experiments	951
8. Conclusion	958
Appendix A. Proof of Lemma 1	958
Appendix B. Proof of Theorem 2	963
Appendix C. Proof of Corollaries 1 and 2	964
Appendix D. ASEG for the convex case	966
Bibliography	969

1. Introduction

Empirical risk minimization is a key concept in supervised machine learning [34]. This paradigm is used to train a wide range of models, such as linear and logistic regression [40], support vector machines [30], tree-based models [18], neural networks [33], and many more. In essence, this is an optimization of the parameters based on the average losses in data. By minimizing the empirical risk, the model aims to generalize well to unseen samples and make accurate predictions. In order to enhance the overall predictive performance and effectively tackle advanced tasks, it is imperative to train on large datasets. By leveraging such datasets during the training process, we aim to maximize the generalization capabilities of models and equip them with the robustness and adaptability required to excel in challenging real-world scenarios [14]. The richness and variety of samples in large datasets provides the depth needed to learn complex patterns, relationships, and nuances, enabling them to make accurate and reliable predictions across a wide range of inputs and conditions. As datasets grow in size and complexity, traditional optimization algorithms may struggle to converge in a reasonable amount of time. This increases the interest in developing distributed methods [42]. Formally, the objective function in the minimization problem is a finite sum over nodes, each containing its own possibly private dataset. In the most general form we have

$$\min_{x \in \mathbb{R}^d} \left[r(x) = \frac{1}{M} \sum_{m=1}^M r_m(x) = \frac{1}{M} \sum_{m=1}^M \left(\frac{1}{n_m} \sum_{j=1}^{n_m} \ell(x, y_j^m) \right) \right], \quad (1)$$

where M refers to the number of nodes/clients/machines/devices, n_m is the size of the m th local dataset, x is the vector representation of a model using d features, y_j^m is the j th data point on the m th node, and ℓ is the loss function. We consider the most computationally powerful device that can communicate with all others to be a server. Without loss of generality we assume that r_1 is responsible for the server, and the other r_m for other nodes.

This formulation of the problem entails a number of challenges. In particular, transferring information between devices incurs communication overhead [36]. This can result in increased resource utilization, which affects the overall efficiency of the training process. There are different ways to deal with communication costs [29], [23], [3], [9]. One of the key approaches is exploiting the similarity of local data and

hence the corresponding losses. One way to formally express it is to use Hessian similarity [39]:

$$\|\nabla^2 r_m(x) - \nabla^2 r(x)\| \leq \delta, \quad (2)$$

where $\delta > 0$ measures the degree of similarity between the corresponding Hessian matrices. The goal is to build methods with communication complexity depending on δ because it is known that $\delta \sim 1/\sqrt{n}$ for non-quadratic losses and $\delta \sim 1/n$ for quadratic ones [19]. Here n is the size of the training set. Hence the more data on nodes, the more statistically ‘similar’ the losses are and the less communication is needed. The basic method in this case is **Mirror Descent** with an appropriate Bregman divergence [35], [19]:

$$x_{k+1} = \arg \min_{x \in \mathbb{R}^d} \left(r_1(x) + \langle \nabla r(x_k) - \nabla r_1(x_k), x \rangle + \frac{\delta}{2} \|x - x_k\|^2 \right).$$

The computation of ∇r requires communication with clients. Our goal is to reduce it. The key idea is that by offloading nodes, we move the most computationally demanding work to the server. Each node performs one computation per iteration (to collect ∇r on the server), while server solves a local subproblem and accesses the local oracle ∇r_1 much more frequently. While this approach provides improvements in communication complexity over a simple distributed version of **Gradient Descent**, there is room for improvement in various aspects. In particular, the problem of constructing an optimal algorithm was open for a long time [35], [5], [45], [38]. It was recently solved by the introduction of **Accelerated ExtraGradient** [24].

Despite the fact that there is an optimal algorithm for communication, there are still open challenges: since each dataset is stored on its own node, it is expensive to compute the full gradient. Moreover, by transmitting it, we lose the ability to keep the procedure private [43], since in this case it is possible to recover local data. One of the best known cheap ways to protect privacy is to add noise to the information communicated [1]. To summarize the above, in this paper, we are interested in stochastic modifications of **Accelerated ExtraGradient** [24] which would be particularly resistant to the imposition of noise on the transmitted packages.

2. Related works

Since our goal is to investigate stochastic versions of algorithms that exploit data similarity (in particular, to speed up the computation of local gradients and to protect privacy), we give a brief literature review of related areas.

2.1. Distributed methods under similarity. In 2014 the scalable **DANE** algorithm was introduced [35]. It was a novel Newton-type method, which used $\mathcal{O}((\delta/\mu)^2 \log(1/\varepsilon))$ communication rounds for μ -strongly convex quadratic losses by taking a step appropriate to the geometry of the problem. This work can be considered as the first that started to apply Hessian similarity in the distributed setting. A year later $\Omega(\sqrt{\delta/\mu} \log(1/\varepsilon))$ was found to be a lower bound for communication rounds under Hessian similarity [5]. Since then, extensive work has been done to achieve the outlined optimal complexity. The idea of **DANE** was developed by adding momentum acceleration to the step, thereby reducing the complexity for quadratic loss to $\mathcal{O}(\sqrt{\delta/\mu} \log(1/\varepsilon))$ [45].

Later, **Accelerated SONATA** achieved $\mathcal{O}(\sqrt{\delta/\mu} \log^2(1/\varepsilon))$ for non-quadratic losses by the **Catalyst** acceleration [39]. Finally, there is **Accelerated ExtraGradient**, which achieves $\mathcal{O}(\sqrt{\delta/\mu} \log(1/\varepsilon))$ communication complexity by using ideas of Nesterov acceleration, sliding, and **ExtraGradient** [24].

Other ways to introduce the definition of similarity are also found in the literature. In [22] the authors study the local technique in the distributed and federated optimization, and

$$\sigma_{\text{sim}}^2 = \frac{1}{M} \sum_{m=1}^M \|\nabla r_m(x_*)\|^2$$

is taken as a measure of gradient similarity. Here x_* is the solution to problem (1). This kind of data heterogeneity is also examined in [44]. The case of gradient similarity of the form $\|\nabla r_m(x) - \nabla r(x)\| \leq \delta$ is discussed in [17]. Moreover, several papers use gradient similarity in the analysis of distributed saddle point problems [6], [10], [8], [7].

2.2. Privacy preserving. A brief overview of the issue of privacy in modern optimization is presented in [15]. In this paper, we are interested in differential privacy. In essence, differential privacy guarantees that results of a data analysis process remain largely unaffected by the addition or removal of any single individual's data. This is achieved by introducing randomness or noise into the data in a controlled manner, so that the statistical properties of the data remain intact while protecting the privacy of individual samples [2]. Differential privacy involves randomized perturbations of the information sent between nodes and the server [32]. In this case it is important to design an algorithm that is resistant to transformations.

2.3. Stochastic optimization. The simplest stochastic methods, for example, SGD, face a common problem: the approximation of the gradient does not tend to zero when searching for the optimum [16]. The solution to this issue appeared in 2014 with the invention of variance reduction technique. There are two main approaches to its implementation: **SAGA** [13] and **SVRG** [20]. The first maintains a gradient history and the second introduces a reference point at which the full gradient is computed, and updates it infrequently. The methods listed above do not use acceleration. In recent years a number of accelerated variance reduction methods emerged [31], [26], [25], [4].

The idea of stochastic methods with variance reduction also carries over to distributed setup. In particular, stochasticity helps to break through lower bounds under the Hessian similarity condition. For example, methods that use variance reduction to implement client sampling are constructed. **Catalyzed SVRP** enjoys $\mathcal{O}((1 + M^{-1/4} \cdot \sqrt{\delta/\mu}) \log^2(1/\varepsilon))$ complexity in terms of the amount of server-node communication [21]. This method combines stochastic proximal point evaluation, client sampling, variance reduction, and acceleration via the **Catalyst** framework [28]. However, it requires strong convexity of each r_m . It is possible to move to weaker assumptions on the local functions. There is **AccSVRS**, which achieves $\mathcal{O}((1 + M^{-1/4} \cdot \sqrt{\delta/\mu}) \log(1/\varepsilon))$ communication complexity [27]. In addition, variance reduction turns out to be useful to design methods with compression. There is **Three Pillars Algorithm**, which achieves $\mathcal{O}((1 + M^{-1/2} \cdot (\delta/\mu)) \log(1/\varepsilon))$ for variational inequalities [11].

The weakness of the accelerated variance reduction methods is that they require a small step size inversely proportional to $\sqrt{M}$ [11], [21], [27]. Thus, one has to ‘pay’ for convergence to an arbitrary ε -solution by increasing the iteration complexity of the method. However, there is the gain, mentioned already, in terms of the number of server-node communications. Meanwhile, accelerated versions of the classic SGD do not have this flaw [41]. In some regimes the accuracy that can be achieved without the use of variance reduction can be sufficient: these include distributed problems under similarity conditions.

3. Our contribution

- **New method.** We propose a new method based on the deterministic **Accelerated ExtraGradient** (AEG) [24]. Unlike AEG, we add the ability to sample computational nodes. This saves a significant amount of runtime. As mentioned above, sampling was already implemented in **Three Pillars Algorithm** [11], **Catalyzed SVRP** [21], and **AccSVRS** [27]. However, this was done using SVRG-like approaches. We are based instead on SGD, which does not require decreasing the step size as the number of computational nodes increases.

- **Privacy via noise.** We consider stochasticity not only in the choice of nodes, but also in the additive noise in the transmission of information from the client to the server. This makes it much harder to steal private data in the case of an attack on the algorithm.

- **Theoretical analysis.** We show that Hessian similarity implies gradient one with an additional term. Therefore, it is possible to estimate the variance associated with sampling nodes more optimistically. Thus, in some cases ASEG converges quite accurately before getting ‘stuck’ near the solution.

- **Analysis of a subproblem.** We consider different approaches to the solution of the subproblem that occurs on a server. Our analysis allows us to estimate the number of iterations required to ensure convergence of the method.

- **Experiments.** We validate the theory constructed by numerical experiments on several real datasets.

4. Stochastic version of Accelerated ExtraGradient

Algorithm 1 (ASEG) is a modification of AEG [24]. In one iteration the server communicates with clients twice. In the first round random machines are selected (Line 6). They perform a computation and send the gradient to the server (Line 15). Then the server solves a local subproblem in Line 16 and the solution is used to compute the stochastic (extra)gradient at the second round of communication (Line 17 and Line 26). This is followed by a gradient step with momentum in Line 27.

Let us summarize briefly the main differences from AEG. Line 16 of Algorithm 1 uses a gradient over local batches of nodes, rather than over all of them, as done in the baseline method. Moreover, the case where local gradients are noisy is supported. This is needed for privacy purposes. Similarly in Line 27. Note that we consider the most general case where different devices are involved in solving the subproblem and performing the gradient step.

ASEG requires $\Theta(2B)$ communication rounds (Line 15 and Line 26), where B is the number of nodes involved in the iteration. Unlike our method, AEG does

Algorithm 1: Accelerated Stochastic ExtraGradient (ASEG)

```

1: Input:  $x^0 = x_f^0 \in \mathbb{R}^d$ 
2: Parameters:  $\tau \in (0, 1]$ ,  $\eta, \theta, \alpha > 0$ ,  $N \in \{1, 2, \dots\}$ ,  $B \in \mathbb{N}$ 
3: for  $k = 0, 1, 2, \dots, N - 1$  do:
4:   for server do:
5:      $x_g^k = \tau x^k + (1 - \tau)x_f^k$ 
6:     Generate a set of  $|I| = B$  numbers uniformly from  $[2, \dots, n]$ 
7:     Send to each device  $m$  one bit  $b_k^m$ : 1 if  $m \in I$ , 0 otherwise
8:   end for
9:   for each device  $m$  in parallel do:
10:    if  $b_k^m = 1$  then
11:      Send  $\nabla r_m(x_g^k, \xi_m^k)$  to server
12:    end if
13:   end for
14:   for server do:
15:      $s_k = \frac{1}{B} \sum_{m \in I} (\nabla r_m(x_g^k, \xi_m^k) - \nabla r_1(x_g^k))$ 
16:      $x_f^{k+1} \approx \arg \min_{x \in \mathbb{R}^d} \left[ \bar{A}_\theta^k(x) := \langle s_k, x - x_g^k \rangle + \frac{1}{2\theta} \|x - x_g^k\|^2 + r_1(x) \right]$ 
17:     Generate set  $|I| = B$  numbers uniformly from  $[1, \dots, n]$ 
18:     Send to each device  $m$  one bit  $c_k^m$ : 1 if  $m \in I$ , 0 otherwise
19:   end for
20:   for each device  $m$  in parallel do:
21:    if  $c_k^m = 1$  then
22:      Send  $\nabla r_m(x_f^{k+1}, \xi_m^{k+1/2})$  to server
23:    end if
24:   end for
25:   for server do:
26:      $t_k = \frac{1}{B} \sum_{m \in I} \nabla r_m(x_f^{k+1}, \xi_m^{k+1/2})$ 
27:      $x^{k+1} = x^k + \eta \alpha (x_f^{k+1} - x^k) - \eta t_k$ 
28:   end for
29: end for
30: Output:  $x^N$ 

```

$\Theta(2M - 2)$ rounds per iteration. AccSVRS does $\Theta(2M)$ rounds on average. Thus, in experiments either the difference between the methods is invisible, or AEG loses. ASEG deals with the local gradient $\nabla r_m(x, \xi)$, which can have the structure

$$\nabla r_m(x, \xi) = \nabla r_m(x) + \xi,$$

where, for example, $\xi \in N(0, \sigma^2)$ or $\xi \in U(-c, c)$. This is the additive noise overlay technique, mentioned previously and popular in federated learning.

5. Theoretical analysis

Before we delve into analysis, let us introduce some definitions and assumptions to build the convergence theory.

In this paper complexity is understood in the sense of the number of server-node communications. This means that we consider how many times the server exchanges vectors with clients. In this case, a single contact is counted as one communication.

We work with functions of a class widely used in the literature.

