CONSTRAINED AND COMPOSITE OPTIMIZATION VIA ADAPTIVE SAMPLING METHODS

YUCHEN XIE*, RAGHU BOLLAPRAGADA†, RICHARD BYRD ‡, AND JORGE NOCEDAL §

Abstract. The motivation for this paper stems from the desire to develop an adaptive sampling method for solving constrained optimization problems in which the objective function is stochastic and the constraints are deterministic. The method proposed in this paper is a proximal gradient method that can also be applied to the composite optimization problem min $f(x) + h(x)$, where f is stochastic and h is convex (but not necessarily differentiable). Adaptive sampling methods employ a mechanism for gradually improving the quality of the gradient approximation so as to keep computational cost to a minimum. The mechanism commonly employed in unconstrained optimization is no longer reliable in the constrained or composite optimization settings because it is based on pointwise decisions that cannot correctly predict the quality of the proximal gradient step. The method proposed in this paper measures the result of a complete step to determine if the gradient approximation is accurate enough; otherwise a more accurate gradient is generated and a new step is computed. Convergence results are established both for strongly convex and general convex f. Numerical experiments are presented to illustrate the practical behavior of the method.

1. Introduction. In this paper, we study the solution of constrained and composite optimization problems in which the objective function is stochastic and the constraints or regularizers are deterministic. We propose methods that automatically adjust the quality of the gradient estimate so as to keep computational cost at a minimum while ensuring a fast rate of convergence. Methods of this kind have been studied in the context of unconstrained optimization but their extension to the constrained and composite optimization settings is not simple because the projections or proximal operators used in the methods introduce discontinuities. This renders existing rules for the control of the gradient unreliable. Whereas in the unconstrained setting pointwise decisions suffice to estimate the quality of a gradient approximation, in the presence of constraints or nonsmooth regularizers one must analyze the result of a complete step.

Let us begin by considering the optimization problem

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad x \in \Omega, \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a stochastic objective function and Ω is a deterministic convex set. Automatic rules for controlling the quality of the gradient when $\Omega = \mathbb{R}^n$ have been studied from a theoretical perspective and have been successfully applied to expected risk minimization problems arising in machine learning. Since in that context the gradient approximation is controlled by the sample size, these methods have been called "adaptive sampling" methods. A fundamental mechanism for controlling the quality of the gradient in the unconstrained setting is the *norm test* [7], which lies behind most algorithms and theory of adaptive sampling methods.

To describe this test, let $\Omega = \mathbb{R}^n$, and consider the iteration

$$x_{k+1} = x_k - \alpha_k g_k, \tag{1.2}$$

where $\alpha_k > 0$ is a steplength and g_k is an approximation to the gradient $\nabla f(x_k)$. To determine if g_k is sufficiently accurate to ensure that iteration (1.2) is convergent, one can test the inequality

*Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, USA. This author was supported by the Office of Naval Research grant N00014-14-1-0313 P00003, and by National Science Foundation grant DMS-1620022.

†Department of Mechanical Engineering, University of Texas, Austin, USA.

‡Department of Computer Science, University of Colorado, Boulder, CO, USA. This author was supported by National Science Foundation grant DMS-1620070.

§Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, USA. This author was supported by the Office of Naval Research grant N00014-14-1-0313 P00003, and National Science Foundation grant DMS-1620022.

[7]:

$$\mathbb{E}[\|g_k - \nabla f(x_k)\|_2^2] \leq \xi \|\nabla f(x_k)\|_2^2, \quad \xi > 0, \tag{1.3}$$

where the expectation is taken with respect to the choice of g_k at iteration k. If (1.3) is satisfied, g_k is deemed accurate enough; otherwise a new and more accurate gradient approximation is computed. We refer to this procedure as the norm test to distinguish it from tests based on angles [6].

The norm test is, however, not adequate in the constrained setting. To see this, suppose that we apply the gradient projection method, $x_{k+1} = P_\Omega[x_k - \alpha_k g_k]$, to solve problem (1.1) when $\Omega \neq \mathbb{R}^n$. A condition such as (1.3) on the quality of the gradient approximation at one point cannot always predict the quality of the full step because the latter is based on a projection of the gradient, which may be much smaller. This is illustrated in Figure 1.1, where we consider the minimization of a strongly convex quadratic function subject to a linear constraint:

$$\min_{x \in \mathbb{R}^n} \tfrac{1}{2} x^T Q x + b^T x + c \qquad \text{s.t. } a^T x \leq 0.$$

In Figure 1.1, $\widehat{x}^*$ denotes the unconstrained minimizer and x^* the solution of the constrained problem. We let the iterate x_k lie on the boundary of the constraint, very close to the solution x^*, and observe that $\|\nabla f(x_k)\|$ is large, and stays large as x_k approaches x^*. Thus, (1.3) does not force the error in g_k to zero as $x_k \to x^*$.

The instance of g_k shown in Figure 1.1 satisfies $\|g_k - \nabla f(x_k)\| < \|\nabla f(x_k)\|$, but results in a poor step. Clearly satisfaction of (1.3) allows for many such steps. Note, however, that $\|g_k - \nabla f(x_k)\|$ is greater than the norm of the projected gradient $P_\Omega[g_k]$, which is a more appropriate measure. Thus, since we are concerned about the error in the total step, and in this

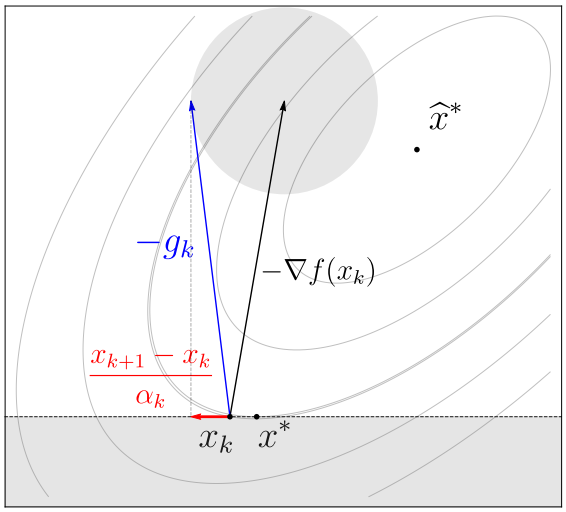

FIG. 1.1. *Failure of norm test for constrained problems.*

example the step is given by $x_{k+1} - x_k = -\alpha_k P_\Omega[g_k]$, it makes sense to compare $\|g_k - \nabla f(x_k)\|$ to $\|P_\Omega[g_k]\| = \|x_{k+1} - x_k\|/\alpha_k$.

We generalize this idea and propose the following procedure for measuring the quality of the gradient approximation. We first compute a projected step $\overline{x}_{k+1} = P_\Omega[x_k - \alpha_k g_k]$ based on the

current gradient estimate g_k, and regard g_k to be acceptable if the following inequality holds:

$$\mathbb{E}[\|g_k - \nabla f(x_k)\|_2^2] \equiv \mathrm{Var}_k\left[g_k\right] \leq \xi \left\| \frac{\mathbb{E}_k\left[\overline{x}_{k+1}\right] - x_k}{\alpha_k} \right\|_2^2, \quad \xi > 0. \tag{1.4}$$

Otherwise, we compute a more accurate gradient estimate g_k, and recompute the step to obtain the new iterate x_{k+1}.

In this strategy one must therefore look ahead, suggesting that a convenient framework for the design and analysis of adaptive sampling methods for constrained optimization is the *proximal gradient method*. In addition to its versatility, the proximal gradient method allows us to expand the range of our investigation to include the composite optimization problem

$$\min_{x \in \mathbb{R}^n} \ \phi(x) = f(x) + h(x), \tag{1.5}$$

where f is a stochastic function and h is a convex (but not necessarily smooth or finite-valued) function. The constrained optimization problem (1.1) can be written in the form (1.5) by defining h to be the convex indicator function for the set Ω.

The goal of this paper is to design an adaptive mechanism for gradually improving the gradient accuracy that can be regarded as an extension of the norm test (1.3) to problems (1.1) and (1.5). We argue in Section 3 that the condition (1.4), with ϕ replacing f, can be used to build such a mechanism within a proximal gradient method. Although condition (1.4) appears to be impractical since it involves $\mathbb{E}_k\left[\overline{x}_{k+1}\right]$, we show how to approximate it in practice. The proposed algorithm reacts to information observed during the course of the iteration, as opposed to methods that dictate the increase in the gradient size *a priori*. Specifically, it has been established in [13] that for a stochastic proximal gradient method in which the sample size grows like a^k, with $a > 1$, convergence can be assured in the convex case. However, the behavior of the algorithm depends very strongly on the value of a, and there are no clear guidelines on how to choose it for a given problem.

N.B. As this paper was being readied for publication, we became aware that [1], which deals with a similar subject, had just been posted. The two papers differ, however, in various ways in their treatment of the topic.

1.1. Literature Review. A deterministic version of the norm test was used by Carter [8] in the design of a trust region method for unconstrained optimization that employs inexact gradients. Friedlander and Schmidt [11] propose increasing the sample size geometrically for the solution of the finite-sum problem, establish a linear convergence result, and report numerical tests with a quasi-Newton method. Byrd et al. [7] studied the expected risk minimization problem and provide a complexity result for the geometric growth condition. That paper also introduces the stochastic version of the norm test (1.3), and reports results with a Newton-like method. Bollapragada et al. [5] introduced a variant of the norm test, called the the inner product test, which is designed to improve the practical efficiency of the method at the price of weakening the theoretical convergence guarantees. (An adaption of this test to problems (1.1) and (1.5) is presented in Section 4.) Adaptive sampling methods have also been studied by Cartis and Scheinberg [9], who establish a global rate of convergence of unconstrained optimization methods that (implicitly) satisfy the norm condition. Pasupathy et al. [16] study sampling rates in stochastic recursions. Roosta et al. [17, 18] analyze sub-sampled Newton methods with adaptive sampling, and De et al. [10] study automatic inference with adaptive sampling.

