Adaptive Sampling Strategies for Stochastic Optimization

Raghu Bollapragada[*] Richard Byrd [†] Jorge Nocedal [‡]

October 30, 2017

Abstract

In this paper, we propose a stochastic optimization method that adaptively controls the sample size used in the computation of gradient approximations. Unlike other variance reduction techniques that either require additional storage or the regular computation of full gradients, the proposed method reduces variance by increasing the sample size as needed. The decision to increase the sample size is governed by an *inner product test* that ensures that search directions are descent directions with high probability. We show that the inner product test improves upon the well known *norm test*, and can be used as a basis for an algorithm that is globally convergent on nonconvex functions and enjoys a global linear rate of convergence on strongly convex functions. Numerical experiments on logistic regression problems illustrate the performance of the algorithm.

[*] 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.

[†] 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 National Science Foundation grant DMS-0810213, and by Department of Energy grant DE-FG02-87ER25047.

1 Introduction

This paper presents a first-order stochastic optimization method that progressively changes the size of the sample used in the gradient approximation with the aim of achieving overall efficiency. The algorithm starts by choosing a small sample, and increases it as needed so that the gradient approximation is accurate enough to yield a linear rate of convergence for strongly convex functions. Adaptive sampling methods of this type are appealing because they enjoy optimal complexity properties [6, 11] and have the potential of being effective on a wide range of applications. Theoretical guidelines for controlling the sample size have been established in the literature [6, 11, 21], but the design of practical implementations has proven to be difficult. For example, the mechanism studied in [6, 8, 14], although intuitively appealing, is often inefficient in practice for reasons discussed below.

The problem of interest is

$$\min_{x\in\mathbb{R}^d} \ \mathbb{E}[\phi(x;\xi)],$$

where $\phi : \mathbb{R}^d \to R$ is a smooth function and ξ is a random variable. A particular instance of this problem arises in machine learning, where it takes the form

$$\min_{x\in\mathbb{R}^d} F(x) = \int f(x;z,y)dP(z,y). \tag{1.1}$$

In this setting, f is the composition of a prediction function (parametrized by a vector x) and a smooth loss function, and (z, y) are random input-output pairs with probability distribution $P(z, y)$. We call F the *expected risk.*

Often, problem (1.1) cannot be tackled directly because the joint probability distribution $P(z, y)$ is unknown. In this case, one draws a data set $\{(z^i, y^i)\}$, $i = 1, \ldots, N$, from the distribution $P(z, y)$, and minimizes the *empirical risk*

$$R(x) = \frac{1}{N}\sum_{i=1}^{N} f(x; z^i, y^i).$$

We define $F_i(x) \overset{\text{def}}{=} f(x; z^i, y^i)$, so that the empirical risk can be written conveniently as

$$R(x) = \frac{1}{N}\sum_{i=1}^{N} F_i(x). \tag{1.2}$$

One may view the optimization algorithm as being applied directly to the expected risk F or to the empirical risk R. We state our algorithm and establish a convergence result in terms of the minimization of F. Later on, in Section 4, we discuss a practical implementation designed to minimize R.

An approximation to the gradient of F can be obtained by sampling. At the iterate x_k, we define

$$\nabla F_{S_k}(x_k) = \frac{1}{|S_k|}\sum_{i\in S_k} \nabla F_i(x_k), \tag{1.3}$$

where the set $S_k \subset \{1, 2, \ldots\}$ indexes certain data points (z^i, y^i). A first-order method based on this gradient approximation is then given by

$$x_{k+1} = x_k - \alpha_k \nabla F_{S_k}(x_k), \qquad \alpha_k > 0. \tag{1.4}$$

In our approach, the sample S_k changes at every iteration, and its size $|S_k|$ is determined by a mechanism described in the next section. It is based on an *inner product test* that ensures that the search direction in (1.4) is a descent direction with high probability. In contrast to the test studied in [6-8, 14], which we call *the norm test*, and which controls both the direction and length of the gradient approximation and promotes search directions that are close to the true gradient, the inner product test places more emphasis on generating descent directions and allows more freedom in their length.

The numerical results presented in Section 5 suggest that the inner product test is efficient in practice, but in order to establish a Q-linear convergence rate for strongly convex functions, we must reinforce it with an additional mechanism that prevents search directions from becoming nearly orthogonal to the true gradient $\nabla F(x_k)$. More precisely, we introduce an *orthogonality test* that ensures that the variance of sampled gradients along the direction orthogonal to $\nabla F(x_k)$ is properly controlled. The orthogonality test is invoked infrequently in practice and should be regarded as a safeguard against rare difficult cases.

An important component of Algorithm (1.4) is the selection of the steplength α_k. One option is to use a fixed value α that is selected for each problem after careful experimentation. An alternative that we explore in more depth is a backtracking line search that imposes sufficient decrease in the sampled function

$$F_{S_k}(x) = \frac{1}{|S_k|} \sum_{i \in S_k} F_i(x), \tag{1.5}$$

and that is controlled by an adaptive estimate of the Lipschitz constant L of the gradient. Similar strategies have been considered for deterministic problems (see e.g. [1]), but the stochastic setting provides some challenges and opportunities that we explore in our line search procedure.

This paper is organized into five sections. A literature review and a summary of our notation are presented in the rest of this section. In Section 2, we describe the inner product test, and in Section 3 we introduce the orthogonality test and establish convergence analysis of an adaptive sampling algorithm that employs both tests. In Section 4, we discuss some practical implementation issues, and present a full description of the algorithm. Numerical results are presented in Section 5, and in Section 6 we make some concluding remarks.

1.1 Literature Review

Optimization methods that progressively increase sample sizes have been studied in [6, 9, 11, 15, 21–24]. Friedlander and Schmidt [11] consider the finite sum problem (1.2) and show linear convergence by increasing $|S_k|$ at a geometric rate. They also experiment with a quasi-Newton version of their algorithm. Byrd et al. [6] study the minimization of expected risk (1.1) and show linear convergence when the sample size grows geometrically, and provide computational complexity bounds. They propose the *norm test* as a practical procedure for controlling the sample size. Pasupathy et al. [21] study more generally the effect of sampling rates on the convergence and complexity of various optimization methods. Hashemi et al. [14] consider a test that is similar to the norm test, which is reinforced by a back up mechanism that ensures a geometric increase in the sample size. They motivate this approach from a stochastic simulation perspective and using variance-bias ratios. Cartis and Scheinberg [8] relax the norm test by allowing it to be violated with a probability less than 0.5, and this ensures that the search directions are successful descent directions more than 50% of the time. They use techniques from stochastic processes and analyze algorithms that perform a line search using the true function values $F(x_k)$. Bollapragada et al. [4] study methods that sample the gradient and Hessian, and establish conditions for global linear convergence. They also provide a superlinear convergence result in the case when the gradient samples are increased at rates faster than geometric and Hessian samples are increased without bound (at any rate).

The adaptive sampling methods studied here can be regarded as variance reducing methods; see the survey [5]. Other noise reducing methods include stochastic aggregated gradient methods, such as SAG [25], SAGA [10], and SVRG [16]. These methods either compute the full gradient at regular intervals, as in SVRG, or require storage of the component gradients, as in SAG or SAGA. These methods have gained much popularity in recent years, as they are able to achieve a linear rate of convergence for the finite sum problem, with a very low iteration cost.

1.2 Notation

We denote the variables of the optimization problem by $x \in \mathbb{R}^d$, and a minimizer of the objective F as x^*. Throughout the paper, $\|\cdot\|$ denotes the ℓ_2 vector norm. The notation $A \preceq B$ means that $B - A$ is a symmetric and positive semi-definite matrix.

2 The Inner Product Test

Let us consider how to select the sample size in the first-order stochastic optimization method

$$x_{k+1} = x_k - \alpha_k \nabla F_{S_k}(x_k). \tag{2.1}$$

Here $\alpha_k > 0$ is the steplength parameter and the sampled gradient $\nabla F_{S_k}(x_k)$ is defined in (1.3). We propose to determine the sample size $|S_k|$ at every iteration through the *inner product test* described below, which aims to ensure that the algorithm generates descent directions sufficiently often. We recall that the search direction of algorithm (2.1) is a descent direction for F if

$$\nabla F_{S_k}(x_k)^T \nabla F(x_k) > 0.$$

This condition will not hold at every iteration of our algorithm, but if S_k is chosen uniformly at random from $\{1, 2, \ldots\}$, it will hold in expectation, i.e.,

$$\mathbb{E}\left[\nabla F_{S_k}(x_k)^T \nabla F(x_k)\right] = \|\nabla F(x_k)\|^2 > 0. \tag{2.2}$$

We must in addition control the variance of the term on the left hand side to guarantee that the iteration (2.1) is convergent. We do so by requiring that the sample size $|S_k|$ be large enough so that the following condition is satisfied

$$\mathbb{E}\left[\left(\nabla F_{S_k}(x_k)^T \nabla F(x_k) - \|\nabla F(x_k)\|^2\right)^2\right] \leq \theta^2 \|\nabla F(x_k)\|^4, \quad \text{for some} \quad \theta > 0. \tag{2.3}$$

The left hand side is difficult to compute but can be bounded by the true variance of individual gradients, i.e.,

$$\mathbb{E}\left[\left(\nabla F_{S_k}(x_k)^T \nabla F(x_k) - \|\nabla F(x_k)\|^2\right)^2\right] \leq \frac{\mathbb{E}\left[\left(\nabla F_i(x_k)^T \nabla F(x_k) - \|\nabla F(x_k)\|^2\right)^2\right]}{|S_k|}. \tag{2.4}$$

Therefore, the following condition ensures (2.3)

$$\frac{\mathbb{E}\left[\left(\nabla F_i(x_k)^T \nabla F(x_k) - \|\nabla F(x_k)\|^2\right)^2\right]}{|S_k|} \leq \theta^2 \|\nabla F(x_k)\|^4. \tag{2.5}$$

We refer to (2.5) as the *(exact variance) inner product test.* In large-scale applications, the computation of $\nabla F(x_k)$ can be prohibitively expensive, but we can approximate the variance on the left side of (2.5) with the sample variance and the gradient $\nabla F(x_k)$ on the right side with a sampled gradient, to obtain

$$\frac{\mathrm{Var}_{i \in S_k}(\nabla F_i(x_k)^T \nabla F_{S_k}(x_k))}{|S_k|} \leq \theta^2 \|\nabla F_{S_k}(x_k)\|^4, \tag{2.6}$$

where

$$\mathrm{Var}_{i \in S_k}\left(\nabla F_i(x_k)^T \nabla F_{S_k}(x_k)\right) = \frac{1}{|S_k| - 1} \sum_{i \in S_k} \left(\nabla F_i(x_k)^T \nabla F_{S_k}(x_k) - \|\nabla F_{S_k}(x_k)\|^2\right)^2.$$

Condition (2.6) will be called the *(approximate) inner product test.* Whenever it is not satisfied, we increase the sample size $|S_k|$ to one that we predict will satisfy (2.6). An outline of this approach is given in Algorithm 1.