Definition 1. We say that a function $f(x): \mathbb{R}^d \rightarrow \mathbb{R}$ is μ -strongly convex on $\mathbb{R}^d$ if

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

If $\mu = 0$, we call f a convex function.

Definition 2. We say that a function $f(x): \mathbb{R}^d \rightarrow \mathbb{R}$ is L -smooth on $\mathbb{R}^d$ if

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

Definition 3. We say that $f_m(\cdot)$ is δ -related to $f(\cdot)$ if

$$\|\nabla^2 f_m(x) - \nabla^2 f(x)\| \leq \delta \quad \forall x \in \mathbb{R}^d.$$

We assume a strongly convex optimization problem (1) with possibly non-convex components.

Assumption 1. The function r is μ -strongly convex, and r_1 is convex and L_1 -smooth.

This assumption does not require the convexity of local functions. Indeed, δ -relatedness is a sufficiently strong property to allow them to be non-convex. In this paper we are interested in the case where $\mu < \delta$, since $\mu \ll \delta \ll L$ for large datasets in practice [38].

Proposition 1. Consider the δ -relatedness of $r_m(\cdot)$ to $r(\cdot)$ (Definition 3) for each $m \in [1, M]$. In this case we have

$$\|\nabla r_m(x) - \nabla r(x)\|^2 \leq \sigma_{\text{sim}}^2 + 2\delta^2 \|x - x_*\|^2,$$

where $\sigma_{\text{sim}}^2 = 2 \max_{m \in [1, M]} \|\nabla r_m(x_*)\|^2$ and x_* is the solution of problem (1).

Proof. It is obvious that Definition 3 implies the smoothness of $r_m - r$:

$$\|\nabla(r_m - r)(x) - \nabla(r_m - r)(x_*)\| \leq \delta \|x - x_*\| \quad \forall m \in [1, M], \quad x \in \mathbb{R}^d.$$

Since x_* is the optimum, it follows that $\nabla r(x_*) = 0$. It is known that $\|x - y\|^2 \leq 2\|x\|^2 + 2\|y\|^2$. We obtain

$$\|\nabla r_m(x) - \nabla r(x)\|^2 \leq 2\|\nabla r_m(x_*)\|^2 + 2\delta^2 \|x - x_*\|^2. \quad \square$$

Thus, from the similarity of the Hessians follows the similarity of the gradients with some correction. Let us introduce a more general assumption on gradient similarity.

Assumption 2. For all $m \in [1, M]$ Definition 3 is satisfied and the following inequality holds:

$$\|\nabla r_m(x) - \nabla r(x)\|^2 \leq \sigma_{\text{sim}}^2 + 2\delta^2 \|x - x_*\|^2 \cdot \zeta,$$

where $\zeta \geq 0$.

Assumption 2 is a generalization of Proposition 1, where the correction term is multiplied by some constant ζ . Formally, it can take any non-negative value, but it is important to analyze only two cases corresponding to different degrees of similarity, $\zeta = 0$ and $\zeta = 1$. Indeed, any non-negative values of ζ give asymptotically the same complexity.

A stochastic oracle is available for the function at each node. We impose the following standard assumption.

Assumption 3. For all $m \in \{1, \dots, n\}$ the stochastic oracle $\nabla r_m(x, \xi)$ is unbiased and has a bounded variance:

$$\mathbb{E}_\xi[\nabla r_m(x, \xi) \mid x] = \nabla r_m(x), \quad \mathbb{E}_\xi[\|\nabla r_m(x, \xi) - \nabla r_m(x)\|^2 \mid x] \leq \sigma_{\text{noise}}^2 \quad \forall x \in \mathbb{R}^d.$$

Remark 1. The Hessian similarity of r_m and r for $m \neq 1$ is only used to formulate Proposition 1 and is not needed anywhere else. The next proofs require only that the server expresses the average nature of the data, while the data at nodes can be heterogeneous.

We want to obtain a criterion that guarantees the convergence of Algorithm 1. The obvious idea is to try to find one for the new algorithm by analogy with how it was done in the non-stochastic case [24].

Lemma 1. Consider Algorithm 1 for problem (1) under Assumptions 1, 2, and 3 with the following tuning:

$$\alpha = \frac{\mu}{3}, \quad \eta \leq \frac{1}{3\alpha}, \quad \eta \leq \frac{\theta}{3\tau}, \quad \theta \leq \frac{1}{3\delta}, \quad (3)$$

and let x_f^{k+1} satisfy

$$\|\nabla \bar{A}_\theta^k(x_f^{k+1})\|^2 \leq \frac{9\delta^2}{11} \left\| x_g^k - \arg \min_{x \in \mathbb{R}^d} \bar{A}_\theta^k(x) \right\|^2. \quad (4)$$

Then the following inequality holds:

$$\begin{aligned} & \mathbb{E} \left[\frac{1}{\eta} \|x^{k+1} - x^*\|^2 + \frac{2}{\tau} [r(x_f^{k+1}) - r(x^*)] \right] \\ & \leq \left(1 - \alpha\eta + \frac{12\theta\delta^2\eta}{B} \cdot \zeta \right) \mathbb{E} \left[\frac{1}{\eta} \|x^k - x^*\|^2 \right] + (1 - \tau) \mathbb{E} \left[\frac{2}{\tau} [r(x_f^k) - r(x^*)] \right] \\ & \quad + \mathbb{E} \left[\left(\frac{12\theta\delta^2}{B\tau} - \frac{\mu}{6} \right) \cdot \zeta (\|x_f^k - x_*\|^2 + \|x_f^{k+1} - x_*\|^2) \right] + \frac{4\theta}{B\tau} \sigma^2. \end{aligned}$$

See the proof in Appendix A. First we deal with the case under the gradient similarity condition ($\zeta = 0$), since it removes two additional constraints on the parameters and makes the analysis simpler.

The following theorem is obtained by rewriting the result of Lemma 1 for a finer tuning of the parameters. We take $\mu < \delta$ into account and find that

$$1 - \frac{\sqrt{\mu\theta}}{3} \geq 1 - \frac{1}{3}.$$

Theorem 1. *In Lemma 1 consider Assumption 2 for $\zeta = 0$ and adjust the parameters:*

$$\alpha = \frac{\mu}{3}, \quad \eta = \min\left\{\frac{1}{3\alpha}, \frac{\theta}{3\tau}\right\}, \quad \tau = \frac{\sqrt{\mu\theta}}{3}, \quad \theta \leq \frac{1}{3\delta}. \quad (5)$$

Then

$$\mathbb{E}[\Phi_{k+1}] \leq \left(1 - \frac{\sqrt{\mu\theta}}{3}\right) \mathbb{E}[\Phi_k] + \frac{4\theta\sigma^2}{B},$$

where

$$\Phi_k = \frac{\tau}{\eta} \|x^k - x^*\|^2 + 2[r(x_f^k) - r(x^*)].$$

The theorem asserts convergence only to some neighbourhood of the solution. To guarantee convergence with an arbitrary accuracy, a suitable parameter θ must be chosen. We propose a corollary of Theorem 1.

Corollary 1. *Consider the conditions of Theorem 1. For an appropriate tuning of θ Algorithm 1 requires*

$$\mathcal{O}\left(\frac{B}{M} \sqrt{\frac{\delta}{\mu}} \log \frac{1}{\varepsilon} + \frac{\sigma_{\text{sim}}^2 + \sigma_{\text{noise}}^2}{\mu M \varepsilon}\right) \text{ communications}$$

to reach an arbitrary ε -solution.

See the proof in Appendix C. Note that the linear part of the estimate obtained reproduces the convergence of AEG. However, for example, for $B = 1$ an iteration of ASEG costs $\Theta(2)$ communications instead of $\Theta(2M - 2)$. Thus, our method requires significantly less communication to achieve the value determined by the sublinear term. Moreover, as mentioned before, in Catalyzed SVRP and AccSVRS estimates an additional $M^{-1/4}$ multiplier arises due to the use of variance reduction. Comparing $M^{-1/4} \cdot \sqrt{\delta/\mu}$ with $BM^{-1} \cdot \sqrt{\delta/\mu}$, we notice the superiority of ASEG in terms of the approach taken to measure communication complexity.

Note that if the noise of the gradients sent to the server is weak ($\sigma_{\text{noise}} = 0$) and the functions are absolutely similar in the gradient sense ($\sigma_{\text{sim}} = 0$), then ASEG has $M^{3/4}/B$ times less complexity in terms of number of communications.

In the case of a convex objective it is not possible to implement the method under Assumption 2 for $\zeta \neq 0$, since the correction terms arising cannot be eliminated. Under Assumption 2 for $\zeta = 0$ we propose a modification of ASEG for convex objective. In this case we need

$$\mathcal{O}\left(\frac{B}{M} \frac{\sqrt{\delta} \|x_0 - x_*\|}{\sqrt{\varepsilon}} + \frac{(\sigma_{\text{sim}}^2 + \sigma_{\text{noise}}^2) \|x_0 - x_*\|^2}{M \varepsilon^2}\right) \text{ communications}$$

to reach an arbitrary ε -solution. See Appendix D.

The case $\zeta = 1$ is more complex and requires more fine-tuned analysis.

Theorem 2. *In Lemma 1 consider Assumption 2 for $\zeta = 1$ and tune the parameters as follows:*

$$\alpha = \frac{\mu}{3}, \quad \eta = \min\left\{\frac{1}{3\alpha}, \frac{\theta}{3\tau}\right\}, \quad \tau = \sqrt{\mu\theta}, \quad \theta \leq \min\left\{\frac{1}{3\delta}, \frac{\mu^3 B^2}{5184 \delta^4}\right\}. \quad (6)$$

Then

$$\mathbb{E}[\Phi_{k+1}] \leq \left(1 - \frac{\sqrt{\mu\theta}}{18}\right) \mathbb{E}[\Phi_k] + \frac{4\theta\sigma^2}{B},$$

where

$$\Phi_k = \frac{\tau}{\eta} \|x^k - x^*\|^2 + 2[r(x_f^k) - r(x^*)].$$

See the proof in Appendix B.

Corollary 2. *Consider the conditions of Theorem 2. For an appropriate tuning of θ Algorithm 1 requires*

$$\mathcal{O}\left(\frac{B}{M} \sqrt{\frac{\delta}{\mu}} \log \frac{1}{\varepsilon} + \frac{\delta^2}{\mu^2 M} \log \frac{1}{\varepsilon} + \frac{\sigma_{\text{sim}}^2 + \sigma_{\text{noise}}^2}{\mu M \varepsilon}\right) \text{ communications}$$

to reach an arbitrary ε -solution.

See the proof in Appendix C. It can be seen from this proof that Assumption 2 for $\zeta = 1$ leads to worse estimates. Nevertheless, the optimal communication complexity can be achieved if we choose a sufficiently large B which equalizes two linear terms (at least $72\delta^{3/2}/\mu^{3/2}$). It comes from the equality

$$\frac{1}{3\delta} = \frac{\mu^3 B^2}{5184 \delta^4}.$$

Thus, even in problems with a poor ratio of δ to μ , if the number of available nodes is large enough, ASEG outperforms AEG, Catalyzed SVRP, and AccSVRS.

6. Analysis of the subproblem

In this section we focus on the quantity

$$\min_{x \in \mathbb{R}^d} \left[\bar{A}_\theta^k(x) := \langle s_k, x - x_g^k \rangle + \frac{1}{2\theta} \|x - x_g^k\|^2 + r_1(x) \right], \quad (7)$$

defined in Line 16 of Algorithm 1. Note that problem (7) is solved on the server and does not affect the communication complexity. Obviously,

$$\|\nabla \bar{A}_\theta^k(x) - \nabla \bar{A}_\theta^k(y)\| = \left\| \frac{x - y}{\theta} + \nabla r_1(x) - \nabla r_1(y) \right\| \leq \left(\frac{1}{\theta} + L_1 \right) \|x - y\|.$$

Moreover,

$$\nabla^2 \bar{A}_\theta^k(x) = \frac{1}{\theta} I + \nabla^2 r_1(x) \succ 0.$$

Thus, subproblem is L_A -smooth, where $L_A = 1/\theta + L_1$ and $1/\theta$ strongly convex.

Despite the fact that problem (7) does not affect the number of communications, we would still like to spend less server time solving it. An obvious solution is to use stochastic approaches.

6.1. The SGD approach. The basic approach is to use **Stochastic Gradient Descent** (SGD) for the strongly convex smooth objective. In this case the convergence rate of the subproblem by argument norm is given by the following theorem [37].

Theorem 3. *Let Assumptions 1 and 3 hold for problem (7). Consider T iterations of SGD with step size $\gamma \leq 1/(2L_A)$. Then the following estimate holds:*

$$\mathbb{E} \left[\|x_T - \arg \min_{x \in \mathbb{R}^d} \bar{A}_\theta^k(x) \|^2 \mid s_k \right] \leq \left(1 - \frac{\gamma}{\theta} \right)^T \|x_0 - \arg \min_{x \in \mathbb{R}^d} \bar{A}_\theta^k(x) \|^2 + \theta \gamma \sigma_1^2.$$

Using the smoothness of the subproblem we observe that

$$\mathbb{E} [\| \nabla \bar{A}_\theta^k(x_T) \|^2 \mid s_k] \leq L_A^2 \mathbb{E} [\|x_T - \arg \min_{x \in \mathbb{R}^d} \bar{A}_\theta^k(x) \|^2 \mid s_k].$$

Let us formulate a consequence of Theorem 3 for this case.

Corollary 3. *Under Assumptions 1 and 3 consider problem (7) and T iterations of SGD. Then the following estimate holds:*

$$\mathbb{E} [\| \nabla \bar{A}_\theta^k(x_T) \|^2 \mid s_k] \leq L_A^2 \left(1 - \frac{\gamma}{\theta} \right)^T \left\| x_0 - \arg \min_{x \in \mathbb{R}^d} \bar{A}_\theta^k(x) \right\|^2 + L_A^2 \theta \gamma \sigma_1^2.$$

Our interest is to choose a step such that it is possible to avoid getting stuck in a neighbourhood of a solution of the subproblem. In accordance with [37], we note the following.

