Robust stochastic optimization with the proximal point method

Damek Davis[*]   Dmitriy Drusvyatskiy[†]

Abstract

Standard results in stochastic convex optimization bound the number of samples that an algorithm needs to generate a point with small function value in expectation. In this work, we show that a wide class of such algorithms on strongly convex problems can be augmented with high confidence bounds at an overhead cost that is only logarithmic in the confidence level and polylogarithmic in the condition number. We discuss consequences both for streaming and offline algorithms.

1 Introduction

Stochastic convex optimization lies at the core of modern statistical and machine learning. Standard results in the subject bound the number of samples that an algorithm needs to generate a point with small function value in *expectation*. More nuanced *high probability* guarantees are rarer, and typically either rely on "light-tails" assumptions or exhibit worse sample complexity. To address this issue, we show that a wide class of stochastic algorithms for strongly convex problems can be augmented with high confidence bounds at an overhead cost that is only logarithmic in the confidence level and polylogarithmic in the condition number. We discuss consequences both for streaming and offline algorithms. The procedure we propose, called `proxBoost`, is elementary and combines two well-known ingredients: robust distance estimation and the proximal point method.

To illustrate the proposed procedure, consider the optimization problem

$$\min_{x}\ f(x)$$

where $f\colon \mathbf{R}^d \to \mathbf{R}$ is a μ-strongly convex function with L-Lipschitz continuous gradient. We will later consider the more general class of convex composite problems. We aim to develop generic procedures that equip stochastic algorithms with high confidence guarantees. Consequently, it will be convenient to treat such algorithms as black boxes. More

[*]School of ORIE, Cornell University, Ithaca, NY 14850, USA; `people.orie.cornell.edu/dsd95/`.

[†]Department of Mathematics, U. Washington, Seattle, WA 98195; Microsoft Research, Redmond, WA 98052; `www.math.washington.edu/~ddrusv`. Research of Drusvyatskiy was supported by the NSF DMS 1651851 and CCF 1740551 awards.

formally, suppose that the function f may only be accessed through a *minimization oracle* $\mathcal{M}_f(\cdot)$, which on input $\epsilon > 0$, returns a point x_ϵ satisfying the low confidence bound

$$\mathbb{P}(f(x_\epsilon) - \min f \leq \epsilon) \geq \frac{2}{3}. \tag{1.1}$$

By Markov's inequality, minimization oracles arise from any algorithm that can generate a point x_ϵ satisfying $\mathbb{E}f(x_\epsilon) - \min f \leq \epsilon/3$. For example, oracles for minimizing an expectation $f(x) = \mathbb{E}_z[f(x,z)]$ may be constructed from streaming algorithms or from offline empirical risk minimization methods.

The procedure introduced in this paper executes a minimization oracle multiple times in order to boost its confidence. To quantify this overhead, let $\mathcal{C}_{\mathcal{M}}(\epsilon, f)$ denote the cost of the oracle call $\mathcal{M}_f(\epsilon)$. It is natural to assume that the cost is decreasing in ϵ and increasing in the condition number $\tau := L/\mu$. The cost may also depend on other parameters, such as initialization quality and bounds on optimal value, but we ignore these for a moment. Given a minimization oracle and its cost, we investigate the following question:

> Is there a procedure within this oracle model of computation that returns a point x_ϵ satisfying the high confidence bound
>
> $$\mathbb{P}(f(x_\epsilon) - \min f \leq \epsilon) \geq 1 - p \tag{1.2}$$
>
> at a total cost that is only a "small" multiple of $\mathcal{C}_{\mathcal{M}}(\epsilon, f) \cdot \ln(\frac{1}{p})$?

We will see that the answer is yes, with the total cost on the order of

$$\log\left(\frac{\log(\tau)}{p}\right) \log(\tau) \cdot \mathcal{C}_{\mathcal{M}}\left(\tfrac{\epsilon}{\log(\tau)}, f\right).$$

Thus, high probability bounds are achieved with a small cost increase, which depends only logarithmically on $1/p$ and polylogarithmically on the condition number τ.

Before introducing our approach, we discuss two techniques for boosting the confidence of a minimization oracle, both of which have limitations. As a first approach, one may query the oracle $\mathcal{M}_f(\epsilon)$ multiple times and pick the "best" iterate from the batch. This approach is flawed since often one cannot test which iterate is "best" without increasing sample complexity. To illustrate, consider estimating the expectation $f(x) = \mathbb{E}_z[f(x,z)]$ to ϵ-accuracy for a fixed point x. This task amounts to approximate mean estimation, which may require on the order of $1/\epsilon^2$ samples, even under sub-Gaussian assumptions [8]. In this paper, the cost $\mathcal{C}_{\mathcal{M}}(\epsilon, f)$ will typically scale at worst as $1/\epsilon$, and therefore mean estimation would significantly degrade the overall sample complexity.

As the second approach, strong convexity immediately implies the distance estimate

$$\mathbb{P}(\|x_\epsilon - \bar{x}\| \leq \sqrt{2\epsilon/\mu}) \geq \frac{2}{3},$$

where $\bar{x}$ is the minimizer of f. Given this bound, one may apply the *robust distance estimation* technique of [35, p. 243] and [19] to choose a point near $\bar{x}$: Run m trials of

$\mathcal{M}_f(\epsilon)$ and find one iterate x_{i^*} around which the other points "cluster". Then the point x_{i^*} will be within a distance of $3\sqrt{2\epsilon/\mu}$ from $\bar{x}$ with probability $1 - \exp(-m/18)$. The downside of this strategy is that when converting naively back to function values, the suboptimality gap becomes $f(x_{i^*}) - \min f \leq \frac{L}{2}\|x_{i^*} - \bar{x}\|^2 \leq 9\tau\epsilon$. Thus the function gap at x_{i^*} may be significantly larger than the expected function gap at x_ϵ, by a factor of the condition number. Therefore, robust distance estimation exhibits a trade-off between robustness and efficiency.

The trade-off between robustness and efficiency disappears for perfectly conditioned losses. Therefore, it appears plausible that one might avoid the τ factor through an iterative algorithm that solves a sequence of nearby, better conditioned problems. This is the strategy we explore here. The `proxBoost` procedure embeds the robust distance estimation technique inside a proximal point method. The algorithm begins by declaring the initial point x_0 to be the output of the robust distance estimator on f. Then the better conditioned function

$$f^t(x) := f(x) + \frac{\mu 2^t}{2}\|x - x_t\|^2,$$

is formed and the next iterate x_{t+1} is declared to be the output of the robust distance estimator on f^t. The procedure is effective since the conditioning of f^t rapidly improves with t, which makes the robust distance estimator more efficient as the counter t grows.