Algorithm 1 Basic Version

```
Input: Initial iterate $x_0$, initial sample $S_0$, and a constant $\theta > 0$.
Set $k \leftarrow 0$
Repeat until a convergence test is satisfied:
    Compute $d_k = -\nabla F_{S_k}(x_k)$
    Choose a steplength $\alpha_k > 0$
    Compute new iterate: $x_{k+1} = x_k + \alpha_k d_k$
    Set $k \leftarrow k + 1$
    Choose a new sample $S_k$ such that the condition (2.6) is satisfied
```

In Section 4, we discuss how to implement the inner product test in practice, how to choose the parameter θ and the stepsize α_k, as well as the strategy for increasing the size of a new sample S_k, when the algorithm calls for it.

It is illuminating to compare the inner product test[1] with a related rule studied in the literature [6, 8, 14] that we call the *norm test.* The comparison can be most simply and clearly seen in the deterministic setting with the gradient based method $x_{k+1} = x_k - \alpha_k g_k$, where g_k is some approximation to the gradient $\nabla F(x_k)$. In this context, the deterministic analog of (2.3) is

$$\left(g_k^T \nabla F(x_k) - \|\nabla F(x_k)\|^2\right)^2 \leq \theta^2 \|\nabla F(x_k)\|^4. \tag{2.7}$$

In contrast, the norm test corresponds to

$$\|g_k - \nabla F(x_k)\|^2 \leq \theta^2 \|\nabla F(x_k)\|^2, \qquad \text{for some} \quad \theta \in [0,1). \tag{2.8}$$

This rule was studied by Carter [7] in the context of trust region methods with inaccurate gradients. It is easy to see that (2.8) ensures that g_k is a descent direction, but it is not a necessary condition; in fact (2.8) is more restrictive than (2.7) because it requires approximate gradients to lie in a ball centered at the true gradient $\nabla F(x_k)$, whereas (2.7) allows gradients that are within an infinite band around the true gradient, as illustrated in Figure 2.1.

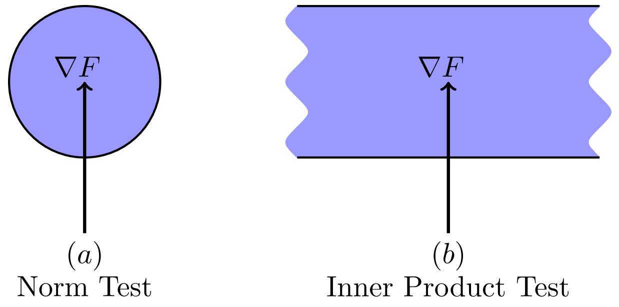

Figure 2.1: Deterministic setting. Given a gradient ∇F, the shaded areas denote the set of vectors g satisfying (a): the norm condition (2.8); (b) the deterministic inner product condition (2.7).

In the stochastic setting, the norm condition (2.8) becomes

$$\mathbb{E}[\|\nabla F_{S_k}(x_k) - \nabla F(x_k)\|^2] \leq \theta^2 \|\nabla F(x_k)\|^2. \tag{2.9}$$

Following the same reasoning as in (2.4), this condition will be satisfied if we impose instead

$$\frac{\mathbb{E}[\|\nabla F_i(x_k) - \nabla F(x_k)\|^2]}{|S_k|} \leq \theta^2 \|\nabla F(x_k)\|^2. \tag{2.10}$$

[1] We use the term *inner product test* to refer to (2.5) or (2.6) when the distinction is not important in the discussion.

This *norm test* is used in [6] to control the sample size: if (2.10) is not satisfied, then sample size is increased.

Numerical experience indicates that the norm test can be unduly restrictive, often leading to a very fast increase in the sample size, negating the benefits of adaptive sampling. An indication that the inner product test increases the sample size more slowly than the norm test can be see through the following argument. Let $|S_i|$, $|S_n|$ represent the minimum number of samples required to satisfy the inner product test (2.5) and the norm test (2.10), respectively, at any given iterate x, using the same value of θ. A simple computation (see Appendix A) shows that

$$\frac{|S_i|}{|S_n|} = \beta(x) \leq 1, \tag{2.11}$$

where

$$\beta(x) = \frac{\mathbb{E}[\|\nabla F_i(x)\|^2 \cos^2(\chi_i)] - \|\nabla F(x)\|^2}{\mathbb{E}[\|\nabla F_i(x)\|^2] - \|\nabla F(x)\|^2} \tag{2.12}$$

and χ_i is the angle between $\nabla F_i(x_k)$ and $\nabla F(x_k)$. The quantity $\beta(x)$ is the ratio of the variance the individual gradients along the true gradient direction and the total variance of the individual gradients. The numerical results presented in Section 5 are consistent with this observation and show that $\beta(x_k)$ is often much less than 1.

3 Analysis

In order to establish linear convergence for this method, it is necessary to introduce an additional condition that has only a slight effect on the algorithm in practice, but guarantees the quality of the search direction in difficult cases. In this section, we first describe this test, and in the second part we establish some results on convergence rates.

3.1 Orthogonality Test

Establishing a convergence rate usually involves showing that the step direction is bounded away from orthogonality to $\nabla F(x_k)$. However, iteration (2.1) with a sample S_k satisfying the inner product condition (2.5) does not necessarily enjoy this property. The possible near orthogonality corresponds to the case where the ratio $\beta(x)$ defined above is near zero and occurs when the variance in the individual gradients is very large compared to the variance in the individual gradients along the true gradient direction. Although we have not observed very small values of $\beta(x_k)$ in our numerical tests, this is harmful in principle and to prove convergence we must be able to avoid this possibility. We propose a test that imposes a loose bound on the component of $\nabla F_{S_k}(x_k)$ orthogonal to the true gradient. This test, together with (2.5), allows us to prove a linear convergence result for the adaptive sampling algorithm, when F is strongly convex.

To motivate the orthogonality test, we note that the component of $\nabla F_{S_k}(x_k)$ orthogonal to $\nabla F(x_k)$ is 0 in expectation, i.e.,

$$\mathbb{E}\left[\nabla F_{S_k}(x_k) - \frac{\nabla F_{S_k}(x_k)^T \nabla F(x_k)}{\|\nabla F(x_k)\|^2} \nabla F(x_k)\right] = 0,$$

but that is not sufficient. We must bound the variance of this orthogonal component, and to achieve this we require that the sample size $|S_k|$ be large enough to satisfy,

$$\mathbb{E}\left[\left\|\nabla F_{S_k}(x_k) - \frac{\nabla F_{S_k}(x_k)^T \nabla F(x_k)}{\|\nabla F(x_k)\|^2}\nabla F(x_k)\right\|^2\right] \leq \nu^2 \, \|\nabla F(x_k)\|^2, \tag{3.1}$$

for some positive constant ν whose choice is discussed in Section 4. For a given sample size, this condition can be expressed, using the true variance of individual gradients, as

$$\frac{\mathbb{E}\left[\left\|\nabla F_i(x_k) - \frac{\nabla F_i(x_k)^T \nabla F(x_k)}{\|\nabla F(x_k)\|^2}\nabla F(x_k)\right\|^2\right]}{|S_k|} \leq \nu^2 \|\nabla F(x_k)\|^2. \tag{3.2}$$

The *(exact variance) orthogonality test* states that if this inequality is not satisfied, the sample size $|S|$ should be increased.

Reasoning as in (2.6), we can derive a variant of the orthogonality test based on sample approximations. This *(approximate) orthogonality test* is given by

$$\frac{1}{|S_k|-1}\frac{\sum_{i\in S_k}\left\|\nabla F_i(x_k) - \frac{\nabla F_i(x_k)^T \nabla F_{S_k}(x_k)}{\|\nabla F_{S_k}(x_k)\|^2}\nabla F_{S_k}(x_k)\right\|^2}{|S_k|} \leq \nu^2 \|\nabla F_{S_k}(x_k)\|^2. \tag{3.3}$$

It is interesting to note that, since (2.3) implies that the root mean square of the component of the step along the gradient is bounded below by $\sqrt{1-\theta^2}\|\nabla F(x_k)\|$, imposition of (3.2) will tend to keep the tangent of the angle χ_k between $\nabla F_{S_k}(x_k)$ and $\nabla F(x_k)$ below $\nu/\sqrt{1-\theta^2}$, providing a limit on the near orthogonality of these two vectors.

3.2 Convergence Analysis

The orthogonality test, in conjunction with the inner product test allows the algorithm to make sufficient progress at every iteration, in expectation. More precisely, we now establish three convergence results for the *exact* versions of these two tests, namely (2.5) and (3.2). Our results apply to iteration (2.1) with a fixed steplength. We start by establishing a technical lemma.

Lemma 3.1. *Suppose that F is twice continuously differentiable and that there exists a constant $L > 0$ such that*

$$\nabla^2 F(x) \preceq LI, \quad \forall x \in \mathbb{R}^d. \tag{3.4}$$

Let $\{x_k\}$ be the iterates generated by iteration (2.1) with any x_0, where $|S_k|$ is chosen such that the (exact variance) inner product test (2.5) and the (exact variance) orthogonality test (3.2) are satisfied at each iteration for any given constants $\theta > 0$ and $\nu > 0$. Then, for any k,

$$\mathbb{E}\left[\|\nabla F_{S_k}(x_k)\|^2\right] \leq (1+\theta^2+\nu^2)\|\nabla F(x_k)\|^2. \tag{3.5}$$

Moreover, if the steplength satisfies

$$\alpha_k = \alpha \leq \frac{1}{(1+\theta^2+\nu^2)L}, \tag{3.6}$$

we have that

$$\mathbb{E}[F(x_{k+1})] \leq \mathbb{E}[F(x_k)] - \frac{\alpha}{2}\|\nabla F(x_k)\|^2. \tag{3.7}$$

Proof. Since (3.2) is satisfied, we have that (3.1) holds. Thus, recalling (2.2) we have that (3.1) can be written as

$$\begin{aligned}
&\mathbb{E}\left[\left\|\nabla F_{S_k}(x_k) - \frac{\nabla F_{S_k}(x_k)^T\nabla F(x_k)}{\|\nabla F(x_k)\|^2}\nabla F(x_k)\right\|^2\right] \\
&\quad = \mathbb{E}\left[\|\nabla F_{S_k}(x_k)\|^2\right] - \frac{2\mathbb{E}\left[\left(\nabla F_{S_k}(x_k)^T\nabla F(x_k)\right)^2\right]}{\|\nabla F(x_k)\|^2} + \frac{\mathbb{E}\left[\left(\nabla F_{S_k}(x_k)^T\nabla F(x_k)\right)^2\right]}{\|\nabla F(x_k)\|^2} \\
&\quad = \mathbb{E}\left[\|\nabla F_{S_k}(x_k)\|^2\right] - \frac{\mathbb{E}\left[\left(\nabla F_{S_k}(x_k)^T\nabla F(x_k)\right)^2\right]}{\|\nabla F(x_k)\|^2} \\
&\quad \leq \nu^2\,\|\nabla F(x_k)\|^2.
\end{aligned}$$

Therefore,

$$\mathbb{E}\left[\|\nabla F_{S_k}(x_k)\|^2\right] \leq \frac{\mathbb{E}\left[\left(\nabla F_{S_k}(x_k)^T\nabla F(x_k)\right)^2\right]}{\|\nabla F(x_k)\|^2} + \nu^2\|\nabla F(x_k)\|^2. \tag{3.8}$$