Theorem 4. *Under Assumptions 1 and 3, consider the problem (7) to be solved using SGD. There exist decreasing step sizes $\gamma_t \leq 1/(2L_A)$ such that for T iterations we obtain*

$$\begin{aligned} \mathbb{E} [\| \nabla \bar{A}_\theta^k(x_T) \|^2 \mid s_k] &\leq 64\theta L_A^3 \left\| x_0 - \arg \min_{x \in \mathbb{R}^d} \bar{A}_\theta^k(x) \right\|^2 \exp \left\{ -\frac{T}{4L_A\theta} \right\} \\ &\quad + \frac{36L_A^2\sigma_1^2\theta}{T}. \end{aligned}$$

In this way we can achieve the optimal convergence rate of SGD. However, we would like to enjoy linear convergence. The idea is to use variance reduction methods.

6.2. The SVRG approach. Another approach to the solution of the subproblem is to use variance reduction methods, for example, **Stochastic Variance Reduced Gradient** (SVRG). Let us take x_g^k as a starting point and look at the first epoch. It is known from [16] that

$$\mathbb{E} \left[\bar{A}_\theta^k \left(\frac{1}{J} \sum_{j=1}^J x_j \right) - \bar{A}_\theta^k(x_*) \right] \leq \frac{2(\mu^{-1} + 2J\gamma^2 L_A)}{J\gamma(1 - 2\gamma L_A)} \mathbb{E} [\bar{A}_\theta^k(x_g^k) - \bar{A}_\theta^k(x_*)],$$

where J is the epoch size. The next point after the algorithm finishes the current

epoch is $x = \frac{1}{J} \sum_{j=1}^J x_j$. We can rewrite for the T th epoch:

$$\mathbb{E} [\bar{A}_\theta^k(\hat{x}_T) - \bar{A}_\theta^k(x_*)] \leq \left(\frac{2(\mu^{-1} + 2J\gamma^2 L_A)}{J\gamma(1 - 2\gamma L_A)} \right)^T \mathbb{E} [\bar{A}_\theta^k(x_g^k) - \bar{A}_\theta^k(x_*)].$$

Then we write out the strong convexity property:

$$\mathbb{E}[\bar{A}_\theta^k(x_g^k) - \bar{A}_\theta^k(x_*)] \leq \frac{1}{2\mu} \mathbb{E}[\|\nabla \bar{A}_\theta^k(x_g^k)\|^2].$$

Using the L_A -smoothness of $\bar{A}_\theta^k$ we obtain

$$\begin{aligned} \mathbb{E}[\|\nabla \bar{A}_\theta^k(\hat{x}_T)\|^2] &\leq 2L\mathbb{E}[\bar{A}_\theta^k(\hat{x}_T) - \bar{A}_\theta^k(x_*)] \\ &\leq \frac{L_A}{\mu} \left(\frac{2(\mu^{-1} + 2J\gamma^2 L_A)}{J\gamma(1 - 2\gamma L_A)} \right)^T \mathbb{E}[\|\nabla \bar{A}_\theta^k(x_g^k)\|^2]. \end{aligned}$$

Thus, for a suitable combination of the epoch size and step, we expect to obtain linear convergence in the subproblem, which is consistent with the experimental results.

6.3. The loopless approach. In this subsection we fill in some gaps in the analysis of the subproblem. The decision criterion for ASEG is (4). Assume that we have obtained

$$\|\nabla A_\theta^k(x_T)\|^2 = \mathcal{O}\left(\frac{1}{T^\alpha}\right), \quad \alpha > 0. \quad (8)$$

We use this to obtain a lower bound on T . The important question is whether it is possible to take fewer steps so that the criterion necessary for the convergence of the whole algorithm is met. In other words, we want to check whether the loopless version of ASEG turns out to be better than the usual one, as in the case of SVRG and KATYUSHA [25].

We are interested in comparing the deterministic and random choices of T . Let ξ be a random variable describing the number of steps; ξ takes positive discrete values. Let A be the number of steps for the subproblem of ASEG defined deterministically and satisfying (4). Assume that

$$\frac{1}{\xi^\alpha} \leq \frac{1}{A^\alpha}. \quad (9)$$

Then from (8) we know that

$$\|\nabla A_\theta^k(x_\xi)\|^2 \leq \|\nabla A_\theta^k(x_A)\|^2,$$

where x_ξ is the point obtained in ξ solver steps. Obviously, the converse is also true. In other words, condition (9) is a criterion for ξ to solve the initial subproblem. At the same time, we want to minimize $\mathbb{E}\xi$ to take as few steps as possible.

Proposition 2. *If $\mathbb{E}[\xi] < A$, then $\mathbb{E}\left[\frac{1}{\xi^\alpha}\right] > \frac{1}{A^\alpha}$.*

Proof. Let p_i be the probability to choose $\xi = i$. Then we have

$$\mathbb{E}\left[\frac{1}{\xi^\alpha}\right] = \sum_{i=1}^T p_i \frac{1}{i^\alpha}.$$

Since $\alpha > 0$, we see that $\phi(i) = 1/i^\alpha$ is a convex function. Using Jensen's inequality we obtain

$$\mathbb{E}\left[\frac{1}{\xi^\alpha}\right] \geq \frac{1}{(\mathbb{E}[\xi])^\alpha} > \frac{1}{A^\alpha}. \quad \square$$

Thus, in the worst case, if the expectation of ξ is less than the deterministic number of iterations, the criterion cannot be met and there is no probabilistic method of choosing the number of iterations that is better than the deterministic one.

It is worth noting that the same can be said about the pair of **SVRG** and **L-SVRG**. It is obvious that **SVRG** could be reduced to **L-SVRG** if the number of iterations before updating the gradient is considered as a random variable. Thus, the epoch size J is from the geometric distribution, and one can compare which algorithm gives a more accurate solution after performing the inner loop. For the **SVRG**-rate we have [16]

$$\mathbb{E}[\bar{A}_\theta^k(x_{J(t+1)}) - \bar{A}_\theta^k(x_*)] \leq \left(\frac{1}{\eta\gamma(1-2L\gamma)J} + \frac{2L\gamma}{1-2L\gamma} \right) \mathbb{E}[\bar{A}_\theta^k(x_{Jt}) - \bar{A}_\theta^k(x_*)], \quad (10)$$

where γ is the step and $\eta > 0$ is some constant. From Proposition 2 and inequality (10) we obtain

$$\frac{1}{\eta\gamma(1-2L\gamma)} \mathbb{E}\left[\frac{1}{J}\right] + \frac{2L\gamma}{1-2L\gamma} \leq \frac{1}{\eta\gamma(1-2L\gamma)J} + \frac{2L\gamma}{1-2L\gamma}.$$

Thus, in the worst case loopless versions converges no faster than ordinary ones.

7. Numerical experiments

To support the theoretical analysis and insights, we solve both the logistic regression problem

$$r(x) = \frac{1}{M} \sum_{m=1}^M \frac{1}{n} \sum_{j=1}^n \ln(1 + \exp\{-b_j^m \langle x, a_j^m \rangle\}) + \lambda \|x\|^2$$

and the quadratic regression problem

$$r(x) = \frac{1}{M} \sum_{m=1}^M \frac{1}{n} \sum_{j=1}^n (\langle x, a_j^m \rangle - b_j^m)^2 + \lambda \|x\|^2.$$

Here x refers to the parameters of the model, a_j^m is the feature description, and b_j^m is the corresponding label. To avoid overfitting, we use l_2 -regularization. The penalty parameter λ is set to be $L/100$, where L is the smoothness constant (see Definition 2). We use **SVRG** and **SARAH** as the subproblem solvers in Algorithm 1. In all tests the original datasets are merged into batches $\{r_b\}_{b=1}^{M+N}$. A function $r_1(x)$ is created from part of the acquired ones:

$$r_1(x) = \frac{1}{N} \sum_{b=1}^N r_b(x), \quad N < M.$$

Due to Hessian similarity we have the following expression for the smoothness constant:

$$\|\nabla^2 r(x) - \nabla^2 r_1(x)\| \leq \delta,$$

where

$$r(x) = \frac{1}{M} \sum_{i=1}^M r_i(x).$$

Then for the quadratic problem we have

$$\delta = \left\| \frac{1}{N} \sum_{i=1}^N A_i - \frac{1}{M} \sum_{j=1}^M A_j \right\|.$$

For logistic regression δ is estimated by enumerating 100 points in a neighbourhood of the solution. In both cases the value obtained is multiplied by $3/2$. In order to demonstrate correctly how the theory works in practice, we solve problem (1) for various parameters δ , μ , M , and N . We consider different tabular datasets from LibSVM [12]: **Mushrooms** (binary classification, 112 features, 8 124 objects), **W8A** (binary classification, 300 features, 49 749 objects), and **A9A** (binary classification, 123 features, 32 561 objects). They allow for a wide range of different problem parameters and are most commonly used to illustrate the convergence of methods in the literature.

- We investigate how the size of the batches affects the convergence of **ASEG** on problems with different parameters. Fig. 1 shows the dependence of untuned **ASEG** convergence on the batch size. In the first experiment the method on 50 nodes solves the problem for $\mu = 0.105$ and $\delta = 1.45$. According to Corollary 2, one should choose $B > 72(\delta/\mu)^{3/2} \approx 3695$ to obtain the optimal complexity. That is, for any size of the batch there is a non-optimal rate. In the second experiment the method on 200 nodes solves the problem for $\mu = 0.06$ and $\delta = 0.07$. In this case **ASEG** has the guaranteed optimal complexity for $B > 57$ and converges to rather small values of the criterion.

- We compare **ASEG** with **AccSVRS** (Figs. 2 and 3). 50 nodes are available for **Mushrooms** and 200 nodes are available for **A9A**. If the relation between μ , δ , and B yields the optimal complexity according to Corollary 2, then due to similarity there is an optimistic estimate for the batching noise providing a significant gain over **AccSVRS**. Otherwise, it is worth using full batching and obtaining a less appreciable gain.

- In **ASEG** stability experiments (Figs. 4–7) it is tested how white noise in each $\nabla r_m(x)$ affects convergence. It can be seen that increasing the randomness does not worsen convergence. This can be explained by the fact that convergence is mainly determined by the accuracy of the solver chosen.

- **ASEG** experiments test how changing the epoch size changes the convergence of **ASEG** and the convergence in the subproblem (Line 16 of Algorithm 1) for the **SVRG** and **SARAH** solvers (Fig. 8–15).

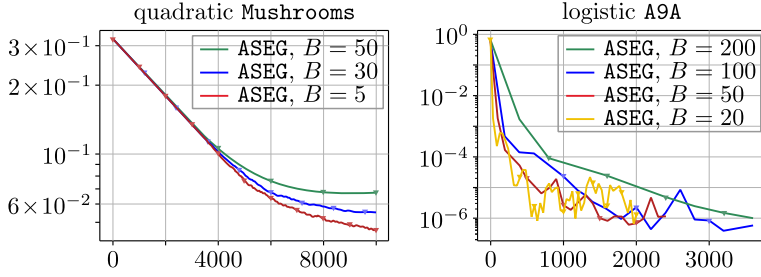


Figure 1. ASEG with different batch sizes. (The number of communications on the horizontal axis and the value of $\|\nabla f(x)\|^2$ on the vertical axis.)

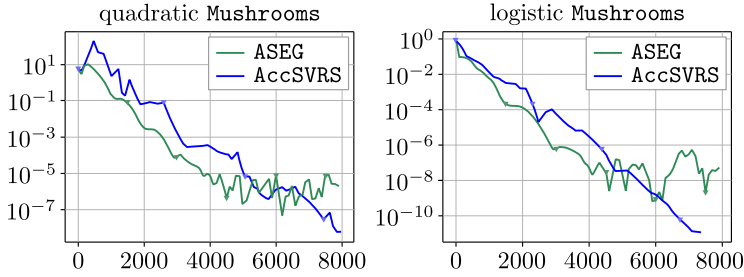


Figure 2. ASEG vs. AccSVRS on data with $\delta = 10.15$ for the quadratic problem and $\delta = 1.45$ for the logistic one. (The number of communications on the horizontal axis and the value of $\|\nabla f(x)\|^2$ on the vertical axis.)

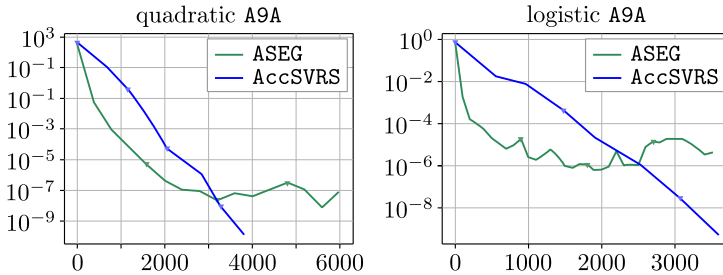


Figure 3. ASEG vs. AccSVRS on data with $\delta = 0.064$ for the quadratic problem and $\delta = 0.061$ for the logistic one. (The number of communications on the horizontal axis and the value of $\|\nabla f(x)\|^2$ on the vertical axis.)

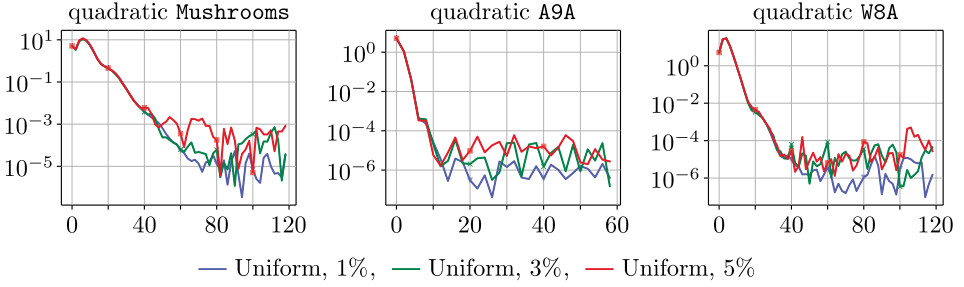


Figure 4. The stability of **ASEG** with the **SVRG**-solver and uniform noise. The comparison is made by solving the quadratic problem on $M = 200$ nodes. (The number of communication rounds on the horizontal axis and the value of $\|\nabla f(x)\|^2$ on the vertical axis.)