The `proxBoost` procedure can be applied to a wide class of stochastic minimization oracles, for example, streaming or empirical risk minimization (ERM) algorithms. For these problems, the loss f takes the form

$$f(x) = \mathbb{E}_{z\sim\mathbb{P}}\left[f(x,z)\right], \tag{1.3}$$

where the population data z follows a fixed unknown distribution $\mathbb{P}$ and the loss $f(\cdot, z)$ is convex and smooth for a.e. $z \in \mathbb{P}$. The cost of streaming or ERM oracles is then measured by the number of samples drawn from $\mathbb{P}$. We now illustrate the consequences of `proxBoost` for these oracles.

1.1 Streaming Oracles

Stochastic gradient methods can be treated as minimization oracles $\mathcal{M}_f(\epsilon)$ with cost $\mathcal{C}_{\mathcal{M}}(\epsilon, f)$ that is measured by the number stochastic gradient estimates needed to reach functional accuracy ϵ in expectation. An algorithm with minimal such cost was proposed by Ghadimi and Lan [16]. It generates a point x_ϵ satisfying $\mathbb{E}\left[f(x_\epsilon) - \min f\right] \leq \epsilon$ with

$$\mathcal{O}\left(\sqrt{\tau}\ln\left(\frac{\Delta_{\text{in}}}{\epsilon}\right) + \frac{\sigma^2}{\mu\epsilon}\right), \tag{1.4}$$

stochastic gradient evaluations, where the quantity σ^2 is an upper bound on the variance of the stochastic gradient estimator $\nabla f(x,z)$ and Δ_{in} is a known upper bound on the initial function gap $\Delta_{\text{in}} \geq f(x_0) - f^*$. A simpler algorithm with a similar efficiency estimate was recently presented by Kulunchakov and Mairal [24], and was based on estimate sequences.

Aybat et al. [4] present an algorithm with similar efficiency, but in contrast to previous work, it does not require the variance σ^2 and the initial gap Δ_{in} as inputs.

It is intriguing to ask if one can equip the stochastic gradient method and its accelerated variant with high confidence guarantees. In their original work [15,16], Ghadimi and Lan provide an affirmative answer under the additional assumption that the stochastic gradient estimator has light tails. The very recent preprint of Juditsky-Nazin-Nemirovsky-Tsybakov [22] shows that one can avoid the light tail assumption for the basic stochastic gradient method, and for mirror descent more generally, by truncating the gradient estimators. High confidence bounds for the accelerated method, without light tail assumptions, remain open.

In this work, the optimal method of [16] will be used as a minimization oracle within `proxBoost`, allowing us to nearly match the efficiency estimate (1.4) without "light-tail" assumptions. Equipped with this oracle, `proxBoost` returns a point x satisfying

$$\mathbb{P}[f(x) - f^* \leq \epsilon] \geq 1 - p,$$

and the overall cost of the procedure is

$$\widetilde{\mathcal{O}}\left(\log\left(\frac{1}{p}\right)\left(\sqrt{\tau}\ln\left(\frac{\Delta_{\text{in}}}{\epsilon} \vee \tau\right) + \frac{\sigma^2}{\mu\epsilon}\right)\right).$$

Here, $\widetilde{\mathcal{O}}(\cdot)$ only suppresses logarithmic dependencies in τ; see Section 5 for details. Thus for small ϵ, the sample complexity of the robust procedure is roughly $\log(1/p)$ times the efficiency estimate (1.4) of the low-confidence algorithm. In this paper, we also provide similar accelerated guarantees for additive convex composite problems, by using the routine of [22] in the last step of `proxBoost`.

1.2 Empirical Risk Minimization Oracles

An alternative approach to streaming algorithms, such as the stochastic gradient method, is based on empirical risk minimization (ERM). Namely, we may draw i.i.d. samples $z_1, \ldots, z_n \sim \mathbb{P}$ and minimize the empirical average

$$\min_x \; f_S(x) := \frac{1}{n}\sum_{i=1}^{n} f(x, z_i). \tag{1.5}$$

A key question is to determine the number n of samples that would ensure that the minimizer x_S of the empirical risk f_S has low generalization error $f(x_S) - \min f$, with reasonably high probability. There is a vast literature on this subject; see for example [5,19,38,39]. We build here on the work of Hsu-Sabato [19], who focused on high confidence guarantees for nonnegative losses $f(x,z)$. They showed that the empirical risk minimizer x_S yields a robust distance estimator of the true minimizer of f, by the aforementioned resampling technique. As a consequence they deduced that the ERM learning rule can find a point x_S satisfying the relative error guarantee

$$\mathbb{P}[f(x_S) \leq (1+\gamma)f^*] \geq 1 - p,$$

with the sample complexity n on the order of

$$\mathcal{O}\left(\log\left(\frac{1}{p}\right)\cdot\frac{\hat{\tau}\tau}{\gamma}\right).$$

Loosely speaking, here τ and $\hat{\tau}$ are the condition numbers of f and f_S, respectively. By embedding empirical risk minimization within `proxBoost`, we obtain an algorithm with the much better sample complexity

$$\widetilde{\mathcal{O}}\left(\log\left(\frac{1}{p}\right)\left(\frac{\hat{\tau}}{\gamma}+\hat{\tau}\right)\right),$$

where the symbol $\widetilde{\mathcal{O}}$ only suppresses polylogarithmic dependence on τ and $\hat{\tau}$.

Related literature

Our paper rests on two pillars: the proximal point method and robust distance estimation. Both techniques have been well studied in the optimization and statistics literature. The proximal point method was introduced by Martinet [30, 31] and further popularized by Rockafellar [37]. The construction is also closely related to the smoothing function of Moreau [33]. Recently, there has been a renewed interest in the proximal point method, most notably due to its uses in accelerating variance reduced methods for minimizing finite sums of convex functions [13,26,27,40]. The proximal point method has also featured prominently as a guiding principle in nonconvex optimization, with the works of [2, 3, 10–12]. The stepsize schedule we use within the proximal point method is geometrically decaying, in contrast to the more conventional polynomially decaying schemes. Geometrically decaying schedules for subgradient methods were first used by Goffin [18] and have regained some attention recently due to their close connection to the popular step-decay schedule in stochastic optimization [4, 14, 41, 42].