To bound the first term on the right side of this inequality, we use the inner product test. Since $|S_k|$ satisfies (2.5), the inequality (2.3) holds, and this in turn yields

$$\begin{aligned}
\mathbb{E}\left[\left(\nabla F_{S_k}(x_k)^T\nabla F(x_k)\right)^2\right] &\leq \|\nabla F(x_k)\|^4 + \theta^2\|\nabla F(x_k)\|^4 \\
&= (1+\theta^2)\|\nabla F(x_k)\|^4.
\end{aligned}$$

Substituting in (3.8), we get the following bound on the length of the search direction:

$$\mathbb{E}\left[\|\nabla F_{S_k}(x_k)\|^2\right] \leq (1+\theta^2+\nu^2)\|\nabla F(x_k)\|^2,$$

which proves (3.5). Using this inequality, (2.1), (3.4), and (3.6) we have,

$$\begin{aligned}
\mathbb{E}[F(x_{k+1})] &\leq \mathbb{E}[F(x_k)] - \mathbb{E}\left[\alpha\nabla F_{S_k}(x_k)^T\nabla F(x_k)\right] + \mathbb{E}\left[\frac{L\alpha^2}{2}\|\nabla F_{S_k}(x_k)\|^2\right] \\
&= \mathbb{E}[F(x_k)] - \alpha\|\nabla F(x_k)\|^2 + \frac{L\alpha^2}{2}\mathbb{E}[\|\nabla F_{S_k}(x_k)\|^2] \\
&\leq \mathbb{E}[F(x_k)] - \alpha\|\nabla F(x_k)\|^2 + \frac{\alpha^2}{2}L(1+\theta^2+\nu^2)\|\nabla F(x_k)\|^2 \\
&\leq \mathbb{E}[F(x_k)] - \frac{\alpha}{2}\|\nabla F(x_k)\|^2.
\end{aligned}$$

□

We now show that iteration (2.1), using a fixed steplength α, is linearly convergent when F is strongly convex. In the discussion that follows, x^* denotes the minimizer of F.

Theorem 3.2. *(Strongly Convex Objective.) Suppose that F is twice continuously differentiable and that there exist constants $0 < \mu \leq L$ such that*

$$\mu I \preceq \nabla^2 F(x) \preceq LI, \quad \forall x \in \mathbb{R}^d. \tag{3.9}$$

Let $\{x_k\}$ be the iterates generated by iteration (2.1) *with any x_0, where $|S_k|$ is chosen such that the (exact variance) inner product test* (2.5) *and the (exact variance) orthogonality test* (3.2) *are satisfied at each iteration for any given constants $\theta > 0$ and $\nu > 0$. Then, if the steplength satisfies* (3.6) *we have that*

$$\mathbb{E}[F(x_k) - F(x^*)] \leq \rho^k (F(x_0) - F(x^*)), \tag{3.10}$$

where

$$\rho = 1 - \mu\alpha. \tag{3.11}$$

In particular, if α takes its maximum value in (3.6), *i.e., $\alpha = 1/(1+\theta^2+\nu^2)L$, we have*

$$\rho = 1 - \frac{\mu}{L(1+\theta^2+\nu^2)}. \tag{3.12}$$

Proof. It is well known [2] that for strongly convex functions

$$\|\nabla F(x_k)\|^2 \geq 2\mu[F(x_k) - F(x^*)].$$

Substituting this in (3.7) and subtracting $F(x^*)$ from both sides we obtain,

$$\mathbb{E}[F(x_{k+1}) - F(x^*)] \leq \mathbb{E}[F(x_k) - F(x^*)](1 - \mu\alpha),$$

from which the theorem follows. □

Note that when $\theta = \nu = 0$ we recover the classical result for the exact gradient method. We now consider the case when F is convex, but not strongly convex.

Theorem 3.3. *(General Convex Objective.) Suppose that F is twice continuously differentiable and convex, and that there exists a constant $L > 0$ such that*

$$\nabla^2 F(x) \preceq LI, \quad \forall x \in \mathbb{R}^d. \tag{3.13}$$

Let $\{x_k\}$ be the iterates generated by iteration (2.1) with any x_0, where $|S_k|$ is chosen such that the exact variance inner product test (2.5) and orthogonality test (3.2) are satisfied at each iteration for any given constants $\theta > 0$ and $\nu > 0$. Then, if the steplength satisfies the strict version of (3.6), that is

$$\alpha_k = \alpha < \frac{1}{(1+\theta^2+\nu^2)L}, \tag{3.14}$$

we have for any positive integer T,

$$\min_{0\leq k\leq T-1} \mathbb{E}\left[F\left(x_k\right)\right] - F^* \leq \frac{1}{2\alpha c T}\|x_0 - x^*\|^2,$$

where F^* is the optimal function value, the constant $c > 0$ is given by $c = 1 - L\alpha(1+\theta^2+\nu^2)$, and $x^* \in \{x : x = \arg\min_x F(x)\}$.

Proof. From Lemma 3.1 we have that

$$\mathbb{E}\left[\|\nabla F_{S_k}(x_k)\|^2\right] \leq (1+\theta^2+\nu^2)\|\nabla F(x_k)\|^2.$$

Using this inequality and (3.13) and considering any $x^* \in X^*$ we have,

$$\begin{aligned}\mathbb{E}[\|x_{k+1} - x^*\|^2] &= \|x_k - x^*\|^2 - 2\alpha\mathbb{E}[\nabla F_{S_k}(x_k)^T(x_k - x^*)] + \alpha^2\mathbb{E}[\|\nabla F_{S_k}(x_k)\|^2] \\ &\leq \|x_k - x^*\|^2 - 2\alpha\nabla F(x_k)^T(x_k - x^*) + \alpha^2(1+\theta^2+\nu^2)\|\nabla F(x_k)\|^2 \\ &\leq \|x_k - x^*\|^2 - 2\alpha(F(x_k) - F^*) + \alpha^2(1+\theta^2+\nu^2)\|\nabla F(x_k)\|^2, \qquad (3.15)\end{aligned}$$

where the last inequality follows from convexity of F.

Now, for any function having Lipschitz continuous gradients

$$\|\nabla F(x)\|^2 \leq 2L(F(x) - F^*). \tag{3.16}$$

This result is shown in [20], but for the sake of completeness we give here a proof. Since F has a Lipschitz continuous gradient,

$$\begin{aligned}F\left(x - \frac{1}{L}\nabla F(x)\right) &\leq F(x) + \nabla F(x)^T\left(x - \frac{1}{L}\nabla F(x) - x\right) + \frac{L}{2}\left\|x - \frac{1}{L}\nabla F(x) - x\right\|^2 \\ &= F(x) - \frac{1}{L}\|\nabla F(x)\|^2 + \frac{1}{2L}\|\nabla F(x)\|^2 \\ &= F(x) - \frac{1}{2L}\|\nabla F(x)\|^2.\end{aligned}$$

Recalling that F^* is the optimal function value we have,

$$F^* \le F\left(x - \frac{1}{L}\nabla F(x)\right) \le F(x) - \frac{1}{2L}\|\nabla F(x)\|^2,$$

which proves (3.16).

Substituting (3.16) in (3.15) we obtain

$$\begin{aligned}\mathbb{E}[\|x_{k+1} - x^*\|^2] &\le \|x_k - x^*\|^2 - 2\alpha(F(x_k) - F^*) + 2L\alpha^2(1+\theta^2+\nu^2)(F(x_k) - F^*)\\ &= \|x_k - x^*\|^2 - 2\alpha(F(x_k) - F^*)\left(1 - L\alpha(1+\theta^2+\nu^2)\right)\\ &= \|x_k - x^*\|^2 - 2\alpha c(F(x_k) - F^*),\end{aligned}$$

by the definition of c. We can write this inequality as

$$\mathbb{E}[F(x_k)] - F^* \le \frac{1}{2\alpha c}\left(\mathbb{E}[\|x_k - x^*\|^2] - \mathbb{E}[\|x_{k+1} - x^*\|^2]\right),$$

and summing, we obtain

$$\begin{aligned}\min_{0\le k\le T-1} \mathbb{E}\left[F\left(x_k\right)\right] - F^* &\le \sum_{k=0}^{T-1}\frac{1}{T}(\mathbb{E}[F(x_k)] - F^*)\\ &\le \frac{1}{2\alpha cT}(\mathbb{E}[\|x_0 - x^*\|^2] - \mathbb{E}[\|x_T - x^*\|^2])\\ &\le \frac{1}{2\alpha cT}\|x_0 - x^*\|^2.\end{aligned}$$

□

This result establishes a sublinear rate of convergence in function values by referencing the best function value obtained after every T iterates. We now consider the case when F is nonconvex and bounded below.

Theorem 3.4. *(Nonconvex Objective.) Suppose that F is twice continuously differentiable and bounded below, and that there exist a constant $L > 0$ such that*

$$\nabla^2 F(x) \preceq LI, \quad \forall x \in \mathbb{R}^d. \tag{3.17}$$

Let $\{x_k\}$ be the iterates generated by iteration (2.1) *with any x_0, where $|S_k|$ is chosen such that the (exact variance) inner product test* (2.5) *and the (exact variance) orthogonality test* (3.2) *are satisfied at each iteration for any given constants $\theta > 0$ and $\nu > 0$. Then, if the steplength satisfies*

$$\alpha_k = \alpha \le \frac{1}{(1+\theta^2+\nu^2)L}, \tag{3.18}$$

then

$$\lim_{k\to\infty} \mathbb{E}[\|\nabla F(x_k)\|^2] \to 0. \tag{3.19}$$

Moreover, for any positive integer T we have that

$$\min_{0\leq k\leq T-1} \mathbb{E}[\|\nabla F(x_k)\|^2] \leq \frac{2}{\alpha T}(F(x_0) - F_{min}),$$

where F_{min} is a lower bound on F in $\mathbb{R}^d$.

Proof. From Lemma 3.1 we have

$$\mathbb{E}[F(x_{k+1})] \leq \mathbb{E}[F(x_k)] - \frac{\alpha}{2}\mathbb{E}[\|\nabla F(x_k)\|^2],$$

and hence

$$\mathbb{E}[\|\nabla F(x_k)\|^2] \leq \frac{2}{\alpha}\mathbb{E}[F(x_k) - F(x_{k+1})].$$

Summing both sides of this inequality from $k = 0$ to $T - 1$, and since F is bounded below by F_{min}, we get

$$\sum_{k=0}^{T-1} \mathbb{E}[\|\nabla F(x_k)\|^2] \leq \frac{2}{\alpha}\mathbb{E}[F(x_0) - F(x_{\mathrm{T}})] \leq \frac{2}{\alpha}[F(x_0) - F_{min}].$$

Taking limits, we obtain

$$\lim_{T\to\infty} \sum_{k=0}^{T-1} \mathbb{E}[\|\nabla F(x_k)\|^2] < \infty,$$

which implies (3.19). We can also conclude that

$$\min_{0\leq k\leq T-1} \mathbb{E}[\|\nabla F(x_k)\|^2] \leq \frac{1}{T}\sum_{k=0}^{T} \mathbb{E}[\|\nabla F(x_k)\|^2] \leq \frac{2}{\alpha T}(F(x_0) - F_{min}).$$

□

This theorem shows that the sequence of gradients $\{\|\nabla F(x_k)\|\}$ converges to zero, in expectation. It also establishes a global sublinear rate of convergence of the smallest gradients generated after every T steps. Results with a similar flavor have been established by Ghadimi and Lan [12] in the context of nonconvex stochastic programming.