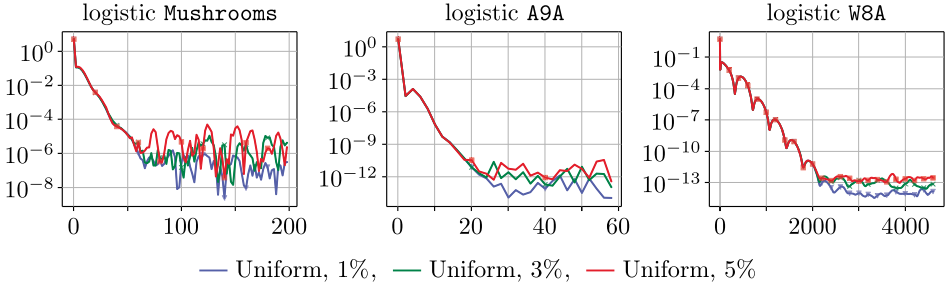


Figure 5. The stability of **ASEG** with the **SVRG**-solver and uniform noise. The comparison is made by solving the logistic problem on $M = 200$ nodes. (The number of communication rounds on the horizontal axis and the value of $\|\nabla f(x)\|^2$ on the vertical axis.)

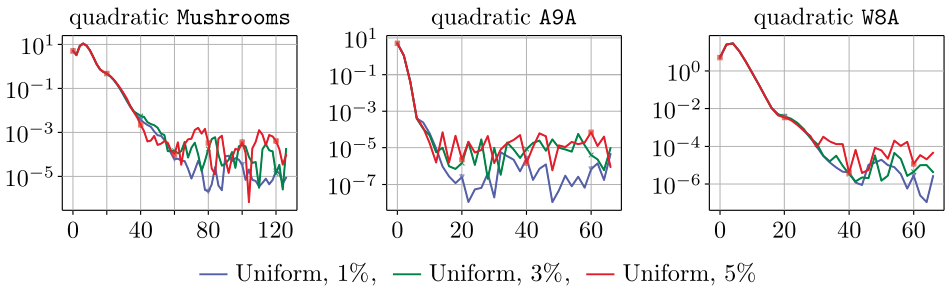


Figure 6. The stability of **ASEG** with the **SARAH**-solver and uniform noise. The comparison is made by solving the quadratic problem on $M = 200$ nodes. (The number of communication rounds on the horizontal axis and the value of $\|\nabla f(x)\|^2$ on the vertical axis.)

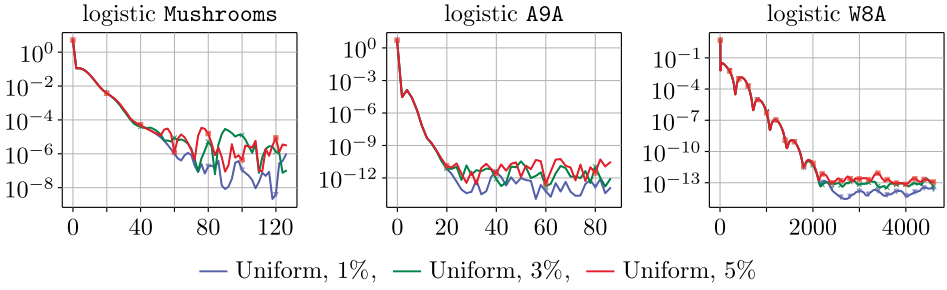


Figure 7. The stability of ASEG with the SARAH-solver and uniform noise. The comparison is made by solving logistic problem on $M = 200$ nodes. (The number of communication rounds on the horizontal axis and the value of $\|\nabla f(x)\|^2$ on the vertical axis.)

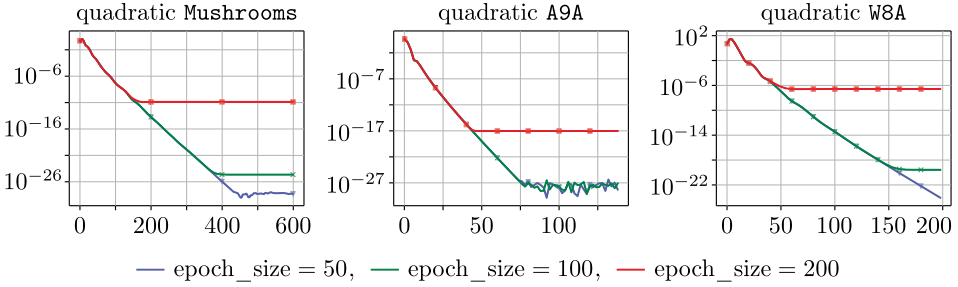


Figure 8. ASEG with the SVRG-solver. The comparison is made by solving the quadratic problem on $M = 200$ nodes with different epoch sizes of the solver. (The number of communication rounds on the horizontal axis and the value of $\|\nabla f(x)\|^2$ on the vertical axis.)

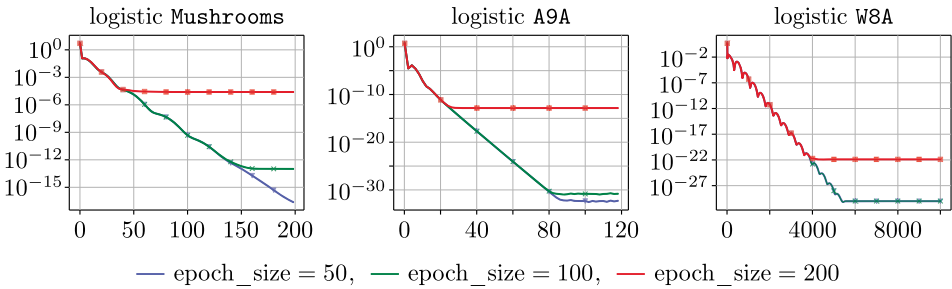


Figure 9. ASEG with the SVRG-solver. The comparison is made by solving the logistic problem on $M = 200$ nodes with different epoch sizes of the solver. (The number of communication rounds on the horizontal axis and the value of $\|\nabla f(x)\|^2$ on the vertical axis.)

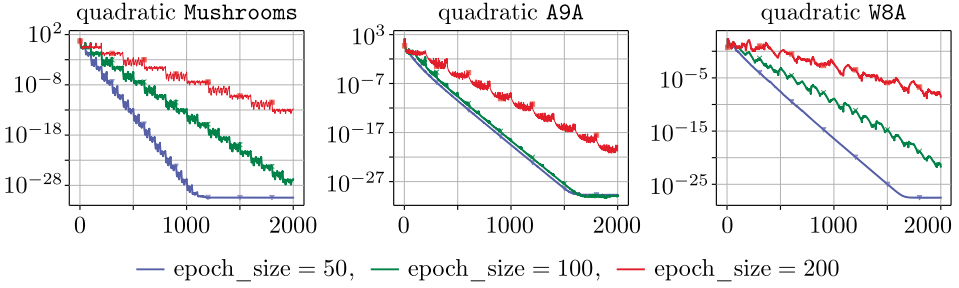


Figure 10. The subproblem with the **SVRG**-solver. The comparison is made by solving quadratic problem on $M = 200$ nodes with different epoch sizes of the solver. (The number of server iterations on the horizontal axis and the value of $\|\nabla f(x)\|^2$ on the vertical axis.)

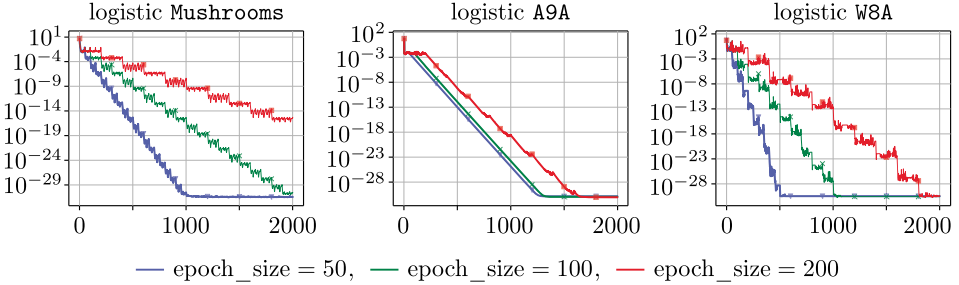


Figure 11. The subproblem with the **SVRG**-solver. The comparison is made by solving logistic problem on $M = 200$ nodes with different epoch sizes of the solver. (The number of server iterations on the horizontal axis and the value of $\|\nabla f(x)\|^2$ on the vertical axis.)

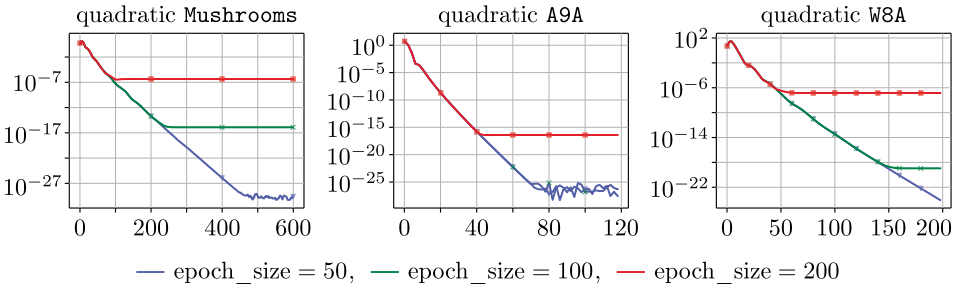


Figure 12. **ASEG** with the **SARAH**-solver. The comparison is made by solving quadratic problem on $M = 200$ nodes with different epoch sizes of the solver. (The number of communication rounds on the horizontal axis and the value of $\|\nabla f(x)\|^2$ on the vertical axis.)

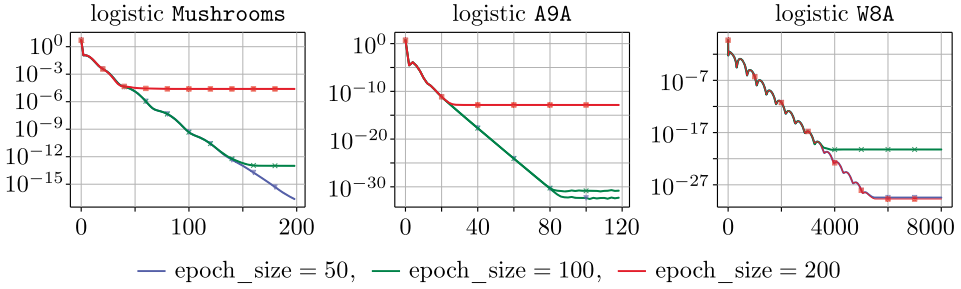


Figure 13. ASEG with the SARAH-solver. The comparison is made by solving logistic problem on $M = 200$ nodes with different epoch sizes of the solver. (The number of communication rounds on the horizontal axis and the value of $\|\nabla f(x)\|^2$ on the vertical axis.)

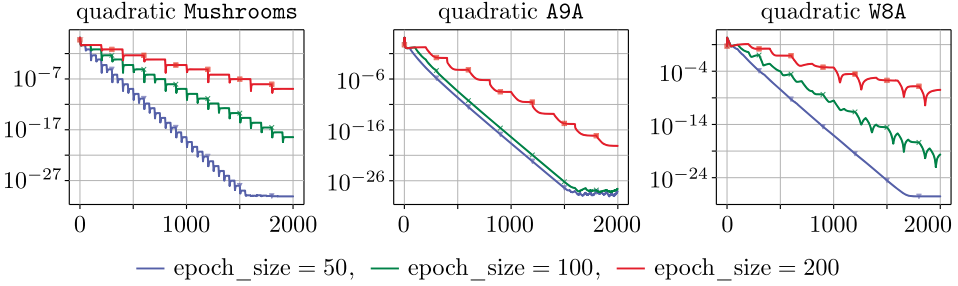


Figure 14. The subproblem with the SARAH-solver. The comparison is made by solving the quadratic problem on $M = 200$ nodes with different epoch sizes of the solver. (The number of server iterations on the horizontal axis and the value of $\|\nabla f(x)\|^2$ on the vertical axis.)

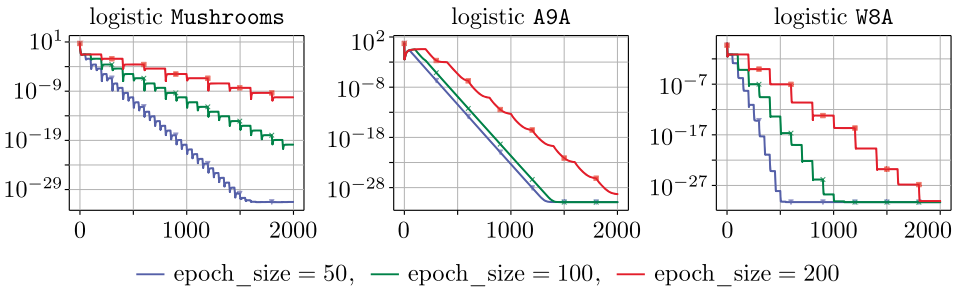


Figure 15. The subproblem with the SARAH-solver. The comparison is made by solving the logistic problem on $M = 200$ nodes with different epoch sizes of the solver. (The number of server iterations on the horizontal axis and the value of $\|\nabla f(x)\|^2$ on the vertical axis.)

8. Conclusion

Based on the SGD-like approach, we have adapted **Accelerated ExtraGradient** to solve Federated Learning problems by adding two stochasticities: sampling of computational nodes and the application of noise to local gradients. Numerical experiments show that for small values of δ/μ or a sufficiently large number of available nodes Algorithm 1 (**ASEG**) is significantly superior to accelerated variance reduction methods (Fig. 3), which is consistent with the theory (Corollary 1). If the problem parameters are not ‘good’ enough, there is a $[\delta^2/(\mu^2 M)] \cdot \log(1/\varepsilon)$ term in the communication complexity of batched **ASEG** (Corollary 2). Nevertheless, even in this case the superiority in the number of communications remains (Figs. 2 and 3). Experiments also show the robustness of our method against noise (Figs. 4–7). The server subproblem has also been studied numerically (Figs. 10, 11, 14, and 15) and theoretically. The analysis shows the inefficiency of the randomly choice of the number of solver iterations for the internal subproblem (Proposition 2).