Robust distance estimation has a long history. The estimator we use was first introduced in [35, p. 243], and can be viewed as a multivariate generalization of the median of means estimator [1, 20]. Robust distance estimation was further investigated in [19] with a focus on high probability guarantees for empirical risk minimization. A different generalization based on the geometric median was studied in [32]. Other recent articles related to the subject include median of means tournaments [28], robust multivariate mean estimators [21, 29], and bandits with heavy tails [7].

One of the main applications of our techniques is to streaming algorithms. Most currently available results that establish high confidence convergence guarantees make sub-Gaussian assumptions on the stochastic gradient estimator [15, 17, 23, 34]. More recently there has been renewed interest in obtaining robust guarantees without the light-tails assumption. For example, the two works [9, 43] make use of the geometric median of means technique to robustly estimate the gradient in distributed optimization. A different technique was recently developed by Juditsky et al. [22], where the authors establish high confidence guarantees for mirror descent type algorithms by truncating the gradient.

The outline of the paper is as follows. Section 2 presents the problem setting. Section 3 develops the `proxBoost` procedure. Section 4 presents consequences for empirical risk minimization, while Section 5 discusses consequences for streaming algorithms.

2 Problem setting

Throughout, we follow standard notation of convex optimization, as set out for example in the monographs [6, 36]. We let $\mathbf{R}^d$ denote an Euclidean space with inner product $\langle \cdot, \cdot \rangle$ and the induced norm $\|x\| = \sqrt{\langle x, x\rangle}$. The symbol $B_\epsilon(x)$ will stand for the closed ball around x of radius $\epsilon > 0$. We will use the shorthand interval notation $[1, m] := \{1, \ldots, m\}$ for any number $m \in \mathbb{N}$.

Consider a function $f\colon \mathbf{R}^d \to \mathbf{R} \cup \{+\infty\}$. The effective domain of f, denoted dom f, consists of all points where f is finite. The function f is called μ-strongly convex if the perturbed function $f - \frac{\mu}{2}\|\cdot\|^2$ is convex. We say that f is L-smooth if it differentiable with L-Lipschitz continuous gradient. If f is both μ-strongly convex and L-smooth, then the two sided bound holds:

$$\frac{\mu}{2}\|x - \bar{x}\|^2 \leq f(x) - f(\bar{x}) \leq \frac{L}{2}\|x - \bar{x}\|^2 \qquad \text{for all } x, \tag{2.1}$$

where $\bar{x}$ is the minimizer of f. We then define the condition number of f to be $\tau := L/\mu$.

Assumption 2.1. Throughout this work, we consider the optimization problem

$$\min_{x \in \mathbf{R}^d} \ f(x) \tag{2.2}$$

where the function $f\colon \mathbf{R}^d \to \mathbf{R} \cup \{+\infty\}$ is μ-strongly convex. We denote the minimizer of f by $\bar{x}$ and its minimal value by $f^* := \min f$.

Let us suppose for the moment that the only access to f is by querying a black-box procedure that estimates $\bar{x}$. Namely following [19] we will call a procedure $\mathcal{D}(\epsilon)$ a *weak distance oracle* for the problem (2.2) if it returns a point x satisfying

$$\mathbb{P}[\|x - \bar{x}\| \leq \epsilon] \geq \frac{2}{3}. \tag{2.3}$$

We will moreover assume that when querying $\mathcal{D}(\epsilon)$ multiple times, the returned vectors are all statistically independent. Weak distance oracles arise naturally in stochastic optimization both in streaming and offline settings. We will discuss specific examples in Sections 4 and 5. The numerical value 2/3 plays no real significance and can be replaced by any fraction greater than a half.

It is well known from [35, p. 243] and [19] that the low-confidence estimate (2.3) can be improved to a high confidence guarantee by a resampling trick. Following [19], we define the *robust distance estimator* $\mathcal{D}(\epsilon, m)$ to be the following procedure

Algorithm 1: Robust Distance Estimation $\mathcal{D}(\epsilon, m)$
Input: trial count m, access to a weak distance oracle $\mathcal{D}(\epsilon)$ Query m times the oracle $\mathcal{D}(\epsilon)$ and let $Y = \{y_1, \ldots, y_m\}$ consist of the responses. **Step** $i = 1, \ldots, m$: Compute $r_i = \min\{r \geq 0 : \lvert B_r(y_i) \cap Y \rvert > \frac{m}{2}\}$. Set $i^* = \operatorname{argmin}_{i \in [1,m]} r_i$ **Return** y_{i^*}

Thus the robust distance estimator $\mathcal{D}(\epsilon, m)$ first generates m statistically independent random points $y_1, \ldots, y_m$ by querying m times the weak distance oracle $\mathcal{D}(\epsilon)$. Then the procedure computes the smallest radius ball around each point y_i that contains more than half of the generated points $\{y_1, \ldots, y_m\}$. Finally, the point y_{i^*} corresponding to the smallest such ball is returned. The intuition behind this procedure is that by Chernoff's bound, with high probability, the ball $B_\epsilon(\bar{x})$ will contain at least $m/2$ of the generated points. Therefore in this event, the estimate $r_{i^*} < 2\epsilon$ holds. Moreover since the two sets, $B_\epsilon(\bar{x})$ and $B_{r_{i^*}}(y_{i^*})$ intersect, it follows that $\bar{x}$ and y_{i^*} are within a distance of 3ϵ of each other. For a complete argument, see [19, Propositions 8,9].