4 Practical Implementation

In this section, we describe our line search procedure, and present a technique for making the algorithm robust in the early stages of a run when sample variances and sample gradients are unreliable. We also discuss a heuristic that determines how much to increase the sample size $|S_k|$. We then present the complete algorithm, followed by a discussion of the choice of some important algorithmic parameters.

4.1 Line Search

We select the steplength parameter α_k by a backtracking line search based on the sampled function F_{S_k} and an adaptive estimate L_k of the Lipschitz constant of the gradient. The initial value of the steplength at the k-th iteration of the algorithm is given by $\alpha_k = 1/L_k$. If sufficient decrease in F_{S_k} is not obtained, L_k is increased by a constant factor until such a decrease is achieved.

Now, since overestimating the Lipschitz constant leads to unnecessarily small steps, at every outer iteration of the algorithm the initial value L_k is set to a fraction of the previous estimate L_{k-1}. This reset strategy has been used in deterministic convex optimization, but requires some attention in the stochastic setting. Specifically, decreasing the Lipschitz constant by a fixed fraction at every iteration may result in an inadequate steplength. We propose a variance-based rule described below to compute a contraction factor ζ_k at every iteration. Our line search strategy is summarized in Algorithm 2.

Algorithm 2 Backtracking Line Search

```
Input: $L_{k-1} > 0$, $\eta > 1$
1: Compute $\zeta_k$ as given in (4.3)
2: Set $L_k = L_{k-1}/\zeta_k$                                        ▷ Decrease the Lipschitz constant
3: Compute $F_{new} = F_{S_k}\left(x_k - \frac{1}{L_k}\nabla F_{S_k}(x_k)\right)$
4: while $F_{new} > F_{S_k}(x_k) - \frac{1}{2L_k}\|\nabla F_{S_k}(x_k)\|^2$ do      ▷ Sufficient decrease
5:     Set $L_k = \eta L_{k-1}$                                      ▷ Increase the Lipschitz constant
6:     Compute $F_{new} = F_{s_k}\left(x_k - \frac{1}{L_k}\nabla F_{S_k}(x_k)\right)$
7: end while
```

The expansion factor η in Step 5 is set to $\eta = 1.5$ in our experiments. To determine the contraction factor ζ_k, we reason as follows. From (2.1) we have

$$\mathbb{E}[F(x_{k+1})] - F(x_k) \leq -\alpha_k \|\nabla F(x_k)\|^2 + \frac{\alpha_k^2 L}{2}\mathbb{E}[\|\nabla F_{S_k}(x_k)\|_2^2].$$

Thus we can guarantee an expected decrease in the true objective function if the right hand side is negative, i.e.,

$$\frac{L\alpha_k^2}{2}\left(\operatorname{Var}\left(\nabla F_{S_k}(x_k)\right) + \|\nabla F(x_k)\|^2\right) \leq \alpha_k \|\nabla F(x_k)\|^2, \tag{4.1}$$

where

$$\operatorname{Var}\left(\nabla F_{S_k}(x_k)\right) = \mathbb{E}[\|\nabla F_{S_k}(x_k) - \nabla F(x_k)\|^2].$$

This is the same variance as in the norm test (2.9). As was done in that context we first note that (4.1) holds if

$$\frac{L\alpha_k^2}{2}\left(\operatorname{Var}\left(\nabla F_i(x_k)\right)/|S_k| + \|\nabla F(x_k)\|^2\right) \leq \alpha_k \|\nabla F(x_k)\|^2.$$

Next, we approximate the true gradient and true variance using a sampled gradient and a sampled variance to obtain

$$\frac{L\alpha_k^2}{2}\left(\mathrm{Var}_{i\in S_k}\left(\nabla F_i(x_k)\right)/|S_k| + \|\nabla F_{S_k}(x_k)\|^2\right) \leq \alpha_k\|\nabla F_{S_k}(x_k)\|^2, \tag{4.2}$$

where

$$\mathrm{Var}_{i\in S_k}\left(\nabla F_i(x_k)\right) = \frac{1}{|S_k|-1}\sum_{i\in S_k}\|\nabla F_i(x_k) - \nabla F(x_k)\|^2.$$

We wish to compute an appropriate value L_k and set $\alpha_k = 1/L_k$. Therefore, since we can assume that L is large enough so that $L_{k-1} < L$, inequality (4.2) gives

$$\frac{1}{L_k}\|\nabla F_{S_k}(x_k)\|^2 \geq \frac{L_{k-1}}{2L_k^2}(\mathrm{Var}_{i\in S_k}\left(\nabla F_i(x_k)\right)/|S_k| + \|\nabla F_{S_k}(x_k)\|^2)$$

or

$$L_k \geq \frac{L_{k-1}}{2}\left(\frac{\mathrm{Var}_{i\in S_k}\left(\nabla F_i(x_k)\right)}{|S_k|\|\nabla F_{S_k}(x_k)\|^2} + 1\right) \equiv \frac{L_{k-1}}{2}a_k.$$

This indicates that when decreasing the Lipschitz estimate by the rule $L_k = L_{k-1}/\zeta_k$ at the start of each iteration, ζ_k may be chosen as

$$\zeta_k = \max\left(1, \frac{2}{a_k}\right) \tag{4.3}$$

Therefore, $\zeta_k \in [1,2]$, meaning L_k could be left unchanged ($\zeta_k = 1$), or reduced by a factor of at most 2.

Algorithm 2 is similar in form to the one proposed in [1] for deterministic functions. However, our line search operates with sampled (e.g. inaccurate) function values, and the adaptive setting has allowed us to employ a variance-based rule described above for estimating the initial value of L_k at every iteration. A line search based on sampled function values that is more close related to ours is described in [25], but it differs from ours in that they use a fixed contraction factor ζ throughout the algorithm.

4.2 Sample Control in the Noisy Regime

In the previous sections, we presented two forms of the inner product and orthogonality tests: one based on the population statistics and one based on samples. Since in many applications only sample statistics are available, our practical implementation of the algorithm imposes the inner product and orthogonality tests by verifying (2.6) instead of (2.5), and (3.3) instead of (3.2), respectively. The *augmented inner product test* consists of the inner product test (2.6) together with the orthogonality test (3.3).

Our numerical experience indicates that these sample approximations are sufficiently accurate, except if we choose to start the algorithm with a very small sample size, say 3, 5, 10. In this highly noisy regime, our conditions may not control the sample correctly because for small samples, $\|\nabla F_{S_k}(x_k)\|$ is often much larger than the true gradient $\|\nabla F(x_k)\|$, and therefore the tests (2.6) and (3.3) are too easily satisfied, preventing increases in the sample size. (These difficulties can also arise when using the norm test.)

To obtain a more accurate estimate of $\nabla F_{S_k}(x_k)$ to use in (2.6) and (3.3), we employ the following strategy. Whenever the sample sizes remain constant for a certain number of iterations, say r, we compute the running average of the most recent sample gradients:

$$g_{\text{avg}} \overset{\text{def}}{=} \frac{1}{r}\Big[\sum_{j=k-r+1}^{k} \nabla F_{S_j}(x_j)\Big]. \tag{4.4}$$

Ideally, r should be chosen such that the iterates in this summation are close enough to provide a good approximation of the gradient at x_k and so that there are enough samples for g_{avg} to be meaningful. (A reasonable default value could be $r = 10$.) If the length of g_{avg} is small compared with the length of $\nabla F_{S_k}(x_k)$, we view this as an indication that the latter is not accurate and the sample size should be increased. That is, if

$$\|g_{\text{avg}}\| < \gamma \|\nabla F_{S_k}(x_k)\|, \quad \gamma \in (0,1), \tag{4.5}$$

then the sample size is increased using a rule given in Section 4.3. The choice of the parameter γ is important, we choose it so that under some assumptions inequality (4.5) implies

$$\|\nabla F_{S_k}(x_k)\| \geq \omega \|\nabla F(x_k)\|, \tag{4.6}$$

for some constant ω (say 10). To relate γ and ω, note that since we used r times more samples in computing g_{avg} compared to $\nabla F_{S_k}(x_k)$, $\|g_{\text{avg}}\|$ should be a better estimate of $\|\nabla F(x_k)\|$, by a factor of $\sqrt{r}$. Although this may be optimistic since the r points x_j used in the running average do not coincide, we nevertheless assume that

$$\|g_{\text{avg}}\| - \|\nabla F(x_k)\| \approx \frac{1}{\sqrt{r}}\left(\|\nabla F_{S_k}(x_k)\| - \|\nabla F(x_k)\|\right)$$

so that

$$\frac{\|g_{\text{avg}}\|}{\|\nabla F_{S_k}(x_k)\|} \approx \frac{1}{\sqrt{r}} + \left(1 - \frac{1}{\sqrt{r}}\right)\frac{\|\nabla F(x_k)\|}{\|\nabla F_{S_k}(x_k)\|}.$$

Recalling (4.5) and (4.6), we then set

$$\gamma = \frac{1}{\sqrt{r}} + \left(1 - \frac{1}{\sqrt{r}}\right)\frac{1}{\omega}.$$

For example, if $\omega = 10$ and $r = 10$ then satisfaction of (4.5) with $\gamma = 0.38$ should correspond to satisfaction of (4.6). We choose this value in practice.

4.3 Increasing the Sample Size

When increasing the sample size, it is natural to require that the new sample satisfy the inner product and the orthogonal tests at the current iterate. We employ the following heuristic approach that aims to achieve this goal.

