Two Losses Are Better Than One: Faster Optimization Using a Cheaper Proxy

Blake Woodworth [1]   Konstantin Mishchenko [2]   Francis Bach [1]

Abstract

We present an algorithm for minimizing an objective with hard-to-compute gradients by using a related, easier-to-access function as a proxy. Our algorithm is based on approximate proximal-point iterations on the proxy combined with relatively few stochastic gradients from the objective. When the difference between the objective and the proxy is δ-smooth, our algorithm guarantees convergence at a rate matching stochastic gradient descent on a δ-smooth objective, which can lead to substantially better sample efficiency. Our algorithm has many potential applications in machine learning, and provides a principled means of leveraging synthetic data, physics simulators, mixed public and private data, and more.

1. Introduction

In many machine learning problems, it is difficult to access the objective function, e.g., to compute its gradients for use in training. This difficulty can come from a wide range of sources. For example, it might be costly or time consuming to collect labelled training examples, leading to a small training set. In some applications, computing the gradient of the loss might require waiting for an actual robot to execute a policy in a physical environment, which could be very slow. Moreover, physical execution can risk damage to the robot or its surroundings. Finally, individuals' data may be subject to privacy or legal constraints, limiting access for gradient computations. In all of these cases, we face challenges that complicate training, and can lead to poor performance using off-the-shelf algorithms like stochastic gradient descent (SGD).

A natural solution to the problems above is to find a second,

easier-to-access "proxy" function that can be used in place of the real objective. When few samples are available, synthetic data could be used to approximate the objective, or in applications with sensitive data, we could train a generative model using a privacy-preserving algorithm and define the proxy using synthetic samples. To mitigate the cost and risk of operating a physical robot, one could use a simulator that is faster, cheaper, and safer.

Unfortunately, directly optimizing a hand-crafted proxy objective might not improve performance on the original problem, because the surrogate loss can introduce a bias. We aim to remedy this in our work, and our main contribution is a general algorithm that exploits curvature information from the proxy loss in order to better optimize the objective that we care about.

1.1. The problem

Our ultimate aim is to solve unconstrained convex optimization problems of the form

$$\min_{\mathbf{w}\in\mathbb{R}^d} L(\mathbf{w})\,. \tag{1}$$

Here, L could represent the population loss of a machine learning model parametrized by weights $\mathbf{w}$; it could also be the training loss, or any other objective function to be minimized. To solve (1), we use a proxy function $\hat{L}$ and stochastic gradients $\mathbf{g}_k$ approximating $\nabla L(\mathbf{w}_k)$ and update

$$\mathbf{w}_{k+1} \approx \arg\min_{\mathbf{w}}\bigg\{ \underbrace{\hat{L}(\mathbf{w})}_{\text{Proxy loss}} + \underbrace{\langle\mathbf{g}_k - \nabla\hat{L}(\mathbf{w}_k), \mathbf{w} - \mathbf{w}_k\rangle}_{\text{Bias correction}}$$
$$+ \underbrace{\frac{1}{2\eta}\|\mathbf{w} - \mathbf{w}_k\|^2}_{\text{Regularization}}\bigg\}. \tag{2}$$

Thus, each iteration uses just a single stochastic gradient from L, but requires a heavier computation based on $\hat{L}$. If we take $\hat{L} \equiv 0$, we note that the update (2) is equivalent to one stochastic gradient descent step

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \eta\mathbf{g}_k\,. \tag{3}$$

Therefore, SGD is a natural point of comparison for our algorithm, and the main question is to what extent updating using (2) (with non-zero $\hat{L}$) rather than (3) is worth the additional computational cost. We will show that when $\hat{L}$

is sufficiently "similar" to L, our algorithm's updates, (2), can converge to the minimum of L in substantially fewer iterations than would be needed for SGD, (3). Accordingly, the number of stochastic gradients from L that our algorithm needs is less than what would be needed by SGD.

For this to work, we obviously require that $\hat{L}$ somehow resembles L, and the particular notion of similarity that we consider is that the function $h := L - \hat{L}$ is differentiable and has δ-Lipschitz continuous gradients, which, in the case that L and $\hat{L}$ are twice-differentiable, is equivalent to requiring that $\forall_{\mathbf{w}} \|\nabla^2 L(\mathbf{w}) - \nabla^2 \hat{L}(\mathbf{w})\|_{\mathrm{op}} \le \delta$. This measure, sometimes referred to as "δ-Hessian similarity", is common in the optimization literature (see, e.g., Mairal, 2013a; 2015; Arjevani & Shamir, 2015; Kovalev et al., 2022; Chayti & Karimireddy, 2022). For certain problems, it can be easy to construct a proxy that satisfies this similarity condition for small δ. For instance, the Hessian of a least-squares objective is simply the feature covariance matrix, so an appropriate proxy can be defined using any estimate of the covariance matrix that is δ-accurate in the operator norm. In Section 3.1 we further show that for least squares and logistic regression, proxies with $\delta = 0$ can be constructed *without using any labels*.

1.2. Our contributions

We present Algorithm 1, which substitutes cheaper accesses to a proxy loss in place of costly accesses to the objective itself. In Section 2.2, we prove convergence guarantees for our algorithm in the convex and strongly convex settings, and in Section 2.4, we show that these guarantees imply statistical optimality as well as a dependence on the similarity parameter, δ, in the place of other properties of L itself, such as its potentially much larger smoothness parameter. On a technical level, our analysis improves over many similar methods by allowing each iteration's proximal-point subproblem to be solved inexactly, with the inexactness captured by a simple criterion that can be evaluated at the time of execution. Furthermore, in Section 2.3 we show that this criterion can be satisfied efficiently owing to the strong convexity of the proximal-point subproblem's objective. In Section 3, we discuss potential applications of our algorithm where the target objective is costly to access and a suitable proxy is available. In Section 4, we conduct experiments to show the efficacy of our algorithm in realistic problems, including one with a non-convex objective. Finally, in Section 5, we discuss several possible extensions of our work and we provide a preliminary analysis of our algorithm for the non-convex setting.

1.3. Background and related work

Using a more tractable surrogate in the place of the actual target loss is a very old idea in statistics and machine learning,

and some type of surrogate is used in almost any application. For instance, convex loss functions like the hinge or logistic loss are often used as surrogates for the discontinuous 0-1 classification loss. This avoids the computational intractability of minimizing the 0-1 loss (see, e.g., Arora et al., 1997), but note that this does not per se mitigate other difficulties with accessing the objective in the sense that we have discussed, e.g., limited access to training samples. Another classic example is empirical risk minimization (Vapnik & Chervonenkis, 1968) where the training loss is used as a surrogate for the real objective, the population loss. This more closely aligns with our motivation since it replaces the hard-to-evaluate population loss—which is defined by an unknown data distribution—with the training loss, which we can directly evaluate, compute gradients of, etc.

However, both of these examples rely on i.i.d. samples from the target distribution while our approach can exploit additional "side-information"—synthetic data, simulators, etc.—to complement a smaller collection of i.i.d. samples. Moreover, the way in which surrogates are classically used differs from our approach: typically, the surrogate is directly minimized and the result is taken as an estimate of the minimum of the target objective, whereas we use the proxy to facilitate optimization using stochastic gradients from the objective itself. As such, the classic approach relies upon minima of the surrogate approximating minima of the target, while our method only requires the proxy and objective to have similar second derivatives. This is not *necessarily* easier to achieve, but it is easier in certain cases (see, e.g., Section 3.1), and it is much more amenable to applications involving synthetic data, simulators, etc., where the proxy loss could have different minima due to idiosyncrasies of the synthetic data or simulator.

In more closely related work, Hendrikx et al. (2020) propose an algorithm, SPAG, for a strongly convex setting, which builds upon a previous method, DANE (Shamir et al., 2014), and which resembles an accelerated version of Algorithm 1. Their guarantees are analogous to ours up to acceleration, however both SPAG and DANE require exact proximal-point updates, while our algorithm and analysis allow these updates to be inexact. Motivated by the problem of minimizing over a subset of parameters, Parpas (2017) assumed that the surrogate subproblem is defined over a different space, with projection into that space and back only introducing a multiplicative error. In other related work, Chayti & Karimireddy (2022) propose several algorithms that use a surrogate for non-convex optimization, one of which resembles a momentum variant of Algorithm 1.

Mairal (2013a; 2015) studies the "majorization-minimization" meta-algorithm, where an upper bound on the objective is minimized in each iteration. Our method can be viewed as an instance of majorization-minimization

Algorithm 1 PROXYPROX

1: **Input:** initialization $\mathbf{w}_0 \in \mathbb{R}^d$, stepsize $\eta > 0$
2: **for** $k = 0, 1, \ldots, K - 1$ **do**
3:     Sample $\mathbf{g}_k$ such that $\mathbb{E}[\mathbf{g}_k \mid \mathbf{w}_k] = \nabla L(\mathbf{w}_k)$
4:     Set $\mathbf{w}_{k+1} \approx \arg\min_{\mathbf{w}} \varphi_k(\mathbf{w})$ where

$$\varphi_k(\mathbf{w}) := \langle \mathbf{g}_k, \, \mathbf{w} \rangle + D_{\hat{L}}(\mathbf{w}; \mathbf{w}_k) + \frac{1}{2\eta}\|\mathbf{w} - \mathbf{w}_k\|^2.$$

with the surrogate being used to define the upper bound on the loss. Mairal analyzes this method under the same δ-similarity condition that we use and prove similar convergence rates, but the key difference between this work and ours is that Mairal's methods require a guaranteed upper bound on the objective—or on components of the objective in the case of finite sum structured problems—and require that this upper bound be exactly minimized in each iteration, whereas we only require an in-expectation upper bound and allow for only approximate minimization. In this way, our setting is more appropriate for the stochastic optimization problems that arise in machine learning.

Finally, in the most closely related work, Mairal (2013b) study a stochastic majorization-minimization approach which is applicable to the stochastic optimization problems we are interested in. However, Mairal's approach requires a guaranteed upper bound of the loss function evaluated on each sample and requires that the δ-similarity bound hold almost surely for each sample. In contrast, we only require an in-expectation upper bound and that the δ-similarity hold in expectation, making it easier to find a suitable surrogate. Furthermore, our convergence guarantees have a better scaling with the number of iterations, K.

2. The algorithm and its analysis

We now present our algorithm and prove convergence guarantees in the convex and strongly convex settings. We will also argue that our method can provide substantially better guarantees than SGD when $\hat{L}$ is sufficiently similar to L.

2.1. Setting and notation

Throughout this paper, we will use $\|\mathbf{u}\|$ to denote the Euclidean norm of the vector $\mathbf{u}$. We will assume the minimum value of L, which we denote L^*, is realized, with $\mathbf{w}^*$ denoting an arbitrary minimizer. We use $B^2 := \mathbb{E}\|\mathbf{w}_0 - \mathbf{w}^*\|^2$ as shorthand to denote the distance between the (possibly random) initialization, $\mathbf{w}_0$, and this minimizer.

A function f is μ-strongly convex if for all $\mathbf{u}, \mathbf{v}$

$$f(\mathbf{v}) \geq f(\mathbf{u}) + \langle \nabla f(\mathbf{u}), \, \mathbf{v} - \mathbf{u} \rangle + \frac{\mu}{2}\|\mathbf{u} - \mathbf{v}\|^2, \quad (4)$$

where, in the event that f is not differentiable, $\nabla f(\mathbf{u})$ de-

notes here an arbitrary subgradient of f. When this holds for $\mu = 0$, f is merely convex. A function f is H-smooth if it is differentiable and ∇f is H-Lipschitz continuous or, equivalently, for all $\mathbf{u}, \mathbf{v}$

$$|f(\mathbf{v}) - f(\mathbf{u}) - \langle \nabla f(\mathbf{u}), \, \mathbf{v} - \mathbf{u} \rangle| \leq \frac{H}{2}\|\mathbf{u} - \mathbf{v}\|^2. \quad (5)$$

We define the Bregman divergence with potential ψ as

$$D_\psi(\mathbf{u}; \mathbf{v}) := \psi(\mathbf{u}) - \psi(\mathbf{v}) - \langle \nabla \psi(\mathbf{v}), \, \mathbf{u} - \mathbf{v} \rangle. \quad (6)$$

While Bregman divergences are typically defined only for strictly convex potentials ψ, in this paper we allow ψ to be any differentiable function with D_ψ simply defined as in (6). The key property of Bregman divergences for our purposes is the three-point identity:[1] for all $\mathbf{u}, \mathbf{v}, \mathbf{w}$

$$D_\psi(\mathbf{u}; \mathbf{v}) - D_\psi(\mathbf{u}; \mathbf{w}) - D_\psi(\mathbf{w}; \mathbf{v})$$
$$= \langle \nabla \psi(\mathbf{v}) - \nabla \psi(\mathbf{w}), \, \mathbf{w} - \mathbf{u} \rangle. \quad (7)$$

Our algorithm relies upon two main assumptions; one which captures the similarity between the objective L and the proxy $\hat{L}$, and one which captures the level of noise in the stochastic gradients $\mathbf{g}_k$. First, we assume

Assumption 2.1. The function $h := L - \hat{L}$ is differentiable and ∇h is δ-Lipschitz continuous.