There is a large literature on proximal gradient methods for solving composite optimization problems; see e.g. [2] and the references therein. Some of these studies consider inexact gradients

[19], but these studies do not propose an automatic procedure for improving the quality of the gradient. [1]

2. Outline of the Algorithm. Since the constrained optimization problem (1.1) is a special case of the composite problem (1.5), we focus on the latter and state the problem under consideration as

$$\min_{x\in\mathbb{R}^n} \ \phi(x) = f(x) + h(x), \quad \text{where } f(x) = \mathbb{E}_{\theta\sim\Theta}\left[F(x,\theta)\right]. \tag{2.1}$$

Here, $F(\cdot,\theta) : \mathbb{R}^n \to \mathbb{R}$ is a smooth function, θ is a random variable with support Θ, and $h : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is a convex (and generally nonsmooth) function. A popular method for solving composite optimization problems is the proximal gradient method (see e.g. [3]), which in the context of problem (2.1) is given as

$$x_{k+1} \leftarrow \operatorname*{argmin}_{x\in\mathbb{R}^n} \ f(x_k) + g_k^T(x - x_k) + \frac{1}{2\alpha_k}\|x - x_k\|^2 + h(x), \quad \text{with } \ 0 < \alpha_k \leq \tfrac{1}{L}, \tag{2.2}$$

where g_k is an unbiased estimator of $\nabla f(x_k)$ and L is a Lipschitz constant defined below. Here and henceforth, $\|\cdot\|$ denotes the Euclidean norm. As is well known, we can also write this iteration as

$$x_{k+1} = \operatorname{prox}_{\alpha_k h}\left(x_k - \alpha_k g_k\right), \tag{2.3}$$

where

$$\operatorname{prox}_{\alpha_k h}(z_k) = \operatorname*{argmin}_{x\in\mathbb{R}^d} \ h(x) + \frac{1}{2\alpha_k}\left\|x - z_k\right\|^2. \tag{2.4}$$

The proposed adaptive sampling proximal gradient algorithm proceeds in two stages. At a given iterate x_k, it first computes a gradient approximation (using the current sample size) as well as a proximal gradient step. Based on information gathered from this step, it computes a second proximal gradient step that determines the new iterate x_{k+1}. An outline of this method is given in Algorithm 1.

[1] An exception is [1], which as mentioned above, was released very shortly before this paper was posted.

Algorithm 1: Outline of Adaptive Sampling Algorithm for Solving Problem (2.1):

Input: x_0, sample size $S \in \mathbb{N}^+$, and sequence $\{\alpha_k > 0\}$.

For k=1, . . .:

1. Draw S i.i.d. samples $\{\theta_0, \theta_1, \cdots, \theta_{S-1}\}$ from Θ, compute

$$\overline{g}_k = \frac{1}{S} \sum_{i=0}^{S-1} \nabla_x F(x, \theta_i), \tag{2.5}$$

and the proximal gradient step

$$\overline{x}_{k+1} = \text{prox}_{\alpha_k h}\left(x_k - \alpha_k \overline{g}_k\right). \tag{2.6}$$

2. Determine the new sample size $S_k \geq S$ (see the next section).
3. **If** $S_k > S$

re-sample S_k i.i.d. samples $\left\{\hat{\theta}_0, \hat{\theta}_1, \cdots, \hat{\theta}_{S_k} - 1\right\}$ from Θ, and compute:

$$g_k = \frac{1}{S_k} \sum_{i=0}^{S_k-1} \nabla_x F(x, \hat{\theta}_i) \tag{2.7}$$

$$x_{k+1} = \text{prox}_{\alpha_k h}\left(x_k - \alpha_k g_k\right) \tag{2.8}$$

Else

$$x_{k+1} = \overline{x}_{k+1}.$$

4. Set $S \leftarrow S_k$

End For

As discussed in Section 3.5, when $S_k > S$, one can reuse the samples from Step 1, and in Step 3 only gather $(S_k - S)$ additional i.i.d. samples $\left\{\hat{\theta}_S, \hat{\theta}_{S+1}, \cdots, \hat{\theta}_{S_k-1}\right\}$ from Θ.

The unspecified parts of this algorithm are the steplength sequence $\{\alpha_k\}$ and the determination of a sample size S_k in Step 2. The analysis in the next section provides the elements for making those decisions. One requirement of the strategy used in Step 2-3 is that, when h is not present, Algorithm 1 should reduce to the iteration (1.2)-(1.3), i.e., to an adaptive sampling gradient method using the norm test to control the sample size.

3. Derivation of the Algorithm. To motivate our approach for determining the sample size S_k, we begin by deriving a fundamental condition (see (3.6) below) that ensures that the steps are good enough to ensure convergence in expectation. The rest of the derivation of the algorithm consist of devising a procedure for approximating condition (3.6) in practice.

3.1. A Fundamental Inequality. We recall that a function $f : \mathbb{R}^n \to \mathbb{R}$ is μ-strongly convex (with $\mu > 0$) iff

$$f(\gamma x + (1-\gamma)y) \leq \gamma f(x) + (1-\gamma) f(y) - \frac{\mu}{2}\gamma(1-\gamma)\left\|y - x\right\|^2, \ \forall x, y \in \mathbb{R}^n, \ \forall \gamma \in [0,1]. \tag{3.1}$$

We also have that if f is a strongly convex and differentiable function, then

$$f(x) \geq f(y) + \nabla f(y)^T (x - y) + \frac{\mu}{2}\|x - y\|^2, \ \forall x \in \mathbb{R}^n. \tag{3.2}$$

If a function is continuously differentiable, μ-strongly convex, and has a Lipschitz continuous gradient with Lipschitz constant L, we say that f is $[\mu, L]$–smooth. We make the following assumptions about problem (2.1).

ASSUMPTIONS 3.1. *$f : \mathbb{R}^n \to \mathbb{R}$ is a $[\mu, L]$-smooth function and $h : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is a closed convex and proper function.*

These assumptions imply that the objective function ϕ defined in (2.1) is strongly convex, and we denote its minimizer by x^* and the minimum objective value by ϕ^*. We consider the stochastic proximal gradient method (2.2) where g_k is an unbiased estimator of $\nabla f(x_k)$ adapted to the filtration $\mathbb{T}$ generated as $\mathbb{T}_k = \sigma\left(x_0, g_0, g_1, \cdots, g_{k-1}\right)$. In other words, we assume that

$$\mathbb{E}\left(g_k | \mathbb{T}_k\right) = \nabla f(x_k). \tag{3.3}$$

For simplicity, we denote conditional expectation as $\mathbb{E}_k\left[\cdot\right] = \mathbb{E}\left(\cdot|\mathbb{T}_k\right)$ and conditional variance as $\mathrm{Var}_k\left[\cdot\right] = \mathbb{E}\left[\|\cdot\|^2 |\mathbb{T}_k\right] - \|\mathbb{E}\left(\cdot|\mathbb{T}_k\right)\|^2$. In what follows, we let $f_k, \nabla f_k$ denote $f(x_k), \nabla f(x_k)$, and similarly for other functions. We begin by establishing a technical lemma that provides the first stepping stone in our analysis.

LEMMA 3.2. *Suppose that Assumptions 3.1 hold and that $\{x_k\}$ is generated by iteration (2.2), where g_k satisfies (3.3). Then,*

$$\begin{aligned}\mathbb{E}_k\left[\phi_{k+1} - \phi^*\right] \leq (1 - \mu\alpha_k)(\phi_k - \phi^*) + \mathbb{E}_k\left[(\nabla f_k - g_k)^T(x_{k+1} - x_k)\right] \\ - \left(\frac{1}{2\alpha_k} - \frac{L}{2}\right)\mathbb{E}_k\left[\|x_{k+1} - x_k\|^2\right].\end{aligned}$$

Proof. By Assumptions 3.1, we have that for any fixed $x_k \in \mathbb{R}^n$,

$$\begin{aligned}\phi_{k+1} \leq\ & f_k + \nabla f_k^T(x_{k+1} - x_k) + \frac{L}{2}\|x_{k+1} - x_k\|^2 + h_{k+1} \\ =\ & f_k + g_k^T(x_{k+1} - x_k) + \frac{1}{2\alpha_k}\|x_{k+1} - x_k\|^2 + h_{k+1} + (\nabla f_k - g_k)^T(x_{k+1} - x_k) \\ & - \left(\frac{1}{2\alpha_k} - \frac{L}{2}\right)\|x_{k+1} - x_k\|^2 \\ \leq\ & f_k + g_k^T(x - x_k) + \frac{1}{2\alpha_k}\|x - x_k\|^2 + h(x) + (\nabla f_k - g_k)^T(x_{k+1} - x_k) \\ & - \left(\frac{1}{2\alpha_k} - \frac{L}{2}\right)\|x_{k+1} - x_k\|^2 \quad \text{(for any } x \in \mathbb{R}^n \text{ by definition (2.2) of } x_{k+1}) \\ =\ & f_k + \nabla f_k^T(x - x_k) + \frac{1}{2\alpha_k}\|x - x_k\|^2 + h(x) + (\nabla f_k - g_k)^T(x_{k+1} - x_k) \\ & + (g_k - \nabla f_k)^T(x - x_k) - \left(\frac{1}{2\alpha_k} - \frac{L}{2}\right)\|x_{k+1} - x_k\|^2 \\ \leq\ & \phi(x) + \left(\frac{1}{2\alpha_k} - \frac{\mu}{2}\right)\|x - x_k\|^2 + (\nabla f_k - g_k)^T(x_{k+1} - x_k) \\ & + (g_k - \nabla f_k)^T(x - x_k) - \left(\frac{1}{2\alpha_k} - \frac{L}{2}\right)\|x_{k+1} - x_k\|^2 \quad \text{(by (3.2))}.\end{aligned}$$

This inequality holds for any $x \in \mathbb{R}^n$. Let us substitute

$$x \leftarrow \widetilde{x}_k = \beta x^* + (1 - \beta)x_k, \quad \text{with} \quad \beta = \mu\alpha_k, \tag{3.4}$$

in the relation above. Recalling the definition (3.1) of strong convexity, we obtain

$$\begin{aligned}\phi_{k+1} \leq\ & \phi(\widetilde{x}_k) + \left(\frac{1}{2\alpha_k} - \frac{\mu}{2}\right)\|\widetilde{x}_k - x_k\|^2 + (\nabla f_k - g_k)^T(x_{k+1} - x_k) \\ & + (g_k - \nabla f_k)^T(\widetilde{x}_k - x_k) - \left(\frac{1}{2\alpha_k} - \frac{L}{2}\right)\|x_{k+1} - x_k\|^2 \\ \leq\ & \beta\phi^* + (1-\beta)\phi_k \underbrace{-\frac{\mu}{2}\beta(1-\beta)\|x^* - x_k\|^2 + \left(\frac{1}{2\alpha_k} - \frac{\mu}{2}\right)\|\widetilde{x}_k - x_k\|^2}_{\text{term 1}} \\ & + (\nabla f_k - g_k)^T(x_{k+1} - x_k) + (g_k - \nabla f_k)^T(\widetilde{x}_k - x_k) - \left(\frac{1}{2\alpha_k} - \frac{L}{2}\right)\|x_{k+1} - x_k\|^2.\end{aligned}$$