Suppose we wish to choose a larger sample size $|\hat{S}_k|$. We assume that increases in sample sizes are gradual enough so that, for any given iterate x_k,

$$\mathrm{Var}_{i\in\hat{S}_k}(\nabla F_i(x_k)^T\nabla F_{\hat{S}_k}(x_k)) \approxeq \mathrm{Var}_{i\in S_k}(\nabla F_i(x_k)^T\nabla F_{S_k}(x_k)).$$

The same logic can be applied to the orthogonality test, i.e.,

$$\begin{aligned}&\mathrm{Var}_{i\in\hat{S}_k}\left(\nabla F_i(x_k) - \frac{\nabla F_i(x_k)^T\nabla F_{\hat{S}_k}(x_k)}{\|\nabla F_{\hat{S}_k}(x_k)\|^2}\nabla F_{\hat{S}_k}(x_k)\right)\\ &\approxeq \mathrm{Var}_{i\in S_k}\left(\nabla F_i(x_k) - \frac{\nabla F_i(x_k)^T\nabla F_{S_k}(x_k)}{\|\nabla F_{S_k}(x_k)\|^2}\nabla F_{S_k}(x_k)\right),\end{aligned}$$

and let us also assume that

$$\|\nabla F_{\hat{S}_k}(x_k)\| \approxeq \|\nabla F_{S_k}(x_k)\|.$$

Under these simplifying assumptions, we see that the inner product test (2.6) and the orthogonality test (3.3) are satisfied if we choose $|\hat{S}_k|$ to be

$$|\hat{S}_k| = \max\left(\frac{\mathrm{Var}_{i\in S_k}(\nabla F_i(x_k)^T\nabla F_{S_k}(x_k))}{\theta^2\|\nabla F_{S_k}(x_k)\|^4}, \frac{\mathrm{Var}_{i\in S_k}\left(\nabla F_i(x_k) - \frac{\nabla F_i(x_k)^T\nabla F_{S_k}(x_k)}{\|\nabla F_{S_k}(x_k)\|^2}\nabla F_{S_k}(x_k)\right)}{\nu^2\|\nabla F_{S_k}(x_k)\|^2}\right). \tag{4.7}$$

In the case when the sample size is very small and we use the running average (4.4), we increase the sample size according to

$$|\hat{S}_k| = \max\left(\frac{\mathrm{Var}_{i\in S_k}(\nabla F_i(x_k)^T g_{\mathrm{avg}})}{\theta^2\|g_{\mathrm{avg}}\|^4}, \frac{\mathrm{Var}_{i\in S_k}\left(\nabla F_i(x_k) - \frac{\nabla F_i(x_k)^T g_{\mathrm{avg}}}{\|g_{\mathrm{avg}}\|^2}g_{\mathrm{avg}}\right)}{\nu^2\|g_{\mathrm{avg}}\|^2}\right). \tag{4.8}$$

Although these heuristics can be unreliable when the assumptions made in their derivation are not satisfied, they have worked well in our experiments.

4.4 The Complete Algorithm

The final version of the adaptive sampling algorithm that incorporates all the practical implementation techniques described in this section is given in Algorithm 3.

Algorithm 3 Adaptive Sampling Method

```
Input: Initial iterate $x_0$, initial sample $S_0$.
Sample Test Parameters: $\theta > 0, \nu > 0, r > 0, \gamma \in (0,1)$
Steplength Parameters: $L_0 > 0$, $\eta > 1$.
Set $k \leftarrow 0$
Repeat until a convergence test is satisfied:
Compute $d_k = -\nabla F_{S_k}(x_k)$
Compute $L_k$ using Algorithm (2)
Set $\alpha_k = 1/L_k$
Compute new iterate: $x_{k+1} = x_k + \alpha_k d_k$
Set $k \leftarrow k + 1$
Set $|S_k| = |S_{k-1}|$ and choose a new sample $S_k$
if condition (2.6) or condition (3.3) is not satisfied then
    Compute $|S_k|$ using (4.7) and choose a new sample $S_k$
end if
if $(|S_k| = |S_{k-1}| = \cdots = |S_{k-r}|)$ then
    Compute the running average gradient $g_{\text{avg}}$ given in (4.4)
    if $\|g_{\text{avg}}\| < \gamma \|\nabla F_{S_k}(x_k)\|$ then
        if condition (2.6) or condition (3.3) is not satisfied using $g_{\text{avg}}$ instead of $\nabla F_{S_k}(x_k)$ then
            Compute $|S_k|$ using (4.8) and choose a new sample $S_k$
        end if
    end if
end if
```

Let us now discuss the choice of the parameters θ and ν that govern the inner product and orthogonality tests. Both have statistical significance, and have considerable impact on the algorithm. For example, θ affects the value of the steplength α in the analysis presented in the previous section (see (3.6)), and consequently the linear convergence constant (see (3.11), (3.12)).

Concerning the parameter θ in the inner product test (2.6), we know that the sample mean is a random quantity that approximately follows a normal distribution, if a significant number of samples are used. In our case, this means,

$$\frac{\nabla F_{S_k}(x_k)^T \nabla F(x_k) - \|\nabla F(x_k)\|^2}{\left(\frac{\sigma}{\sqrt{|S_k|}}\right)} \sim \mathcal{N}(0,1), \tag{4.9}$$

where $\sigma^2 = \mathbb{E}\left[\left(\nabla F_i(x_k)^T \nabla F(x_k) - \|\nabla F(x_k)\|^2\right)^2\right]$ is the population variance employed in (2.5).

The one-sided $100(1-\varrho)\%$ confidence interval for $\nabla F_{S_k}(x_k)^T \nabla F(x_k)$ is given by

$$\left(\|\nabla F(x_k)\|^2 - z_\varrho \frac{\sigma}{\sqrt{|S_k|}}, \infty\right).$$

Since our goal is to have $\nabla F_{S_k}(x_k)^T \nabla F(x_k) > 0$, we propose that the lower value be nonnegative, which implies

$$\frac{\sigma^2}{|S_k|} \leq \frac{1}{z_\varrho^2}\|\nabla F(x_k)\|^4.$$

Therefore, we see from (2.6) that θ is related to z_ϱ. Smaller θ values increase the probability of obtaining a descent direction, and at the same time promote larger sample sizes. In particular, the value $\theta = 0.9$ corresponds to a probability of about 0.8; we have observed that this value works well in our numerical tests.

Let us now consider the choice of ν, which governs the orthogonality test (3.3). As discussed in the paragraph that follows (3.3), ν affects the tangent of the angle χ_k between $\nabla F_{S_k}(x_k)$ and $\nabla F(x_k)$. In practice, we choose ν to be sufficiently large such that the orthogonality test does not influence the sample size selection for most of the run, while at the same time ensuring that the angle χ_k is bounded away from 90°. This reasoning leads us to the choice $\nu = \tan 80^\circ = 5.84$. Although in theory one can choose much larger values of ν, this value works well in our tests — but like θ, ν should be regarded as a tunable parameter in the algorithm.

5 Numerical Experiments

In this section, we report the results of numerical experiments that illustrate the performance of the method proposed in this paper. We consider binary classification problems where the objective function is given by the logistic loss with ℓ_2 regularization:

$$R(x) = \frac{1}{N}\sum_{i=1}^{N} \log(1 + \exp(-z^i x^T y^i)) + \frac{\lambda}{2}\|x\|^2, \quad \text{with} \quad \lambda = \frac{1}{N}. \tag{5.1}$$

We used the datasets listed in Table 5.1.

Data Set	Data Points N	Variables d	Reference
gisette	6000	5000	[13]
synthetic	7000	50	[19]
mushroom	8124	112	[18]
sido	12678	4932	[18]
RCV1	20242	47236	[18]
ijcnn	35000	22	[18]
MNIST	60000	784	[17]
real-sim	65078	20958	[18]
covertype	581012	54	[3]

Table 5.1: Characteristics of the binary datasets used in the experiments.

Our goal is to compare the performance of the *augmented inner product test* implemented in Algorithm 3, which imposes both (2.6) and (3.3), and the norm test (2.10), which as mentioned above, dates back to Carter [7] and has recently received much attention. The practical form of the norm test is

$$\frac{1}{|S_k|-1}\frac{\sum_{i\in S_k}\|\nabla F_i(x_k)-\nabla F_{S_k}(x_k)\|^2}{|S_k|}\le \theta^2\|\nabla F_{S_k}(x_k)\|^2. \tag{5.2}$$

An approximation R^* of the optimal function value was computed for each problem by running the L-BFGS method until $\|\nabla R(x_k)\|_\infty \le 10^{-8}$. For all the experiments, we provide plots with respect to effective gradient evaluations and iterations. Effective gradient evaluations represent the total number of equivalent full gradients ∇R and full functions R computed in a run. All runs were terminated when $\|\nabla R(x_k)\|_\infty \le 10^{-6}$ or if 100 epochs (passes through the whole data set) are performed. We observed that for the finite sum problem (5.1), best performance is attained when the samples are chosen at random *without* replacement.

5.1 Tests with Fixed Steplengths

We first test the algorithm using a constant steplength rather than the line search (Algorithm 2) in order to evaluate the adaptive sampling strategies in a simple setting. Using a fixed steplength $\alpha_k = \alpha$, and tuning it for each problem so as to achieve best performance is a popular strategy in machine learning. In our experiments we select α from the set $\{2^{-10}, 2^{-7}, \cdots, 2^{15}\}$. All other parameters of the algorithms are set as follows: $\theta = 0.9$, $\nu = \tan(80^o) = 5.84$, $r = 10$, $\gamma = 0.38$. The initial sample size was set to $|S_0| = 2$.

Figure 5.1 reports the performance of the two methods for the `synthetic` dataset. The vertical axis measures the error in the function, $R(x) - R^*$, and the horizontal axis the number of effective gradient evaluations or iterations. (Results for the other datasets are given in Appendix B.) For this problem, the best steplength for both methods was $\alpha = 2^0$. The inner product test is clearly more efficient in terms of effective gradient evaluations, which is indicative of the total computational work and CPU time, but the norm test requires fewer iterations for reasons discussed in the next experiment.

In Figure 5.2, we plot sample size $|S_k|$ (as a percentage of the total dataset) at each iteration. We observe that the augmented inner product test increases the sample size slower than the norm test. The latter approximates the full gradient method earlier on, explaining the smaller number of iterations observed in Figure 5.1.

We also plot, in Figure 5.3, the angle (in degrees) between the sampled gradient $\nabla F_{S_k}(x_k)$ and true gradient $\nabla F(x_k)$, at each iteration. Note that the norm test rapidly reduces the angle between these two vectors, whereas the augmented inner product test allows for more freedom, with the angle varying significantly, but averaging about 70°. Results on the remaining data sets are given in the Appendix B.

5.2 Performance of the Complete Algorithm

We now compare the performance of Algorithm 3 against the variant that employs the norm test (5.2). Both utilize the line search procedure described in Algorithm 2. As before, we set $\theta = 0.9, \nu = 5.84, r = 10, \gamma = 0.38$. In Algorithm 2, the initial estimate of the

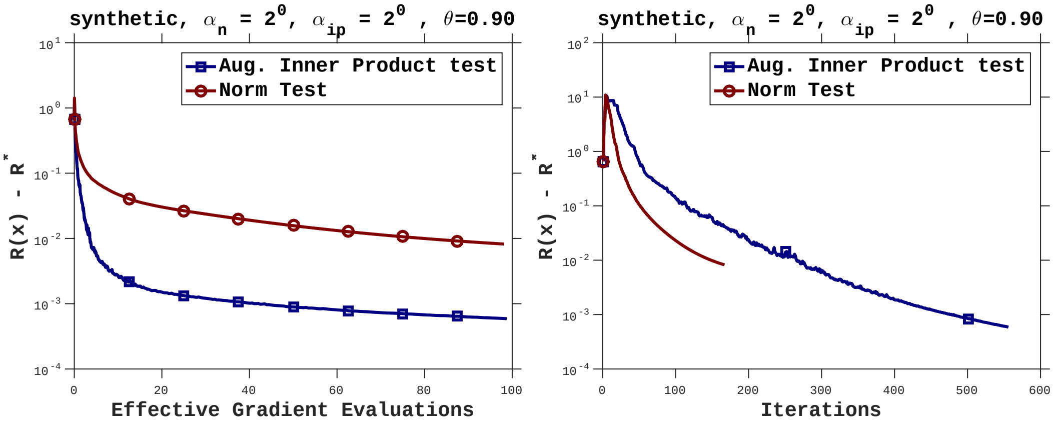

Figure 5.1: `synthetic` dataset: Performance of the augmented inner product test and norm test. Left: Function error vs. effective gradient evaluations; Right: Function error vs. iterations.