That is, we require that our proxy $\hat{L}$ has similar curvature to the objective L. In light of (5), Assumption 2.1 implies that

$$|D_h(\mathbf{u}; \mathbf{v})| = \left| D_L(\mathbf{u}; \mathbf{v}) - D_{\hat{L}}(\mathbf{u}; \mathbf{v}) \right| \leq \frac{\delta}{2}\|\mathbf{u} - \mathbf{v}\|^2.$$

This justifies the definition of φ_k in Algorithm 1 because, when δ is small

$$\varphi_k(\mathbf{w}) \approx \langle \mathbf{g}_k, \, \mathbf{w} \rangle + D_L(\mathbf{w}; \mathbf{w}_k) + \frac{1}{2\eta}\|\mathbf{w} - \mathbf{w}_k\|^2,$$

which corresponds to (inexact, stochastic) proximal-point iterations directly on the objective L, which is well-known to have good performance (see, e.g., Barré et al., 2022).

We note that Assumption 2.1 can accommodate non-differentiable objectives, e.g., if $L(\mathbf{w}) = \ell(\mathbf{w}) + \psi(\mathbf{w})$ and $\hat{L}(\mathbf{w}) = \hat{\ell}(\mathbf{w}) + \psi(\mathbf{w})$ with ℓ and $\hat{\ell}$ differentiable, because ψ does not appear in $L - \hat{L}$. In this way, our algorithm is compatible, e.g., with non-smooth regularizers such as an L_1 penalty as long as this is incorporated into the proxy.

In addition to Assumption 2.1, we require the following:

Assumption 2.2. For each k, $\mathbb{E}[\mathbf{g}_k \mid \mathbf{w}_k] = \nabla L(\mathbf{w}_k)$ and $\mathbb{E}\left[\|\mathbf{g}_k - \nabla L(\mathbf{w}_k)\|^2 \mid \mathbf{w}_k\right] \leq \sigma^2$. If L is not differentiable but convex, this holds for some subgradient in $\partial L(\mathbf{w}_k)$.

---

[1]This does not require convexity (Chen & Teboulle, 1993, Lemma 3.1).

The assumption that the stochastic gradients $\mathbf{g}_k$ are unbiased and have bounded variance is very common in the optimization literature (Nemirovskii & Yudin, 1983; Dekel et al., 2012; Bubeck, 2015).

2.2. Convergence guarantees and proof sketch

We begin with our main result:

Theorem 2.3. *Under Assumptions 2.1 and 2.2, let L be μ-strongly convex and let $\eta \leq \frac{1}{4\delta}$. If for each k*

$$\mathbb{E}\|\nabla\varphi_k(\mathbf{w}_{k+1})\|^2 \leq \frac{\mu}{4\eta}\mathbb{E}\|\mathbf{w}_{k+1} - \mathbf{w}_k\|^2 + G^2,$$

for some G, then a geometrically weighted average of the iterates of Algorithm 1, $\bar{\mathbf{w}}_K$ (see (25)), satisfies

$$\mathbb{E}L(\bar{\mathbf{w}}_K) - L^* \leq \frac{5B^2}{8\eta}\left(1 + \frac{2\eta\mu}{5}\right)^{1-K} + 2\eta\sigma^2 + \frac{G^2}{\mu},$$

where the expectation is taken over the randomness in both the stochastic gradients and the selection of the iterates.

A detailed proof can be found in Appendices A and B. We start with a straightforward algebraic manipulation of $\nabla\varphi_k$. Recalling that $h = L - \hat{L}$, we write[2]

$$\nabla\varphi_k(\mathbf{w}_{k+1}) = -\nabla h(\mathbf{w}_{k+1}) + \nabla h(\mathbf{w}_k) + \nabla L(\mathbf{w}_{k+1})$$
$$+ \mathbf{g}_k - \nabla L(\mathbf{w}_k) + \frac{\mathbf{w}_{k+1} - \mathbf{w}_k}{\eta}. \quad (8)$$

Next, (6) implies

$$L(\mathbf{w}_{k+1}) - L^*$$
$$= \langle \nabla L(\mathbf{w}_{k+1}), \mathbf{w}_{k+1} - \mathbf{w}^* \rangle - D_L(\mathbf{w}^*; \mathbf{w}_{k+1}). \quad (9)$$

Rearranging (8) allows us to express $\nabla L(\mathbf{w}_{k+1})$ in terms of $\nabla\varphi_k(\mathbf{w}_{k+1})$ and the other quantities, which we substitute in the place of $\nabla L(\mathbf{w}_{k+1})$ in (9). After several straightforward algebraic manipulations, including using the three-point identity (7), we arrive at the key identity

$$L(\mathbf{w}_{k+1}) - L^* \quad (10)$$

$$= \frac{1}{2\eta}\|\mathbf{w}_k - \mathbf{w}^*\|^2 - D_h(\mathbf{w}^*; \mathbf{w}_k) \left.\right\} \text{(Telescope)}$$
$$- \frac{1}{2\eta}\|\mathbf{w}_{k+1} - \mathbf{w}^*\|^2 + D_h(\mathbf{w}^*; \mathbf{w}_{k+1})$$
$$+ \langle \nabla L(\mathbf{w}_k) - \mathbf{g}_k, \mathbf{w}_{k+1} - \mathbf{w}^* \rangle \quad \} \quad \text{(Noise)}$$
$$+ \langle \nabla\varphi_k(\mathbf{w}_{k+1}), \mathbf{w}_{k+1} - \mathbf{w}^* \rangle \quad \} \quad \text{(Inexact)}$$
$$+ D_h(\mathbf{w}_{k+1}; \mathbf{w}_k) - \frac{1}{2\eta}\|\mathbf{w}_{k+1} - \mathbf{w}_k\|^2 \left.\right\} \text{(Cancel)}$$
$$- D_L(\mathbf{w}^*; \mathbf{w}_{k+1}).$$

---

[2]Here, for simplicity, assume L and $\hat{L}$ are differentiable, but see Appendix A for the general case.

Although the expression looks unwieldy at first, it leads us directly towards a strategy for analyzing Algorithm 1.

The Telescope terms resemble those typically found in the analysis of convex optimization algorithms. In the course of proving Theorem 2.3, we, roughly speaking, average (10) over k, causing all but two of these terms to cancel out, and we show that under Assumption 2.1 with $\eta \leq \frac{1}{4\delta}$, their total contribution to the error is small.

The Noise term captures the effect of using noisy gradients $\mathbf{g}_k$ rather than $\nabla L(\mathbf{w}_k)$. Under Assumption 2.2, its expectation is at most $2\eta\sigma^2 + \frac{1}{4\eta}\mathbb{E}\|\mathbf{w}_{k+1} - \mathbf{w}_k\|^2$. Importantly, since $\mathbf{g}_k$ is unbiased, we can bound this in terms of $\|\mathbf{w}_{k+1} - \mathbf{w}_k\|$ rather than $\|\mathbf{w}_{k+1} - \mathbf{w}^*\|$.

The Inexact term captures the effect of only approximately computing the proximal-point update. If we could just set $\mathbf{w}_{k+1} = \arg\min_{\mathbf{w}} \varphi_k(\mathbf{w})$, the Inexact term would be zero, but this is impossible in practice. Instead, a simple application of Young's inequality bounds this term by $\frac{1}{\mu}\|\nabla\varphi_k(\mathbf{w}_{k+1})\|^2 + \frac{\mu}{4}\|\mathbf{w}_{k+1} - \mathbf{w}^*\|^2$.

Nowhere above did we use any property of L besides those implied by Assumptions 2.1 and 2.2, but to address the Cancel terms, which we use to eliminate unwanted quantities introduced above, we finally use that in the μ-strongly convex setting, $D_L(\mathbf{w}^*; \mathbf{w}_{k+1}) \geq \frac{\mu}{2}\|\mathbf{w}_{k+1} - \mathbf{w}^*\|^2$. So, when $\mathbf{w}_{k+1}$ is chosen so that $\|\nabla\varphi_k(\mathbf{w}_{k+1})\|^2$ is sufficiently small, this allows us to cancel out everything in Noise and Inexact except for $2\eta\sigma^2$, and this leaves an extra $-\frac{\mu}{4}\|\mathbf{w}_{k+1} - \mathbf{w}^*\|^2$ to be incorporated into the Telescope terms to generate geometric shrinkage. Here, the strong convexity of L is crucially needed to cancel the dependence of the Inexact term on $\|\mathbf{w}_{k+1} - \mathbf{w}^*\|$—without strong convexity, it is not immediately clear how the Inexact term can be controlled without setting $\nabla\varphi_k(\mathbf{w}_{k+1}) = 0$. Filling in the details and using standard arguments completes the proof.

While Theorem 2.3 requires L to be μ-strongly convex, a similar guarantee can be achieved when L is convex via regularization:

Theorem 2.4. *Under Assumptions 2.1 and 2.2, let L be convex and let $\eta \leq \frac{1}{4\delta}$. Then applying Algorithm 1 to the regularized objectives $L^{(\mu)}(\mathbf{w}) := L(\mathbf{w}) + \frac{\mu}{2}\|\mathbf{w} - \mathbf{w}_0\|^2$ and $\hat{L}^{(\mu)}(\mathbf{w}) := \hat{L}(\mathbf{w}) + \frac{\mu}{2}\|\mathbf{w} - \mathbf{w}_0\|^2$ with $\mu = \frac{1}{\eta K}$ and any $\eta \leq \frac{1}{4\delta}$ ensures that if for each k*

$$\mathbb{E}\|\nabla\varphi_k(\mathbf{w}_{k+1})\|^2 \leq \frac{1}{4\eta^2 K}\mathbb{E}\|\mathbf{w}_{k+1} - \mathbf{w}_k\|^2 + G^2$$

for some G, then

$$\mathbb{E}L\left(\frac{1}{K}\sum_{k=1}^{K}\mathbf{w}_k\right) - L^* \leq \frac{9B^2}{8\eta K} + 2\eta\sigma^2 + \eta K G^2.$$

The proof, which we defer to Appendix B, simply observes

that $L^{(\mu)}$ and $\hat{L}^{(\mu)}$ satisfy the conditions needed for Theorem 2.3, with larger μ leading to faster convergence. At the same time, when μ is small enough, optimizing $L^{(\mu)}$ is essentially equivalent to optimizing L itself. We conclude by choosing μ to balance between these competing goals.