Appendix A. Proof of Lemma 1

In all appendices we consider

$$r_1(x) = q(x) \quad \text{and} \quad p(x) = r(x) - r_1(x).$$

Expectations are implied in the context of the proofs.

In this section we prove Lemma 1. First, we need the following assertion.

Lemma 2. *Consider Algorithm 1. Let θ be defined as in Lemma 1: $\theta \leq 1/(3\delta)$. Then under Assumptions 1 and 3 the following inequality holds for all $\bar{x} \in \mathbb{R}^d$:*

$$\begin{aligned} & \mathbb{E}[2\langle \bar{x} - x_g^k, t_k \rangle \mid x^k, x_f^{k+1}] \\ & \leq 2[r(\bar{x}) - r(x_f^{k+1})] - \mu\|x_f^{k+1} - \bar{x}\|^2 - \theta\|\nabla r(x_f^{k+1})\|^2 \\ & \quad + \frac{11\theta}{3} \left(\|\nabla \bar{A}_\theta(x_f^{k+1})\|^2 - \frac{9\delta^2}{11} \left\| x_g^k - \arg \min_{x \in \mathbb{R}^d} \bar{A}_\theta(x) \right\|^2 \right) \\ & \quad + 3\theta\|\nabla p(x_g^k) - s_k\|^2. \end{aligned} \tag{11}$$

Proof. Using the μ -strong convexity of $r(x)$ we obtain

$$\begin{aligned} \mathbb{E}[2\langle \bar{x} - x_g^k, t_k \rangle \mid x^k, x_f^{k+1}] &= 2\langle \bar{x} - x_g^k, \mathbb{E}[t_k \mid x^k, x_f^{k+1}] \rangle \\ &= 2\langle \bar{x} - x_g^k, \nabla r(x_f^{k+1}) \rangle \\ &= 2\langle \bar{x} - x_f^{k+1}, \nabla r(x_f^{k+1}) \rangle + 2\langle x_f^{k+1} - x_g^k, \nabla r(x_f^{k+1}) \rangle \\ &\leq 2[r(\bar{x}) - r(x_f^{k+1})] - \mu\|x_f^{k+1} - \bar{x}\|^2 \\ & \quad + 2\langle x_f^{k+1} - x_g^k, \nabla r(x_f^{k+1}) \rangle \\ &= 2[r(\bar{x}) - r(x_f^{k+1})] - \mu\|x_f^{k+1} - \bar{x}\|^2 \\ & \quad + 2\theta\langle \theta^{-1}(x_f^{k+1} - x_g^k), \nabla r(x_f^{k+1}) \rangle. \end{aligned}$$

By the properties of the norm we can write

$$\begin{aligned} \left\| \frac{x_f^{k+1} - x_g^k}{\theta} + \nabla r(x_f^{k+1}) \right\|^2 &= \frac{1}{\theta^2} \|x_f^{k+1} - x_g^k\|^2 + \|\nabla r(x_f^{k+1})\|^2 \\ &\quad + \frac{2}{\theta} \langle x_f^{k+1} - x_g^k, \nabla r(x_f^{k+1}) \rangle. \end{aligned}$$

Thus,

$$\begin{aligned} \mathbb{E}[2\langle \bar{x} - x_g^k, t_k \rangle \mid x^k, x_f^{k+1}] &= 2[r(\bar{x}) - r(x_f^{k+1})] - \mu \|x_f^{k+1} - \bar{x}\|^2 - \frac{1}{\theta} \|x_f^{k+1} - x_g^k\|^2 \\ &\quad - \theta \|\nabla r(x_f^{k+1})\|^2 + \theta \|\theta^{-1}(x_f^{k+1} - x_g^k) + \nabla r(x_f^{k+1})\|^2. \end{aligned}$$

The definition of $\bar{A}_\theta^k(x)$ and the δ -Lipschitzness of ∇p give

$$\begin{aligned} \mathbb{E}[2\langle \bar{x} - x_g^k, t_k \rangle \mid x^k, x_f^{k+1}] &\leq 2[r(\bar{x}) - r(x_f^{k+1})] - \mu \|x_f^{k+1} - \bar{x}\|^2 - \frac{1}{\theta} \|x_f^{k+1} - x_g^k\|^2 \\ &\quad + \theta \|\nabla \bar{A}_\theta^k(x_f^{k+1}) + \nabla p(x_f^{k+1}) - \nabla p(x_g^k) + \nabla p(x_g^k) - s_k\|^2 \\ &\quad - \theta \|\nabla r(x_f^{k+1})\|^2. \end{aligned}$$

Note that

$$\|a + b + c\|^2 \leq 3\|a\|^2 + 3\|b\|^2 + 3\|c\|^2,$$

and Hessian similarity (Definition 3) implies that

$$\|\nabla p(x_f^{k+1}) - \nabla p(x_g^k)\|^2 \leq \delta^2 \|x_f^{k+1} - x_g^k\|^2.$$

Thus,

$$\begin{aligned} \mathbb{E}[2\langle \bar{x} - x_g^k, t_k \rangle \mid x^k, x_f^{k+1}] &\leq 2[r(\bar{x}) - r(x_f^{k+1})] - \mu \|x_f^{k+1} - \bar{x}\|^2 - \frac{1}{\theta} \|x_f^{k+1} - x_g^k\|^2 \\ &\quad - \theta \|\nabla r(x_f^{k+1})\|^2 + \theta \|\nabla \bar{A}_\theta^k(x_f^{k+1})\|^2 \\ &\quad + 3\theta \delta^2 \|x_f^{k+1} - x_g^k\|^2 + 3\theta \|\nabla p(x_g^k) - s_k\|^2 \\ &= 2[r(\bar{x}) - r(x_f^{k+1})] - \mu \|x_f^{k+1} - \bar{x}\|^2 \\ &\quad - \frac{1}{\theta} (1 - 3\theta^2 \delta^2) \|x_f^{k+1} - x_g^k\|^2 - \theta \|\nabla r(x_f^{k+1})\|^2 \\ &\quad + 3\theta \|\nabla \bar{A}_\theta^k(x_f^{k+1})\|^2 + 3\theta \|\nabla p(x_g^k) - s_k\|^2. \end{aligned}$$

For $\theta \leq 1/(3\delta)$ we have

$$\begin{aligned} \mathbb{E}[2\langle \bar{x} - x_g^k, t_k \rangle \mid x^k, x_f^{k+1}] &\leq 2[r(\bar{x}) - r(x_f^{k+1})] - \mu \|x_f^{k+1} - \bar{x}\|^2 - \frac{2}{3\theta} \|x_f^{k+1} - x_g^k\|^2 \\ &\quad - \theta \|\nabla r(x_f^{k+1})\|^2 + 3\theta \|\nabla \bar{A}_\theta^k(x_f^{k+1})\|^2 + 3\theta \|\nabla p(x_g^k) - s_k\|^2 \\ &\leq 2[r(\bar{x}) - r(x_f^{k+1})] - \mu \|x_f^{k+1} - \bar{x}\|^2 - \theta \|\nabla r(x_f^{k+1})\|^2 \\ &\quad + 3\theta \|\nabla \bar{A}_\theta^k(x_f^{k+1})\|^2 + 3\theta \|\nabla p(x_g^k) - s_k\|^2 \\ &\quad - \frac{1}{3\theta} \left\| x_g^k - \arg \min_{x \in \mathbb{R}^d} \bar{A}_\theta^k(x) \right\|^2 + \frac{2}{3\theta} \left\| x_f^{k+1} - \arg \min_{x \in \mathbb{R}^d} \bar{A}_\theta^k(x) \right\|^2. \end{aligned}$$

Here we have used the fact that

$$-\|a - c\|^2 \leq -\frac{1}{2}\|a - b\|^2 + \|b - c\|^2.$$

One can observe that $A_\theta^k(x)$ is $(1/\theta)$ -strongly convex. Hence

$$\begin{aligned} \mathbb{E}[2\langle \bar{x} - x_g^k, t_k \rangle \mid x^k, x_f^{k+1}] &\leq 2[r(\bar{x}) - r(x_f^{k+1})] - \mu\|x_f^{k+1} - \bar{x}\|^2 - \theta\|\nabla r(x_f^{k+1})\|^2 \\ &\quad + 3\theta\|\nabla \bar{A}_\theta^k(x_f^{k+1})\|^2 + 3\theta\|\nabla p(x_g^k) - s_k\|^2 \\ &\quad - \frac{1}{3\theta}\left\|x_g^k - \arg \min_{x \in \mathbb{R}^d} \bar{A}_\theta^k(x)\right\|^2 + \frac{2\theta}{3}\|\nabla \bar{A}_\theta^k(x_f^{k+1})\|^2. \end{aligned}$$

Now we are ready to construct an expression with the convergence criterion:

$$\begin{aligned} \mathbb{E}[2\langle \bar{x} - x_g^k, t_k \rangle \mid x^k, x_f^{k+1}] &\leq 2[r(\bar{x}) - r(x_f^{k+1})] - \mu\|x_f^{k+1} - \bar{x}\|^2 - \theta\|\nabla r(x_f^{k+1})\|^2 \\ &\quad + \frac{11\theta}{3}\left(\|\nabla \bar{A}_\theta^k(x_f^{k+1})\|^2 - \frac{1}{11\theta^2}\left\|x_g^k - \arg \min_{x \in \mathbb{R}^d} \bar{A}_\theta^k(x)\right\|^2\right) + 3\theta\|\nabla p(x_g^k) - s_k\|^2 \\ &= 2[r(\bar{x}) - r(x_f^{k+1})] - \mu\|x_f^{k+1} - \bar{x}\|^2 - \theta\|\nabla r(x_f^{k+1})\|^2 \\ &\quad + \frac{11\theta}{3}\left(\|\nabla \bar{A}_\theta^k(x_f^{k+1})\|^2 - \frac{9\delta^2}{11}\left\|x_g^k - \arg \min_{x \in \mathbb{R}^d} \bar{A}_\theta^k(x)\right\|^2\right) + 3\theta\|\nabla p(x_g^k) - s_k\|^2. \quad \square \end{aligned}$$

Let us move on to the proof of Lemma 1.

Lemma 3 (Lemma 1). *Under Assumptions 1, 2, and 3, condition (4), and the tuning (3) we obtain*

$$\begin{aligned} \mathbb{E}\left[\frac{1}{\eta}\|x^{k+1} - x^*\|^2 + \frac{2}{\tau}[r(x_f^{k+1}) - r(x^*)]\right] \\ \leq \left(1 - \alpha\eta + \frac{12\theta\delta^2\eta}{B} \cdot \zeta\right)\mathbb{E}\left[\frac{1}{\eta}\|x^k - x^*\|^2\right] + (1 - \tau)\mathbb{E}\left[\frac{2}{\tau}[r(x_f^k) - r(x^*)]\right] \\ + \mathbb{E}\left[\left(\frac{12\theta\delta^2}{B\tau} - \frac{\mu}{6}\right) \cdot \zeta(\|x_f^k - x^*\|^2 + \|x_f^{k+1} - x^*\|^2)\right] + \frac{4\theta}{B\tau}\sigma^2. \end{aligned}$$

Proof. Using Line 27 of Algorithm 1 we write

$$\begin{aligned} \frac{1}{\eta}\|x^{k+1} - x^*\|^2 &= \frac{1}{\eta}\|x^k - x^*\|^2 + \frac{2}{\eta}\langle x^{k+1} - x^k, x^k - x^* \rangle + \frac{1}{\eta}\|x^{k+1} - x^k\|^2 \\ &= \frac{1}{\eta}\|x^k - x^*\|^2 + 2\alpha\langle x_f^{k+1} - x^k, x^k - x^* \rangle - 2\langle t_k, x^k - x^* \rangle \\ &\quad + \frac{1}{\eta}\|x^{k+1} - x^k\|^2 \\ &= \frac{1}{\eta}\|x^k - x^*\|^2 + \alpha\|x_f^{k+1} - x^*\|^2 - \alpha\|x^k - x^*\|^2 \\ &\quad - \alpha\|x_f^{k+1} - x^k\|^2 - 2\langle t_k, x^k - x^* \rangle + \frac{1}{\eta}\|x^{k+1} - x^k\|^2; \end{aligned}$$

here we have used the properties of the norm to write out the inner product. Line 5 of Algorithm 1 gives

$$\begin{aligned} \frac{1}{\eta} \|x^{k+1} - x^*\|^2 &= \left(\frac{1}{\eta} - \alpha\right) \|x^k - x^*\|^2 + \alpha \|x_f^{k+1} - x^*\|^2 + \frac{1}{\eta} \|x^{k+1} - x^k\|^2 \\ &\quad - \alpha \|x_f^{k+1} - x^k\|^2 + 2\langle t_k, x^* - x_g^k \rangle + \frac{2(1-\tau)}{\tau} \langle t_k, x_f^k - x_g^k \rangle. \end{aligned}$$

Using (11) for $\bar{x} = x^*$ and $\bar{x} = x_f^k$ we obtain

$$\begin{aligned} \mathbb{E} \left[\frac{1}{\eta} \|x^{k+1} - x^*\|^2 \mid x^k, x_f^{k+1} \right] &\leq \left(\frac{1}{\eta} - \alpha\right) \|x^k - x^*\|^2 + \alpha \|x_f^{k+1} - x^*\|^2 \\ &\quad + \mathbb{E} \left[\frac{1}{\eta} \|x^{k+1} - x^k\|^2 \mid x^k, x_f^{k+1} \right] - \alpha \|x_f^{k+1} - x^k\|^2 \\ &\quad + 2[r(x^*) - r(x_f^{k+1})] - \mu \|x_f^{k+1} - x^*\|^2 \\ &\quad + \frac{2(1-\tau)}{\tau} [r(x_f^k) - r(x_f^{k+1})] - \frac{\theta}{\tau} \|\nabla r(x_f^{k+1})\|^2 \\ &\quad - \mu \|x_f^{k+1} - x_f^k\|^2 + \frac{3\theta}{\tau} \|\nabla p(x_g^k) - s_k\|^2 \\ &\quad + \frac{11\theta}{3\tau} \left(\|\nabla \bar{A}_\theta^k(x_f^{k+1})\|^2 - \frac{9\delta^2}{11} \left\| x_g^k - \arg \min_{x \in \mathbb{R}^d} \bar{A}_\theta^k(x) \right\|^2 \right). \end{aligned}$$