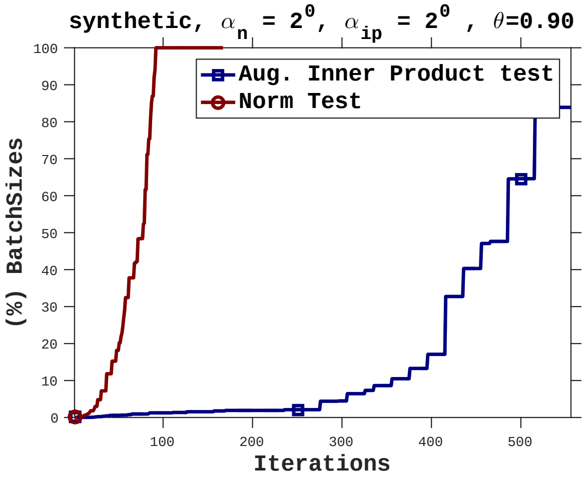

Figure 5.2: `synthetic` dataset: Growth of the sample (batch) size $|S_k|$ for the two methods, as a function of iterations.

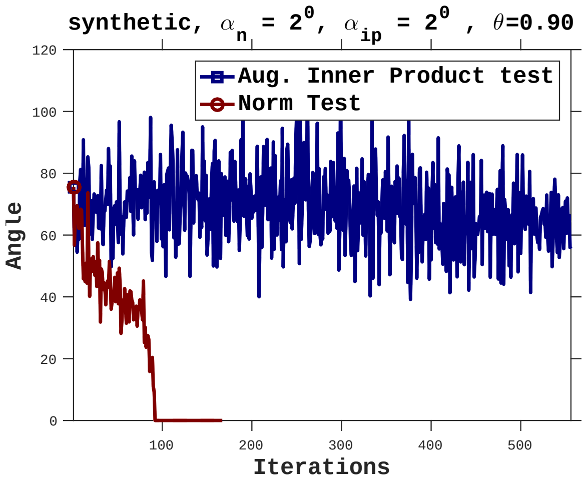

Figure 5.3: `synthetic` dataset: Angle between sampled gradient and true gradient vs. iterations for the two methods.

Lipschitz constant is $L_0 = 1$, and we use $\eta = 1.5$; we observed that these two values give good performance for all our test problems.

The plots in Figure 5.4 report function error vs. effective gradient evaluations, and batch sizes vs. iterations, respectively. Effective gradient evaluations include the function evaluations during the line search procedure, i.e., Algorithm 2. We observe that the augmented inner product test outperforms the norm test and increases the sample sizes less aggressively.

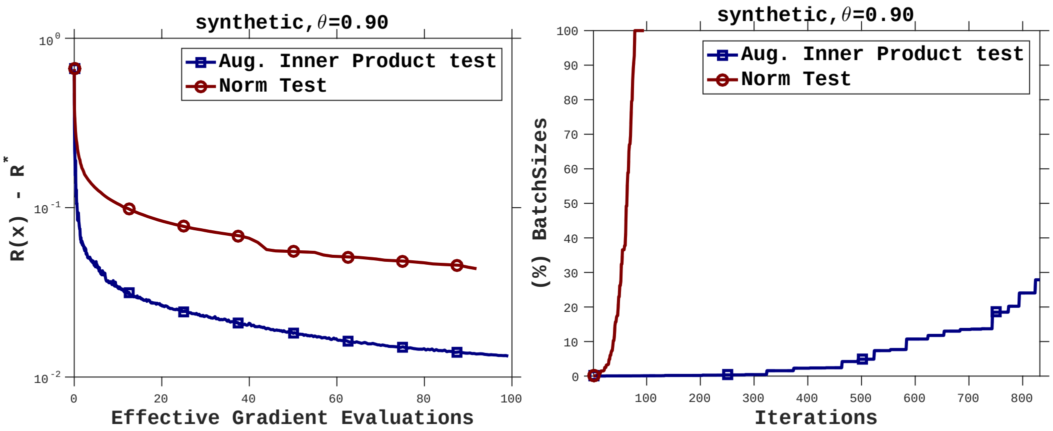

Figure 5.4: `synthetic` dataset: Performance of Algorithm 3 using the augmented inner product test and using the norm test. Left: Function error vs effective gradient evaluations; Right: Batch size $|S_k|$ vs. iterations.

To illustrate the performance of the line search procedure, we report in Figure 5.5 the steplengths chosen at each iteration by the two versions of the algorithm. As a reference, we also plot the best value ($\alpha = 1$) for the case when a fixed steplength is used. We observe that Algorithm 3, using the augmented inner product test, produced smaller steplengths initially, and that they were increased gradually until they approached 1.

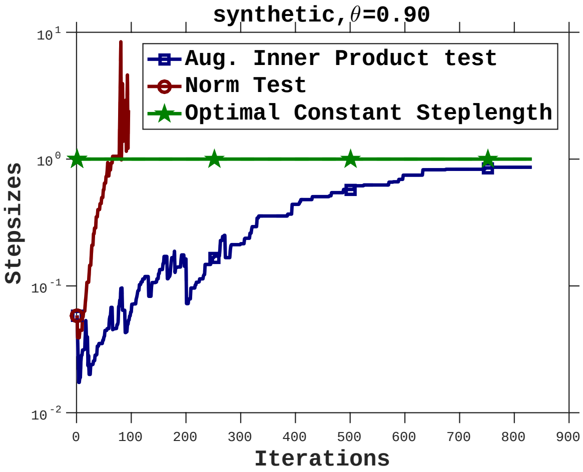

Figure 5.5: `synthetic` dataset: Performance of Algorithm 3 using the inner product test and using the norm test. Stepsizes vs. iterations

The results in Appendix B show the same pattern as in the `synthetic` dataset: Algorithm 3 using the inner product test is more efficient, and increases the sample size more gradually, than the norm test.

6 Final Remarks

Stochastic optimization algorithms that increase the sample size during the course of a run are interesting from a practical and theoretical perspective. We regard them as one of the most promising variance reducing methods for optimization applications involving very large data sets, as is the case in machine learning. Complexity analysis [6] has shown that adaptive sampling algorithms can be efficient in terms of total computational work. However, implementing them in practice has been challenging because a natural criterion (*the norm test*) for determining when gradient approximations should be improved is difficult to control and often leads to inefficient behavior. In this paper, we propose a different strategy that provides more freedom in the choice of the search direction; we refer to it as the *inner product test*. We argue that this technique is algorithmically appealing, and is more efficient in practice than the classical norm test strategy.

In this paper, we considered a first-order method but our methodology is amenable to the development of second-order methods [4, 11, 23]. An interesting open question is how to employ our inner product test in sub-sampled Newton and quasi-Newton methods for stochastic optimization. More generally, we believe that the concept underlying the inner product test can be the basis for new variance reducing optimization algorithms.

Acknowledgement. We thank Albert Berahas for many useful suggestions on how to improve the manuscript.

References

[1] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

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

[3] 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.

[4] Raghu Bollapragada, Richard Byrd, and Jorge Nocedal. Exact and inexact subsampled Newton methods for optimization. *arXiv preprint arXiv:1609.08502*, 2016.

[5] Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *arXiv preprint arXiv:1606.04838*, 2016.

[6] 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.

[7] 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.

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

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

[10] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pages 1646–1654, 2014.

[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] Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

[13] 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.

[14] Fatemeh S Hashemi, Soumyadip Ghosh, and Raghu Pasupathy. On adaptive sampling rules for stochastic recursions. In *Simulation Conference (WSC), 2014 Winter*, pages 3959–3970. IEEE, 2014.

[15] Tito Homem-De-Mello. Variable-sample methods for stochastic optimization. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 13(2):108–133, 2003.

[16] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems 26*, pages 315–323, 2013.

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

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

[19] Indraneel Mukherjee, Kevin Canini, Rafael Frongillo, and Yoram Singer. Parallel boosting with momentum. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 17–32. Springer Berlin Heidelberg, 2013.

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

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

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

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

[24] Johannes O Royset and Roberto Szechtman. Optimal budget allocation for sample average approximation. *Operations Research*, 61(3):762–776, 2013.

[25] Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming (MAPR), 2017.*, 2013.

A Proof of equations (2.11)-(2.12)

Proof. The numerator of the left hand side term in the condition (2.5) can be simplified by expanding the variance term and the inner product. That is,

$$\begin{aligned}\mathbb{E}\left[\left(\nabla F_i(x)^T\nabla F(x_k) - \|\nabla F(x)\|^2\right)^2\right] &= \mathbb{E}\left[\left(\nabla F_i(x)^T\nabla F(x)\right)^2\right] - \|\nabla F(x)\|^4 \\ &= \mathbb{E}\left[\|\nabla F_i(x)\|^2\cos^2(\chi_i)\right]\|\nabla F(x)\|^2 - \|\nabla F(x)\|^4.\end{aligned}$$

Substituting this equality in the inner product condition (2.5) we obtain,

$$\frac{\mathbb{E}\left[\|\nabla F_i(x)\|^2\cos^2(\chi_i)\right] - \|\nabla F(x)\|^2}{|S|} \le \theta^2\|\nabla F(x)\|^2. \tag{A.1}$$

Let$|S_i|$ be the minimum number of samples such that this condition is satisfied. Therefore,

$$|S_i| = \frac{\mathbb{E}\left[\|\nabla F_i(x)\|^2\cos^2(\chi_i)\right] - \|\nabla F(x)\|^2}{\theta^2\|\nabla F(x)\|^2}.$$

Now, we consider the norm test (2.10) and expand the variance in the numerator of the left hand side of the inequality to obtain

$$\frac{\mathbb{E}\left[\|\nabla F_i(x)\|^2\right] - \|\nabla F(x)\|^2}{|S|} \le \theta^2\|\nabla F(x)\|^2. \tag{A.2}$$

If $|S_n|$ is the minimum number of samples such that this condition is satisfied, we have

$$|S_n| = \frac{\mathbb{E}\left[\|\nabla F_i(x)\|^2\right] - \|\nabla F(x)\|^2}{\theta^2\|\nabla F(x)\|^2}.$$

Therefore,

$$\frac{|S_i|}{|S_n|} = \frac{\mathbb{E}[\|\nabla F_i(x)\|^2\cos^2(\chi_i)] - \|\nabla F(x)\|^2}{\mathbb{E}[\|\nabla F_i(x)\|^2] - \|\nabla F(x)\|^2} \tag{A.3}$$

Now since,

$$\mathbb{E}[\|\nabla F_i(x)\|^2\cos^2(\chi_i)] \le \mathbb{E}[\|\nabla F_i(x)\|^2],$$

we have that $\beta(x) \le 1$. □

B Additional Numerical Results

We present numerical results for the rest of the datasets listed in Table 5.1.