Term 1 can be written as

$$\begin{aligned}& -\frac{\mu}{2}\beta(1-\beta)\|x^* - x_k\|^2 + \left(\frac{1}{2\alpha_k} - \frac{\mu}{2}\right)\beta^2\|x^* - x_k\|^2 \\ =& -\frac{\mu}{2}\beta(1-\beta)\|x^* - x_k\|^2 + \left(\frac{1-\mu\alpha_k}{2\alpha_k}\right)\beta^2\|x^* - x_k\|^2 \\ =& \beta(1-\beta)\|x^* - x_k\|^2\left(\frac{\beta}{2\alpha_k} - \frac{\mu}{2}\right) \\ =& 0 \qquad (3.5)\end{aligned}$$

since $\beta = \mu\alpha_k$. Therefore,

$$\begin{aligned}\phi_{k+1} \leq\ & \beta\phi^* + (1-\beta)\phi_k + (\nabla f_k - g_k)^T(x_{k+1} - x_k) + (g_k - \nabla f_k)^T(\widetilde{x}_k - x_k) \\ & - \left(\frac{1}{2\alpha_k} - \frac{L}{2}\right)\|x_{k+1} - x_k\|^2.\end{aligned}$$

Taking conditional expectation, noting that $\widetilde{x}_k \in \mathbb{T}_k$, and recalling (3.3), we have

$$\begin{aligned}\mathbb{E}_k[\phi_{k+1}] \leq\ & \beta\phi^* + (1-\beta)\phi_k + \mathbb{E}_k\left[(\nabla f_k - g_k)^T(x_{k+1} - x_k)\right] \\ & - \left(\frac{1}{2\alpha_k} - \frac{L}{2}\right)\mathbb{E}_k\left[\|x_{k+1} - x_k\|^2\right],\end{aligned}$$

and by the definition of β we conclude that

$$\begin{aligned}\mathbb{E}_k[\phi_{k+1} - \phi^*] \leq\ & (1-\mu\alpha_k)(\phi_k - \phi^*) + \mathbb{E}_k\left[(\nabla f_k - g_k)^T(x_{k+1} - x_k)\right] \\ & - \left(\frac{1}{2\alpha_k} - \frac{L}{2}\right)\mathbb{E}_k\left[\|x_{k+1} - x_k\|^2\right].\end{aligned}$$

∎

From this result, we can readily establish conditions under which the proximal gradient iteration, with a fixed steplength $\alpha_k = \alpha$, achieves Q-linear convergence of ϕ_k, in expectation.

THEOREM 3.3. *Suppose that Assumptions 3.1 hold, that $\{x_k\}$ is generated by (2.2) with $\alpha_k = (1-\eta)/L$ for $\eta \in (0,1)$, and that g_k satisfies (3.3). If we have that for all k,*

$$\alpha_k\mathbb{E}_k\left[(\nabla f_k - g_k)^T(x_{k+1} - x_k)\right] \leq \frac{\eta}{2}\mathbb{E}_k\left[\|x_{k+1} - x_k\|^2\right] \qquad (3.6)$$

then

$$\mathbb{E}_k[\phi_{k+1} - \phi^*] \leq \left[1 - (1-\eta)\frac{\mu}{L}\right](\phi_k - \phi^*). \qquad (3.7)$$

We note that when $h = 0$ and the iteration becomes (1.2), condition (3.6) reduces to the norm test (1.3) with ξ given by $\eta/(1-\eta)$.

The assumption on α_k in Theorem 3.3 is fairly standard. The key is inequality (3.6), which is the most general condition we have identified for ensuring linear convergence. However, it does not seem to be possible to enforce this condition in practice, even approximately, for the composite optimization problem (which includes convex constrained optimization). Therefore, we seek an implementable version of (3.6), even if it is more restrictive. Before doing so, we show that condition (3.6) can also be used to establish convergence in the case when f is convex, but not strongly convex.

3.2. Convergence for General Convex f. We now show that when f is convex, the sequence of function values $\{\phi(x_k)\}$ converges to the optimal value ϕ^* of problem (2.1) at a sublinear rate, in expectation. To establish this result, we make the following assumptions.

ASSUMPTIONS 3.4. *$f : \mathbb{R}^n \to \mathbb{R}$ is convex, differentiable and has an $L-$ Lipschitz continuous gradient, and $h : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is a closed, convex and proper function.*

We begin by proving a technical lemma.

LEMMA 3.5. *Suppose that Assumptions (3.4) hold and that $\{x_k\}$ is generated by iteration (2.2), where g_k satisfies (3.3) and $\alpha_k = (1-\eta)/L$ for $\eta \in (0,1)$. If in addition condition (3.6) is satisfied, we have that for any given $z \in \mathbb{T}_k$,*

$$\mathbb{E}_k[\phi_{k+1}] \leq \phi(z) + \frac{1}{\alpha_k}\mathbb{E}_k\left[(x_k - x_{k+1})^T(x_k - z)\right] - \frac{1}{2\alpha_k}\mathbb{E}_k[\|x_k - x_{k+1}\|^2]. \tag{3.8}$$

Proof. By Assumptions (3.4), we have that

$$\begin{aligned}
\phi_{k+1} &\leq f_k + \nabla f_k^T(x_{k+1} - x_k) + \frac{L}{2}\|x_{k+1} - x_k\|^2 + h_{k+1} \\
&\leq f(z) - \nabla f_k^T(z - x_k) + \nabla f_k^T(x_{k+1} - x_k) + \frac{L}{2}\|x_{k+1} - x_k\|^2 + h_{k+1} \quad \text{(by convexity of } f\text{)} \\
&\leq f(z) - \nabla f_k^T(z - x_k) + \nabla f_k^T(x_{k+1} - x_k) + \frac{L}{2}\|x_{k+1} - x_k\|^2 + h(z) \\
&\quad - \left(\frac{x_k - x_{k+1}}{\alpha_k} - g_k\right)^T (z - x_{k+1}) \quad \text{(by convexity of } h \text{ and definition of } x_{k+1}\text{)} \\
&= \phi(z) - \nabla f_k^T(z - x_k) + \nabla f_k^T(x_{k+1} - x_k) + \frac{L}{2}\|x_{k+1} - x_k\|^2 \\
&\quad - \left(\frac{x_k - x_{k+1}}{\alpha_k} - g_k\right)^T (z - x_k + x_k - x_{k+1}) \\
&= \phi(z) + (g_k - \nabla f_k)^T(z - x_k) + \frac{1}{\alpha_k}(x_k - x_{k+1})^T(x_k - z) \\
&\quad + \left(\frac{L}{2} - \frac{1}{\alpha_k}\right)\|x_k - x_{k+1}\|^2 + (\nabla f_k - g_k)^T(x_{k+1} - x_k), \quad \text{(rearraging terms)}
\end{aligned}$$

where the third inequality follows from the fact that $0 \in g_k + \partial h_{k+1} + \frac{x_k - x_{k+1}}{\alpha_k}$. Taking conditional

expectation and using (3.6), we have

$$\begin{aligned}
\mathbb{E}_k[\phi_{k+1}] &\leq \phi(z) + \mathbb{E}_k[(g_k - \nabla f_k)^T(z - x_k)] + \frac{1}{\alpha_k}\mathbb{E}_k[(x_k - x_{k+1})^T(x_k - z)] \\
&\quad + \left(\frac{L}{2} - \frac{1}{\alpha_k}\right)\mathbb{E}_k[\|x_k - x_{k+1}\|^2] + \mathbb{E}_k[(\nabla f_k - g_k)^T(x_{k+1} - x_k)] \\
&= \phi(z) + \frac{1}{\alpha_k}\mathbb{E}_k[(x_k - x_{k+1})^T(x_k - z)] + \left(\frac{L}{2} - \frac{1}{\alpha_k}\right)\mathbb{E}_k[\|x_k - x_{k+1}\|^2] \\
&\quad + \mathbb{E}_k[(\nabla f_k - g_k)^T(x_{k+1} - x_k)] \quad (\text{since } z \in \mathbb{T}_k\ ) \\
&\leq \phi(z) + \frac{1}{\alpha_k}\mathbb{E}_k[(x_k - x_{k+1})^T(x_k - z)] - \left(\frac{1}{\alpha_k} - \frac{L}{2} - \frac{\eta}{2\alpha_k}\right)\mathbb{E}_k[\|x_k - x_{k+1}\|^2] \quad (\text{by (3.6)}) \\
&= \phi(z) + \frac{1}{\alpha_k}\mathbb{E}_k[(x_k - x_{k+1})^T(x_k - z)] - \frac{1}{2\alpha_k}\mathbb{E}_k[\|x_k - x_{k+1}\|^2,
\end{aligned}$$

where the last equality is due to $\alpha_k = \frac{1-\eta}{L}$. ∎

THEOREM 3.6. *Suppose that Assumptions (3.4) hold and that $\{x_k\}$ is generated by iteration (2.2), where g_k satisfies (3.3) and $\alpha_k = \alpha = (1-\eta)/L$ for $\eta \in (0,1)$. If in addition (3.6) is satisfied, we have*

$$\mathbb{E}[\phi_k - \phi^*] \leq \frac{L\|x_0 - x^*\|^2}{2(1-\eta)k}, \tag{3.9}$$

where x^* is any optimal solution of problem (2.1).