Finally, choosing η optimally in the strongly convex and convex settings yields the following complexity guarantee, which we prove in Appendix C:

Corollary 2.5. *In the setting of Theorem 2.3, there is a universal constant c s.t. for any $\varepsilon > 0$, when $G^2 \leq \frac{1}{2}\mu\varepsilon$ and*

$$\eta = \frac{5}{\mu(K-1)}\Big(1 + \ln\Big(\frac{5B^2\mu(K-1)}{\varepsilon}\Big)\Big),$$

then the output, $\hat{\mathbf{w}}$, of Algorithm 1 will have suboptimality at most $\mathbb{E}L(\hat{\mathbf{w}}) - L^* \leq \varepsilon$ whenever

$$K \geq c \cdot \Big(1 + \frac{\delta}{\mu} + \frac{\sigma^2}{\mu\varepsilon}\Big)\ln\Big(e + \frac{(\mu+\delta)B^2}{\varepsilon} + \frac{\sigma^2 B^2}{\varepsilon^2}\Big).$$

Likewise, in the setting of Theorem 2.4, there is a universal constant c s.t. for any $\varepsilon > 0$, when $G^2 \leq \frac{\sigma^2}{K} + \frac{\delta^2 B^2}{K^2}$ and

$$\eta = \min\Big\{\frac{1}{4\delta}, \frac{B}{\sigma\sqrt{K}}\Big\},$$

then the output, $\hat{\mathbf{w}}$, of Algorithm 1 will have suboptimality at most $\mathbb{E}L(\hat{\mathbf{w}}) - L^* \leq \varepsilon$ whenever

$$K \geq c \cdot \Big(\frac{\delta B^2}{\varepsilon} + \frac{\sigma^2 B^2}{\varepsilon^2}\Big).$$

2.3. Making $\nabla\varphi_k$ small

Theorems 2.3 and 2.4 require that each update satisfy

$$\mathbb{E}\|\nabla\varphi_k(\mathbf{w}_{k+1})\|^2 \leq \frac{\mu}{4\eta}\mathbb{E}\|\mathbf{w}_{k+1} - \mathbf{w}_k\|^2 + G^2,$$

with $\mu = \frac{1}{\eta K}$ in the convex setting. This is a convenient inexactness criterion, because it consists entirely of quantities that can be evaluated at the time of execution, and similar bounds on the inaccuracy have been studied in the context of the classic proximal-point algorithm (see, e.g., Rockafellar, 1976; Barré et al., 2022). Nevertheless, this raises the question of how difficult it is to find such a $\mathbf{w}_{k+1}$. It turns out that this can be quite easy due to the $\frac{1}{\eta}$-strong convexity of the subproblem objective φ_k. For example, we could use stochastic gradient descent:

Proposition 2.6. *Let $\hat{L}$ be convex and H-smooth and let stochastic gradients be available for $\hat{L}$ with variance at most ρ^2. Then for constants c, c', the output, $\hat{\mathbf{w}}$, of*

$$T \geq c(1 + H\eta)\max\Big\{\ln\Big(\frac{c'(1+H\eta)^2}{\mu\eta}\Big), \frac{\rho^2}{G^2}\Big\}$$

steps of SGD *on φ_k initialized at $\mathbf{w}_k$ satisfies*

$$\mathbb{E}\|\nabla\varphi_k(\hat{\mathbf{w}})\|^2 \leq \frac{\mu}{4\eta}\|\hat{\mathbf{w}} - \mathbf{w}_k\|^2 + G^2.$$

The proof, which we defer to Appendix E, is based on the observation that $\hat{L}$ being H-smooth implies that φ_k is $(H+\frac{1}{\eta})$-smooth, so $\|\nabla\varphi_k(\hat{\mathbf{w}})\|^2 \leq 2(H+\frac{1}{\eta})(\varphi_k(\hat{\mathbf{w}})-\varphi_k^*)$, and plugging in the standard suboptimality guarantee for T steps of SGD on a smooth and strongly convex objective eventually leads to the desired bound.

There is nothing particularly special about using SGD to compute the update $\mathbf{w}_{k+1}$. An advantage of our method is that any algorithm for approximately minimizing φ_k will suffice when $\hat{L}$ is smooth, so we can exploit structure in φ_k by using a specialized optimization algorithm. There are also other, more exotic SGD variants that directly target a solution with a small gradient norm (Allen-Zhu, 2018; Foster et al., 2019), and can have a better dependence on ρ^2.

2.4. Comparison against directly optimizing L

To get a better sense of what Theorems 2.3 and 2.4 mean, we will compare them against what we could do by just directly optimizing the objective L using stochastic gradients. Suppose that Assumption 2.2 holds, the objective L is H-smooth, $\|\mathbf{w}_0 - \mathbf{w}^*\|^2 = B^2$, and L is either convex or μ-strongly convex. In this case, a natural and popular approach to minimizing L using a total of K stochastic gradients would be to execute K steps of SGD. Ignoring constant factors, it is well-known that this would yield an ε-accurate solution when (Nemirovskii & Yudin, 1983)

$$K \geq \frac{HB^2}{\varepsilon} + \frac{\sigma^2 B^2}{\varepsilon^2} \tag{11}$$

$$K \geq \frac{H}{\mu}\ln\frac{HB^2}{\varepsilon} + \frac{\sigma^2}{\mu\varepsilon}. \tag{12}$$

Instead, given a proxy $\hat{L}$ satisfying Assumption 2.1, we could use the same number of stochastic gradients from L to execute K steps of Algorithm 1. Ignoring constants and a logarithmic factor in the strongly convex setting, Corollary 2.5 shows that this yields an ε-accurate solution when

$$K \geq \frac{\delta B^2}{\varepsilon} + \frac{\sigma^2 B^2}{\varepsilon^2} \tag{13}$$

$$K \geq \frac{\delta}{\mu}\ln\frac{\delta B^2}{\varepsilon} + \frac{\sigma^2}{\mu\varepsilon}. \tag{14}$$

Comparing (11) and (12) against (13) and (14), we see that Algorithm 1 essentially replaces SGD's dependence on H, the smoothness constant of L, with δ, the smoothness constant of $h = L - \hat{L}$. So if a proxy $\hat{L}$ can be found with $\delta \ll H$, Algorithm 1 may have a much better guarantee than SGD.

On the other hand, the second, "statistical terms" of our guarantees match those of SGD. These statistical terms capture information-theoretic limits to optimizing L using only K stochastic gradients with variance σ^2, and these are

known to be unimprovable in the worst case (Nemirovskii & Yudin, 1983), so our algorithm is statistically optimal (up to a logarithmic factor in the strongly convex setting). In fact, long-standing lower bounds show that obtaining an ε-accurate solution using *any* algorithm that only uses K stochastic gradients from L requires at least (Nemirovskii & Yudin, 1983)

$$K \geq \frac{HB^2}{\sqrt{\varepsilon}} + \frac{\sigma^2 B^2}{\varepsilon^2}$$
$$K \geq \sqrt{\frac{H}{\mu}} \ln \frac{HB^2}{\varepsilon} + \frac{\sigma^2}{\mu\varepsilon}.$$

Therefore, when $\delta \leq \frac{H}{\sqrt{\varepsilon}}$ in the convex setting or $\delta \lesssim \sqrt{H\mu}$ in the strongly convex case, then Algorithm 1 can surpass this lower bound by exploiting the proxy $\hat{L}$ in order to achieve a better guarantee than could be achieved by *any* algorithm that only uses K stochastic gradients from L.

Although Algorithm 1 only requires one stochastic gradient from L for each iteration, each update does require finding an approximate stationary point of the function φ_k, which is more computationally expensive than one SGD update. That said, in the μ-strongly convex case (the conclusion is similar in the merely convex setting), if we ignore all constant and logarithmic factors, Corollary 2.5 shows that for $K := \frac{\delta}{\mu} + \frac{\sigma^2}{\mu\epsilon}$, $G^2 = \mu\epsilon$, and $\eta = \frac{1}{\mu K}$, Algorithm 1 will find an $O(\epsilon)$-approximate minimizer of L. Furthermore, Proposition 2.6 (with $\rho = \sigma$) shows that the proximal subproblem in each iteration of Algorithm 1 can be solved to sufficient accuracy using $T := \frac{K}{\mu K} + \frac{H\sigma^2}{\mu^2 K\epsilon}$ steps of SGD on φ_k. Therefore, the total number of gradient update computations needed to implement Algorithm 1 is $TK = \frac{H}{\mu} + \frac{H}{\mu} \cdot \frac{\sigma^2}{\mu\epsilon}$. In contrast, (12) implies that SGD directly on L would require $S := \frac{H}{\mu} + \frac{\sigma^2}{\mu\epsilon}$ gradient update computations to reach an $O(\epsilon)$-approximate minimizer of L.

So, SGD directly on L could be computationally cheaper than Algorithm 1 in terms of the total number of floating point operations needed since $S < TK$. However, crucially, Algorithm 1 only requires K stochastic gradients from L itself (the other gradients come from $\hat{L}$), while SGD needs $S > K$. So to summarize, executing Algorithm 1 could require more computation than SGD on L, but it can make up for this by using fewer stochastic gradients from L, which could yield significant savings when L is harder to access.

2.5. Connection to Preconditioning

A useful way to understand Algorithm 1 is to consider the special case where the proxy $\hat{L}(\mathbf{w}) = \frac{1}{2}\mathbf{w}^\top \mathbf{P}\mathbf{w} + \mathbf{b}^\top \mathbf{w}$ is quadratic. In this case, exactly solving the proximal

subproblems in Algorithm 1 amounts to solving

$$\mathbf{g}_k + \nabla\hat{L}(\mathbf{w}_{k+1}) - \nabla\hat{L}(\mathbf{w}_k) + \frac{1}{\eta}(\mathbf{w}_{k+1} - \mathbf{w}_k) = 0$$
$$\iff \mathbf{w}_{k+1} = \mathbf{w}_k - \eta(\mathbf{I} + \eta\mathbf{P})^{-1}\mathbf{g}_k. \quad (15)$$

That is, each update in Algorithm 1 amounts to a single preconditioned SGD update with preconditioning matrix $(\mathbf{I} + \eta\mathbf{P})^{-1}$. When a quadratic proxy that satisfies Assumption 2.1 can be found, then these updates with a fixed preconditioning matrix will be very effective at minimizing the objective.

However, for many objectives, the Hessian $\nabla^2 L$ is not constant (nor close to constant) and so it cannot be well-approximated everywhere by a matrix $\mathbf{P}$ and thus any fixed preconditioner may not be effective. In this more general case, we can view each step of Algorithm 1 as approximating a locally preconditioned SGD update with the preconditioning matrix $(\mathbf{I} + \eta\nabla^2\hat{L}(\mathbf{w}_k))^{-1}$. This is because for sufficiently smooth $\hat{L}$, $\nabla\hat{L}(\mathbf{w}_{k+1}) - \nabla\hat{L}(\mathbf{w}_k) \approx \nabla^2\hat{L}(\mathbf{w}_k)(\mathbf{w}_{k+1} - \mathbf{w}_k)$ so

$$\mathbf{g}_k + \nabla\hat{L}(\mathbf{w}_{k+1}) - \nabla\hat{L}(\mathbf{w}_k) + \frac{1}{\eta}(\mathbf{w}_{k+1} - \mathbf{w}_k) = 0$$
$$\implies \mathbf{w}_{k+1} \approx \mathbf{w}_k - \eta(\mathbf{I} + \eta\nabla^2\hat{L}(\mathbf{w}_k))^{-1}\mathbf{g}_k. \quad (16)$$

Of course, this is not a perfect correspondence, but it demonstrates the role of $\hat{L}$, which is effectively to adapt each update along the $\mathbf{g}_k$ direction to the local curvature of $\hat{L}$. Furthermore, when the curvature of $\hat{L}$ is close enough to that of L, this also indicates why Algorithm 1's updates should be more effective than SGD directly on L.

3. Applications

Algorithm 1 allows us to reduce the number of stochastic gradients needed from the objective L by approximately solving a sequence of strongly convex subproblems based on $\hat{L}$. Accordingly, this approach is most advantageous when $\hat{L}$ is a faithful approximation of L in the sense that δ is small, and also when the model $\hat{L}$ is more accessible than the true objective L. This can arise in numerous situations, and in this section we will discuss several examples.

3.1. Regression and classification with costly labels

In supervised learning, we want to train using input-label pairs $(x, y) \sim \mathcal{D}$. Often, we can easily collect a large sample of i.i.d. inputs $x_1, x_2, \ldots$, but obtaining labels for these inputs can be much more costly. For example, labelling whether an MRI scan contains a tumor may require consulting with (and likely paying) a trained medical technician. So, we would like to learn using as few labels as possible, and our algorithm can provide a means of doing this.

For inputs $x \in \mathbb{R}^d$ and labels $y \in \mathbb{R}$, least squares regression, where $L(\mathbf{w}) = \frac{1}{2}\mathbb{E}_{(x,y)\sim\mathcal{D}}[(\mathbf{w}^\top x - y)^2]$, is one of the main workhorses of statistical analysis. Conveniently, the Hessian of this loss, $\nabla^2 L(\mathbf{w}) = \mathbb{E}_{x\sim\mathcal{D}_x}[xx^\top]$, does not depend on the labels y, so we can construct a proxy $\hat{L}(\mathbf{w}) = \frac{1}{2}\mathbf{w}^\top \mathbb{E}_{x\sim\mathcal{D}_x}[xx^\top]\mathbf{w}$ that satisfies Assumption 2.1 with $\delta = 0$ without using any labels! Thus, we can implement Algorithm 1 using just one labelled sample (x_k, y_k) per iteration to estimate $\mathbf{g}_k = (\mathbf{w}^\top x_k - y_k)x_k$, and then approximately minimizing φ_k only requires x samples.