B.1 Tests with Fixed Steplengths

First, we consider the use of a fixed steplength parameter α, which is selected for each problem and each method so as to give optimal performance. In the figures that follow, α_n, α_{ip} denote the steplengths used in conjunction with the norm and inner product tests, respectively. In all experiments the parameter θ in (2.6) and (5.2) was set to $\theta = 0.9$, and the parameter ν in (3.3) was set to $\nu = \tan 80° = 5.84$.

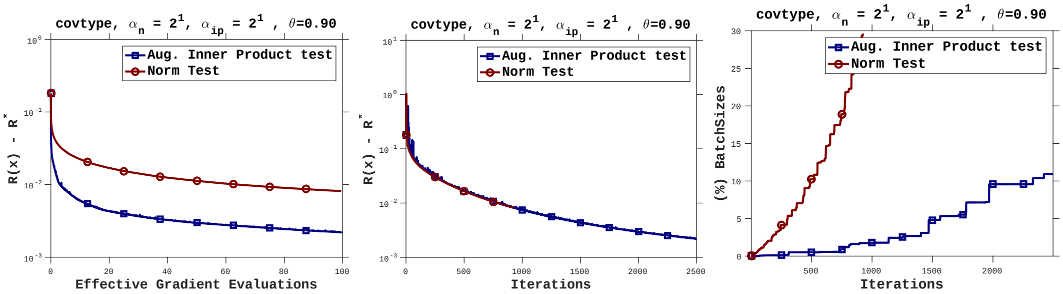

Figure B.1: `covertype` dataset: Performance of the adaptive sampling algorithm using the augmented inner product test and the norm test. The steplengths were chosen as $\alpha_n = 2^1$, $\alpha_{ip} = 2^1$. Left: Function error vs. effective gradient evaluations; Middle: Function error vs. iterations; Right: Batch size $|S_k|$ vs. iterations.

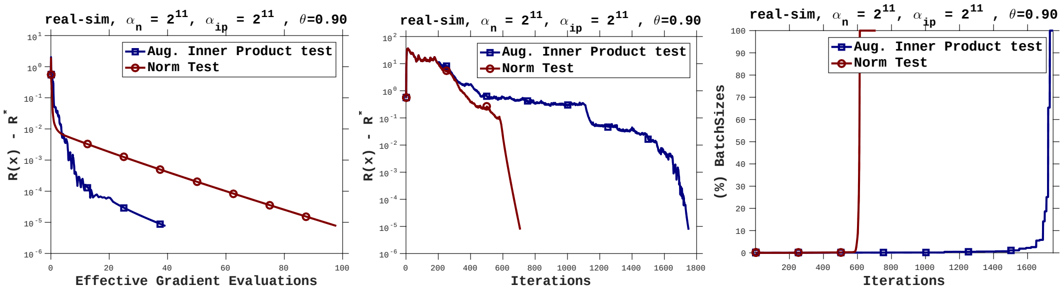

Figure B.2: `real-sim` dataset: Performance of the adaptive sampling algorithm using the augmented inner product test and the norm test. The steplengths were chosen as $\alpha_n = 2^{11}$, $\alpha_{ip} = 2^{11}$. Left: Function error vs. effective gradient evaluations; Middle: Function error vs. iterations; Right: Batch size $|S_k|$ vs. iterations.

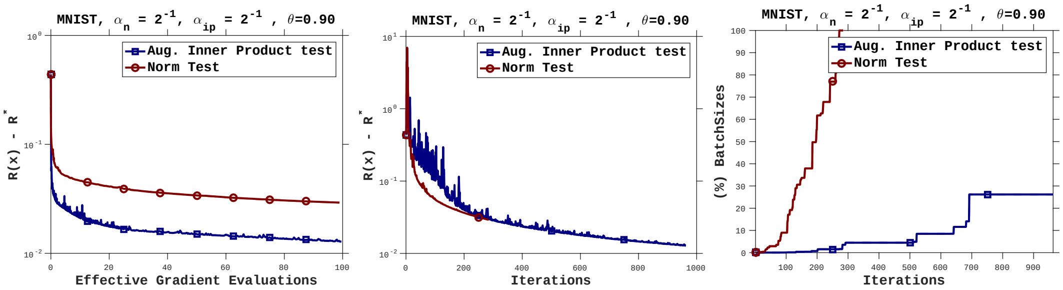

Figure B.3: `MNIST` dataset: Performance of the adaptive sampling algorithm using the augmented inner product test and the norm test. The steplengths were chosen as $\alpha_n = 2^{-1}$, $\alpha_{ip} = 2^{-1}$. Left: Function error vs. effective gradient evaluations; Middle: Function error vs. iterations; Right: Batch size $|S_k|$ vs. iterations.

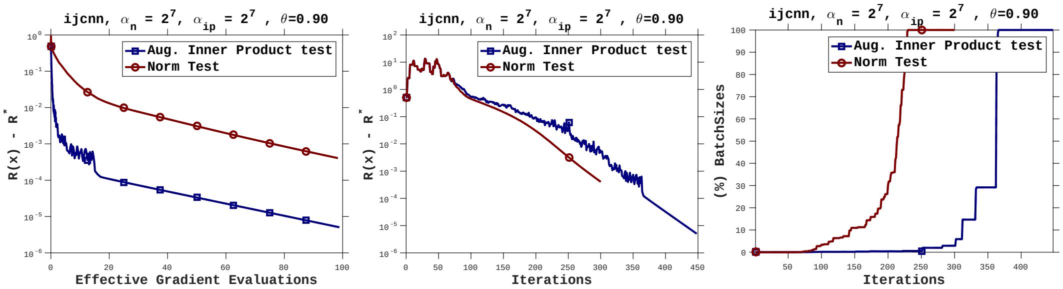

Figure B.4: `ijcnn` dataset: Performance of the adaptive sampling algorithm using the augmented inner product test and the norm test. The steplengths were chosen as $\alpha_n = 2^7$, $\alpha_{ip} = 2^7$.Left: Function error vs. effective gradient evaluations; Middle: Function error vs. iterations; Right: Batch size $|S_k|$ vs. iterations.

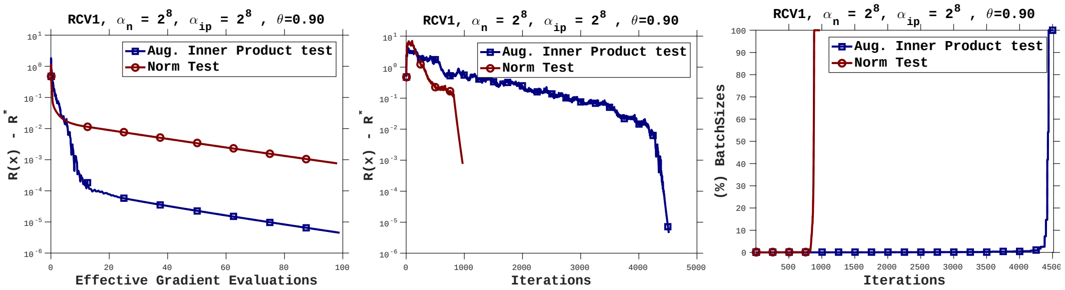

Figure B.5: `RCV1` dataset: Performance of the adaptive sampling algorithm using the augmented inner product test and the norm test. The steplengths were chosen as $\alpha_n = 2^8$, $\alpha_{ip} = 2^8$. Left: Function error vs. effective gradient evaluations; Middle: Function error vs. iterations; Right: Batch size $|S_k|$ vs. iterations.

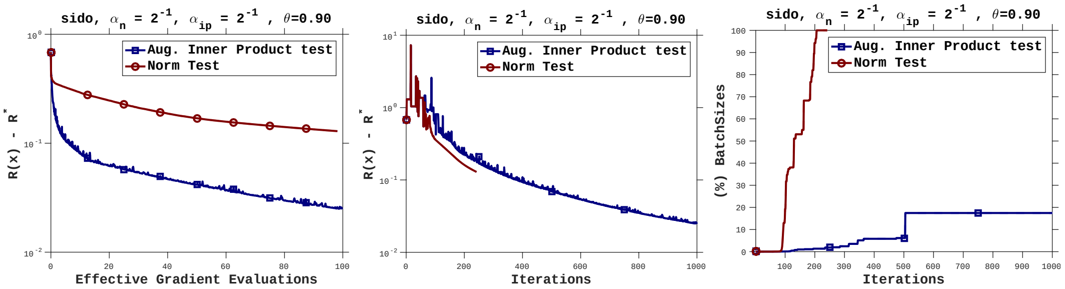

Figure B.6: `sido` dataset: Performance of the adaptive sampling algorithm using the augmented inner product test and the norm test. The steplengths were chosen as $\alpha_n = 2^{-1}$, $\alpha_{ip} = 2^{-1}$. Left: Function error vs. effective gradient evaluations; Middle: Function error vs. iterations; Right: Batch size $|S_k|$ vs. iterations.

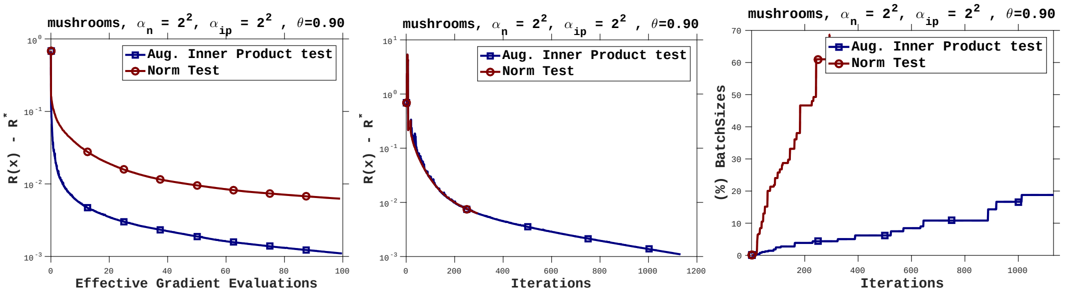

Figure B.7: `mushrooms` dataset: Performance of the adaptive sampling algorithm using the augmented inner product test and the norm test. The steplengths were chosen as $\alpha_n = 2^2$, $\alpha_{ip} = 2^2$. Left: Function error vs. effective gradient evaluations; Middle: Function error vs. iterations; Right: Batch size $|S_k|$ vs. iterations.

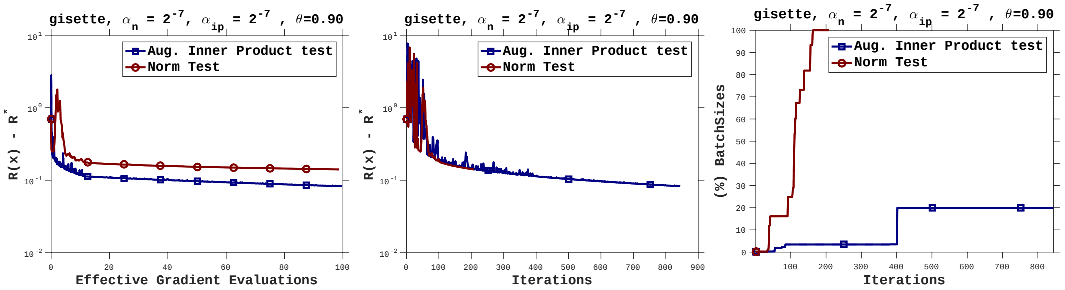

Figure B.8: `Gisette` dataset: Performance of the adaptive sampling algorithm using the augmented inner product test and the norm test. The steplengths were chosen as $\alpha_n = 2^{-7}$, $\alpha_{ip} = 2^{-7}$. Left: Function error vs. effective gradient evaluations; Middle: Function error vs. iterations; Right: Batch size $|S_k|$ vs. iterations.