Proof. From Lemma 3.5, for any $z \in \mathbb{T}_k$, we have

$$\mathbb{E}_k[\phi_{k+1}] \leq \phi(z) + \frac{1}{\alpha}\mathbb{E}_k\left[(x_k - x_{k+1})^T(x_k - z)\right] - \frac{1}{2\alpha}\mathbb{E}_k[\|x_k - x_{k+1}\|^2].$$

Now substituting $z = x^* \in \mathbb{T}_k$ and taking full expectations, we have

$$\begin{aligned}
\mathbb{E}[\phi_{k+1} - \phi^*] &\leq \frac{1}{\alpha}\mathbb{E}\left[(x_k - x_{k+1})^T(x_k - x^*)\right] - \frac{1}{2\alpha}\mathbb{E}[\|x_k - x_{k+1}\|^2] \\
&= \frac{1}{2\alpha}\mathbb{E}\left[2(x_k - x_{k+1})^T(x_k - x^*) - \|x_k - x_{k+1}\|^2\right] \\
&= \frac{1}{2\alpha}\mathbb{E}\left[\|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2\right].
\end{aligned}$$

Summing the above inequality for $k = 0$ to $k-1$, we get

$$\begin{aligned}
\frac{1}{k}\sum_{t=0}^{k-1}\mathbb{E}[\phi_{t+1} - \phi^*] &\leq \frac{1}{2\alpha k}\mathbb{E}\left[\|x_0 - x^*\|^2 - \|x_k - x^*\|^2\right] \\
&\leq \frac{\|x_0 - x^*\|^2}{2\alpha k}.
\end{aligned}$$

By substituting $z = x_k \in \mathbb{T}_k$ in Lemma 3.5, we have that the sequence of expected function values is a deceasing sequence; specifically,

$$\mathbb{E}_k[\phi_{k+1}] \leq \phi_k - \frac{1}{2\alpha}\mathbb{E}_k[\|x_k - x_{k+1}\|^2].$$

Therefore, we have,

$$\mathbb{E}[\phi_k - \phi^*] \leq \frac{1}{k}\sum_{t=0}^{k-1}\mathbb{E}[\phi_{t+1} - \phi^*] \leq \frac{\|x_0 - x^*\|^2}{2\alpha k}.$$ ∎

3.3. A Practical Condition. To obtain a condition that is more amenable to computation than (3.6), we look for an upper bound for the left hand side of this inequality and a lower bound for the right hand side. Imposing an inequality between these two bounds will imply (3.6).

THEOREM 3.7. *Suppose that Assumptions 3.1 hold, that $\{x_k\}$ is generated by (2.2) with $\alpha_k = (1-\eta)/L$ for $\eta \in (0,1)$, and that g_k satisfies (3.3). If g_k additionally satisfies*

$$Var_k\left[g_k\right] \leq \frac{\eta}{2}\left\|\frac{\mathbb{E}_k\left[x_{k+1}\right]-x_k}{\alpha_k}\right\|^2, \tag{3.10}$$

then (3.6) holds and hence

$$\mathbb{E}_k\left[\phi_{k+1}-\phi^*\right] \leq \left[1-(1-\eta)\frac{\mu}{L}\right](\phi_k-\phi^*).$$

If instead of Assumptions 3.1, the weaker Assumptions 3.4 hold, then

$$\mathbb{E}[\phi_k-\phi^*] \leq \frac{L\|x_0-x^*\|^2}{2(1-\eta)k},$$

where x^* is any optimal solution of problem (2.1).

Proof. We first note that in (3.6),

$$\begin{aligned}\mathbb{E}_k\left[\left\|x_{k+1}-x_k\right\|^2\right] &= \left\|\mathbb{E}_k\left[x_{k+1}\right]-x_k\right\|^2 + \mathrm{Var}_k\left[x_{k+1}-x_k\right]\\ &\geq \left\|\mathbb{E}_k\left[x_{k+1}\right]-x_k\right\|^2,\end{aligned}$$

which is a quantity that we can approximate with a sample estimation; as we argue in Section 3.4.

On the other hand, let

$$\hat{x}_{k+1} = \mathrm{prox}_{\alpha_k h}\left(x_k-\alpha_k\nabla f_k\right) \in \mathbb{T}_k.$$

Then, since the prox operator is a contraction mapping,

$$\begin{aligned}&\mathbb{E}_k\left[(\nabla f_k-g_k)^T(x_{k+1}-x_k)\right]\\ &= \mathbb{E}_k\left[(\nabla f_k-g_k)^T(x_{k+1}-\hat{x}_{k+1})\right]+\mathbb{E}_k\left[(\nabla f_k-g_k)^T(\hat{x}_{k+1}-x_k)\right]\\ &= \mathbb{E}_k\left[(\nabla f_k-g_k)^T(x_{k+1}-\hat{x}_{k+1})\right] \quad (\text{since } \hat{x}_{k+1}\in\mathbb{T}_k)\\ &\leq \mathbb{E}_k\left[\|\nabla f_k-g_k\|\,\|x_{k+1}-\hat{x}_{k+1}\|\right]\\ &= \mathbb{E}_k\left[\|\nabla f_k-g_k\|\,\left\|\mathrm{prox}_{\alpha_k h}\left(x_k-\alpha_k g_k\right)-\mathrm{prox}_{\alpha_k h}\left(x_k-\alpha_k\nabla f_k\right)\right\|\right]\\ &\leq \alpha_k\mathbb{E}_k\left[\|\nabla f_k-g_k\|^2\right]\\ &= \alpha_k\mathrm{Var}_k\left[g_k\right].\end{aligned}$$

Thus, we have obtained both

$$\frac{\eta}{2}\|\mathbb{E}_k\left[x_{k+1}\right]-x_k\|^2 \leq \frac{\eta}{2}\mathbb{E}_k\left[\|x_{k+1}-x_k\|^2\right]$$

and

$$\alpha_k\mathbb{E}_k\left[(\nabla f_k-g_k)^T(x_{k+1}-x_k)\right] \leq \alpha_k^2\mathrm{Var}_k\left[g_k\right].$$

Therefore, if we require that

$$\mathrm{Var}_k\left[g_k\right] \leq \frac{\eta}{2}\left\|\frac{\mathbb{E}_k\left[x_{k+1}\right]-x_k}{\alpha_k}\right\|^2,$$

it follows that condition (3.6) is satisfied. ∎

The significance of Theorem 3.7 is that it establishes the convergence of the algorithm under condition (3.10) which, although being more restrictive than condition (3.6), can be approximated empirically, as shown in Section 3.4. Again, it is reassuring that when $h = 0$, condition (3.10) reduces to the norm test (1.3).

3.4. Choice of the Sample Size S_k. We now discuss how to ensure that condition (3.10) is satisfied. At iterate x_k, suppose we select S_k i.i.d. samples $\{\theta_0, \theta_1, \cdots, \theta_{S_k-1}\}$ from Θ, and set

$$g_k = \frac{1}{S_k}\sum_{i=0}^{S_k-1} \nabla_x F(x_k, \theta_i). \tag{3.11}$$

Clearly, (3.3) holds and the variance of g_k is given by

$$\mathrm{Var}_k\left[g_k\right] = \frac{\mathbb{E}_k\left[\|\nabla_x F(x_k,\theta)-\nabla f(x_k)\|^2\right]}{S_k}.$$

Therefore, (3.10) holds if S_k satisfies

$$\frac{\mathbb{E}_k\left[\|\nabla_x F(x_k,\theta)-\nabla f(x_k)\|^2\right]}{S_k} \leq \frac{\eta}{2}\left\|\frac{\mathbb{E}_k\left[x_{k+1}\right]-x_k}{\alpha_k}\right\|^2, \tag{3.12}$$

or

$$S_k \geq \mathbb{E}_k\left[\|\nabla_x F(x_k,\theta)-\nabla f(x_k)\|^2\right] \Bigg/ \frac{\eta}{2}\left\|\frac{\mathbb{E}_k\left[x_{k+1}\right]-x_k}{\alpha_k}\right\|^2. \tag{3.13}$$

This is the theoretical condition suggested by our analysis. Based on this condition we can state

COROLLARY 3.8. *Suppose that Assumptions 3.1 hold, that $\{x_k\}$ is generated by (2.2) with $\alpha_k = (1-\eta)/L$ for $\eta \in (0,1)$ and with g_k given by (3.11). If the sample sizes S_k are chosen to satisfy (3.13) for all k, then (3.6) is satisfied and hence (3.7) holds. If instead of Assumptions 3.1, the weaker Assumptions 3.4 are satisfied, then (3.9) holds.*

In practice, we need to estimate both quantities on the right hand side of (3.13) As was done e.g. in [6, 7] the population variance term in the numerator can be approximated by a sample average:

$$\mathbb{E}_k\left[\|\nabla_x F(x_k,\theta)-\nabla f(x_k)\|^2\right] \approx \frac{1}{S-1}\sum_{i=0}^{S-1}\|\nabla_x F(x_k,\theta_i)-\overline{g}_k\|^2, \tag{3.14}$$

where $\overline{g}_k$ is defined in (2.5). We handle the denominator on the right hand side of (3.13) by approximating $\|\mathbb{E}_k[x_{k+1}]-x_k\|$ with the norm of the trial step $\|\overline{x}_{k+1}-x_k\|$ defined in (2.6), i.e.,

$$\overline{x}_{k+1} = \mathrm{prox}_{\alpha_k h}\left(x_k - \alpha_k \overline{g}_k\right).$$

This is somewhat analogous to the approximations made in the unconstrained case in [6, 7], the main difference being the presence here of the prox operator, which is a contraction. Given this, we can replace (3.13) by

$$S_k \geq \frac{1}{S-1}\sum_{i=0}^{S-1}\|\nabla_x F(x,\theta_i)-\overline{g}_k\|^2 \Bigg/ \frac{\eta}{2}\left\|\frac{\overline{x}_{k+1}-x_k}{\alpha_k}\right\|^2. \tag{3.15}$$

Our algorithm will use condition (3.15) to control the sample size.

3.5. The Practical Adaptive Sampling Algorithm. We now summarize the algorithm proposed in this paper, which employs the aforementioned approximations.

Algorithm 2: Complete Algorithm for Solving Problem (2.1):

Input: x_0, initial sample size $S \in \mathbb{N}^+$, and sequence $\{\alpha_k > 0\}$.