Lemma 2.2 (Robust Distance Estimator)**.** *The point x returned by $\mathcal{D}(\epsilon, m)$ satisfies*

$$\mathbb{P}[\|x - \bar{x}\| \leq 3\epsilon] \geq 1 - \exp\left(-\frac{m}{18}\right).$$

We seek to understand how one may use a robust distance estimator $\mathcal{D}(\epsilon, m)$ to compute a point x satisfying $f(x) - \min f \leq \delta$ with high probability, where $\delta > 0$ is a specified accuracy. As motivation, consider the case when f is L-smooth. Then one immediate approach is to appeal to the upper bound in (2.1). Hence the point $x = \mathcal{D}(\epsilon, m)$, with $\epsilon = \sqrt{\frac{2\delta}{9L}}$, satisfies the guarantee

$$\mathbb{P}\left(f(x) - f^* \leq \delta\right) \geq 1 - \exp\left(-\frac{m}{18}\right).$$

We will follow an alternative approach, which in concrete circumstances can significantly decrease the overall cost. The optimistic goal is to replace the accuracy $\epsilon \approx \sqrt{\frac{\delta}{L}}$ used in the call to $\mathcal{D}(\epsilon, m)$ by the potentially much larger quantity $\sqrt{\frac{\delta}{\mu}}$. The strategy we propose will apply a robust distance estimator $\mathcal{D}$ to a sequence of optimization problems that are better and better conditioned, thereby amortizing the overall cost. In the initial step, we will simply apply $\mathcal{D}$ to f with the low accuracy $\sqrt{\frac{\delta}{\mu}}$. In step i, we will apply $\mathcal{D}$ to a new function f^i, which has condition number $\tau_i \approx \frac{L+\mu 2^i}{\mu+\mu 2^i}$, with accuracy $\epsilon_i \approx \sqrt{\frac{\delta}{\mu+\lambda_i}}$. Continuing this process for $T \approx \log_2(\tau)$ rounds, we arrive at accuracy $\epsilon_T \approx \sqrt{\frac{\delta}{\mu+L}}$ and a function f^T that is nearly perfectly conditioned with $\tau_T \leq 2$. In this way, the total cost is amortized over the sequence of optimization problems. The key of course is to control the error incurred by varying the optimization problems along the iterations.

3 Main result

The procedure outlined at the end of the previous section can be succinctly described within the framework of an *inexact proximal point method*. Henceforth fix an increasing sequence of penalties $\lambda_0, \ldots, \lambda_T$ and a sequence of centers $x_0, \ldots, x_T$. For each index $i = 0, \ldots, T$, define the quadratically perturbed functions and their minimizers:

$$f^i(x) := f(x) + \frac{\lambda_i}{2}\|x - x_i\|^2, \qquad \bar{x}_{i+1} := \operatorname*{argmin}_x f^i(x).$$

The exact proximal point method [30, 31, 37] proceeds by inductively declaring $x_i = \bar{x}_i$ for $i \geq 1$. Since computing $\bar{x}_i$ is in general impossible, we will instead monitor the error $\|\bar{x}_i - x_i\|$. The following elementary result will form the basis for the rest of the paper. To simplify notation, we will set $\bar{x}_0 := \operatorname{argmin} f$ and $\lambda_{-1} := 0$, throughout.

Theorem 3.1 (Inexact proximal point method)**.** *For all $j \geq 0$, the estimates hold:*

$$f^j(\bar{x}_{j+1}) - f^* \leq \sum_{i=0}^{j} \frac{\lambda_i}{2}\|\bar{x}_i - x_i\|^2, \tag{3.1}$$

Consequently, we have the error decomposition:

$$\boxed{f(x_{j+1}) - \min f \leq (f^j(x_{j+1}) - f^j(\bar{x}_{j+1})) + \sum_{i=0}^{j} \frac{\lambda_i}{2}\|\bar{x}_i - x_i\|^2}. \tag{3.2}$$

Moreover, if f is L-smooth, then for all $j \geq 0$ the estimate holds:

$$f(x_j) - f^* \leq \frac{L + \lambda_{j-1}}{2}\|\bar{x}_j - x_j\|^2 + \sum_{i=0}^{j-1} \frac{\lambda_i}{2}\|\bar{x}_i - x_i\|^2. \tag{3.3}$$

Proof. We first establish (3.1) by induction. To see the base case $j = 0$, observe

$$f^0(\bar{x}_1) \leq f^0(\bar{x}_0) = f^* + \frac{\lambda_0}{2}\|\bar{x}_0 - x_0\|^2.$$

As the inductive assumption, suppose the estimate (3.1) holds up to iteration $j - 1$. We then conclude

$$\begin{aligned} f^j(\bar{x}_{j+1}) \leq f^j(\bar{x}_j) &= f(\bar{x}_j) + \frac{\lambda_j}{2}\|\bar{x}_j - x_j\|^2 \\ &\leq f^{j-1}(\bar{x}_j) + \frac{\lambda_j}{2}\|\bar{x}_j - x_j\|^2 \leq f^* + \sum_{i=0}^{j} \frac{\lambda_i}{2}\|\bar{x}_i - x_i\|^2, \end{aligned}$$

where the last inequality follows by the inductive assumption. This completes the proof of (3.1). To see (3.2), we observe using (3.1) the estimate

$$\begin{aligned} f(x_{j+1}) - f^* \leq f^j(x_{j+1}) - f^* &= (f^j(x_{j+1}) - f^j(\bar{x}_{j+1})) + f^j(\bar{x}_{j+1}) - f^* \\ &\leq (f^j(x_{j+1}) - f^j(\bar{x}_{j+1})) + \sum_{i=0}^{j} \frac{\lambda_i}{2}\|\bar{x}_i - x_i\|^2. \end{aligned}$$

Inequality (3.3) for index $j = 0$ follows from smoothness, while the general case $j \geq 1$ follows from using the bound $f^j(x_{j+1}) - f^j(\bar{x}_{j+1}) \leq \frac{L+\lambda_j}{2}\|\bar{x}_{j+1} - x_{j+1}\|^2$ in (3.2). □

The main conclusion of Theorem 3.1 is the decomposition of the functional error described in (3.2). Namely, the estimate (3.2) upper bounds the error $f(x_{j+1}) - \min f$ as the sum of the suboptimality in the last step $f^T(x_{T+1}) - f^T(\bar{x}_{T+1})$ and the errors $\frac{\lambda_i}{2}\|\bar{x}_i - x_i\|^2$ incurred along the way. By choosing T sufficiently large, we can be sure that the function f^T is well-conditioned. Moreover in order to ensure that each term in the sum $\frac{\lambda_i}{2}\|\bar{x}_i - x_i\|^2$ is of order δ, it sufficies to guarantee $\|\bar{x}_i - x_i\| \leq \sqrt{\frac{2\delta}{\lambda_i}}$ for each index i. Since λ_i is an increasing sequence, it follows that we may gradually decrease the tolerance on the errors $\|\bar{x}_i - x_i\|$, all the while improving the conditioning of the functions we encounter. With this intuition in mind, we introduce the `proxBoost` procedure (Algorithm 2).