B.2 Performance of the Complete Algorithm

We now consider the performance of Algorithm 3 which uses a line search procedure to select the steplength α_k at every iteration. The parameters θ and ν were chosen as stated above. The initial estimate of the Lipschitz constant was $L_0 = 1$, and we set $\eta = 1.5$.

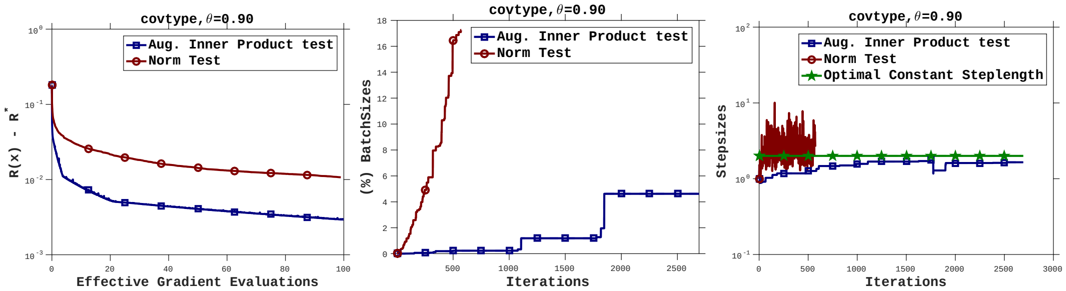

Figure B.9: `covertype` dataset: Performance of Algorithm 3 using the inner product test and using the norm test. Left: Function error vs. effective gradient evaluations; Middle: Batch size $|S_k|$ vs. iterations; Right: Stepsize vs. iterations.

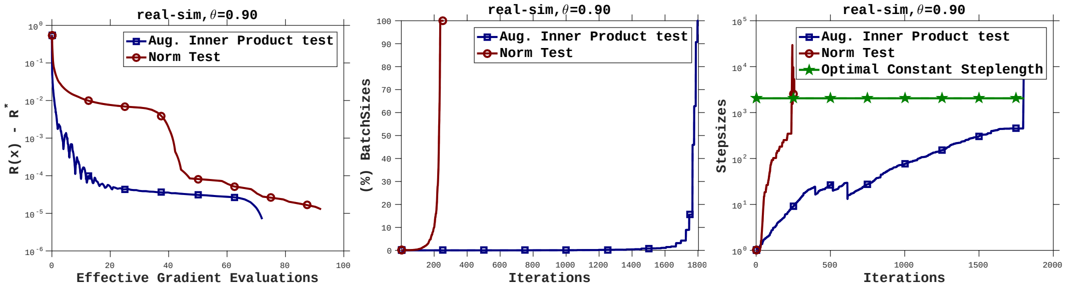

Figure B.10: `real-sim` dataset: Performance of Algorithm 3 using the inner product test and using the norm test. Left: Function error vs. effective gradient evaluations; Middle: Batch size $|S_k|$ vs. iterations; Right: Stepsize vs. iterations.

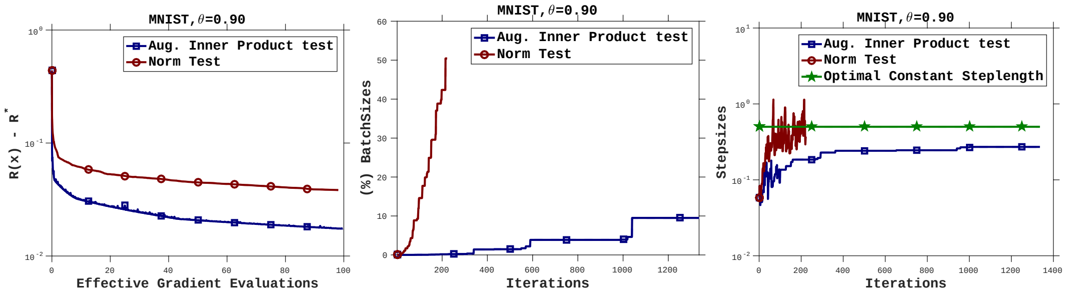

Figure B.11: `MNIST` dataset: Performance of Algorithm 3 using the inner product test and using the norm test. Left: Function error vs. effective gradient evaluations; Middle: Batch size $|S_k|$ vs. iterations; Right: Stepsize vs. iterations.

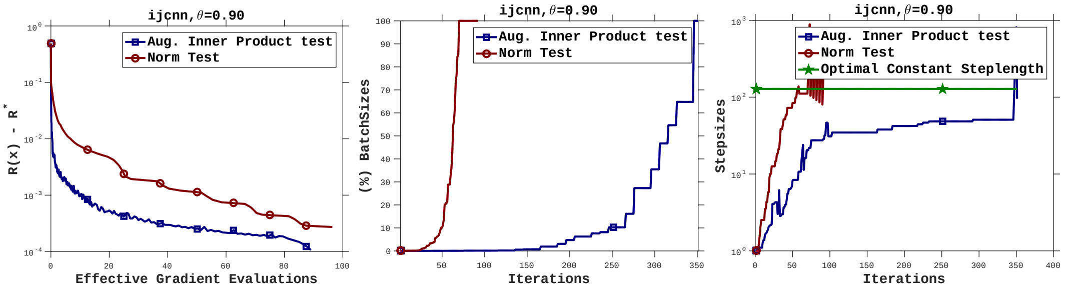

Figure B.12: `ijcnn` dataset: Performance of Algorithm 3 using the inner product test and using the norm test. Left: Function error vs. effective gradient evaluations; Middle: Batch size $|S_k|$ vs. iterations; Right: Stepsize vs. iterations.

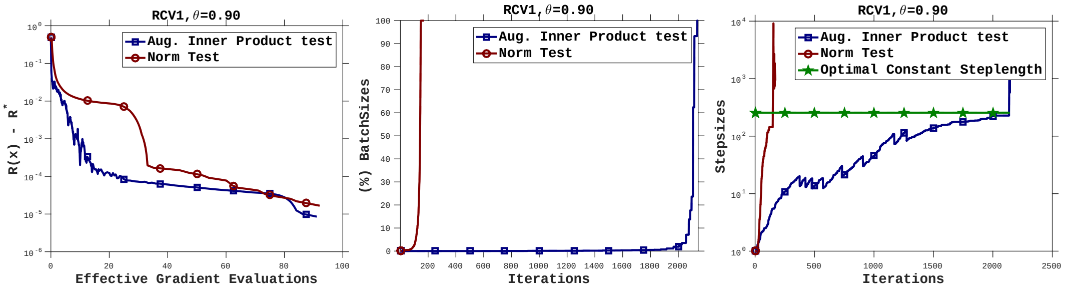

Figure B.13: `RCV1` dataset: Performance of Algorithm 3 using the inner product test and using the norm test. Left: Function error vs. effective gradient evaluations; Middle: Batch size $|S_k|$ vs. iterations; Right: Stepsize vs. iterations.

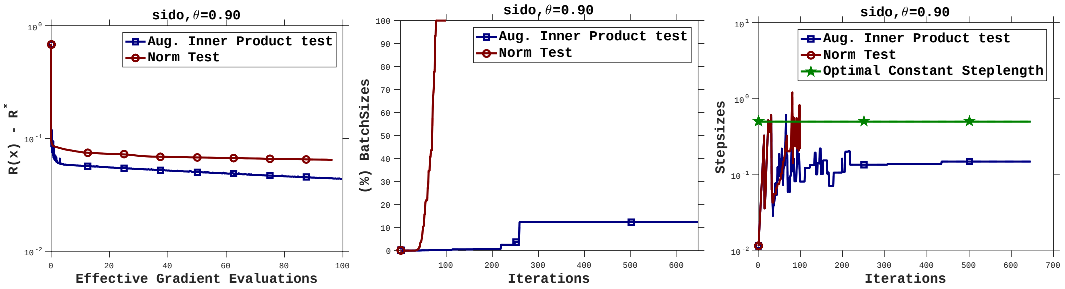

Figure B.14: `sido` dataset: Performance of Algorithm 3 algorithm using the inner product test and using the norm test. Left: Function error vs. effective gradient evaluations; Middle: Batch size $|S_k|$ vs. iterations; Right: Stepsize vs. iterations.

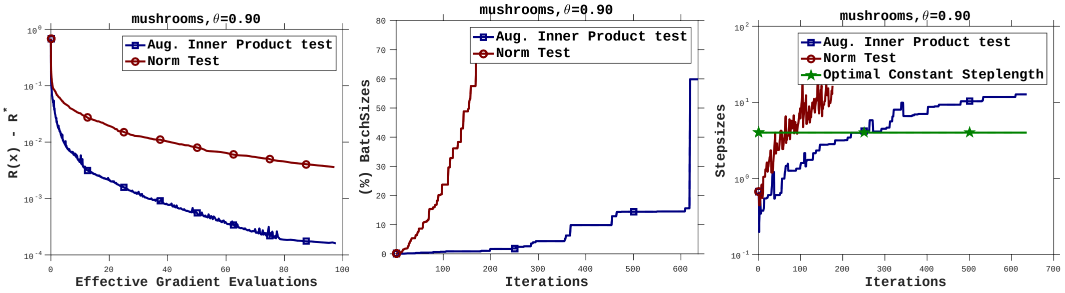

Figure B.15: `mushrooms` dataset: Performance of Algorithm 3 using the inner product test and using the norm test. Left: Function error vs. effective gradient evaluations; Middle: Batch size $|S_k|$ vs. iterations; Right: Stepsize vs. iterations.

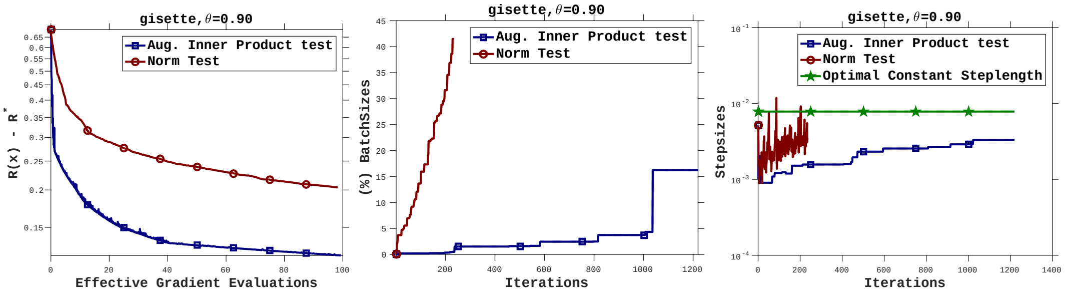

Figure B.16: `Gisette` dataset: Performance of Algorithm 3 using the inner product test and using the norm test. Left: Function error vs. effective gradient evaluations; Middle: Batch size $|S_k|$ vs. iterations; Right: Stepsize vs. iterations.