Similarly, for inputs $x \in \mathbb{R}^d$ and binary labels $y \in \{0, 1\}$, the logistic regression loss function is

$$L(\mathbf{w}) = -\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\big[y \ln s(\mathbf{w}^\top x) + (1 - y)\ln(1 - s(\mathbf{w}^\top x))\big],$$

where $s(z) = 1/(1+\exp(-z))$ denotes the logistic function. As in the previous example, the Hessian of this loss is

$$\nabla^2 L(\mathbf{w}) = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\big[s(\mathbf{w}^\top x)(1 - s(\mathbf{w}^\top x))xx^\top\big],$$

which again does not depend on the labels y. Therefore, we can construct a proxy loss by simply assigning all points the label $y = 1$ (or assigning them any arbitrary labels):

$$\hat{L}(\mathbf{w}) := -\mathop{\mathbb{E}}_{x\sim\mathcal{D}_x}[\ln s(\mathbf{w}^\top x_n)].$$

This ensures $\nabla^2 \hat{L} = \nabla^2 L$ so that Assumption 2.1 holds with $\delta = 0$, despite using no labels. So again, Algorithm 1 can be implemented using as little as one label per iteration, plus sufficient x samples to approximately minimize φ_k.

3.2. Synthetic data, simulators, and meta-learning

In many machine learning problems of interest, it is difficult to collect i.i.d. samples from the target distribution. For instance, as discussed in the previous section, obtaining a label in medical imaging applications may require a consultation with a radiologist and perhaps even an additional biopsy in difficult cases. This can make it costly and time-consuming to obtain a large training set. To mitigate this issue, there has been great interest in using synthetic data as a complement to or substitute for costly samples from the target distribution. Recent advances in generative models like generative adversarial networks (GANs), variational autoencoders, diffusion models, etc. have raised the possibility of using a small seed of real data to generate an essentially unlimited quantity of synthetic training data. For instance, Frid-Adar et al. (2018) used a GAN to augment a small dataset of 182 CT scans of liver lesions, which allowed them to train a classifier with better performance. Similarly, meta-learning (Hospedales et al., 2022), multitask learning (Zhang & Yang, 2021), and transfer learning (Weiss et al., 2016) are all efforts to improve performance on a given task by leveraging data collected for other, related tasks. For

example, when learning to classify images of giraffes, it is common to use another image dataset like ImageNet for pretraining or joint training—even though ImageNet does not contain labelled images of giraffes.

In addition to possible difficulties in obtaining the training data, evaluating the loss or its gradient can be expensive. In robotics, evaluating the reward achieved by a given policy might require waiting for a physical robot to execute the policy for a period of time, and waiting seconds or minutes for each function or gradient evaluation can greatly slow training. In addition, training something like a self-driving car in the real world carries the risk of damaging the vehicle or injuring pedestrians. As a result, it is very common to use simulators like MuJoCo (Todorov et al., 2012) and the OpenAI Gym (Brockman et al., 2016), which make it possible to train models in a virtual environment, allowing for faster, risk-free processing.

Synthetic data and simulators are often used to define a surrogate loss function that is directly minimized to train the model. In meta-learning, multitask learning, or transfer learning, other tasks are usually used as a prior that biases training towards models that also perform well on all tasks simultaneously. Therefore, all of these approaches will only work to the extent that the minima of the surrogate losses correspond to minima of the target loss L itself. Indeed, recent work has cast some doubt on the utility of, e.g., ImageNet pretraining (He et al., 2019; Kornblith et al., 2019) and robots trained on simulators frequently generalize poorly to the real world (Zhao et al., 2020), presumably because these surrogates can bias the model towards worse solutions. In contrast, our approach does not rely on minima of the proxy having any particular correspondence with those of the target; instead, we use the curvature of the proxy to speed up optimization of the objective itself.

3.3. Learning with public and private data

Another natural setting where our algorithm could be useful is when the objective L is based on private data. One approach to learning from this data while maintaining rigorous privacy guarantees is to use a differentially private algorithm to generate a synthetic dataset, which could then be used freely as described in the previous section (see, e.g., Amin et al., 2019; Bowen & Snoke, 2021)

Our method might also be useful when a stock of non-private data is be available in addition to the sensitive data. For example, some patients could elect to provide their medical data to researchers without restriction, while others only do so on the condition that their privacy is rigorously protected. Similarly, different individuals may be subject to different regulatory regimes—data from E.U. residents must be used in accordance with more stringent GDPR rules that do not apply to, e.g., US residents. In these cases, one could use

just the non-private data to learn, and avoid the trouble of using the protected data, but this could be suboptimal if a large fraction of the total data is private. Instead, we could use the non-private data to define $\hat{L}$ while using a privacy-preserving mechanism to compute stochastic gradients of L from the protected samples (Dwork et al., 2014). When the private and non-private samples have similar distributions, we would expect Assumption 2.1 to hold with small δ, meaning that relatively few iterations of Algorithm 1 will be needed to achieve low error, and therefore the private data would need to be accessed fewer times. Since privacy guarantees degrade with the number of accesses, this could allow for better performance and better privacy guarantees than would be possible using the private data alone.

3.4. Handling distribution shifts

Often, we would like to learn in the face of distribution shift—a situation where the training distribution differs from the test distribution. For example, the distribution of inputs and outputs could be changing over time, and the training set is collected today and then the model is deployed tomorrow. In recent work, Lazaridou et al. (2021) argue that the performance of neural language models can degrade over time for this very reason. To keep a model up-to-date, one could periodically collect a new dataset and train a whole new model. However, collecting a large enough quantity of high-quality data could be expensive and time consuming, and it would be better to take advantage of the older data that is already available. To apply our algorithm, we could have L be the loss on the current data distribution, while out-of-date data could be used to define $\hat{L}$. As long as the distribution shift is not too dramatic, we could expect Assumption 2.1 to hold with a relatively small δ, and so to implement our method, only a small amount of new data would needed.

Distribution shifts can also arise when the training distribution is stationary but biased. For example, survey data is often affected by selection, acquiescence, or social desirability biases which can lead to differences between survey results and reality. Relatedly, when learning from a data source in which one group or class is rare, we may up-weight the loss on the rare group in order to prevent the model from learning to ignore it, which corresponds to targeting a test distribution where that group's prevalence is higher. In either case, our algorithm could be applied as described above using relatively few unbiased samples.

4. Experiments

To exhibit the effectiveness of our method in practice, we conducted several experiments on realistic problems.

Logistic regression: First, we consider a simple binary logistic regression problem with the 'mushrooms' dataset, which consists of 8124 samples of dimension 112. In this experiment, the objective L is the logistic regression loss evaluated on the training data plus an ℓ_2 regularizer with weight $\mu = 10^{-6}H$, where H is the smoothness parameter of the objective. Stochastic gradients for L are calculated by evaluating the gradient on a minibatches of size 256 or 1024 drawn uniformly with replacement. To simulate a situation where labels are hard to come by as discussed in Section 3.1, we define the proxy $\hat{L}$ by assigning random labels to the input features. The results, depicted in Figure 1, show that our method PROXYPROX has competitive performance compared with performing SGD directly on the objective. We are also competitive with the AUXMOM and AUXMVR algorithms proposed by Chayti & Karimireddy (2022), which are momentum and variance-reduced analogues of PROXYPROX. In Figure 1, the x-axis counts how many stochastic gradients from L have been used by each method, so this comparison holds constant the amount of access to L, although the computational cost of PROXYPROX, AUXMOM and AUXMVR is higher due to the approximate proximal-point updates in each iteration. We approximately solve each of the φ_k subproblems using gradient descent. For each algorithm in the experiment, we selected hyperparameters like η using grid search to minimize the loss after 250 steps and ran the best option for the full 1000 iterations. We set AUXMOM and AUXMVR's $a = 0.1$, which corresponds to the standard momentum setting of 0.9.

ResNet-18: In this experiment, summarized in Figure 2, we train a ResNet-18 network on CIFAR-10, defining $\hat{L}$ using a 2560 subset of the images, and using the full train dataset of 50000 images for L. While L corresponds to the training loss, we also plot the test loss to show that our method does not overfit. We compare running our method with either 20 and 40 iterations of SGD (equivalent to 1 epoch and 2 epochs on the $\hat{L}$ data) to approximately minimize φ_k, with the stepsize tuned by grid search and ultimately set to 0.01. Figure 2 indicates that a single pass over the subsampled data seems sufficient to minimize φ_k. We use minibatch size of 128 for both $\mathbf{g}_k$ and the SGD updates to minimize φ_k. Our method can be better than ADAMW for the first ~ 25 epochs, and it improves more convincingly over SGD. We use standard stepsizes: 0.1 for SGD, 0.001 for ADAMW, and weight decay of 0.1 for ADAMW, which gave the best test accuracy in a grid search, and all methods also used cosine annealing. We note that just training on $\hat{L}$ using SGD plateaus at test accuracy 65 after just a few passes, so, our method's improvement is indeed coming from using $\hat{L}$ jointly with L.
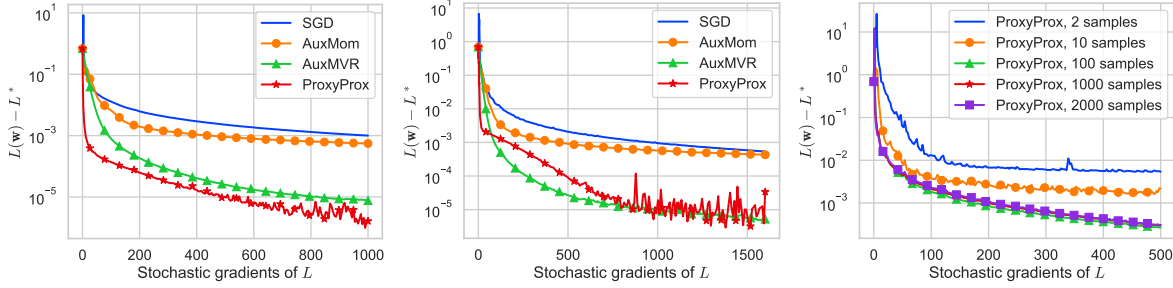
Figure 1. Binary logistic regression on 'mushrooms'. Left: batch size 1024; middle: batch size 256; right: performance of PROXYPROX with $\hat{L}$ subsampled from L, stochastic gradients of L computed with batch size 256.
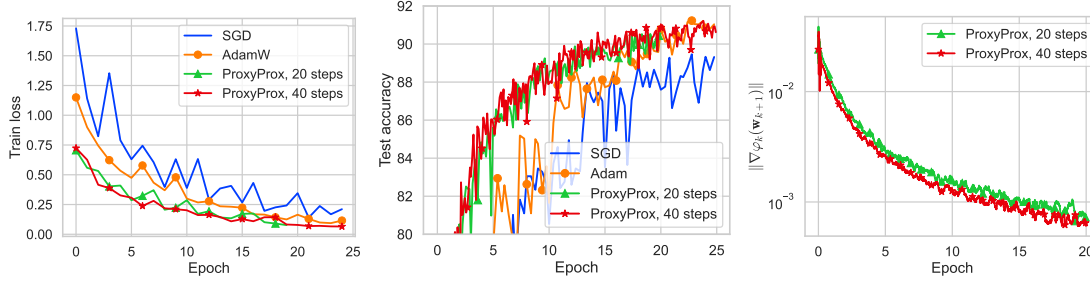

Figure 2. CIFAR-10 experiment. Left: the train loss, L; middle: the test accuracy; right: $\|\nabla\varphi_k(\mathbf{w}_{k+1})\|$ after optimizing φ_k with 20 or 40 SGD steps. The x-axis represents the number of passes over the full dataset defining L.