For k=1, . . . :

1. Draw S i.i.d. samples $\{\theta_0, \theta_1, \cdots, \theta_{S-1}\}$ from Θ, compute

$$\overline{g}_k = \frac{1}{S}\sum_{i=0}^{S-1}\nabla_x F(x,\theta_i),$$

and trial proximal gradient step

$$\overline{x}_{k+1} = \mathrm{prox}_{\alpha_k h}\left(x_k - \alpha_k \overline{g}_k\right). \tag{3.16}$$

2. Compute

$$a = \frac{1}{S-1}\sum_{i=0}^{S-1}\|\nabla_x F(x,\theta_i)-\overline{g}_k\|^2 \Bigg/ \frac{\eta}{2}\left\|\frac{\overline{x}_{k+1}-x_k}{\alpha_k}\right\|^2$$

and set

$$S_k = \max\{a, S\}. \tag{3.17}$$

3. If $S_k > S$, choose $(S_k - S)$ additional i.i.d. samples $\{\theta_S, \theta_{S+1}, \cdots, \theta_{S_k-1}\}$ from Θ, and compute:

$$g_k = \frac{1}{S_k}\sum_{i=0}^{S_k-1}\nabla_x F(x,\theta_i) \tag{3.18}$$

$$x_{k+1} = \mathrm{prox}_{\alpha_k h}\left(x_k - \alpha_k g_k\right) \tag{3.19}$$

Else

$$x_{k+1} = \overline{x}_{k+1}.$$

4. Set $S \leftarrow S_k$

End For

As noted above, for large S, the choice of S_k given by (3.17) approximately ensures the

condition (3.10) is satisfied. Computing S_k involves one evaluation of the proximal operator as well as the evaluation S stochastic gradients.

4. Using an Inner-Product Test in Place of the Norm Test. In the unconstrained setting, Bollapragada et al [6] have observed that the norm test, although endowed with optimal theoretical convergence rates, is too demanding in terms of sample size requirements. They derived a practical test called the *inner-product test*, which ensures that the search directions are descent directions with high probability, and performs well with smaller sample sizes. In this section, we extend these ideas to the constrained (or composite) optimization settings, and derive the equivalent inner-product test for adaptively controlling the sample sizes.

The goal is to choose the sample sizes such that the algorithm step provides descent with high probability. We would like to choose a sample size so that $\bar{d}_k = (\bar{x}_{k+1} - x_k)/\alpha_k$ provides decrease on the objective approximation $\nabla f(x_k)^T d + h(x_k + d)$, and specifically so that $\nabla f(x_k)^T \bar{d}_k + h(x_k + \bar{d}_k) - h(x_k) \leq \beta(\bar{g_k}^T \bar{d}_k + h(x_k + \bar{d}_k) - h(x_k))$ for some $\beta \in (0, 1)$. This means we want to satisfy

$$(\nabla f(x_k) - \bar{g_k})^T \bar{d}_k \leq -(1-\beta)\left(\bar{g_k}^T \bar{d}_k + h(x_k + \bar{d}_k) - h(x_k)\right). \tag{4.1}$$

To estimate the left hand side of (4.1), note that in general, given a vector $p \in \mathbb{R}^n$, since $\mathbb{E}_k\left[(\bar{g} - \nabla f(x_k))^T p\right] = 0$, we can estimate the size of the quantity $(\bar{g_k} - \nabla f(x_k))^T p$ by estimating the variance of $(\bar{g_k} - \nabla f(x_k))^T p$. Since the initial sample size is S this is given by

$$\mathrm{Var}_k\left[\bar{g}_k^T p\right] \approx \frac{1}{S_k}\frac{1}{S-1}\sum_{i=0}^{S-1}\left((\nabla_x F(x,\theta_i) - \overline{g}_k)^T p\right)^2. \tag{4.2}$$

We use this estimate in (4.1) with $p = \bar{d}_k$ and get the condition

$$S_k \geq \frac{1}{S-1}\sum_{i=0}^{S-1}\left((\nabla_x F(x,\theta_i) - \overline{g}_k)^T \bar{d}_k\right)^2 \Bigg/ (1-\beta)^2(\bar{g_k}^T \bar{d}_k + h(x_k + \bar{d}_k) - h(x_k))^2. \tag{4.3}$$

where $(1-\beta)^2$ is analogous to $\eta/2$ in (3.15). We are aware that, in using (4.2) with $p = \bar{d}_k$, we are treating $\bar{g_k}$ and $\bar{d}_k$ as independent while they are not, but the practical success of the inner product test in [6] indicates that this approach is worthy of exploration.

We also note that in a problem with convex constraints, where $h(\cdot)$ involves a convex indicator function with possibly infinite values, the algorithm will only generate feasible points x_k and $x_k + \bar{d}_k$, so that in the above discussion h only takes on finite values at those points.

The version of our algorithm using this approach consists of following Algorithm 2 with the right hand side of (4.3) used in place of the formula for a in Step 2.

5. Numerical Experiments. We conducted numerical experiments to illustrate the performance of the proposed algorithms. We consider binary classification problems where the objective function is given by the logistic loss with ℓ_1-regularization:

$$\phi(x) = \frac{1}{N}\sum_{i=1}^{N}\log(1 + \exp(-y^i x^T z^i)) + \lambda\|x\|_1. \tag{5.1}$$

Here $\{(z^i, y^i),\ i = 1, \ldots, N\}$ are the input output data pairs, and the regularization parameter is chosen as $\lambda = 1/N$. This problem falls into the general category of minimizing composite optimization problems of the form (1.5), where $f(x) = \mathbb{E}\left[\log(1 + \exp(-y^i x^T z^i))\right]$, with expectation taken over a discrete uniform probability distribution defined on the dataset, and $h(x) = \lambda\|x\|_1$. We use the data sets listed in Table 5.1.

Data Set	Data Points N	Variables d	Reference
`covertype`	581012	54	[4]
`gisette`	6000	5000	[12]
`ijcnn`	35000	22	[15]
`MNIST`	60000	784	[14]
`mushrooms`	8124	112	[15]
`sido`	12678	4932	[15]

TABLE 5.1
Characteristics of the binary datasets used in the experiments.

We implemented three different variants of proximal stochastic gradient methods where the batch sizes are either: (1) continuously increased at a geometric rate (labelled `GEOMETRIC`), i.e.,

$$S_k = \left\lceil S_0 \left(1+\gamma\right)^k \right\rceil; \tag{5.2}$$

where $\gamma > 0$ is a parameter that will be varied in the experiments; (2) adaptively chosen based on the norm test (3.15) (labelled `NORM`); or (3) adaptively chosen based on the inner-product test (4.3) (labelled `IP`). The steplength parameter α_k in each method is chosen as the number in the set $\{2^{-10}, 2^{-7}, \cdots, 2^{15}\}$ that leads to best performance. The initial sample size was set to $S_0 = 2$. The methods are terminated if $\|x_{k+1} - x_k\|/\alpha_k \leq 10^{-8}$ or if 100 epochs (passes through entire dataset) are performed. An approximation ϕ^* of the optimal function value was computed for each problem by running the deterministic proximal gradient method for 50,000 iterations.

Figures 5.1 and 5.2 report the performance of the three methods for the dataset `mushroom`, for various values of the parameter γ in (5.2) and η in (3.15) and (4.3) (η in (4.3) corresponds to $2(1-\beta)^2$). Figure 5.1, the vertical axis measures the optimality gap, $\phi(x) - \phi^*$, and the horizontal axis measures the number of effective gradient evaluations, defined as $\sum_{j=0}^{k} S_j/N$. In Figure 5.2, the vertical axis measures the batchsize as a fraction of total number of data points N, and the horizontal axis measures the number of iterations. The results for other datasets in Table 5.1 can be found in Appendix A.

We observe that the inner product test is the most efficient in terms of effective gradient evaluations, which is indicative of the total computational work and CPU time. For the geometric strategy, smaller values of γ typically lead to better performance, as they prevent the batch size from growing too rapidly. The norm test gives a performance comparable to the best runs of the geometric strategy, but the latter has much higher variability. In other words, whereas the geometric strategy can be quite sensitive to the choice of γ, the norm and inner product tests are fairly insensitive to the choice of η.

6. Final Remarks. Algorithms that adaptively improve the quality of the approximate gradient during the optimization process are of interest from theoretical and practical perspectives, and have been well-studied in the context of unconstrained optimization. In this paper, we proposed an adaptive method for solving constrained and composite optimization problems. The cornerstone of the proposed algorithm and its analysis is condition (3.6), which we regard as a natural generalization of the well-known norm test from unconstrained optimization. As this condition is difficult to implement precisely in practice, we approximate it by condition (3.10). We are able to prove convergence for the resulting methods under standard conditions. It remains to be seen whether there is a condition with similar properties as (3.6), that is amenable to computation and less restrictive than (3.10). In this paper, we also proposed a practical inner-product condition (4.1) that extends the ideas proposed in the unconstrained settings, and is more efficient in practice than the norm condition.

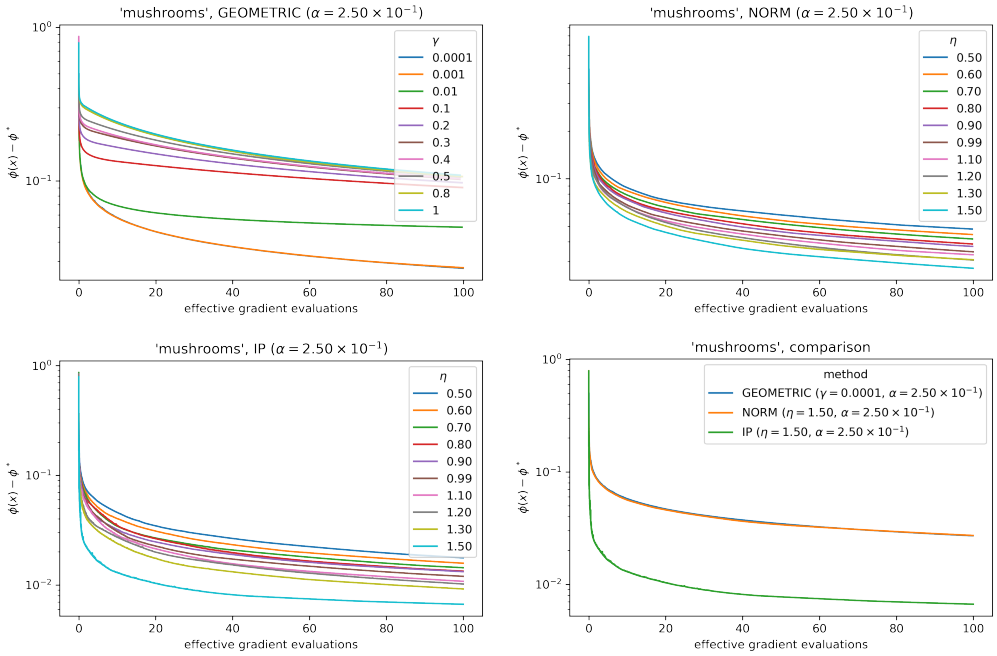

FIG. 5.1. *Optimality gap $\phi(x_k) - \phi^*$ against effective gradient evaluations on dataset* `mushrooms`*, with different strategies to control batch size: geometric increase (top left), norm test (top right), inner-product test (bottom left), and comparison between the best run for each method (bottom right).*

REFERENCES

[1] Florian Beiser, Brendan Keith, Simon Urbainczyk, and Barbara Wohlmuth. Adaptive sampling strategies for risk-averse stochastic optimization with constraints. *arXiv preprint arXiv:2012.03844*, 2020.