The last expression

$$\frac{11\theta}{3\tau} \left(\|\nabla \bar{A}_\theta^k(x_f^{k+1})\|^2 - \frac{9\delta^2}{11} \left\| x_g^k - \arg \min_{x \in \mathbb{R}^d} \bar{A}_\theta^k(x) \right\|^2 \right),$$

can be neglected in view of condition (4). Consider the expectation separately:

$$\begin{aligned} \mathbb{E} \left[\frac{1}{\eta} \|x^{k+1} - x^k\|^2 \mid x^k, x_f^{k+1} \right] &= \frac{1}{\eta} \mathbb{E} [\|\eta\alpha(x_f^{k+1} - x^k) - \eta t_k\|^2 \mid x^k, x_f^{k+1}] \\ &\leq 3\eta\alpha^2 \|x_f^{k+1} - x^k\|^2 + 3\eta \|\nabla r(x_f^{k+1})\|^2 \\ &\quad + 3\eta \mathbb{E} [\|\nabla r(x_f^{k+1}) - t_k\|^2 \mid x^k, x_f^{k+1}]. \end{aligned}$$

Here we have added and subtracted $\eta \nabla r(x_f^{k+1})$ under the sign of norm. After rearranging the expressions, we obtain

$$\begin{aligned} \mathbb{E} \left[\frac{1}{\eta} \|x^{k+1} - x^*\|^2 \mid x^k, x_f^{k+1} \right] &\leq \left(\frac{1}{\eta} - \alpha\right) \|x^k - x^*\|^2 + \left(\alpha - \frac{\mu}{3}\right) \|x_f^{k+1} - x^*\|^2 \\ &\quad + \alpha(3\eta\alpha - 1) \|x_f^{k+1} - x^k\|^2 + \frac{3\theta}{\tau} \|\nabla p(x_g^k) - s_k\|^2 \\ &\quad + \frac{2(1-\tau)}{\tau} [r(x_f^k) - r(x^*)] - \frac{2}{\tau} [r(x_f^{k+1}) - r(x^*)] \\ &\quad + 3\eta \|\nabla r(x_f^{k+1}) - t_k\|^2 - \frac{\mu}{3} \|x_f^{k+1} - x^*\|^2 \\ &\quad - \frac{\mu}{3} \|x_f^{k+1} - x^*\|^2 - \mu \|x_f^{k+1} - x_f^k\|^2 \\ &\quad + \left(3\eta - \frac{\theta}{\tau}\right) \|\nabla r(x_f^{k+1})\|^2. \end{aligned}$$

The choice of α , η , and τ as defined by (3) gives

$$\begin{aligned} \mathbb{E} \left[\frac{1}{\eta} \|x^{k+1} - x^*\|^2 \mid x^k, x_f^{k+1} \right] &\leq \left(\frac{1}{\eta} - \alpha \right) \|x^k - x^*\|^2 + \frac{2(1-\tau)}{\tau} [r(x_f^k) - r(x^*)] \\ &\quad - \frac{2}{\tau} [r(x_f^{k+1}) - r(x^*)] + \frac{3\theta}{\tau} \|\nabla p(x_g^k) - s_k\|^2 \\ &\quad + \frac{\theta}{\tau} \|\nabla r(x_f^{k+1}) - t_k\|^2 - \frac{\mu}{3} \|x_f^{k+1} - x_*\|^2 \\ &\quad - \frac{\mu}{3} \|x_f^{k+1} - x_*\|^2 - \mu \|x_f^{k+1} - x_f^k\|^2. \end{aligned}$$

Note that

$$-\frac{\mu}{3} \|x_f^{k+1} - x_*\|^2 \leq -\frac{\mu}{6} \|x_f^k - x_*\|^2 + \frac{\mu}{3} \|x_f^{k+1} - x_f^k\|^2.$$

Here we need the following technical lemma.

Lemma 4. *Let $p(x) = r(x) - r_1(x)$ and*

$$s_k = \frac{1}{B} \sum_{i=1}^B (\nabla r_{m_i}(x_g^k, \xi) - \nabla r_1(x_g^k)).$$

Then

$$\mathbb{E}_{\xi, i} [\|s_k - \nabla p(x_g^k)\|^2] \leq \frac{2(\sigma_{\text{sim}}^2 + \sigma_{\text{noise}}^2)}{B} + \frac{4\delta^2}{B} \|x_f^{k+1} - x_*\|^2 \cdot \zeta.$$

Proof. Let us apply Young's inequality:

$$\begin{aligned} \mathbb{E}_{\xi, i} [\|s_k - \nabla p(x_g^k)\|^2] &= \mathbb{E}_{\xi, i} \left[\left\| \nabla r(x_g^k) - \frac{1}{B} \sum_{i=1}^B \nabla r_{m_i}(x_g^k, \xi_i) \pm \frac{1}{B} \sum_{i=1}^B \nabla r_{m_i}(x_g^k) \right\|^2 \right] \\ &\leq 2\mathbb{E}_i \left[\left\| \nabla r(x_g^k) - \frac{1}{B} \sum_{i=1}^B \nabla r_{m_i}(x_g^k) \right\|^2 \right] \\ &\quad + \frac{2}{B^2} \mathbb{E}_{i, \xi} \left\| \sum_{i=1}^B (\nabla r_{m_i}(x_g^k) - \nabla r_{m_i}(x_g^k, \xi_i)) \right\|^2. \end{aligned}$$

Next we look at the expression given by the second term:

$$\begin{aligned} \frac{2}{B^2} \sum_{i=1}^B \|\nabla r_{m_i}(x_g^k) - \nabla r_{m_i}(x_g^k, \xi_i)\|^2 \\ + \frac{2}{B^2} \sum_{\substack{1 \leq i, j \leq B \\ i \neq j}} \langle \nabla r_{m_i}(x_g^k) - \nabla r_{m_i}(x_g^k, \xi_i), \nabla r_{m_j}(x_g^k) - \nabla r_{m_j}(x_g^k, \xi_j) \rangle. \end{aligned}$$

Let us use the tower property and take the ξ -expectation first. Since m_i and m_j are independent, we can separately take the expectations of the factors in the second

summand. Then only the first summand remains, which is estimated based on Assumption 2. Using Assumption 2 we obtain

$$\mathbb{E}_{\xi,i} [\|s_k - \nabla p(x_g^k)\|^2] \leq \frac{2(\sigma_{\text{sim}}^2 + \sigma_{\text{noise}}^2)}{B} + \frac{4\delta^2}{B} \|x_f^{k+1} - x_*\|^2 \cdot \zeta. \quad \square$$

We return to the proof of Lemma 1. Set

$$\sigma^2 = 2(\sigma_{\text{sim}}^2 + \sigma_{\text{noise}}^2).$$

Taking the expectation and using (4) we have

$$\begin{aligned} & \mathbb{E} \left[\frac{1}{\eta} \|x^{k+1} - x^*\|^2 + \frac{2}{\tau} [r(x_f^{k+1}) - r(x^*)] \right] \\ & \leq \mathbb{E} \left[\left(\frac{1}{\eta} - \alpha + \frac{12\theta\delta^2}{B} \cdot \zeta \right) \|x^k - x^*\|^2 \right] \\ & \quad + \mathbb{E} \left[\frac{2(1-\tau)}{\tau} [r(x_f^k) - r(x^*)] \right] \\ & \quad + \mathbb{E} \left[\left(\frac{12\theta\delta^2}{B\tau} - \frac{\mu}{6} \right) \|x_f^k - x_*\|^2 \cdot \zeta \right] \\ & \quad + \mathbb{E} \left[\left(\frac{4\theta\delta^2}{B\tau} - \frac{\mu}{3} \right) \|x_f^{k+1} - x_*\|^2 \cdot \zeta \right] \\ & \quad + \frac{4\theta}{B\tau} \sigma^2, \end{aligned}$$

which completes the proof of the lemma. $\square$

Appendix B. Proof of Theorem 2

Theorem 5 (Theorem 2). *In Lemma 1 choose $\zeta = 1$ and the tuning (6). Then*

$$\mathbb{E}[\Phi_{k+1}] \leq \left(1 - \frac{\sqrt{\mu\theta}}{18} \right) \mathbb{E}[\Phi_k] + \frac{4\theta}{B},$$

where

$$\Phi_k = \frac{\tau}{\eta} \|x^k - x^*\|^2 + 2[r(x_f^k) - r(x^*)].$$

Proof. We consider the possible values of η one by one.

- Let $\eta = 1/\mu$. In this case we have

$$1 - \eta\alpha + \frac{12\eta\theta\delta^2}{B} \leq \frac{2}{3} + \frac{12\delta^2}{B\mu} \frac{\mu B\tau}{72\delta^2} \leq \frac{5}{6} = 1 - \frac{1}{6}.$$

- Let $\eta = \theta/(3\tau)$. Then

$$\begin{aligned} 1 - \eta\alpha + \frac{12\eta\theta\delta^2}{B} &= 1 - \frac{\mu\theta}{9\tau} + \frac{4\theta^2\delta^2}{B\tau} \leq 1 - \frac{\mu\theta}{9\tau} + \frac{4\theta\delta^2}{B\tau} \frac{\mu B\tau}{72\delta^2} \leq 1 - \frac{\mu\theta}{18\tau} \\ &= 1 - \frac{\sqrt{\mu\theta}}{18}. \end{aligned}$$

Thus,

$$\mathbb{E}[\Phi_{k+1}] \leq \max \left\{ 1 - \frac{1}{6}, 1 - \frac{\sqrt{\mu\theta}}{18} \right\}.$$

Since $\mu < \delta$, we obtain the required result. $\square$

Appendix C. Proof of Corollaries 1 and 2

Here we present a proof of Corollaries 1 and 2. We use the technique from [37].

Proof. We obtained before the recurrence formula

$$0 \leq (1 - a\sqrt{\theta})\mathbb{E}[\Phi_k] - \mathbb{E}[\Phi_{k+1}] + c\theta.$$

Let us multiply both sides by the weight $w_k = (1 - a\sqrt{\theta})^{-(k+1)}$ and rewrite:

$$0 \leq \frac{1}{\sqrt{\theta}} (\mathbb{E}[\Phi_k]w_{k-1} - \mathbb{E}[\Phi_{k+1}]w_k) + cw_k\sqrt{\theta}.$$

We set

$$W_N = \sum_{k=0}^N w_k (1 - \sqrt{\theta})^{-(N+1)} \sum_{k=0}^N (1 - a\sqrt{\theta})^k \leq \frac{w_N}{a\sqrt{\theta}}$$

and sum up for $k = 0, \dots, N$:

$$\frac{w_N \mathbb{E}[\Phi_{k+1}]}{\sqrt{\theta} W_N} \leq \frac{\Phi_0}{\sqrt{\theta} W_N} + c\sqrt{\theta}.$$

We rewrite this as

$$\mathbb{E}[\Phi_{k+1}] \leq \frac{1}{a\sqrt{\theta}} \exp\{-a\sqrt{\theta}(N+1)\} + \frac{c\sqrt{\theta}}{a}.$$

To complete the proof we need to substitute

$$a = \frac{\sqrt{\mu}}{3} \quad \text{and} \quad c = \frac{4\sigma^2}{B}$$

in the case of Corollary 1 and

$$a = \frac{\sqrt{\mu}}{18} \quad \text{and} \quad c = \frac{4\sigma^2}{B}$$

in the case of Corollary 2 and look at different θ .

For Corollary 1 we have the following.

If

$$\frac{9 \ln^2(\max\{2, B\mu N^2 \Phi_0 / (36\sigma^2)\})}{\mu N^2} \leq \frac{1}{3\delta},$$

where $\sigma^2 = 2(\sigma_{\text{sim}}^2 + \sigma_{\text{noise}}^2)$, then we choose

$$\theta = \frac{9 \ln^2(\max\{2, B\mu N^2 \Phi_0 / (36\sigma^2)\})}{\mu N^2}$$

and obtain

$$\mathbb{E}[\Phi_{N+1}] = \mathcal{O}\left(\frac{24(\sigma_{\text{sim}}^2 + \sigma_{\text{noise}}^2)}{\mu BN}\right).$$

Otherwise, we choose $\theta = 1/(3\delta)$ and obtain

$$\mathbb{E}[\Phi_{N+1}] = \mathcal{O}\left(\sqrt{3\delta} \Phi_0 \exp\left\{-\frac{\sqrt{\mu}}{3\sqrt{3}\sqrt{\delta}} N\right\} + \frac{24(\sigma_{\text{sim}}^2 + \sigma_{\text{noise}}^2)}{\mu BN}\right).$$

It remains to recall that an iteration of the method requires B/M communications.

For Corollary 2 we have the following.