5. Open problems and future directions

In the convex setting, we show in Section 2.2 that under Assumptions 2.1 and 2.2, Algorithm 1 can take advantage of the proxy to achieve a convergence rate comparable to SGD on a δ-smooth function—despite the fact that L may not be smooth, and even if it is smooth, its smoothness parameter may be much larger than δ. However, the complexity guarantees in Corollary 2.5 require knowledge of problem parameters, including δ which may be hard to estimate in practice. In future work, it may therefore be useful to explore whether similar performance could be achieved using a method that is adaptive to unknown problem parameters. Also, the uniform bound on the stochastic gradient variance in Assumption 2.2 can be unrealistic, and in other contexts it can be possible to show similar results under the weaker assumption that the gradient variance is bounded only at $\mathbf{w}^*$ (Bach & Moulines, 2011; Dragomir et al., 2021).

Going beyond the convex setting is also important in machine learning, where many models used in practice, like neural networks, give rise to non-convex training objectives. Conceptually, the idea of using the proxy $\hat{L}$ to guide optimization of the objective L seems sound even when these functions are non-convex, and our CIFAR-10 experiment shows that it can work well. However, there is still work to do in proving theoretical guarantees. We can show that Algorithm 1 converges to the vicinity of a stationary point:

Theorem 5.1. *Under Assumptions 2.1 and 2.2, with L dif-*

ferentiable but potentially non-convex, let $\eta \leq \frac{1}{4\delta}$. If

$$\mathbb{E}\|\varphi_k(\mathbf{w}_{k+1})\|^2 \leq \frac{7}{16\eta^2}\mathbb{E}\|\mathbf{w}_{k+1}-\mathbf{w}_k\|^2+\frac{1}{8}\|\nabla L(\mathbf{w}_{k+1})\|^2,$$

and $\mathbb{E}\varphi_k(\mathbf{w}_{k+1}) \leq \mathbb{E}\varphi_k(\mathbf{w}_k)$ for each k, then

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\|\nabla L(\mathbf{w}_k)\|^2 \leq \frac{48(L(\mathbf{w}_0)-L^*)}{\eta K} + 8\sigma^2.$$

We prove this in Appendix D. Taking $\eta = \frac{1}{4\delta}$, this first term becomes $O(\delta(L(\mathbf{w}_0) - L^*)/K)$, which matches the rate of gradient descent on a δ-smooth objective, paralleling our results in the convex setting. However, the noise term remains $\Omega(\sigma^2)$ regardless of η or K. It is not clear whether this is merely a shortcoming in our analysis, or if this actually reflects the worst case performance of Algorithm 1. To our knowledge, a satisfying analysis of stochastic proximal-point methods in the non-convex setting is lacking more generally, so this situation may not be unique to our method.

Acknowledgements

References

Allen-Zhu, Z. How to make the gradients small stochastically: Even faster convex and nonconvex SGD. *Advances in Neural Information Processing Systems*, 31, 2018. (Cited on page 5)

Amin, K., Dick, T., Kulesza, A., Munoz, A., and Vassilvitskii, S. Differentially private covariance estimation. *Advances in Neural Information Processing Systems*, 32, 2019. (Cited on page 7)

Arjevani, Y. and Shamir, O. Communication complexity of distributed convex learning and optimization. *Advances in Neural Information Processing Systems*, 28, 2015. (Cited on page 2)

Arora, S., Babai, L., Stern, J., and Sweedyk, Z. The hardness of approximate optima in lattices, codes, and systems of linear equations. *Journal of Computer and System Sciences*, 54(2):317–331, 1997. (Cited on page 2)

Bach, F. and Moulines, E. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. *Advances in Neural Information Processing Systems*, 24, 2011. (Cited on page 9)

Barré, M., Taylor, A. B., and Bach, F. Principled analyses and design of first-order methods with inexact proximal operators. *Mathematical Programming*, pp. 1–46, 2022. (Cited on pages 3 and 5)

Bowen, C. M. and Snoke, J. Comparative study of differentially private synthetic data algorithms from the NIST PSCR differential privacy synthetic data challenge. *Journal of Privacy and Confidentiality*, 11:1, 2021. (Cited on page 7)

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. OpenAI gym. *arXiv preprint arXiv:1606.01540*, 2016. (Cited on page 7)

Bubeck, S. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8 (3-4):231–357, 2015. (Cited on page 4)

Chayti, E. M. and Karimireddy, S. P. Optimization with access to auxiliary information. *arXiv preprint arXiv:2206.00395*, 2022. (Cited on pages 2 and 8)

Chen, G. and Teboulle, M. Convergence analysis of a proximal-like minimization algorithm using bregman functions. *SIAM Journal on Optimization*, 3(3):538–543, 1993. (Cited on page 3)

Dekel, O., Gilad-Bachrach, R., Shamir, O., and Xiao, L. Optimal distributed online prediction using mini-batches. *Journal of Machine Learning Research*, 13(Jan):165–202, 2012. (Cited on page 4)

Dragomir, R. A., Even, M., and Hendrikx, H. Fast stochastic Bregman gradient methods: Sharp analysis and variance reduction. In *International Conference on Machine Learning*, pp. 2815–2825, 2021. (Cited on page 9)

Dwork, C., Roth, A., et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014. (Cited on page 8)

Foster, D. J., Sekhari, A., Shamir, O., Srebro, N., Sridharan, K., and Woodworth, B. The complexity of making the gradient small in stochastic convex optimization. In *Conference on Learning Theory*, pp. 1319–1345, 2019. (Cited on page 5)

Frid-Adar, M., Klang, E., Amitai, M., Goldberger, J., and Greenspan, H. Synthetic data augmentation using GAN for improved liver lesion classification. In *International Symposium on Biomedical Imaging (ISBI)*, pp. 289–293, 2018. (Cited on page 7)

He, K., Girshick, R., and Dollár, P. Rethinking imagenet pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4918–4927, 2019. (Cited on page 7)

Hendrikx, H., Xiao, L., Bubeck, S., Bach, F., and Massoulie, L. Statistically preconditioned accelerated gradient method for distributed optimization. In *International Conference on Machine Learning*, pp. 4203–4227, 2020. (Cited on page 2)

Hospedales, T., Antoniou, A., Micaelli, P., and Storkey, A. Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5149–5169, 2022. doi: 10.1109/TPAMI. 2021.3079209. (Cited on page 7)

Kornblith, S., Shlens, J., and Le, Q. V. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2661–2671, 2019. (Cited on page 7)

Kovalev, D., Beznosikov, A., Borodich, E. D., Gasnikov, A., and Scutari, G. Optimal gradient sliding and its application to distributed optimization under similarity. In *Advances in Neural Information Processing Systems*, 2022. (Cited on page 2)

Lazaridou, A., Kuncoro, A., Gribovskaya, E., Agrawal, D., Liska, A., Terzi, T., Gimenez, M., de Masson d'Autume, C., Kocisky, T., Ruder, S., et al. Mind the gap: Assessing temporal generalization in neural language models. *Advances in Neural Information Processing Systems*, 34: 29348–29363, 2021. (Cited on page 8)

Mairal, J. Optimization with first-order surrogate functions. In *International Conference on Machine Learning*, pp. 783–791. PMLR, 2013a. (Cited on pages 2 and 3)

Mairal, J. Stochastic majorization-minimization algorithms for large-scale optimization. *Advances in Neural Information Processing Systems*, 26, 2013b. (Cited on page 3)

Mairal, J. Incremental majorization-minimization optimization with application to large-scale machine learning. *SIAM Journal on Optimization*, 25(2):829–855, 2015. (Cited on page 2)

Nemirovskii, A. S. and Yudin, D. B. *Problem Complexity and Method Efficiency in Optimization*. Wiley-Interscience, 1983. (Cited on pages 4, 5, 6, and 19)

Nesterov, Y. and Vial, J.-P. Confidence level solutions for stochastic programming. *Automatica*, 44(6):1559–1568, 2008. (Cited on page 13)

Parpas, P. A multilevel proximal gradient algorithm for a class of composite optimization problems. *SIAM Journal on Scientific Computing*, 39(5):S681–S701, 2017. (Cited on page 2)

Rockafellar, R. T. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Mathematics of Operations Research*, 1(2):97–116, 1976. (Cited on page 5)

Shamir, O., Srebro, N., and Zhang, T. Communication-efficient distributed optimization using an approximate newton-type method. In *International Conference on Machine Learning*, pp. 1000–1008, 2014. (Cited on page 2)

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012. (Cited on page 7)

Vapnik, V. N. and Chervonenkis, A. Y. The uniform convergence of frequencies of the appearance of events to their probabilities. In *Doklady Akademii Nauk*, volume 181, pp. 781–783. Russian Academy of Sciences, 1968. (Cited on page 2)

Weiss, K., Khoshgoftaar, T. M., and Wang, D. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016. (Cited on page 7)

Zhang, Y. and Yang, Q. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5586–5609, 2021. (Cited on page 7)

Zhao, W., Queralta, J. P., and Westerlund, T. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE symposium series on computational intelligence (SSCI)*, pp. 737–744. IEEE, 2020. (Cited on page 7)

A. Proof of Theorem 2.3

Our analysis requires that $h = L - \hat{L}$ be differentiable under Assumption 2.1, but when L is convex, it is not necessary for L itself to be differentiable. In particular, Assumption 2.2 requires that the expectation of each stochastic gradient $\mathbf{g}_k$ must be a subgradient of L at $\mathbf{w}_k$. Accordingly, in the following proofs, we will use $\nabla L(\mathbf{w}_k) := \mathbb{E}[\mathbf{g}_k \,|\, \mathbf{w}_k]$ as notation to denote the particular subgradient corresponding to the expectation of $\mathbf{g}_k$. Since $h = L - \hat{L}$ is differentiable, we can use this, in turn, to define $\nabla \hat{L}(\mathbf{w}_k) := \nabla L(\mathbf{w}_k) - \nabla h(\mathbf{w}_k)$. Finally, since $0 \in \partial L(\mathbf{w}^*)$, we denote $\nabla L(\mathbf{w}^*) := 0$ even when L is not differentiable at its minimizer.

Lemma A.1. *Let* $\mathbf{w}^* \in \arg\min_{\mathbf{w}} L(\mathbf{w})$ *and* $\eta \leq \frac{1}{4\delta}$. *Then under Assumptions 2.1 and 2.2, for any* $\mathbf{w}_{k+1}, \mathbf{w}_k$

$$
\begin{aligned}
\mathbb{E}[L(\mathbf{w}_{k+1}) - L^*] \leq \ &\frac{1}{2\eta}\mathbb{E}\|\mathbf{w}_k - \mathbf{w}^*\|^2 - \mathbb{E}D_h(\mathbf{w}^*; \mathbf{w}_k) \\
&- \frac{1}{2\eta}\mathbb{E}\|\mathbf{w}_{k+1} - \mathbf{w}^*\|^2 + \mathbb{E}D_h(\mathbf{w}^*; \mathbf{w}_{k+1}) \\
&+ \mathbb{E}\left\langle \nabla\varphi_k(\mathbf{w}_{k+1}), \ \mathbf{w}_{k+1} - \mathbf{w}^* \right\rangle + 2\eta\sigma^2 \\
&- \frac{1}{4\eta}\mathbb{E}\|\mathbf{w}_{k+1} - \mathbf{w}_k\|^2 - \mathbb{E}D_L(\mathbf{w}^*; \mathbf{w}_{k+1}),
\end{aligned}
$$

where the expectation is taken over the randomness in $\mathbf{g}_k$.