Algorithm 2: `proxBoost`(δ, p, T)

Input: $\delta \geq 0$, $p \in (0,1)$, $T \in \mathbb{N}$
Set $\lambda_{-1} = 0$, $\epsilon_{-1} = \sqrt{\frac{2\delta}{\mu}}$
Generate a point x_0 satisfying $\|x_0 - \bar{x}_0\| \leq \epsilon_{-1}$ with probability $1-p$.
for $j = 0, \ldots, T-1$ **do**
  Set $\epsilon_j = \sqrt{\frac{2\delta}{\mu+\lambda_j}}$
  Generate a point x_{j+1} satisfying

$$\mathbb{P}\left[\|x_{j+1} - \bar{x}_{j+1}\| \leq \epsilon_j \mid E_j\right] \geq 1-p, \tag{3.4}$$

  where E_j denotes the event $E_j := \left\{x_i \in B_{\epsilon_{i-1}}(\bar{x}_i) \text{ for all } i \in [0,j]\right\}$.
end
Generate a point x_{T+1} satisfying

$$\mathbb{P}\left[f^T(x_{T+1}) - \min f^T \leq \delta \mid E_T\right] \geq 1-p. \tag{3.5}$$

Return x_{T+1}

Thus the `proxBoost`procedure consists of three stages, which we now examine in detail.

Stage I: Initialization. Algorithm 2 begins by generating a point x_0 that is a distance of $\sqrt{\frac{2\delta}{\mu}}$ away from the minimizer of f with probability $1-p$. This task can be achieved by applying a robust distance estimator on f, as discussed previously.

Stage II: Proximal iterations. In each subsequent iteration, x_{j+1} is defined to be a point that is within a radius of $\epsilon_j = \sqrt{\frac{2\delta}{\mu+\lambda_j}}$ from the minimizer of f^j with probability $1-p$ conditioned on the event E_j. The event E_j encodes that each previous iteration was successful in the sense that the point x_i indeed lies inside $B_{\epsilon_{i-1}}(\bar{x}_i)$ for all $i = 0, \ldots, j$.

Thus x_{j+1} can be determined by a procedure that within the event E_j is a robust distance estimator on the function f^j.

Stage III: Cleanup. In the final step, the algorithm outputs a δ-minimizer of f^T with probability $1-p$ conditioned on the event E_T. In particular, if f is L-smooth then we may use a robust distance estimator on f^T. Namely, taking into account the upper bound (2.1), we may declare x_{T+1} to be any point satisfying

$$\mathbb{P}\left[\|x_{T+1}-\bar{x}_{T+1}\| \le \sqrt{\tfrac{2\delta}{L+\lambda_T}} \mid E_T\right] \ge 1-p.$$

Notice that by choosing T sufficiently large, we may ensure that the condition number $\frac{\mu+\lambda_T}{L+\lambda_T}$ of f^T is arbitrarily close to one. If f is not smooth, such as when constraints are present, we can not use a robust distance estimator in the cleanup stage. We will see in Section 5 a different approach, based on the robust stochastic gradient method of [22].

The following theorem summarizes the guarantees of the `proxBoost` procedure.

Theorem 3.2 (Proximal Boost). *Fix a constant $\delta > 0$, a probability of failure $p \in (0,1)$ and a natural number $T \in \mathbb{N}$. Then with probability at least $1-(T+2)p$, the point $x_{T+1} =$ `proxBoost`(δ, p, T) satisfies*

$$f(x_{T+1}) - \min f \le \delta\left(1 + \sum_{i=0}^{T} \frac{\lambda_i}{\mu+\lambda_{i-1}}\right). \tag{3.6}$$

Proof. We first prove by induction the estimate

$$\mathbb{P}[E_t] \ge 1-(t+1)p \qquad \text{for all } t = 0,\ldots,T. \tag{3.7}$$

The base case $t=0$ is immediate from the definition of x_0. Suppose now that (3.7) holds for some index $t-1$. Then the inductive assumption and the definition of x_t yield

$$\mathbb{P}[E_t] = \mathbb{P}[E_t \big| E_{t-1}]\mathbb{P}[E_{t-1}] \ge (1-p)\,(1-tp) \ge 1-(t+1)p,$$

thereby completing the induction. Thus the inequalities (3.7) hold. Define the event

$$F = \{f^T(x_{T+1}) - \min f^T \le \delta\}.$$

We therefore deduce

$$\mathbb{P}[F \cap E_T] = \mathbb{P}[F \mid E_T]\cdot\mathbb{P}[E_T] \ge (1-(T+1)p)(1-p) \ge 1-(T+2)p.$$

Suppose now that the event $F \cap E_T$ occurs. Then using the estimate (3.2), we conclude

$$f(x_{T+1}) - \min f \le (f^T(x_{T+1}) - f^T(\bar{x}_{T+1})) + \sum_{i=0}^{T} \frac{\lambda_i}{2}\|\bar{x}_i - x_i\|^2 \le \delta + \sum_{i=0}^{T} \frac{\delta\lambda_i}{\mu+\lambda_{i-1}},$$

where the last inequality uses the definitions of x_{T+1} and ϵ_j. This completes the proof. $\square$

Looking at the estimate (3.6), we see that the final error $f(x_{T+1}) - \min f$ is controlled by the sum $\sum_{i=0}^{T} \frac{\lambda_i}{\mu+\lambda_{i-1}}$. A moment of thought yields an appealing choice $\lambda_i = \mu 2^i$ for the proximal parameters. Indeed, then every element in the sum $\frac{\lambda_i}{\mu+\lambda_{i-1}}$ is upper bounded by two. Moreover, if f is L-smooth, then the condition number $\frac{L+\lambda_T}{\mu+\lambda_T}$ of f^T is upper bounded by two after only $T = \lceil \log(L/\mu) \rceil$ rounds.

Corollary 3.3 (Proximal boost with geometric decay)**.** *Fix an iteration count T, a target accuracy $\epsilon > 0$, and a probability of failure $p \in (0,1)$. Define the algorithm parameters:*

$$\delta = \frac{\epsilon}{2+2T} \qquad \text{and} \qquad \lambda_i = \mu 2^i \qquad \forall i \in [0,T].$$

Then the point $x_{T+1} =$ `proxBoost`(δ, p, T) satisfies

$$\mathbb{P}(f(x_{T+1}) - \min f \leq \epsilon) \geq 1 - (T+2)p.$$

In the next two sections, we seed the `proxBoost` procedure with (accelerated) stochastic gradient algorithms and methods based on empirical risk minimization. The reader, however, should keep in mind that `proxBoost` is entirely agnostic to the inner workings of the robust distance estimators it uses. The only point to be careful about is that some distance estimators (e.g. stochastic gradient) require to be passed auxiliary quantities, such as an upper estimate on the function gap at the initial point. Therefore, we may have to update such estimates along the iterations of Algorithm 2.