[2] Dimitri P Bertsekas. *Convex Optimization Algorithms*. Athena Scientific, 2015.

[3] Dimitri P Bertsekas, Angelia Nedić, and Asuman E Ozdaglar. *Convex analysis and optimization*. Athena Scientific Belmont, 2003.

[4] Jock A Blackard and Denis J Dean. Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Computers and electronics in agriculture*, 24(3):131–151, 1999.

[5] Raghu Bollapragada, Richard Byrd, and Jorge Nocedal. Adaptive sampling strategies for stochastic optimization. *arXiv preprint arXiv:1710.11258*, 2017.

[6] Raghu Bollapragada, Richard Byrd, and Jorge Nocedal. Adaptive sampling strategies for stochastic optimization. *SIAM Journal on Optimization*, 28(4):3312–3343, 2018.

[7] Richard H Byrd, Gillian M Chin, Jorge Nocedal, and Yuchen Wu. Sample size selection in optimization methods for machine learning. *Mathematical Programming*, 134(1):127–155, 2012.

[8] Richard G Carter. On the global convergence of trust region algorithms using inexact gradient information. *SIAM Journal on Numerical Analysis*, 28(1):251–265, 1991.

[9] Coralia Cartis and Katya Scheinberg. Global convergence rate analysis of unconstrained optimization methods based on probabilistic models. *Mathematical Programming*, pages 1–39, 2015.

[10] Soham De, Abhay Yadav, David Jacobs, and Tom Goldstein. Automated inference with adaptive batches. In *Artificial Intelligence and Statistics*, pages 1504–1513, 2017.

[11] Michael P Friedlander and Mark Schmidt. Hybrid deterministic-stochastic methods for data fitting. *SIAM Journal on Scientific Computing*, 34(3):A1380–A1405, 2012.

[12] Isabelle Guyon, Steve Gunn, Asa Ben-Hur, and Gideon Dror. Result analysis of the NIPS 2003 feature selection challenge. In *Advances in neural information processing systems*, pages 545–552, 2004.

[13] Afrooz Jalilzadeh, Uday V Shanbhag, Jose H Blanchet, and Peter W Glynn. Optimal smoothed variable sample-size accelerated proximal methods for structured nonsmooth stochastic convex programs. *arXiv preprint arXiv:1803.00718*, 2018.

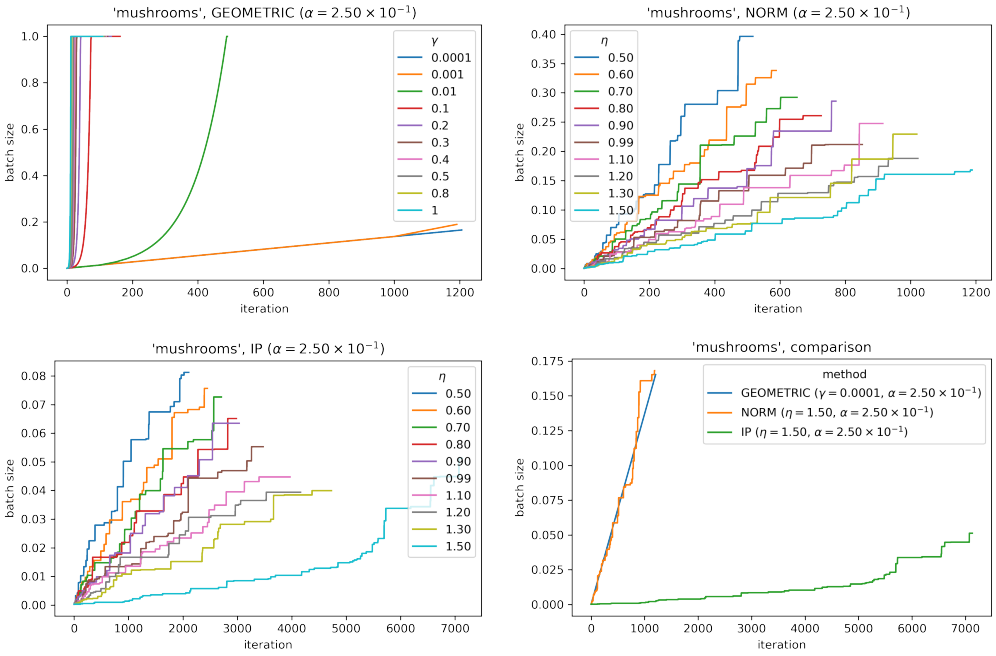

FIG. 5.2. *Batch size (as a fraction of total number of data points N) against iterations on dataset* `mushrooms`*, with different strategies to control batch size: geometric increase (top left), norm test (top right), inner-product test (bottom left), and comparison between the best run for each method (bottom right).*

[14] Yann LeCun, Corinna Cortes, and Christopher JC Burges. MNIST handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2010.

[15] Moshe Lichman. UCI machine learning repository. http://archive.ics.uci.edu/ml, 2013.

[16] Raghu Pasupathy, Peter Glynn, Soumyadip Ghosh, and Fatemeh S Hashemi. On sampling rates in stochastic recursions. 2015. Under Review.

[17] Farbod Roosta-Khorasani and Michael W Mahoney. Sub-sampled Newton methods I: Globally convergent algorithms. *arXiv preprint arXiv:1601.04737*, 2016.

[18] Farbod Roosta-Khorasani and Michael W Mahoney. Sub-sampled Newton methods II: Local convergence rates. *arXiv preprint arXiv:1601.04738*, 2016.

[19] Mark Schmidt, Nicolas Roux, and Francis Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24, pages 1458–1466. Curran Associates, Inc., 2011.

Appendix A. Additional Numerical Experiments. Here we present the numerical experiments for remaining data sets.

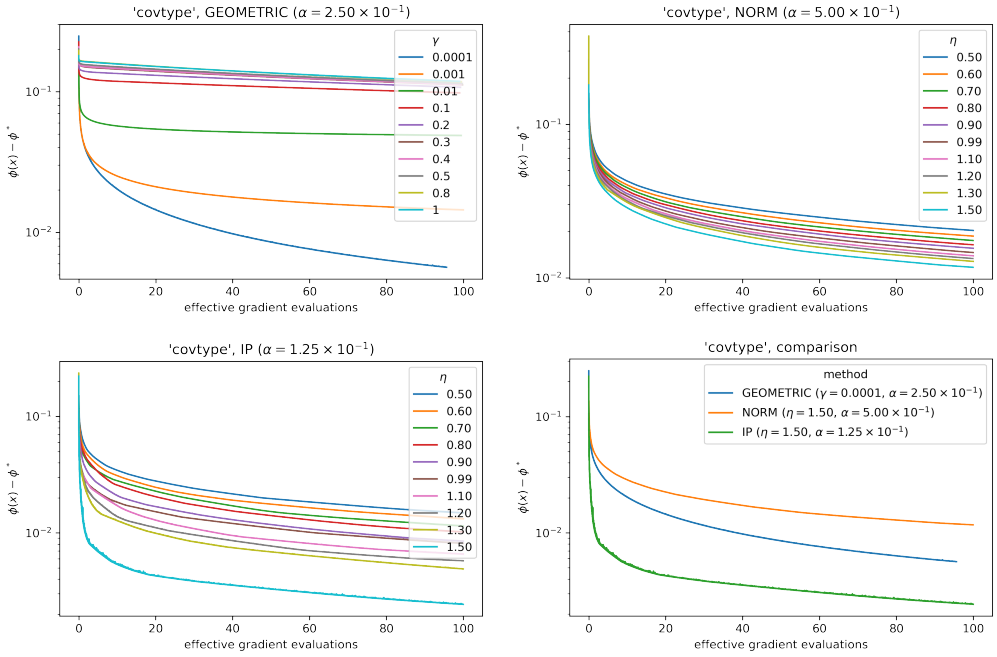

FIG. A.1. *Optimality gap $\phi(x_k) - \phi^*$ against effective gradient evaluations on dataset* `covtype`*, with different strategies to control batch size: geometric increase (top left), norm test (top right), inner-product test (bottom left), and comparison between the best run for each method (bottom right).*