Proof. Computing $\nabla\varphi_k$ and rearranging yields

$$
\nabla\varphi_k(\mathbf{w}_{k+1}) = \nabla\hat{L}(\mathbf{w}_{k+1}) + \mathbf{g}_k - \nabla\hat{L}(\mathbf{w}_k) + \frac{1}{\eta}(\mathbf{w}_{k+1} - \mathbf{w}_k)
$$

$$
\implies \nabla L(\mathbf{w}_{k+1}) = \nabla h(\mathbf{w}_{k+1}) - \nabla h(\mathbf{w}_k) + \nabla\varphi_k(\mathbf{w}_{k+1}) + \nabla L(\mathbf{w}_k) - \mathbf{g}_k - \frac{1}{\eta}(\mathbf{w}_{k+1} - \mathbf{w}_k). \tag{17}
$$

Also, (6) immediately implies

$$
L(\mathbf{w}_{k+1}) - L^* = \langle \nabla L(\mathbf{w}_{k+1}), \ \mathbf{w}_{k+1} - \mathbf{w}^* \rangle - D_L(\mathbf{w}^*; \mathbf{w}_{k+1}). \tag{18}
$$

After substituting (17) into (18), we derive

$$
\begin{aligned}
&L(\mathbf{w}_{k+1}) - L^* \\
&= \langle \nabla L(\mathbf{w}_{k+1}), \ \mathbf{w}_{k+1} - \mathbf{w}^* \rangle - D_L(\mathbf{w}^*; \mathbf{w}_{k+1}) \\
&= \left\langle \nabla h(\mathbf{w}_{k+1}) - \nabla h(\mathbf{w}_k) + \nabla\varphi_k(\mathbf{w}_{k+1}) + \nabla L(\mathbf{w}_k) - \mathbf{g}_k + \frac{1}{\eta}(\mathbf{w}_k - \mathbf{w}_{k+1}), \ \mathbf{w}_{k+1} - \mathbf{w}^* \right\rangle - D_L(\mathbf{w}^*; \mathbf{w}_{k+1}) \\
&= D_h(\mathbf{w}^*; \mathbf{w}_{k+1}) - D_h(\mathbf{w}^*; \mathbf{w}_k) + D_h(\mathbf{w}_{k+1}; \mathbf{w}_k) + \langle \nabla\varphi_k(\mathbf{w}_{k+1}) + \nabla L(\mathbf{w}_k) - \mathbf{g}_k, \ \mathbf{w}_{k+1} - \mathbf{w}^* \rangle \\
&\quad + \frac{1}{2\eta}\|\mathbf{w}_k - \mathbf{w}^*\|^2 - \frac{1}{2\eta}\|\mathbf{w}_{k+1} - \mathbf{w}^*\|^2 - \frac{1}{2\eta}\|\mathbf{w}_{k+1} - \mathbf{w}_k\|^2 - D_L(\mathbf{w}^*; \mathbf{w}_{k+1}), \tag{19}
\end{aligned}
$$

where the third equality uses the three-point property with D_h, (7). To proceed, Assumption 2.2, Young's inequality, and $\eta \leq \frac{1}{4\delta}$, together imply

$$
\begin{aligned}
\mathbb{E}\langle \nabla L(\mathbf{w}_k) - \mathbf{g}_k, \ \mathbf{w}_{k+1} - \mathbf{w}^* \rangle &= \mathbb{E}\langle \nabla L(\mathbf{w}_k) - \mathbf{g}_k, \ \mathbf{w}_{k+1} - \mathbf{w}_k \rangle \\
&\leq \frac{\eta\mathbb{E}\|\nabla L(\mathbf{w}_k) - \mathbf{g}_k\|^2}{1 - 2\eta\delta} + \frac{1 - 2\eta\delta}{4\eta}\|\mathbf{w}_{k+1} - \mathbf{w}_k\|^2 \\
&\leq 2\eta\sigma^2 + \frac{1 - 2\eta\delta}{4\eta}\|\mathbf{w}_{k+1} - \mathbf{w}_k\|^2,
\end{aligned}
$$

where the expectation is taken over the randomness in $\mathbf{g}_k$. Since h is δ-smooth, in light of (5), $|D_h(\mathbf{w}; \mathbf{w}_k)| \leq \frac{\delta}{2}\|\mathbf{w} - \mathbf{w}_k\|^2$, so

$$
\mathbb{E}[D_h(\mathbf{w}_{k+1}; \mathbf{w}_k) + \langle \nabla L(\mathbf{w}_k) - \mathbf{g}_k, \ \mathbf{w}_{k+1} - \mathbf{w}^* \rangle] \leq 2\eta\sigma^2 + \frac{1}{4\eta}\mathbb{E}\|\mathbf{w}_{k+1} - \mathbf{w}_k\|^2.
$$

Taking the expectation of (19) and plugging this in completes the proof. □

Theorem 2.3. *Under Assumptions 2.1 and 2.2, let L be μ-strongly convex and let $\eta \leq \frac{1}{4\delta}$. If for each k*

$$\mathbb{E}\|\nabla\varphi_k(\mathbf{w}_{k+1})\|^2 \leq \frac{\mu}{4\eta}\mathbb{E}\|\mathbf{w}_{k+1} - \mathbf{w}_k\|^2 + G^2,$$

for some G, then a geometrically weighted average of the iterates of Algorithm 1, $\bar{\mathbf{w}}_K$ (see (25)), satisfies

$$\mathbb{E}L(\bar{\mathbf{w}}_K) - L^* \leq \frac{5B^2}{8\eta}\left(1 + \frac{2\eta\mu}{5}\right)^{1-K} + 2\eta\sigma^2 + \frac{G^2}{\mu},$$

where the expectation is taken over the randomness in both the stochastic gradients and the selection of the iterates.

Proof. For brevity, denote $\Delta_k := \frac{1}{2\eta}\|\mathbf{w}_k - \mathbf{w}^*\|^2 - D_h(\mathbf{w}^*; \mathbf{w}_k)$. Because L is strongly convex, it follows immediately from (4) and (6) that $D_L(\mathbf{w}^*; \mathbf{w}_{k+1}) \geq \frac{\mu}{2}\|\mathbf{w}_{k+1} - \mathbf{w}^*\|^2$. Also, in light of (5), it follows under Assumption 2.1 that

$$|D_h(\mathbf{w}^*; \mathbf{w}_{k+1})| \leq \frac{\delta}{2}\|\mathbf{w}_{k+1} - \mathbf{w}^*\|^2 \leq \frac{1}{8\eta}\|\mathbf{w}_{k+1} - \mathbf{w}^*\|^2.$$

This implies that

$$0 \leq \Delta_{k+1} \leq \frac{5}{8\eta}\|\mathbf{w}_{k+1} - \mathbf{w}^*\|^2 \leq \frac{5}{4\eta\mu}D_L(\mathbf{w}^*; \mathbf{w}_{k+1}).$$

Therefore, by Lemma A.1 and Young's inequality

$$\mathbb{E}[L(\mathbf{w}_{k+1}) - L^*]$$
$$\leq \mathbb{E}\Delta_k - \mathbb{E}\Delta_{k+1} + 2\eta\sigma^2 + \mathbb{E}\langle\nabla\varphi_k(\mathbf{w}_{k+1}), \mathbf{w}_{k+1} - \mathbf{w}^*\rangle - \frac{1}{4\eta}\mathbb{E}\|\mathbf{w}_{k+1} - \mathbf{w}_k\|^2 - \mathbb{E}D_L(\mathbf{w}^*; \mathbf{w}_{k+1})$$
$$\leq \mathbb{E}\Delta_k - \left(1 + \frac{2\eta\mu}{5}\right)\mathbb{E}\Delta_{k+1} + 2\eta\sigma^2 + \mathbb{E}\langle\nabla\varphi_k(\mathbf{w}_{k+1}), \mathbf{w}_{k+1} - \mathbf{w}^*\rangle - \frac{1}{4\eta}\mathbb{E}\|\mathbf{w}_{k+1} - \mathbf{w}_k\|^2 - \frac{\mu}{4}\mathbb{E}\|\mathbf{w}_{k+1} - \mathbf{w}^*\|^2.$$
$$(20)$$

So by Young's inequality and the assumed upper bound

$$\|\nabla\varphi_k(\mathbf{w}_{k+1})\|^2 \leq \frac{\mu}{4\eta}\mathbb{E}\|\mathbf{w}_{k+1} - \mathbf{w}_k\|^2 + G^2 \tag{21}$$

we conclude

$$\mathbb{E}\langle\nabla\varphi_k(\mathbf{w}_{k+1}), \mathbf{w}_{k+1} - \mathbf{w}^*\rangle \leq \frac{1}{\mu}\mathbb{E}\|\nabla\varphi_k(\mathbf{w}_{k+1})\| + \frac{\mu}{4}\mathbb{E}\|\mathbf{w}_{k+1} - \mathbf{w}^*\|^2 \tag{22}$$

$$\leq \frac{1}{4\eta}\mathbb{E}\|\mathbf{w}_{k+1} - \mathbf{w}_k\|^2 + \frac{\mu}{4}\mathbb{E}\|\mathbf{w}_{k+1} - \mathbf{w}^*\|^2 + \frac{G^2}{\mu}. \tag{23}$$

Plugging this into (20) yields

$$\mathbb{E}[L(\mathbf{w}_{k+1}) - L^*] \leq \mathbb{E}\Delta_k - \left(1 + \frac{2\eta\mu}{5}\right)\mathbb{E}\Delta_{k+1} + 2\eta\sigma^2 + \frac{G^2}{\mu}. \tag{24}$$

As is typical in strongly convex stochastic optimization, we now introduce weights $\alpha_k = (1 + \frac{2\eta\mu}{5})^{k-1}$ (see Section 5 in Nesterov & Vial, 2008, for a similar weighting) with sum $A_K := \sum_{k=1}^K \alpha_k$ and we will attempt to upper bound the suboptimality of the weighted iterate

$$\bar{\mathbf{w}}_K := \frac{1}{A_K}\sum_{k=1}^K \alpha_k\mathbf{w}_k. \tag{25}$$

13

By the convexity of L and (24),

$$\mathbb{E}L(\bar{\mathbf{w}}_K) - L^* \le \frac{1}{A_K}\sum_{k=1}^K \alpha_k \mathbb{E}[L(\mathbf{w}_k) - L^*]$$

$$\le \frac{\mathbb{E}\Delta_0}{A_K} + 2\eta\sigma^2 + \frac{G^2}{\mu}$$

$$\le \frac{5\mathbb{E}\|\mathbf{w}_0 - \mathbf{w}^*\|^2}{8\eta A_K} + 2\eta\sigma^2 + \frac{G^2}{\mu}.$$

Finally, we lower bound

$$A_K = \frac{5\left(\left(1+\frac{2\eta\mu}{5}\right)^K - 1\right)}{2\eta\mu} \ge \left(1 + \frac{2\eta\mu}{5}\right)^{K-1},$$

which completes the proof. $\qquad\square$

B. Proof of Theorem 2.4

Theorem 2.4. *Under Assumptions 2.1 and 2.2, let L be convex and let $\eta \le \frac{1}{4\delta}$. Then applying Algorithm 1 to the regularized objectives $L^{(\mu)}(\mathbf{w}) := L(\mathbf{w}) + \frac{\mu}{2}\|\mathbf{w} - \mathbf{w}_0\|^2$ and $\hat{L}^{(\mu)}(\mathbf{w}) := \hat{L}(\mathbf{w}) + \frac{\mu}{2}\|\mathbf{w} - \mathbf{w}_0\|^2$ with $\mu = \frac{1}{\eta K}$ and any $\eta \le \frac{1}{4\delta}$ ensures that if for each k*

$$\mathbb{E}\|\nabla\varphi_k(\mathbf{w}_{k+1})\|^2 \le \frac{1}{4\eta^2 K}\mathbb{E}\|\mathbf{w}_{k+1} - \mathbf{w}_k\|^2 + G^2$$

for some G, then

$$\mathbb{E}L\left(\frac{1}{K}\sum_{k=1}^K \mathbf{w}_k\right) - L^* \le \frac{9B^2}{8\eta K} + 2\eta\sigma^2 + \eta K G^2.$$

Proof. By construction, $L^{(\mu)}$ is μ-strongly convex, and yet $L^{(\mu)} - \hat{L}^{(\mu)} = h$, so this difference remains δ-smooth. Therefore, our analysis from Theorem 2.3 can be applied to $L^{(\mu)}$. Denote $\mathbf{w}_\mu^* := \arg\min_{\mathbf{w}} L^{(\mu)}(\mathbf{w})$. In the course of proving Theorem 2.3, we showed in (24) that

$$\mathbb{E}[L^{(\mu)}(\mathbf{w}_{k+1}) - L^{(\mu)*}] \le \mathbb{E}\Delta_k - \left(1 + \frac{2\eta\mu}{5}\right)\mathbb{E}\Delta_{k+1} + 2\eta\sigma^2 + \frac{G^2}{\mu}$$

$$\le \mathbb{E}\Delta_k - \mathbb{E}\Delta_{k+1} + 2\eta\sigma^2 + \frac{G^2}{\mu},$$

where $\Delta_k := \frac{1}{2\eta}\|\mathbf{w}_k - \mathbf{w}_\mu^*\|^2 - D_h(\mathbf{w}_\mu^*; \mathbf{w}_k)$ and $0 \le \Delta_k \le \frac{5}{8\eta}\|\mathbf{w}_k - \mathbf{w}_\mu^*\|^2$ for all k. Furthermore, $L^{(\mu)}(\mathbf{w}_{k+1}) \ge L(\mathbf{w}_{k+1})$ and

$$L^{(\mu)*} \le L^{(\mu)}(\mathbf{w}^*) = L^* + \frac{\mu}{2}\|\mathbf{w}_0 - \mathbf{w}^*\|^2.$$