4 Consequences for empirical risk minimization

In this section, we explore the consequences of the `proxBoost` algorithm for empirical risk minimization. Setting the stage, fix a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and equip $\mathbf{R}^d$ with the Borel σ-algebra. Consider the optimization problem

$$\min_x f(x) = \mathbb{E}_{z\sim\mathbb{P}}\left[f(x,z)\right], \tag{4.1}$$

where $f \colon \mathbf{R}^d \times \Omega \to \mathbf{R}_+$ is a measurable nonnegative function. A common approach to expectation minimization problems is based on empirical risk minimization. Namely, we may form an i.i.d. sample $z_1, \ldots, z_n \sim \mathbb{P}$ and minimize the empirical average

$$\min_x f_S(x) := \frac{1}{n}\sum_{i=1}^{n} f(x, z_i). \tag{4.2}$$

A central question is to determine the number n of samples that would ensure that the minimizer x_S of the empirical risk has low generalization error $f(x_S) - \min f$, with reasonably high probability. There is a vast literature on this subject; some representative works include [5,19,38,39]. We build here on the work of Hsu-Sabato [19], who specifically focused on high confidence guarantees for smooth strongly convex minimization. As in the previous sections, we let $\bar{x}$ be a minimizer of f and define the shorthand $f^* = \min f$.

Assumption 4.1. Following [19], we make the following assumptions on the loss.

1. **(Strong convexity)** There exist a real $\mu > 0$ and a natural number $N \in \mathbb{N}$ such that:
   (a) the population loss f is μ-strongly convex,
   (b) the empirical loss $x \mapsto f_S(x)$ is μ-strongly convex with probability at least 5/6, whenever $|S| \geq N$.
2. **(Smoothness)** There exist constants $L, \hat{L} > 0$ such that:
   (a) for a.e. $z \sim \mathbb{P}$, the loss $x \mapsto f(x, z)$ is $\hat{L}$-smooth,
   (b) the population objective $x \mapsto f(x)$ is L-smooth.

Define the empirical and population condition numbers, $\hat{\tau} := \hat{L}/\mu$ and $\tau = L/\mu$, respectively.

The following result proved in [19, Theorem 15] shows that the empirical risk minimizer is a weak distance oracle for the problem (4.1).

Lemma 4.2. *Fix an i.i.d. sample $z_1, \ldots, z_n \sim \mathbb{P}$ of size $n \geq N$. Then the minimizer x_S of the empirical risk* (4.2) *satisfies the bound:*

$$\mathbb{P}\left[\|x_S - \bar{x}\| \leq \sqrt{\frac{96\hat{L}f(x_*)}{n\mu^2}}\right] \geq 2/3.$$

In particular, using Algorithm 1 we may turn empirical risk minimization into a robust distance estimator for the problem (4) using a total of mn samples. Let us estimate the function value at the generated point by a direct application of smoothness. Appealing to Lemma 2.2 and (2.1), we deduce that with probability $1 - \exp(-m/18)$ the procedure will return a point x satisfying

$$f(x) \leq \left(1 + \frac{432\hat{L}L}{n\mu^2}\right) f^*.$$

Observe that this is an estimate of *relative error*. In particular, let $p \in (0, 1)$ be some acceptable probability of failure and let $\gamma > 0$ be a desired level of relative accuracy. Then setting $m = \lceil 18 \ln(1/p) \rceil$ and $n \geq \max\{\frac{432\hat{\tau}\tau}{\gamma}, N\}$, we conclude that x satisfies

$$\mathbb{P}[f(x) \leq (1 + \gamma)f^*] \geq 1 - p, \tag{4.3}$$

while the overall sample complexity of the procedure is

$$\left\lceil 18 \ln\left(\frac{1}{p}\right)\right\rceil \cdot \max\left\{\left\lceil\frac{432\hat{\tau}\tau}{\gamma}\right\rceil, N\right\}. \tag{4.4}$$

This is exactly the result [19, Corollary 16]. We will now see how to find a point x satisfying (4.3) with significantly fewer samples by embedding empirical risk minimization within the `proxBoost` algorithm. Algorithm 3 encodes the empirical risk minimization process on a quadratically regularized problem. Algorithm 4 is the robust distance estimator induced by Algorithm 3. Finally, Algorithm 5 is the `proxBoost` algorithm specialized to empirical risk minimization.

Algorithm 3: `ERM`(n, λ, x)

Input: sample count $n \in \mathbb{N}$, center $x \in \mathbf{R}^d$, amplitude $\lambda > 0$.
Generate i.i.d. samples $z_1, \ldots, z_n \sim \mathbb{P}$ and compute the minimizer $\bar{y}$ of

$$\min_y \; \frac{1}{n}\sum_{i=1}^{n} f(y, z_i) + \frac{\lambda}{2}\|y - x\|^2.$$

Return $\bar{y}$

Algorithm 4: `ERM-R`(n, m, λ, x)

Input: sample count $n \in \mathbb{N}$, trial count $m \in \mathbb{N}$, center $x \in \mathbf{R}^d$, amplitude $\lambda > 0$.
Query m times ERM(n, λ, x) and let $Y = \{y_1, \ldots, y_m\}$ consist of the responses.
Step $j = 1, \ldots, m$:
Compute $r_i = \min\{r \geq 0 : |B_r(y_i) \cap Y| > \frac{m}{2}\}$.
Set $i^* = \operatorname{argmin}_{i \in [1,m]} r_i$
Return y_{i^*}

Algorithm 5: `BoostERM`(γ, T, m)

Input: $T, m \in \mathbb{N}$, $\gamma > 0$
Set $\lambda_{-1} = 0$, $n_{-1} = \frac{432\hat{L}}{\gamma\mu}$
Step $j = 0, \ldots, T$:
$x_j = \text{ERM-R}(n_{j-1}, m, \lambda_{j-1}, x_{j-1})$
$n_j = 432\left\lceil \frac{\hat{L}+\lambda_j}{\mu+\lambda_j}\left(\frac{1}{\gamma} + \sum_{i=0}^{j}\frac{\lambda_i}{\mu+\lambda_{i-1}}\right)\right\rceil \vee N$
Return $x_{T+1} = \text{ERM-R}(\frac{L+\lambda_T}{\mu+\lambda_T} \cdot n_T, m, \lambda_T, x_T)$

Using Theorem 3.2, we can now prove the following result.