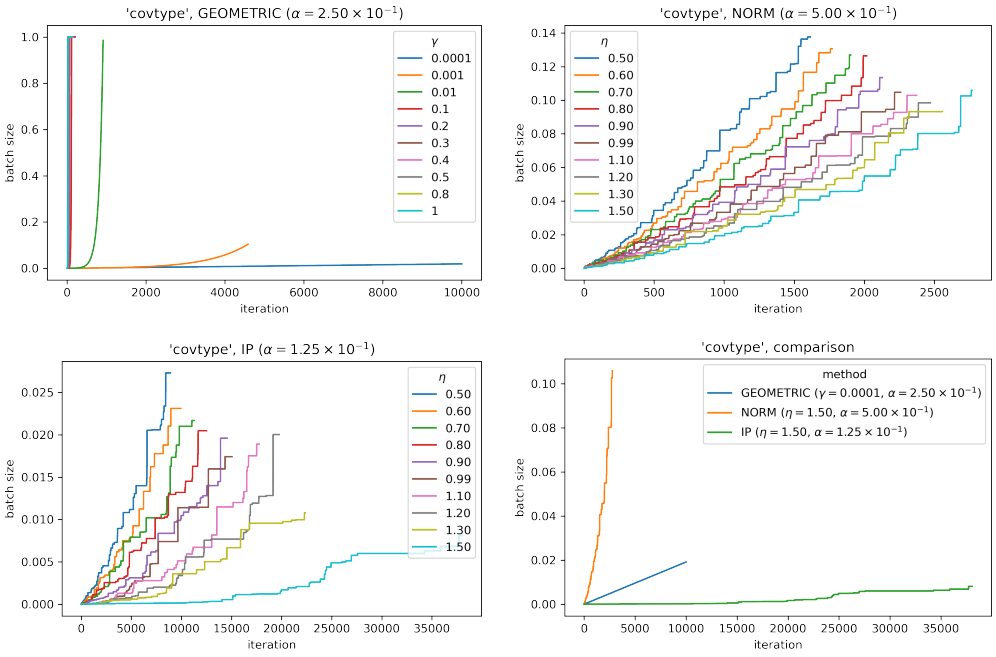

FIG. A.2. *Batch size (as a fraction of total number of data points N) against iterations on dataset* **`covtype`**, *with different strategies to control batch size: geometric increase (top left), norm test (top right), inner-product test (bottom left), and comparison between the best run for each method (bottom right).*

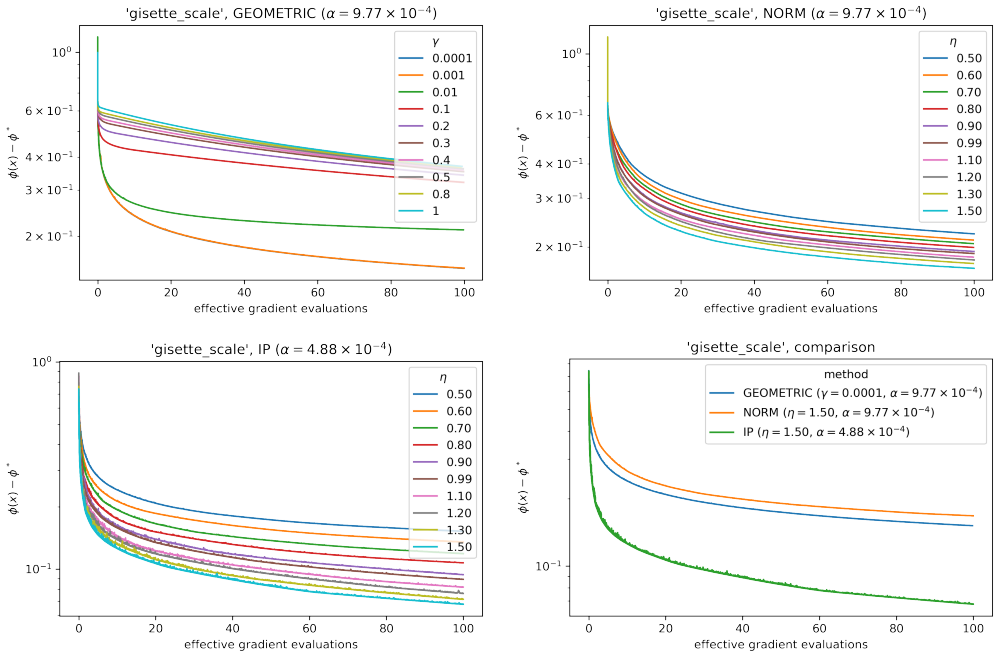

FIG. A.3. *Optimality gap $\phi(x_k) - \phi^*$ against effective gradient evaluations on dataset* `gisette_scale`*, with different strategies to control batch size: geometric increase (top left), norm test (top right), inner-product test (bottom left), and comparison between the best run for each method (bottom right).*

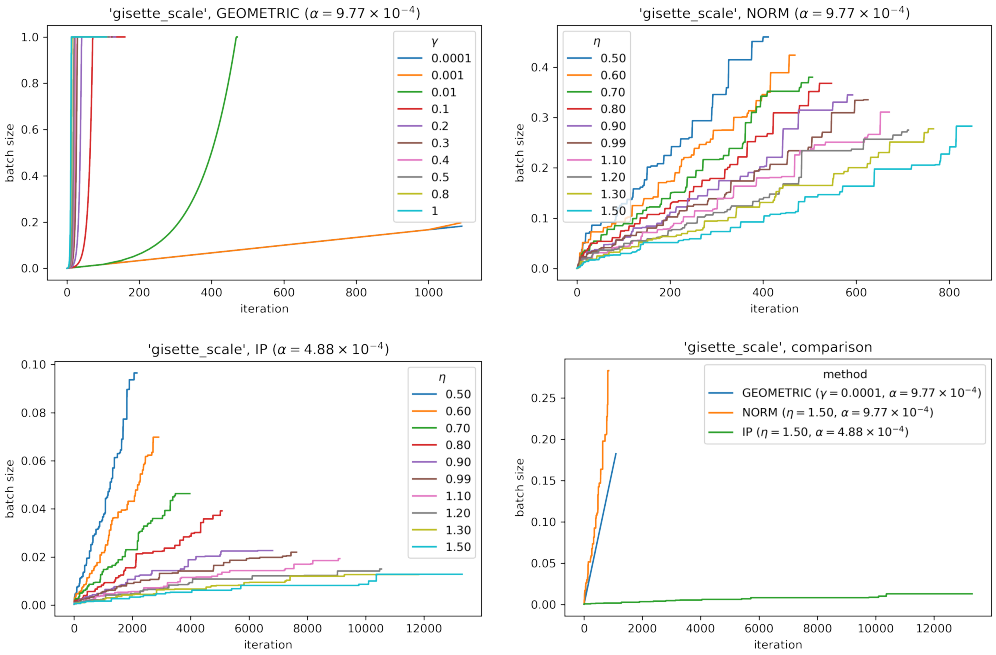

FIG. A.4. *Batch size (as a fraction of total number of data points N) against iterations on dataset* `gisette_scale`*, with different strategies to control batch size: geometric increase (top left), norm test (top right), inner-product test (bottom left), and comparison between the best run for each method (bottom right).*

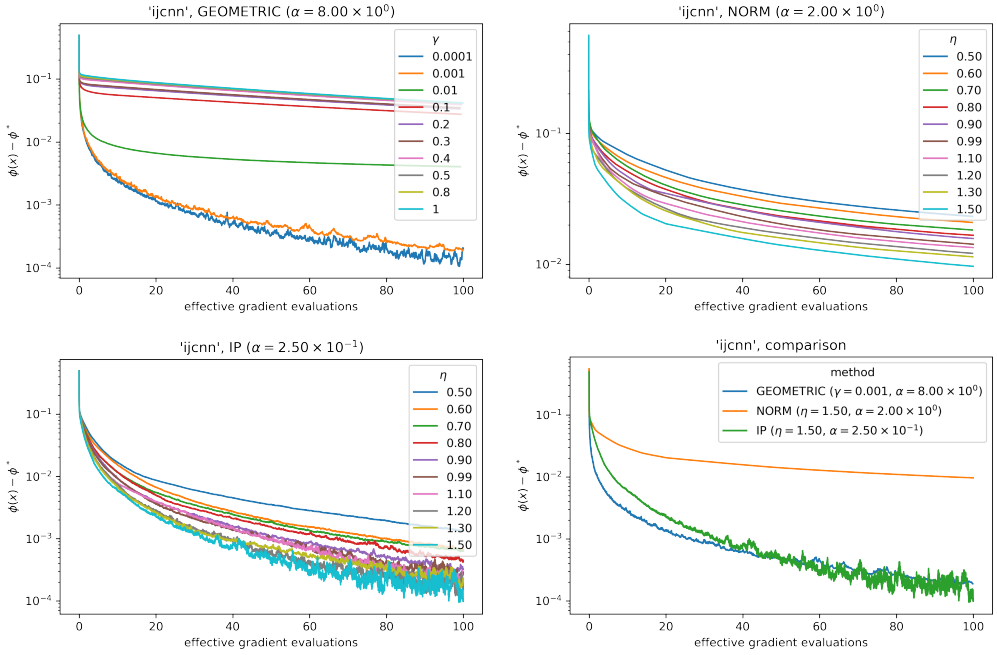

FIG. A.5. *Optimality gap $\phi(x_k) - \phi^*$ against effective gradient evaluations on dataset* **ijcnn**, *with different strategies to control batch size: geometric increase (top left), norm test (top right), inner-product test (bottom left), and comparison between the best run for each method (bottom right).*

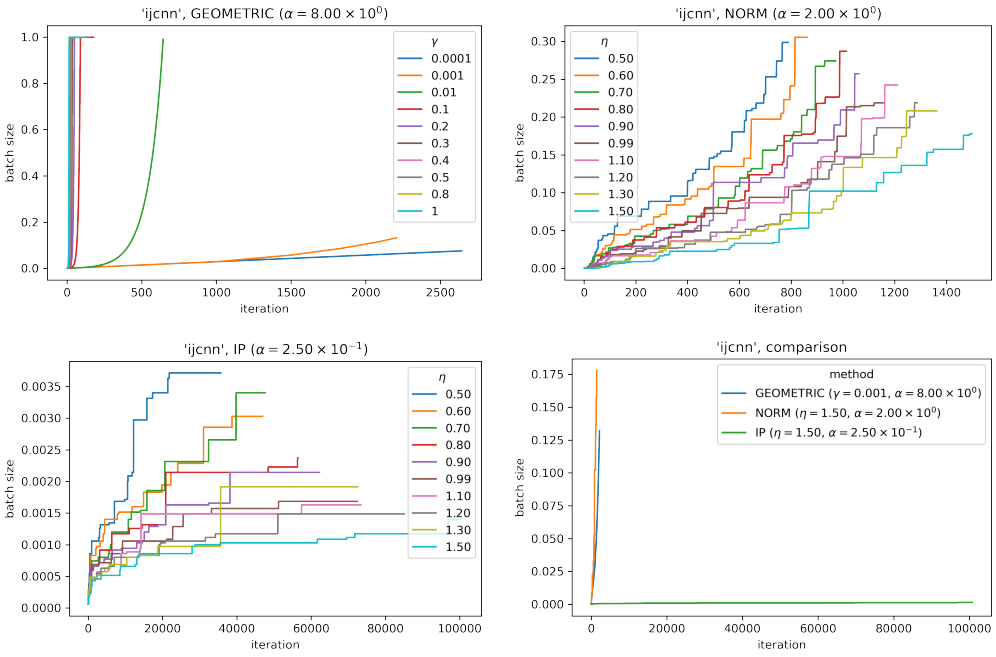

FIG. A.6. *Batch size (as a fraction of total number of data points N) against iterations on dataset* **ijcnn**, *with different strategies to control batch size: geometric increase (top left), norm test (top right), inner-product test (bottom left), and comparison between the best run for each method (bottom right).*