Therefore, by the convexity of L

$$\mathbb{E}L\left(\frac{1}{K}\sum_{k=1}^K \mathbf{w}_k\right) - L^* \le \frac{1}{K}\sum_{k=1}^K \mathbb{E}[L(\mathbf{w}_k) - L^*]$$

$$\le \frac{1}{K}\sum_{k=1}^K \mathbb{E}\left[L^{(\mu)}(\mathbf{w}_k) - L^{(\mu)*} + \frac{\mu}{2}\|\mathbf{w}_0 - \mathbf{w}^*\|^2\right]$$

$$\le \frac{\mathbb{E}\Delta_0}{K} + 2\eta\sigma^2 + \frac{G^2}{\mu} + \frac{\mu}{2}\mathbb{E}\|\mathbf{w}_0 - \mathbf{w}^*\|^2. \qquad (26)$$

Finally,

$$0 = \nabla L^{(\mu)}(\mathbf{w}_\mu^*) = \nabla L(\mathbf{w}_\mu^*) + \mu(\mathbf{w}_\mu^* - \mathbf{w}_0),$$

so,

$$\left\|\mathbf{w}_0 - \mathbf{w}_\mu^*\right\|^2 - \left\|\mathbf{w}_0 - \mathbf{w}^*\right\|^2 = -\left\|\mathbf{w}^* - \mathbf{w}_\mu^*\right\|^2 + 2\left\langle \mathbf{w}_0 - \mathbf{w}_\mu^*, \mathbf{w}^* - \mathbf{w}_\mu^* \right\rangle$$

$$= -\left\|\mathbf{w}^* - \mathbf{w}_\mu^*\right\|^2 - \frac{2}{\mu}\left\langle \nabla L(\mathbf{w}_\mu^*), \mathbf{w}_\mu^* - \mathbf{w}^* \right\rangle$$

$$\leq 0.$$

where for the last line we used the convexity of L. So,

$$\Delta_0 \leq \frac{5}{8\eta}\|\mathbf{w}_0 - \mathbf{w}_\mu^*\|^2 \leq \frac{5}{8\eta}\|\mathbf{w}_0 - \mathbf{w}^*\|^2.$$

When combined with (26), this means

$$\mathbb{E}L\left(\frac{1}{K}\sum_{k=1}^K \mathbf{w}_k\right) - L^* \leq \frac{5\mathbb{E}\|\mathbf{w}_0 - \mathbf{w}^*\|^2}{8\eta K} + 2\eta\sigma^2 + \frac{G^2}{\mu} + \frac{\mu}{2}\mathbb{E}\|\mathbf{w}_0 - \mathbf{w}^*\|^2,$$

and plugging in our choice of $\mu = \frac{1}{\eta K}$ completes the proof. $\qquad\square$

C. Proof of Corollary 2.5

Corollary 2.5. *In the setting of Theorem 2.3, there is a universal constant c s.t. for any $\varepsilon > 0$, when $G^2 \leq \frac{1}{2}\mu\varepsilon$ and*

$$\eta = \frac{5}{\mu(K-1)}\left(1 + \ln\left(\frac{5B^2\mu(K-1)}{\varepsilon}\right)\right),$$

then the output, $\hat{\mathbf{w}}$, of Algorithm 1 will have suboptimality at most $\mathbb{E}L(\hat{\mathbf{w}}) - L^* \leq \varepsilon$ whenever

$$K \geq c \cdot \left(1 + \frac{\delta}{\mu} + \frac{\sigma^2}{\mu\varepsilon}\right)\ln\left(e + \frac{(\mu+\delta)B^2}{\varepsilon} + \frac{\sigma^2 B^2}{\varepsilon^2}\right).$$

Likewise, in the setting of Theorem 2.4, there is a universal constant c s.t. for any $\varepsilon > 0$, when $G^2 \leq \frac{\sigma^2}{K} + \frac{\delta^2 B^2}{K^2}$ and

$$\eta = \min\left\{\frac{1}{4\delta}, \frac{B}{\sigma\sqrt{K}}\right\},$$

then the output, $\hat{\mathbf{w}}$, of Algorithm 1 will have suboptimality at most $\mathbb{E}L(\hat{\mathbf{w}}) - L^* \leq \varepsilon$ whenever

$$K \geq c \cdot \left(\frac{\delta B^2}{\varepsilon} + \frac{\sigma^2 B^2}{\varepsilon^2}\right).$$

Proof. The complexity guarantee in the convex case follows immediately after plugging the chosen η into the guarantee of Theorem 2.4 and then choosing K large enough that the expected suboptimality is less than ε.

Moving on to the strongly convex setting, to apply Theorem 2.3, we require $\eta \leq \frac{1}{4\delta}$. Suppose for now that η also satisfies

$$\eta \geq \frac{1}{2\mu(K-1)}.$$

Then the first term in the guarantee from Theorem 2.3 is at most

$$\frac{5B^2}{8\eta}\left(1 + \frac{2\eta\mu}{5}\right)^{1-K} \leq \frac{5B^2\mu(K-1)}{4}\left(1 + \frac{2\eta\mu}{5}\right)^{1-K}.$$

Next, we note that for $g(z) = \ln(1+z)$, and any $z \geq 0$

$$g'(z) = \frac{1}{1+z} = 1 - \frac{z}{1+z} \geq 1 - z.$$

15

Therefore, for $z \geq 0$

$$\ln(1+z) = \int_0^z g'(t)dt \geq \int_0^z (1-t)dt = z - \frac{1}{2}z^2.$$

Thus,

$$\left(1 + \frac{2\eta\mu}{5}\right)^{1-K} = \exp\left(-(K-1)\ln\left(1 + \frac{2\eta\mu}{5}\right)\right) \leq \exp\left(-\frac{2(K-1)\eta\mu}{5}\left(1 - \frac{\eta\mu}{5}\right)\right)$$

therefore, as long as we choose $\eta \leq \frac{5}{2\mu}$, then it follows that

$$\frac{5B^2}{8\eta}\left(1 + \frac{2\eta\mu}{5}\right)^{1-K} \leq \frac{5B^2\mu(K-1)}{4}\exp\left(-\frac{(K-1)\eta\mu}{5}\right).$$

Therefore, if all of the following constraints on η are satisfied:

$$\eta \leq \frac{1}{4\delta} \tag{27}$$

$$\eta \leq \frac{5}{2\mu} \tag{28}$$

$$\eta \geq \frac{1}{2\mu(K-1)} \tag{29}$$

$$\eta \geq \frac{5}{\mu(K-1)}\ln\left(\frac{5B^2\mu(K-1)}{4\varepsilon}\right) \tag{30}$$

then this first term in the guarantee from Theorem 2.3 is upper bounded by ε. We therefore set

$$\eta = \frac{5}{\mu(K-1)}\left(1 + \ln\left(\frac{5B^2\mu(K-1)}{4\varepsilon}\right)\right). \tag{31}$$

This value always satisfies (29) and (30), and we will proceed to choose K large enough that it also satisfies (27) and (28). In particular, we require K such that

$$K \geq 1 + 2\max\left\{1, \frac{10\delta}{\mu}\right\}\left(1 + \ln\left(\frac{5B^2\mu(K-1)}{4\varepsilon}\right)\right). \tag{32}$$

Plugging the stepsize (31) into the second term of the guarantee from Theorem 2.3 yields

$$2\eta\sigma^2 = \frac{10\sigma^2}{\mu(K-1)}\left(1 + \ln\left(\frac{5B^2\mu(K-1)}{4\varepsilon}\right)\right).$$

So, to make this smaller than ε, we require

$$K \geq 1 + \frac{10\sigma^2}{\mu\varepsilon}\left(1 + \ln\left(\frac{5B^2\mu(K-1)}{4\varepsilon}\right)\right).$$

Combining this with (32), we conclude that with η set according to (31), if

$$K \geq 10\max\left\{1, \frac{\delta}{\mu}, \frac{\sigma^2}{\mu\varepsilon}\right\}\left(1 + \ln\left(\frac{5B^2\mu K}{4\varepsilon}\right)\right) \tag{33}$$

and, in addition $G \leq \sqrt{\mu\varepsilon}$, then the expected suboptimality is at most 3ε, so using $\varepsilon' = \varepsilon/3$ instead above ensures accuracy ε.

To conclude the proof, we note for $A, B > 0$, the inequality

$$X \geq A(1 + \ln(BX))$$

is satisfied by setting

$$X \geq A(1 + 4\ln(e + AB)).$$

To show this, we first note that because $z \mapsto \sqrt{z}$ is a concave function on $z > 0$, we have the inequality

$$\sqrt{z} \leq \sqrt{1} + (z-1)\left(\frac{d}{dz}\sqrt{z}\Big|_{z=1}\right) = 1 + \frac{z-1}{2} = \frac{1+z}{2}.$$

This implies that for $z > 0$

$$\frac{d}{dz}\sqrt{z} = \frac{1}{2\sqrt{z}} \geq \frac{1}{1+z} = \frac{d}{dz}\ln(1+z), \tag{34}$$

Finally, since $\sqrt{0} = \ln(1+0)$, this implies that $\ln(1+z) \leq \sqrt{z}$ for all $z \geq 0$.

So, suppose that

$$X = \alpha A(1 + 4\ln(e + AB)). \tag{35}$$

for some $\alpha \geq 1$. Then,

$$
\begin{aligned}
A(1 + \ln(BX)) &= A(1 + \ln(\alpha) + \ln(AB) + \ln(1 + 4\ln(e + AB))) \\
&\leq A(1 + \ln(\alpha) + \ln(e + AB) + \sqrt{4\ln(e + AB)}) \\
&\leq A(1 + \ln(\alpha) + 3\ln(e + AB)) \\
&\leq \frac{1}{\alpha}X + \ln(c)A \\
&\leq \frac{1 + \ln(\alpha)}{\alpha}X \\
&\leq X,
\end{aligned}
$$

where we used that $1 + \ln(\alpha) \leq \alpha$ for all $\alpha \geq 1$.

Therefore, for a sufficiently large constant c, setting

$$K = c \cdot \left(1 + \frac{\delta}{\mu} + \frac{\sigma^2}{\mu\varepsilon}\right)\ln\left(e + \left(1 + \frac{\delta}{\mu} + \frac{\sigma^2}{\mu\varepsilon}\right)\frac{\mu\mathbb{E}\|\mathbf{w}_0 - \mathbf{w}^*\|^2}{\varepsilon}\right)$$

satisfies (33), completing the proof. □

D. Proof of Theorem 5.1

Lemma D.1. *Under Assumptions 2.1 and 2.2, let $\eta \leq \frac{1}{4\delta}$. Then for any $\mathbf{w}$ such that $\mathbb{E}\varphi_k(\mathbf{w}) \leq \mathbb{E}\varphi_k(\mathbf{w}_k)$,*

$$\mathbb{E}[L(\mathbf{w}) - L(\mathbf{w}_k)] \leq -\frac{1}{4\eta}\mathbb{E}\|\mathbf{w} - \mathbf{w}_k\|^2 + 2\eta\sigma^2.$$

Proof. Under Assumption 2.1, $h = L - \hat{L}$ is δ-smooth. So,

$$L(\mathbf{w}) - \hat{L}(\mathbf{w}) \leq L(\mathbf{w}_k) - \hat{L}(\mathbf{w}_k) + \frac{\delta}{2}\|\mathbf{w} - \mathbf{w}_k\|^2 + \left\langle\nabla L(\mathbf{w}_k) - \nabla\hat{L}(\mathbf{w}_k), \mathbf{w} - \mathbf{w}_k\right\rangle.$$

Rearranging and substituting φ_k, this implies

$$L(\mathbf{w}) - L(\mathbf{w}_k) \leq \varphi_k(\mathbf{w}) - \varphi_k(\mathbf{w}_k) - \frac{1 - \eta\delta}{2\eta}\|\mathbf{w} - \mathbf{w}_k\|^2 + \langle\nabla L(\mathbf{w}_k) - \mathbf{g}_k, \mathbf{w} - \mathbf{w}_k\rangle.$$