• If

$$\frac{1}{3\delta} \leq \frac{\mu^3 B^2}{5184\delta^4} \quad \text{and} \quad \frac{324 \ln^2(\max\{2, B\mu N^2 \Phi_0 / (1296 \sigma^2)\})}{\mu N^2} \leq \frac{1}{3\delta},$$

where $\sigma^2 = 2(\sigma_{\text{sim}}^2 + \sigma_{\text{noise}}^2)$, then we choose

$$\theta = \frac{324 \ln^2(\max\{2, B\mu N^2 \Phi_0 / (1296 \sigma^2)\})}{\mu N^2}$$

and obtain

$$\mathbb{E}[\Phi_{N+1}] = \mathcal{O}\left(\frac{144(\sigma_{\text{sim}}^2 + \sigma_{\text{noise}}^2)}{\mu BN}\right).$$

• If

$$\frac{1}{3\delta} \leq \frac{\mu^3 B^2}{5184\delta^4} \quad \text{and} \quad \frac{324 \ln^2(\max\{2, B\mu N^2 \Phi_0 / (1296 \sigma^2)\})}{\mu N^2} \geq \frac{1}{3\delta},$$

where $\sigma^2 = 2(\sigma_{\text{sim}}^2 + \sigma_{\text{noise}}^2)$, then we choose

$$\theta = \frac{1}{3\delta}$$

and obtain

$$\mathbb{E}[\Phi_{N+1}] = \mathcal{O}\left(\sqrt{3\delta} \Phi_0 \exp\left\{-\frac{\sqrt{\mu}}{18\sqrt{3}\sqrt{\delta}} N\right\} + \frac{144(\sigma_{\text{sim}}^2 + \sigma_{\text{noise}}^2)}{\mu BN}\right).$$

• If

$$\frac{1}{3\delta} \geq \frac{\mu^3 B^2}{5184\delta^4} \quad \text{and} \quad \frac{324 \ln^2(\max\{2, B\mu N^2 \Phi_0 / (1296 \sigma^2)\})}{\mu N^2} \leq \frac{\mu^3 B^2}{5184\delta^4},$$

where $\sigma^2 = 2(\sigma_{\text{sim}}^2 + \sigma_{\text{noise}}^2)$, then we choose

$$\theta = \frac{324 \ln^2(\max\{2, B\mu N^2 \Phi_0 / (1296 \sigma^2)\})}{\mu N^2}$$

and obtain

$$\mathbb{E}[\Phi_{N+1}] = \mathcal{O}\left(\frac{144(\sigma_{\text{sim}}^2 + \sigma_{\text{noise}}^2)}{\mu BN}\right).$$

• If

$$\frac{1}{3\delta} \geq \frac{\mu^3 B^2}{5184 \delta^4} \quad \text{and} \quad \frac{324 \ln^2(\max\{2, B\mu N^2 \Phi_0 / (1296 \sigma^2)\})}{\mu N^2} \geq \frac{\mu^3 B^2}{5184 \delta^4},$$

where $\sigma^2 = 2(\sigma_{\text{sim}}^2 + \sigma_{\text{noise}}^2)$, then we choose

$$\theta = \frac{324 \ln^2(\max\{2, B\mu N^2 \Phi_0 / (1296 \sigma^2)\})}{\mu N^2}$$

and obtain

$$\mathbb{E}[\Phi_{N+1}] = \mathcal{O}\left(\frac{72\delta^2}{B\sqrt{\mu^3}}\Phi_0 \exp\left\{-\frac{B}{1296}\frac{\mu^2}{\delta^2}N\right\} + \frac{144(\sigma_{\text{sim}}^2 + \sigma_{\text{noise}}^2)}{\mu BN}\right).$$

As in the case of Corollary 1, an iteration of the method requires B/M communications. $\square$

Appendix D. ASEG for the convex case

The next algorithm is an adaptation of Algorithm 1 to the convex case. We use time-varying τ_k and η_k .

Theorem 6. *Let Algorithm 2 run for N iterations under Assumption 1 for $\mu = 0$, Assumption 2 for $\zeta = 0$, and Assumption 3, with the following tuning:*

$$\tau_k = \frac{2}{k+1}, \quad \theta \leq \frac{1}{3\delta}, \quad \eta_k = \frac{\theta}{\tau_k},$$

and let x_f^{k+1} in Line 16 satisfy

$$\|\nabla \bar{A}_\theta^k(x_f^{k+1})\|^2 \leq \frac{9\delta^2}{11} \left\|x_g^k - \arg \min_{x \in \mathbb{R}^d} \bar{A}_\theta^k(x)\right\|^2. \quad (12)$$

Then the following inequality holds:

$$\mathbb{E}\left[\frac{\theta}{\tau_{N+1}^2} [r(x_f^{N+1}) - r(x_*)]\right] \leq \|x_0 - x_*\|^2 + \frac{6(\sigma_{\text{sim}}^2 + \sigma_{\text{noise}}^2)}{B} \theta^2 \sum_{i=1}^N \frac{1}{\tau_k^2}.$$

Proof. Set $\sigma^2 = 2(\sigma_{\text{sim}}^2 + \sigma_{\text{noise}}^2)$ again. Here we apply Lemma 2 for $\mu = 0$ and $\zeta = 0$:

$$\begin{aligned} \frac{1}{\eta_{k+1}} \mathbb{E}[\|x_{k+1} - x_*\|^2 \mid x_k] &\leq \frac{1}{\eta_{k+1}} \|x_k - x_*\|^2 + \frac{1}{\eta_{k+1}} \mathbb{E}[\|x_{k+1} - x_k\|^2 \mid x_k] \\ &\quad + 2[r(x_*) - r(x_f^{k+1})] + \frac{2(1 - \tau_{k+1})}{\tau_{k+1}} [r(x_f^k) - r(x_*)] \\ &\quad - \frac{\theta}{\tau_{k+1}} \|\nabla r(x_f^{k+1})\|^2 + \frac{3\theta}{B\tau_{k+1}} \sigma^2 \\ &\leq \frac{1}{\eta_{k+1}} \|x_k - x_*\|^2 + \frac{\eta_{k+1}}{B} \sigma^2 + \frac{3\theta}{B\tau_{k+1}} \sigma^2 \\ &\quad + \left(\eta_{k+1} - \frac{\theta}{\tau_{k+1}}\right) \|\nabla r(x_f^{k+1})\|^2 + \frac{2}{\tau_{k+1}} [r(x_*) - r(x_f^{k+1})] \\ &\quad + \frac{2(1 - \tau_{k+1})}{\tau_{k+1}} [r(x_f^k) - r(x_*)]. \end{aligned}$$

Algorithm 2: Accelerated Stochastic ExtraGradient for convex case

```

1: Input:  $x^0 = x_f^0 \in \mathbb{R}^d$ 
2: Parameters:  $\tau \in (0, 1]$ ,  $\eta, \theta, \alpha > 0$ ,  $N \in \{1, 2, \dots\}$ ,  $B \in \mathbb{N}$ 
3: for  $k = 0, 1, 2, \dots, N - 1$  do:
4:   for server do:
5:     Generate a set of  $|I| = B$  numbers uniformly from  $[2, \dots, n]$ 
6:     Send to each device  $m$  one bit  $b_k^m$ : 1 if  $m \in I$ , 0 otherwise
7:   end for
8:   for each device  $m$  in parallel do:
9:      $x_g^k = \tau_{k+1}x^k + (1 - \tau_{k+1})x_f^k$ 
10:    if  $b_k^m = 1$  then
11:      Send  $\nabla r_m(x_g^k, \xi)$  to server
12:    end if
13:  end for
14:  for server do:
15:     $s_k = \frac{1}{B} \sum_{m \in I} (\nabla r_m(x_g^k, \xi) - \nabla r_1(x_g^k))$ 
16:     $x_f^{k+1} \approx \arg \min_{x \in \mathbb{R}^d} \left[ \bar{A}_\theta^k(x) := \langle s_k, x - x_g^k \rangle + \frac{1}{2\theta} \|x - x_g^k\|^2 + r_1(x) \right]$ 
17:    Generate a set of  $|I| = B$  numbers uniformly from  $[1, \dots, n]$ 
18:    Send to each device  $m$  one bit  $c_k^m$ : 1 if  $m \in I$ , 0 otherwise
19:  end for
20:  for each device  $m$  in parallel do:
21:    if  $c_k^m = 1$  then
22:      Send  $\nabla r_m(x_f^{k+1}, \xi)$  to server
23:    end if
24:  end for
25:  for server do:
26:     $t_k = \frac{1}{B} \sum_{m \in I} \nabla r_m(x_f^{k+1}, \xi)$ 
27:     $x^{k+1} = x^k - \eta_{k+1}t_k$ 
28:  end for
29: end for
30: Output:  $x^N$ 

```

We substitute $\eta_{k+1} = \theta/(2\tau_{k+1})$:

$$\begin{aligned}
\frac{\tau_{k+1}}{\theta} \mathbb{E}[\|x_{k+1} - x_*\|^2 \mid x_k] &\leq \frac{\tau_{k+1}}{\theta} \|x_k - x_*\|^2 + \frac{\theta}{4B\tau_{k+1}} \sigma^2 + \frac{3\theta}{2B\tau_{k+1}} \sigma^2 \\
&\quad + \frac{1}{\tau_{k+1}} [r(x_*) - r(x_f^{k+1})] + \frac{1 - \tau_{k+1}}{\tau_{k+1}} [r(x_f^k) - r(x_*)].
\end{aligned}$$

We multiply both sides by $\theta/(\tau_{k+1})$:

$$\begin{aligned}
\mathbb{E}[\|x_{k+1} - x_*\|^2 \mid x_k] &\leq \|x_k - x_*\|^2 + \frac{3\theta^2}{B\tau_{k+1}^2} \sigma^2 + \frac{\theta}{\tau_{k+1}^2} [r(x_*) - r(x_f^{k+1})] \\
&\quad + \frac{\theta(1 - \tau_{k+1})}{\tau_{k+1}^2} [r(x_f^k) - r(x_*)].
\end{aligned}$$

Let us rewrite the last inequality:

$$\begin{aligned} \mathbb{E}[\|x_{k+1} - x_*\|^2 \mid x_k] &+ \frac{\theta}{\tau_{k+1}^2} [r(x_f^{k+1}) - r(x_*)] \\ &\leq \|x_k - x_*\|^2 + \frac{3\theta^2}{B\tau_{k+1}^2} \sigma^2 + \frac{\theta(1 - \tau_{k+1})}{\tau_{k+1}^2} [r(x_f^k) - r(x_*)]. \end{aligned}$$

Define

$$\Phi_k = \|x_k - x_*\|^2 + \frac{\theta}{\tau_k^2} [r(x_f^k) - r(x_*)].$$

Using $\tau_k = 2/(k+1)$ we obtain

$$\frac{\theta}{\tau_{k+1}^2} [r(x_f^{k+1}) - r(x_*)] \leq \Phi_k + \frac{3\theta^2}{B\tau_{k+1}^2} \sigma^2.$$

Next we apply the previous inequality and assume that $\tau_1 = 1$:

$$\mathbb{E} \left[\frac{\theta}{\tau_{N+1}^2} [r(x_f^{N+1}) - r(x_*)] \right] \leq \|x_0 - x_*\|^2 + \frac{3\sigma^2\theta^2}{B} \sum_{i=1}^N \frac{1}{\tau_i^2}. \quad \square$$

Similarly to the case where $\mu > 0$, it is not possible to ensure convergence by a naive choice of θ . We propose a choice of θ that ensures optimal convergence.

Corollary 4. *Consider Assumption 1 for $\mu = 0$, Assumption 2 for $\zeta = 0$, and Assumption 3. Let Algorithm 2 run for N iterations with the tuning of τ_k and η_k as in Theorem 6. Then the following tuning of θ is proposed.*

If

$$\frac{\sqrt{B} \|x_0 - x_*\|}{\sqrt{3} \sigma (N+1)^{3/2}} \leq \frac{1}{3\delta},$$

then choose

$$\theta = \frac{\sqrt{B} \|x_0 - x_*\|}{\sqrt{3} \sigma (N+1)^{3/2}}$$

and obtain

$$\mathbb{E}[r(x_f^{N+1}) - r(x_*)] \leq \frac{2\sqrt{3}\sigma \|x_0 - x_*\|}{\sqrt{B} \sqrt{N+1}}.$$

Otherwise, choose $\theta = 1/(3\delta)$ and obtain

$$\mathbb{E}[r(x_f^{N+1}) - r(x_*)] \leq \frac{3\delta \|x_0 - x_*\|^2}{(N+1)^2} + \frac{\sqrt{3}\sigma \|x_0 - x_*\|}{\sqrt{B} \sqrt{N+1}}.$$

Proof. Note that $1/\tau_k^2 \leq (N+1)^2/4$. Thus, using Theorem 6 we obtain

$$\mathbb{E}[\theta(N+1)^2 [r(x_f^{N+1}) - r(x_*)]] \leq \|x_0 - x_*\|^2 + \frac{3\sigma^2}{B} \theta^2 (N+1)^3.$$

We divide both sides by $\theta(N+1)^2$:

$$\mathbb{E}[r(x_f^{N+1}) - r(x_*)] \leq \frac{\|x_0 - x_*\|^2}{\theta(N+1)^2} + \frac{3\sigma^2}{B} \theta (N+1).$$

Notice that $\theta = \frac{\sqrt{B} \|x_0 - x_*\|}{\sqrt{3} \sigma (N+1)^{3/2}}$ equalizes the terms on the right-hand side. If $\frac{\sqrt{B} \|x_0 - x_*\|}{\sqrt{3} \sigma (N+1)^{3/2}} \leq \frac{1}{3\delta}$, we choose $\theta = \frac{\sqrt{B} \|x_0 - x_*\|}{\sqrt{3} \sigma (N+1)^{3/2}}$. Then we obtain

$$\mathbb{E}[r(x_f^{N+1}) - r(x_*)] \leq \frac{2\sqrt{3}\sigma\|x_0 - x_*\|}{\sqrt{B}\sqrt{N+1}}.$$

Otherwise, we choose $\theta = 1/(3\delta)$ and obtain

$$\mathbb{E}[r(x_f^{N+1}) - r(x_*)] \leq \frac{3\delta\|x_0 - x_*\|^2}{(N+1)^2} + \frac{\sqrt{3}\sigma\|x_0 - x_*\|}{\sqrt{B}\sqrt{N+1}}. \quad \square$$

Bibliography

- [1] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and Li Zhang, “Deep learning with differential privacy”, *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security* (Vienna 2016), ACM, New York 2016, pp. 308–318.
- [2] M. Aitsam, “Differential privacy made easy”, *2022 International conference on emerging trends in electrical, control, and telecommunication engineering (ETECTE)* (Lahore 2022), IEEE 2022, pp. 1–7.
- [3] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, “QSGD: Communication-efficient SGD via gradient quantization and encoding”, *NIPS’17: Proceedings of the 31st international conference on neural information processing systems*, Adv. Neural Inf. Process. Syst., vol. 30, Curran Associates, Inc., Red Hook, NY 2017, 1707–1718, <https://proceedings.neurips.cc/paper/2017/hash/6c340f25839e6acdc73414517203f5f0-Abstract.html>.
- [4] Z. Allen-Zhu, “Katyusha: The first direct acceleration of stochastic gradient methods”, *J. Mach. Learn. Res.* **18** (2018), 221, 51 pp.
- [5] Y. Arjevani and O. Shamir, “Communication complexity of distributed convex learning and optimization”, *NIPS’15: Proceedings of the 28th international conference on neural information processing systems*, vol. 1, Adv. Neural Inf. Process. Syst., vol. 28, MIT Press, Cambridge, MA 2015, 1756–1764, https://papers.nips.cc/paper_files/paper/2015/hash/7fec306d1e665bc9c748b5d2b99a6e97-Abstract.html.
- [6] B. Barazandeh, Tianjian Huang, and G. Michailidis, “A decentralized adaptive momentum method for solving a class of min-max optimization problems”, *Signal Process.* **189** (2021), 108245, 23 pp.
- [7] A. Beznosikov, P. Dvurechenskii, A. Koloskova, V. Samokhin, S. U. Stich, and A. Gasnikov, “Decentralized local stochastic extra-gradient for variational inequalities”, *NIPS’22: Proceedings of the 36th international conference on neural information processing systems*, Adv. Neural Inf. Process. Syst., vol. 35, Curran Associates, Inc., Red Hook, NY 2022, 38116–38133, 2762, https://proceedings.neurips.cc/paper_files/paper/2022/hash/f9379afacdbabfd6b060972b60f9ab8-Abstract-Conference.html.
- [8] A. Beznosikov and A. Gasnikov, “Compression and data similarity: combination of two techniques for communication-efficient solving of distributed variational inequalities”, *Optimization and applications*, Lecture Notes in Comput. Sci., vol. 13781, Springer, Cham 2022, pp. 151–162.