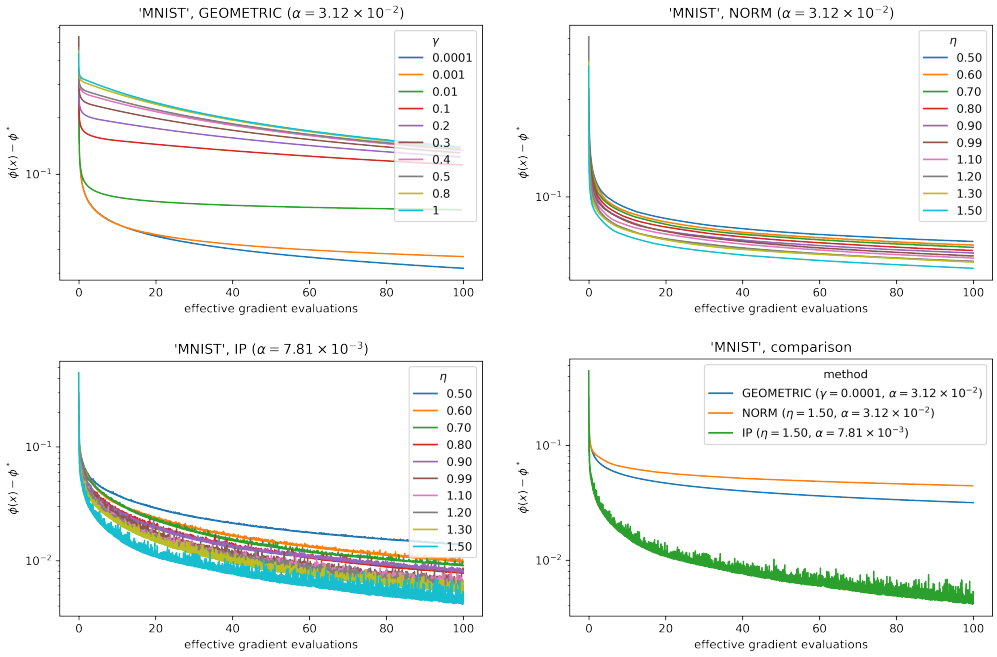

FIG. A.7. *Optimality gap $\phi(x_k) - \phi^*$ against effective gradient evaluations on dataset* `MNIST`*, with different strategies to control batch size: geometric increase (top left), norm test (top right), inner-product test (bottom left), and comparison between the best run for each method (bottom right).*

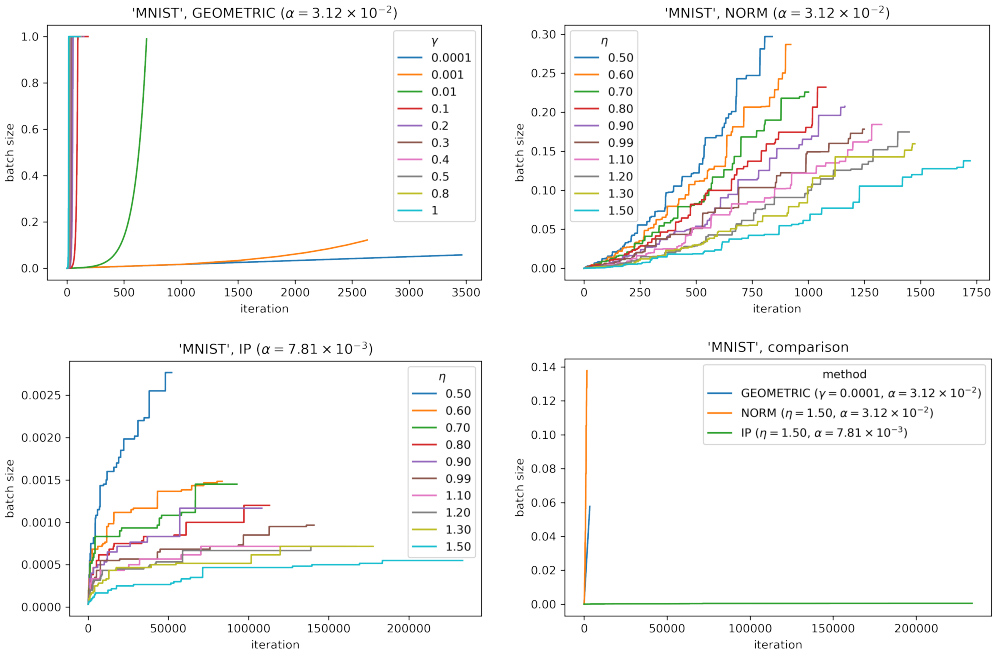

FIG. A.8. *Batch size (as a fraction of total number of data points N) against iterations on dataset* `MNIST`*, with different strategies to control batch size: geometric increase (top left), norm test (top right), inner-product test (bottom left), and comparison between the best run for each method (bottom right).*

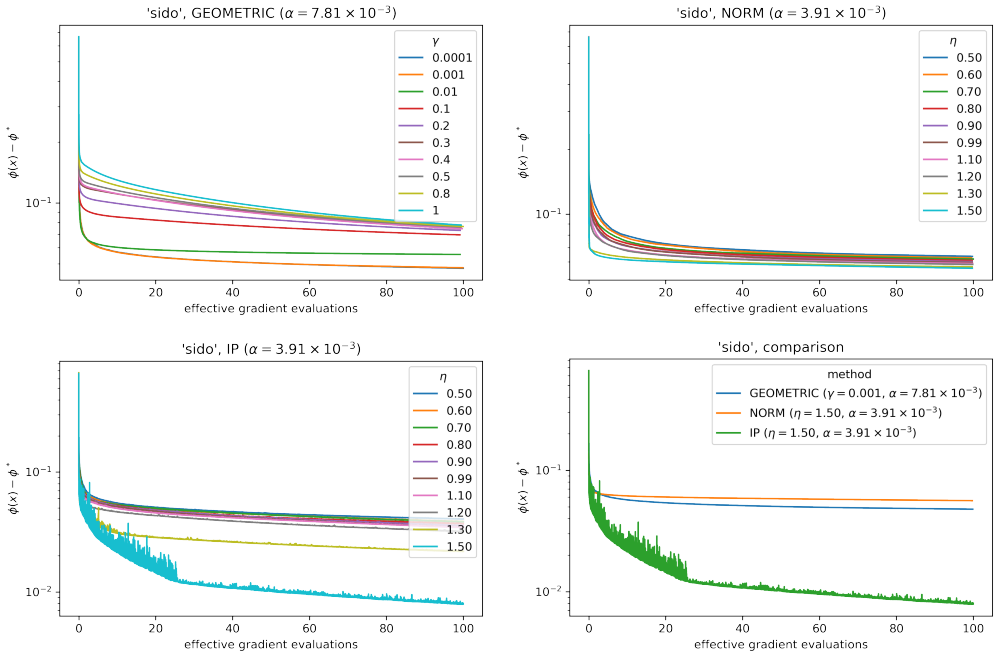

FIG. A.9. *Optimality gap $\phi(x_k) - \phi^*$ against effective gradient evaluations on dataset* **`sido`**, *with different strategies to control batch size: geometric increase (top left), norm test (top right), inner-product test (bottom left), and comparison between the best run for each method (bottom right).*

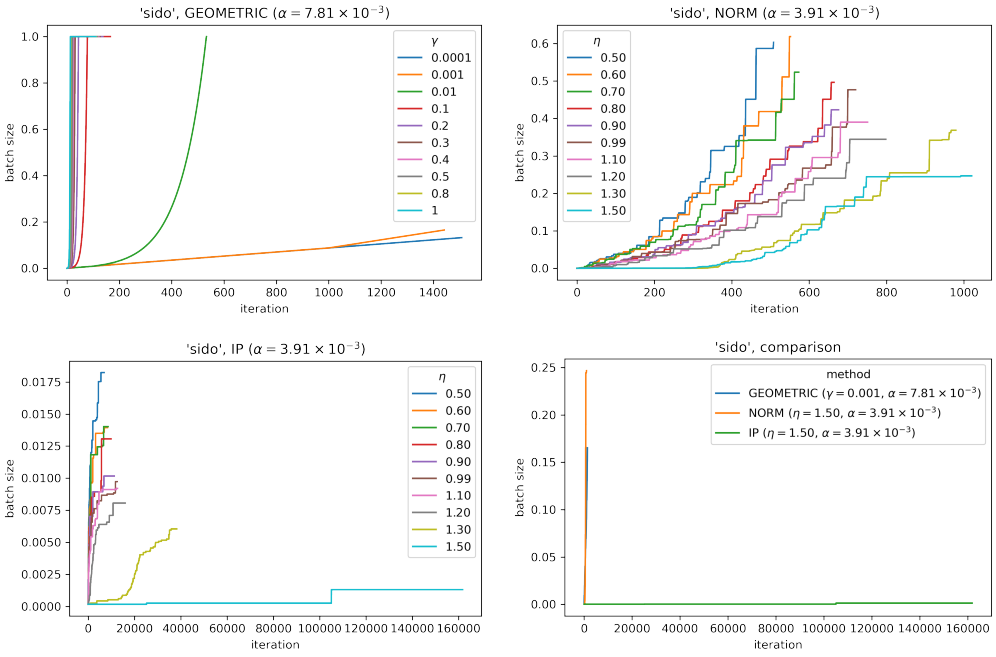

FIG. A.10. *Batch size (as a fraction of total number of data points N) against iterations on dataset* **`sido`**, *with different strategies to control batch size: geometric increase (top left), norm test (top right), inner-product test (bottom left), and comparison between the best run for each method (bottom right).*