Next we use Young's inequality with Assumption 2.2 to upper bound

$$\mathbb{E}\langle\nabla L(\mathbf{w}_k) - \mathbf{g}_k, \mathbf{w} - \mathbf{w}_k\rangle \leq \frac{\eta\mathbb{E}\|\nabla L(\mathbf{w}_k) - \mathbf{g}_k\|^2}{1 - 2\eta\delta} + \frac{1 - 2\eta\delta}{4\eta}\mathbb{E}\|\mathbf{w} - \mathbf{w}_k\|^2 \leq \frac{\eta\sigma^2}{1 - 2\eta\delta} + \frac{1 - 2\eta\delta}{4\eta}\mathbb{E}\|\mathbf{w} - \mathbf{w}_k\|^2.$$

The fact that $\eta \leq \frac{1}{4\delta}$ and $\mathbb{E}\varphi_k(\mathbf{w}) \leq \mathbb{E}\varphi_k(\mathbf{w}_k)$ completes the proof. □

Lemma D.2. *Under Assumptions 2.1 and 2.2, let L be differentiable and let $\eta \leq \frac{1}{4\delta}$. If $\mathbf{w}_{k+1}$ satisfies*

$$\mathbb{E}\|\varphi_k(\mathbf{w}_{k+1})\|^2 \leq \frac{7}{16\eta^2}\mathbb{E}\|\mathbf{w}_{k+1} - \mathbf{w}_k\|^2 + \frac{1}{8}\|\nabla L(\mathbf{w}_{k+1})\|^2$$

then

$$-\frac{1}{4\eta}\|\mathbf{w}_{k+1} - \mathbf{w}_k\|^2 \leq -\frac{\eta}{48}\mathbb{E}\|\nabla L(\mathbf{w}_{k+1})\|^2 + \frac{\eta\sigma^2}{6}.$$

Proof. By the relaxed triangle inequality, for any vectors a, b, c, d,

$$\|a\|^2 = \|a + b + c + d - b - c - d\|^2 \leq 4(\|a + b + c + d\|^2 + \|b\|^2 + \|c\|^2 + \|d\|^2). \tag{36}$$

Furthermore,

$$\nabla\varphi_k(\mathbf{w}_{k+1}) = \nabla\hat{L}(\mathbf{w}_{k+1}) - \nabla L(\mathbf{w}_{k+1}) - \nabla\hat{L}(\mathbf{w}_k) + \nabla L(\mathbf{w}_k) + \mathbf{g}_k - \nabla L(\mathbf{w}_k) + \nabla L(\mathbf{w}_{k+1}) + \frac{1}{\eta}(\mathbf{w}_{k+1} - \mathbf{w}_k). \tag{37}$$

So (36) with

$$a = \nabla L(\mathbf{w}_{k+1})$$
$$b = -\nabla\varphi_k(\mathbf{w}_{k+1})$$
$$c = \nabla\hat{L}(\mathbf{w}_{k+1}) - \nabla L(\mathbf{w}_{k+1}) - \nabla\hat{L}(\mathbf{w}_k) + \nabla L(\mathbf{w}_k)$$
$$d = \mathbf{g}_k - \nabla L(\mathbf{w}_k)$$

so $a + b + c + d = \frac{1}{\eta}(\mathbf{w}_k - \mathbf{w}_{k+1})$, implies that under Assumptions 2.1 and 2.2

$$\frac{1}{\eta^2}\|\mathbf{w} - \mathbf{w}_k\|^2$$

$$\geq \frac{1}{4}\mathbb{E}\|\nabla L(\mathbf{w})\|^2 - \mathbb{E}\|\nabla\varphi_k(\mathbf{w})\|^2 - \mathbb{E}\|\mathbf{g}_k - \nabla L(\mathbf{w}_k)\|^2 - \mathbb{E}\|\nabla\hat{L}(\mathbf{w}) - \nabla L(\mathbf{w}) - \nabla\hat{L}(\mathbf{w}_k) + \nabla L(\mathbf{w}_k)\|^2$$

$$\geq \frac{1}{4}\mathbb{E}\|\nabla L(\mathbf{w})\|^2 - \mathbb{E}\|\nabla\varphi_k(\mathbf{w})\|^2 - \delta^2\mathbb{E}\|\mathbf{w} - \mathbf{w}_k\|^2 - \sigma^2$$

$$\geq \frac{1}{8}\mathbb{E}\|\nabla L(\mathbf{w})\|^2 - \frac{1}{2\eta^2}\mathbb{E}\|\mathbf{w} - \mathbf{w}_k\|^2 - \sigma^2,$$

where we used the upper bound on $\mathbb{E}\|\nabla\varphi_k(\mathbf{w})\|^2$ and the fact that $\eta \leq \frac{1}{4\delta}$. Rearranging completes the proof. $\qquad\square$

Theorem 5.1. *Under Assumptions 2.1 and 2.2, with L differentiable but potentially non-convex, let $\eta \leq \frac{1}{4\delta}$. If*

$$\mathbb{E}\|\varphi_k(\mathbf{w}_{k+1})\|^2 \leq \frac{7}{16\eta^2}\mathbb{E}\|\mathbf{w}_{k+1} - \mathbf{w}_k\|^2 + \frac{1}{8}\|\nabla L(\mathbf{w}_{k+1})\|^2,$$

and $\mathbb{E}\varphi_k(\mathbf{w}_{k+1}) \leq \mathbb{E}\varphi_k(\mathbf{w}_k)$ for each k, then

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\|\nabla L(\mathbf{w}_k)\|^2 \leq \frac{48(L(\mathbf{w}_0) - L^*)}{\eta K} + 8\sigma^2.$$

Proof. By Lemmas D.1 and D.2

$$\mathbb{E}[L(\mathbf{w}_{k+1}) - L(\mathbf{w}_k)] \leq -\frac{1}{4\eta}\mathbb{E}\|\mathbf{w}_{k+1} - \mathbf{w}_k\|^2 + 2\eta\sigma^2 \leq -\frac{\eta}{48}\mathbb{E}\|\nabla L(\mathbf{w}_{k+1})\|^2 + \frac{\eta\sigma^2}{6}. \tag{38}$$

Therefore, rearranging and averaging over k yields

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\|\nabla L(\mathbf{w}_k)\|^2 \leq \frac{48(L(\mathbf{w}_0) - L(\mathbf{w}_K))}{\eta K} + 8\sigma^2 \leq \frac{48(L(\mathbf{w}_0) - L^*)}{\eta K} + 8\sigma^2 \tag{39}$$

which completes the proof. $\qquad\square$

E. Proof of Proposition 2.6

Proposition 2.6. *Let $\hat{L}$ be convex and H-smooth and let stochastic gradients be available for $\hat{L}$ with variance at most ρ^2. Then for constants c, c', the output, $\hat{\mathbf{w}}$, of*

$$T \geq c(1 + H\eta) \max\left\{\ln\left(\frac{c'(1 + H\eta)^2}{\mu\eta}\right), \frac{\rho^2}{G^2}\right\}$$

steps of SGD *on φ_k initialized at $\mathbf{w}_k$ satisfies*

$$\mathbb{E}\|\nabla\varphi_k(\hat{\mathbf{w}})\|^2 \leq \frac{\mu}{4\eta}\|\hat{\mathbf{w}} - \mathbf{w}_k\|^2 + G^2.$$

Proof. Since $\varphi_k(\mathbf{w}) = \hat{L}(\mathbf{w}) + \frac{1}{2\eta}\|\mathbf{w} - \mathbf{w}_k\|^2$ up to an additional affine term, it is easy to see that φ_k is $\frac{1}{\eta}$-strongly convex and $(H + \frac{1}{\eta})$-smooth. Therefore, for any $\mathbf{w}$

$$\|\nabla\varphi_k(\mathbf{w})\|^2 \leq 2\left(H + \frac{1}{\eta}\right)(\varphi_k(\mathbf{w}) - \varphi_k(\mathbf{w}_\varphi^*)), \tag{40}$$

where $\mathbf{w}_\varphi^* = \arg\min_{\mathbf{w}} \varphi_k(\mathbf{w})$. In addition, the output, $\hat{\mathbf{w}}$, of SGD on φ_k using the optimal stepsize and initialized at $\mathbf{w}_k$ is, for universal constants $c, c' \geq 1$ ([Nemirovskii & Yudin, 1983](#))

$$\mathbb{E}[\varphi_k(\hat{\mathbf{w}}) - \varphi_k(\mathbf{w}_\varphi^*)] \leq c\left(H + \frac{1}{\eta}\right)\|\mathbf{w}_k - \mathbf{w}_\varphi^*\|^2 \exp\left(-\frac{T}{c'(1 + H\eta)}\right) + \frac{c\eta\rho^2}{T}. \tag{41}$$

So, by the relaxed triangle inequality and the strong convexity of φ_k

$$\left\|\mathbf{w}_k - \mathbf{w}_\varphi^*\right\|^2 \leq 2\|\hat{\mathbf{w}} - \mathbf{w}_k\|^2 + 2\left\|\hat{\mathbf{w}} - \mathbf{w}_\varphi^*\right\|^2 \tag{42}$$

$$\leq 2\|\hat{\mathbf{w}} - \mathbf{w}_k\|^2 + 4\eta\left(\varphi_k(\hat{\mathbf{w}}) - \varphi_k(\mathbf{w}_\varphi^*)\right) \tag{43}$$

$$\leq 2\|\hat{\mathbf{w}} - \mathbf{w}_k\|^2 + \frac{4c\eta^2\rho^2}{T} + 4c(1 + H\eta)\exp\left(-\frac{T}{c'(1 + H\eta)}\right)\|\mathbf{w}_k - \mathbf{w}_\varphi^*\|^2. \tag{44}$$

Therefore, if

$$T \geq c'(1 + H\eta)\ln(8c(1 + H\eta)) \tag{45}$$

then

$$\left\|\mathbf{w}_k - \mathbf{w}_\varphi^*\right\|^2 \leq 4\|\hat{\mathbf{w}} - \mathbf{w}_k\|^2 + \frac{8c\eta^2\rho^2}{T}, \tag{46}$$

and thus

$$\|\nabla\varphi_k(\hat{\mathbf{w}})\|^2 \leq 2\left(H + \frac{1}{\eta}\right)(\varphi_k(\hat{\mathbf{w}}) - \varphi_k(\mathbf{w}_\varphi^*)) \tag{47}$$

$$\leq 2c\left(H + \frac{1}{\eta}\right)^2 \exp\left(-\frac{T}{c'(1 + H\eta)}\right)\|\mathbf{w}_k - \mathbf{w}_\varphi^*\|^2 + \frac{2c(1 + H\eta)\rho^2}{T} \tag{48}$$

$$\leq 2c\left(H + \frac{1}{\eta}\right)^2 \exp\left(-\frac{T}{c'(1 + H\eta)}\right)\left(4\|\hat{\mathbf{w}} - \mathbf{w}_k\|^2 + \frac{8c\eta^2\rho^2}{T}\right) + \frac{2c(1 + H\eta)\rho^2}{T} \tag{49}$$

$$\leq 8c\left(H + \frac{1}{\eta}\right)^2 \exp\left(-\frac{T}{c'(1 + H\eta)}\right)\|\hat{\mathbf{w}} - \mathbf{w}_k\|^2 + \frac{4c(1 + H\eta)\rho^2}{T}. \tag{50}$$

Therefore, choosing

$$T \geq \max\left\{c'(1 + H\eta)\ln\left(\frac{32c(1 + H\eta)^2}{\mu\eta}\right), \frac{4c(1 + H\eta)\rho^2}{G^2}\right\} \tag{51}$$

ensures that

$$\|\nabla\varphi_k(\hat{\mathbf{w}})\|^2 \leq \frac{\mu}{4\eta}\|\hat{\mathbf{w}} - \mathbf{w}_k\|^2 + G^2. \tag{52}$$

Finally, we note that our choice of T (51) satisfies the condition (45) because $\hat{L}$ being H smooth implies that L is $(H + \delta)$-smooth and therefore

$$\mu \leq H + \delta \leq H + \frac{1}{\eta} \implies \frac{1 + H\eta}{\eta\mu} \geq 1.$$

$\square$