Theorem 4.3 (Efficiency of `BoostERM`)**.** *Fix a target relative accuracy $\gamma > 0$ and numbers $T, m \in \mathbb{N}$. Then with probability at least $1 - (T+2)\exp\left(-\frac{m}{48}\right)$, the point $x_{T+1} =$ `BoostERM`(γ, T, m) satisfies*

$$f(x_{T+1}) - f^* \leq \left(1 + \sum_{i=0}^{T}\frac{\lambda_i}{\mu + \lambda_{i-1}}\right)\gamma f^*.$$

Proof. We will verify that Algorithm 5 is an instantiation of Algorithm 2 with $\delta = \gamma f^*$ and $p = \exp(-\frac{m}{18})$. More precisely, we will prove by induction that with this choice of p

and δ, the iterates x_j satisfy (3.4) for each index $j = 0, \ldots, T$ and x_{T+1} satisfies (3.5). As the base case, consider the evaluation $x_0 = \texttt{ERM-R}(n_{-1}, m, \lambda_{-1}, x_{-1})$. Then Lemma 2.2 and Theorem 4.2 guarantee

$$\mathbb{P}\left[\|x_0 - \bar{x}_0\| \le 3\sqrt{\frac{96\hat{L}f^*}{n_{-1}\mu^2}}\right] \ge 1 - \exp\left(-\frac{m}{18}\right).$$

Taking into account the definition of n_{-1}, we deduce

$$\mathbb{P}\left[\|x_0 - \bar{x}_0\| \le \epsilon_{-1}\right] \ge 1 - p,$$

as claimed. As an inductive hypothesis, suppose that (3.4) holds for the iterates $x_0, x_1, \ldots, x_{j-1}$. We will prove it holds for x_j. To this end, suppose that the event E_{j-1} occurs. Then by the same reasoning as in the base case, the point x_j satisfies

$$\mathbb{P}\left[\|x_j - \bar{x}_j\| \le 3\sqrt{\frac{96(\hat{L}+\lambda_{j-1})f^{j-1}(\bar{x}_j)}{n_{j-1}(\mu+\lambda_{j-1})^2}}\right] \ge 1 - \exp\left(-\frac{m}{18}\right). \tag{4.5}$$

Observe now, using (3.1) and the inductive assumption, the estimate:

$$f^{j-1}(\bar{x}_j) - f^* \le \sum_{i=0}^{j-1} \frac{\lambda_i}{2}\|\bar{x}_i - x_i\|^2 \le \delta \sum_{i=0}^{j-1} \frac{\lambda_i}{\mu + \lambda_{i-1}}.$$

Combining this inequality with (4.5), we conclude that conditioned on the event E_{j-1}, we have with probability $1 - p$ the guarantee

$$\frac{\mu+\lambda_{j-1}}{2}\|x_j - \bar{x}_j\|^2 \le \frac{432(\hat{L}+\lambda_{j-1})(1+\gamma\sum_{i=0}^{j-1}\frac{\lambda_i}{\mu+\lambda_{i-1}})}{n_{j-1}(\mu+\lambda_{j-1})} \cdot f^* = \gamma f^*, \tag{4.6}$$

where the last equality follows from the definition of n_{j-1}. Thus the estimate (3.4) holds for the iterate $x_0, \ldots, x_T$, as needed. Suppose now that that event E_T occurs. Then by exactly the same reasoning that led to (4.6), we have the estimate

$$\frac{\mu+\lambda_T}{2}\|x_{T+1} - \bar{x}_{T+1}\|^2 \le \frac{\mu+\lambda_T}{L+\lambda_T}\gamma f^*.$$

Using smoothness, we therefore deduce $f^T(x_{T+1}) - \min f^T \le \gamma f^* = \delta$, as claimed. An application of Theorem 3.2 completes the proof. □

When using the proximal parameters $\lambda_i = \mu 2^i$, we obtain the following guarantee.

Corollary 4.4 (Efficiency of `BoostERM` with geometric decay)**.** *Fix a target relative accuracy $\gamma' > 0$ and a probability of failure $p \in (0,1)$. Define the algorithm parameters:*

$$T = \lceil \log_2(\tau) \rceil, \qquad m = \left\lceil 18 \ln\left(\frac{T+2}{p}\right)\right\rceil, \qquad \gamma = \frac{\gamma'}{2+2T}, \qquad \lambda_t = \mu 2^t.$$

Then with probability of at least $1-p$, the point $x_{T+1} = \texttt{BoostERM}(\gamma, T, m)$ satisfies $f(x^{T+1}) \le (1+\gamma')f^*$. Moreover, the total number of samples used by the algorithm is

$$\mathcal{O}\left(\ln(\tau)\ln\left(\frac{\ln(\tau)}{p}\right)\cdot \max\left\{\left(1+\tfrac{1}{\gamma'}\right)\hat{\tau}\ln(\tau), N\right\}\right).$$

Notice that the sample complexity provided by Corollary 4.4 is an order of magnitude better than (4.4) in terms of the dependence on the condition numbers $\hat{\tau}$ and τ.

5 Consequences for stochastic approximation

In this section, we investigate the consequences of the `proxBoost` algorithm for stochastic approximation. Namely, we will seed `proxBoost` with the robust distance estimator, induced by the stochastic proximal gradient method and its accelerated variant. An important point is that the sample complexity of stochastic gradient methods depends on the initialization quality $f(x_0) - f^*$. Consequently, in order to know how many iterations are needed to reach a desired accuracy $\mathbb{E}f(x_i) - f^* \le \delta$, we must have available an upper bound on the initialization quality $\Delta \ge f(x_0) - f^*$. Moreover, we will have to update the initialization estimate for each proximal subproblem along the iterations of `proxBoost`. The following assumption formalizes this idea.