- [9] A. Beznosikov, S. Horváth, P. Richtárik, and M. Safaryan, “On biased compression for distributed learning”, *J. Mach. Learn. Res.* **24** (2023), 276, 50 pp.
- [10] A. Beznosikov, G. Scutari, A. Rogozin, and A. Gasnikov, “Distributed saddle-point problems under similarity”, *NIPS’21: Proceedings of the 35th international conference on neural information processing systems*, Adv. Neural Inf. Process. Syst., vol. 34, Curran Associates, Inc., Red Hook, NY 2021, 8172–8184, 625, <https://proceedings.neurips.cc/paper/2021/hash/44e65d3e9bc2f88b2b3d566de51a5381-Abstract.html>.
- [11] A. Beznosikov, M. Takác, and A. Gasnikov, “Similarity, compression and local steps: three pillars of efficient communications for distributed variational inequalities”, *NIPS’23: Proceedings of the 37th international conference on neural information processing systems*, Adv. Neural Inf. Process. Syst., vol. 36, Curran Associates, Inc., Red Hook, NY 2023, 28663–28677, 1246, https://proceedings.neurips.cc/paper_files/paper/2023/hash/5b4a459db23e6db9be2a128380953d96-Abstract-Conference.html.
- [12] Chih-Chung Chang and Chih-Jen Lin, “LIBSVM: A library for support vector machines”, *ACM Trans. Intell. Syst. Technol. (TIST)* **2:3** (2011), 27, 27 pp.
- [13] A. Defazio, F. Bach, and S. Lacoste-Julien, “SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives”, *NIPS’14: Proceedings of the 27th international conference on neural information processing systems*, vol. 1, Adv. Neural Inf. Process. Syst., vol. 27, MIT Press, Cambridge, MA 2014, 1646–1654, https://proceedings.neurips.cc/paper_files/paper/2014/hash/ede7e2b6d13a41ddf9f4bdef84fdc737-Abstract.html.
- [14] Jiashi Feng, Huan Xu, and Shie Mannor, *Distributed robust learning*, 2015 (v1 – 2014), 18 pp., [arXiv:1409.5937](https://arxiv.org/abs/1409.5937).
- [15] S. Gade and N. H. Vaidya, “Private optimization on networks”, *2018 Annual american control conference (ACC)* (Milwaukee, WI 2018), IEEE 2018, pp. 1402–1409.
- [16] E. Gorbunov, F. Hanzely, and P. Richtárik, “A unified theory of SGD: variance reduction, sampling, quantization and coordinate descent”, *Proceedings of the 23rd international conference on artificial intelligence and statistics (AISTATS 2020)*, Proc. Mach. Learn. Res. (PMLR), vol. 108, 2020, pp. 680–690, <https://proceedings.mlr.press/v108/gorbunov20a.html>; 2019, 38 pp., [arXiv:1905.11261](https://arxiv.org/abs/1905.11261).
- [17] E. Gorbunov, F. Hanzely, and P. Richtárik, “Local SGD: unified theory and new efficient methods”, *Proceedings of the 24th international conference on artificial intelligence and statistics (AISTATS 2021)*, Proc. Mach. Learn. Res. (PMLR), vol. 130, 2021, pp. 3556–3564, <https://proceedings.mlr.press/v130/gorbunov21a.html>.
- [18] T. M. Hehn, J. F. Kooij, and F. A. Hamprecht, “End-to-end learning of decision trees and forests”, *Int. J. Comput. Vis.* **128:4** (2020), 997–1011.
- [19] H. Hendrikx, Lin Xiao, S. Bubeck, F. Bach, and L. Massoulié, “Statistically preconditioned accelerated gradient method for distributed optimization”, *Proceedings of the 37th international conference on machine learning (ICML 2020)*, Proc. Mach. Learn. Res. (PMLR), vol. 119, 2020, 4203–4227, <https://proceedings.mlr.press/v119/hendrikx20a.html>.
- [20] R. Johnson and Tong Zhang, “Accelerating stochastic gradient descent using predictive variance reduction”, *NIPS’13: Proceedings of the 26th international conference on neural information processing systems*, vol. 1, Adv. Neural Inf. Process. Syst., vol. 26, Curran Associates Inc., Red Hook, NY

- 2013, 315–323, https://proceedings.neurips.cc/paper_files/paper/2013/hash/ac1dd209cbcc5e5d1c6e28598e8cbb8-Abstract.html.
- [21] A. Khaled and Chi Jin, *Faster federated optimization under second-order similarity*, 2023 (v1 – 2022), 44 pp., [arXiv: 2209.02257](https://arxiv.org/abs/2209.02257).
- [22] A. Khaled, K. Mishchenko, and P. Richtárik, “Tighter theory for local SGD on identical and heterogeneous data”, *Proceedings of the 23rd international conference on artificial intelligence and statistics (AISTATS 2020)*, Proc. Mach. Learn. Res. (PMLR), vol. 108, 2020, pp. 4519–4529, <https://proceedings.mlr.press/v108/bayoumi20a.html>.
- [23] A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, and S. Stich, “A unified theory of decentralized SGD with changing topology and local updates”, *Proceedings of the 37th international conference on machine learning (ICML 2020)*, Proc. Mach. Learn. Res. (PMLR), vol. 119, 2020, 5381–5393, <https://proceedings.mlr.press/v119/koloskova20a.html>.
- [24] D. Kovalev, A. Beznosikov, E. Borodich, A. Gasnikov, and G. Scutari, “Optimal gradient sliding and its application to optimal distributed optimization under similarity”, *NIPS’22: Proceedings of the 36th international conference on neural information processing systems*, Adv. Neural Inf. Process. Syst., vol. 35, Curran Associates, Inc., Red Hook, NY 2022, 33494–33507, 2427, https://proceedings.neurips.cc/paper_files/paper/2022/hash/d88f6f81e1aaf606776ffdd06fdf24ef-Abstract-Conference.html.
- [25] D. Kovalev, S. Horváth, and P. Richtárik, “Don’t jump through hoops and remove those loops: SVRG and Katyusha are better without the outer loop”, *Proceedings of the 31st international conference on algorithmic learning theory (ALT 2020)*, Proc. Mach. Learn. Res. (PMLR), vol. 117, 2020, 451–467, <https://proceedings.mlr.press/v117/kovalev20a.html>.
- [26] Zhize Li, Hongyan Bao, Xiangliang Zhang, and P. Richtárik, “PAGE: A simple and optimal probabilistic gradient estimator for nonconvex optimization”, *Proceedings of the 38th international conference on machine learning (ICML 2021)*, Proc. Mach. Learn. Res. (PMLR), vol. 139, 2021, pp. 6286–6295, <https://proceedings.mlr.press/v139/li21a.html>.
- [27] Dachao Lin, Yuze Han, Haishan Ye, and Zhihua Zhang, “Stochastic distributed optimization under average second-order similarity: algorithms and analysis”, *NIPS’23: Proceedings of the 37th international conference on neural information processing systems*, Adv. Neural Inf. Process. Syst., vol. 36, Curran Associates, Inc., Red Hook, NY 2023, pp. 1849–1862, https://proceedings.neurips.cc/paper_files/paper/2023/hash/05e552739c2629f3324c1063a382b4bd-Abstract-Conference.html.
- [28] Hongzhou Lin, J. Mairal, and Z. Harchaoui, “A universal catalyst for first-order optimization”, *NIPS’15: Proceedings of the 28th international conference on neural information processing systems*, vol. 2, Adv. Neural Inf. Process. Syst., vol. 28, MIT Press, Cambridge, MA 2015, 3384–3392, https://proceedings.neurips.cc/paper_files/paper/2015/hash/c164bbc9d6c72a52c599bbb43d8db8e1-Abstract.html.
- [29] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Arcas, “Communication-efficient learning of deep networks from decentralized data”, *Proceedings of the 20th international conference on artificial intelligence and statistics (AISTATS 2017)*, Proc. Mach. Learn. Res. (PMLR), vol. 54, 2017, 1273–1282, <https://proceedings.mlr.press/v54/mcmahan17a>.
- [30] J. M. Moguerza and A. Muñoz, “Support vector machines with applications”, *Statist. Sci.* **21**:3 (2006), 322–336.

- [31] L. M. Nguyen, Jie Liu, K. Scheinberg, and M. Takáč, “SARAH: A novel method for machine learning problems using stochastic recursive gradient”, *Proceedings of the 34th international conference on machine learning (ICML 2017)*, Proc. Mach. Learn. Res. (PMLR), vol. 70, 2017, pp. 2613–2621, <https://proceedings.mlr.press/v70/nguyen17b.html>.
- [32] E. Nozari, P. Tallapragada, and J. Cortés, “Differentially private distributed convex optimization via functional perturbation”, *IEEE Trans. Control Netw. Syst.* **5**:1 (2018), 395–408.
- [33] S. S. Sannakki and A. A. Kulkarni, “A review paper on artificial neural network in cognitive science”, *Int. J. Engrg. Techn.* **2**:2 (2016), 118–123, <https://oaji.net/articles/2017/1992-1514450980.pdf>.
- [34] S. Shalev-Shwartz, O. Shamir, N. Srebro, and K. Sridharan, “Learnability, stability and uniform convergence”, *J. Mach. Learn. Res.* **11** (2010), 2635–2670.
- [35] O. Shamir, N. Srebro, and Tong Zhang, “Communication-efficient distributed optimization using an approximate Newton-type method”, *Proceedings of the 31st international conference on machine learning (ICML 2014)*, Proc. Mach. Learn. Res. (PMLR), vol. 32, 2014, pp. 1000–1008, <https://proceedings.mlr.press/v32/shamir14.html>.
- [36] S. Sreenivasan, R. Cohen, E. López, Z. Toroczkai, and H. E. Stanley, *Communication bottlenecks in scale-free networks*, 2006, 5 pp., [arXiv:cs/0604023](https://arxiv.org/abs/cs/0604023).
- [37] S. U. Stich, *Unified optimal analysis of the (stochastic) gradient method*, 2019, 11 pp., [arXiv:1907.04232](https://arxiv.org/abs/1907.04232).
- [38] Ying Sun, G. Scutari, and A. Daneshmand, “Distributed optimization based on gradient tracking revisited: enhancing convergence rate via surrogation”, *SIAM J. Optim.* **32**:2 (2022), 354–385.
- [39] Yy Tian, G. Scutari, Tianyu Cao, and A. Gasnikov, “Acceleration in distributed optimization under similarity”, *Proceedings of the 25th international conference on artificial intelligence and statistics (AISTATS 2022)*, Proc. Mach. Learn. Res. (PMLR), vol. 151, 2022, pp. 5721–5756, <https://proceedings.mlr.press/v151/tian22b.html>.
- [40] G. Tripepi, K. J. Jager, F. W. Dekker, and C. Zoccali, “Linear and logistic regression analysis”, *Kidney Int.* **73**:7 (2008), 806–810.
- [41] S. Vaswani, I. Laradji, F. Kunstner, Si Yi Meng, M. Schmidt, and S. Lacoste-Julien, *Adaptive gradient methods converge faster with over-parameterization (but you should do a line-search)*, 2021 (v1 – 2020), 41 pp., [arXiv:2006.06835](https://arxiv.org/abs/2006.06835).
- [42] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, and J. S. Rellermeyer, “A survey on distributed machine learning”, *ACM Comput. Surveys* **53**:2 (2020), 30, 33 pp.
- [43] P. C. Weeraddana and C. Fischione, “On the privacy of optimization”, *IFAC-PapersOnLine* **50**:1 (2017), 9502–9508.
- [44] B. E. Woodworth, K. K. Patel, and N. Srebro, “Minibatch vs local SGD for heterogeneous distributed learning”, *NIPS’20: Proceedings of the 34th international conference on neural information processing systems*, Adv. Neural Inf. Process. Syst., vol. 33, Curran Associates, Inc., Red Hook, NY 2020, 6281–6292, 527, https://proceedings.neurips.cc/paper_files/paper/2020/hash/45713f6ff2041d3fdfae927b82488db8-Abstract.html.

- [45] Xiao-Tong Yuan and Ping Li, “On convergence of distributed approximate Newton methods: globalization, sharper bounds and beyond”, *J. Mach. Learn. Res.* **21** (2020), 206, 51 pp.

Dmitry A. Bylinkin

Moscow Institute of Physics and Technology
(National Research University), Moscow, Russia;
Ivannikov Institute for System Programming
of the Russian Academy of Sciences,
Moscow, Russia

E-mail: bylinkin.da@phystech.edu

Received 15/AUG/24

Kirill D. Degtyarev

Moscow Institute of Physics and Technology
(National Research University), Moscow, Russia

E-mail: degtyarev.kd@phystech.edu

Aleksandr N. Beznosikov

Ivannikov Institute for System Programming
of the Russian Academy of Sciences,
Moscow, Russia;

Sber AI Lab, Moscow, Russia;

Innopolis University, Innopolis, Russia

E-mail: anbeznosikov@gmail.com;
beznosikov.an@phystech.su