Assumption 5.1. Consider the proximal minimization problem

$$\min_y \; g(x) := f(y) + \frac{\lambda}{2}\|y-x\|^2,$$

Let $\Delta > 0$ be a real number satisfying $g(x) - \min g \le \Delta$. We will let $\texttt{Alg}(\delta, \lambda, \Delta, x)$ be a procedure that returns a point y satisfying

$$\mathbb{P}[g(y) - \min g \le \delta] \ge \frac{2}{3}.$$

Clearly, by strong convexity, we may turn $\texttt{Alg}(\cdot)$ into a robust distance estimator on the proximal problems as long as Δ is indeed an upper bound on the initialization error. We record the robust distance estimator induced by $\texttt{Alg}(\cdot)$ as Algorithm 6.

Algorithm 6: $\texttt{Alg-R}(\delta, \lambda, \Delta, x, m)$

Input: accuracy $\delta > 0$, amplitude $\lambda > 0$, upper bound $\Delta > 0$, center $x \in \mathbf{R}^d$, trial count $m \in \mathbb{N}$.
Query m times $\texttt{Alg}(\delta, \lambda, \Delta, x)$ and let $Y = \{y_1, \ldots, y_m\}$ consist of the responses.
Step $j = 1, \ldots, m$:
  Compute $r_i = \min\{r \ge 0 : |B_r(y_i) \cap Y| > \frac{m}{2}\}$.
Set $i^* = \operatorname{argmin}_{i\in[1,m]} r_i$
Return y_{i^*}

When f is L-smooth, it is straightforward to instantiate `proxBoost` with the robust distance estimator `Alg-R`. The situation is more nuanced when f is nonsmooth for two reasons. First, it becomes less clear how to control the initialization quality for each proximal subproblem. Secondly, `Alg-R` can not be used in the cleanup stage of `proxBoost` to generate the last iterate x_{T+1}; instead, we will use a different algorithm [22] in this last stage. In the following two sections, we consider the smooth and nonsmooth settings in order.

5.1 Smooth Setting

Throughout this section, in addition to Assumptions 2.1 and 5.1, we assume that f is L-smooth and set $\tau = \frac{L}{\mu}$. Algorithm 7 seeds the `proxBoost` procedure with `Alg-R`.

Algorithm 7: $\texttt{BoostAlg}(\delta, \Delta, x, T, m)$

Input: accuracy $\delta > 0$, upper bound $\Delta > 0$, center $x \in \mathbf{R}^d$, iterations $m, T \in \mathbb{N}$

Set $\lambda_{-1} = 0$, $\Delta_{-1} = \Delta$, $x_{-1} = x$

Step $j = 0, \ldots, T$:

$\quad x_j = \texttt{Alg-R}(\delta/9, \lambda_{j-1}, \Delta_{j-1}, x_{j-1}, m)$

$\quad \Delta_j = \delta\left(\frac{L+\lambda_{j-1}}{\mu+\lambda_{j-1}} + \sum_{i=0}^{j-1} \frac{\lambda_i}{\mu+\lambda_{i-1}}\right)$

Return $x_{T+1} = \texttt{Alg-R}(\frac{\mu+\lambda_T}{L+\lambda_T} \cdot \frac{\delta}{9}, \lambda_T, \Delta_T, x_T, m)$

We can now prove the following theorem on the efficiency of Algorithm 7. The proof is almost a direct application of Theorem 3.2. The only technical point is to verify that for all indices j, the quantity Δ_j is a valid upper bound on the initialization error $f^j(x_j) - \min f^j$ in the event E_j.

Theorem 5.2 (Efficiency of `BoostAlg`). *Fix an arbitrary point $x_{\text{in}} \in \mathbf{R}^d$ and let Δ_{in} be any upper bound $\Delta_{\text{in}} \geq f(x_{\text{in}}) - \min f$. Fix natural numbers $T, m \in \mathbb{N}$. Then with probability at least $1 - (T+2)\exp\left(-\frac{m}{18}\right)$, the point $x_{T+1} = \texttt{BoostAlg}(\delta, \Delta_{\text{in}}, x_{\text{in}}, T, m)$ satisfies*

$$f(x_{T+1}) - \min f \leq \delta\left(1 + \sum_{i=0}^{T} \frac{\lambda_i}{\mu + \lambda_{i-1}}\right).$$

Proof. We will verify that Algorithm 7 is an instantiation of Algorithm 2 with $p = \exp(-\frac{m}{18})$. More precisely, we will prove by induction that with this choice of p, the iterates x_j satisfy (3.4) for each index $j = 0, \ldots, T$ and x_{T+1} satisfies (3.5). To see the base case, observe that Lemma 2.2 guarantees that with probability $1 - p$, the estimate holds:

$$\|x_0 - \bar{x}_0\| \leq 3\sqrt{\frac{2 \cdot \delta/9}{\mu}} = \epsilon_{-1}.$$

As an inductive hypothesis, suppose that (3.4) holds for the iterates $x_0, x_1, \ldots, x_{j-1}$. We will prove it holds for x_j. To this end, suppose that the event E_{j-1} occurs. Then using

(3.3) we deduce

$$f^{j-1}(x_{j-1}) - \min f^{j-1} \le f(x_{j-1}) - f^* \le \frac{L+\lambda_{j-2}}{2}\|\bar{x}_{j-1} - x_{j-1}\|^2 + \sum_{i=0}^{j-2} \frac{\lambda_i}{2}\|\bar{x}_i - x_i\|^2$$
$$\le \frac{\delta(L+\lambda_{j-2})}{\mu+\lambda_{j-2}} + \sum_{i=0}^{j-2} \frac{\delta\lambda_i}{\mu+\lambda_{i-1}} = \Delta_{j-1}, \qquad (5.1)$$

where the two inequalities follow from the inductive assumption and smoothness. Therefore Lemma 2.2 guarantees that conditioned on E_{j-1} with probability $1-p$, the estimate holds:

$$\|x_j - \bar{x}_j\| \le 3\sqrt{\frac{2\cdot\delta/9}{\mu+\lambda_{j-1}}} = \epsilon_{j-1}.$$

Thus the estimate (3.4) holds for the iterate x_j, as needed.

Now suppose that the event E_T holds. Then exactly the same reasoning that led to (5.1) yields the guarantee $f^T(x_T) - \min f^T \le \Delta_T$. Therefore Lemma 2.2 guarantees that with probability $1-p$ conditioned on E_T, we have

$$\|x_{T+1} - \bar{x}_{T+1}\| \le 3\sqrt{\frac{2}{\mu+\lambda_T}\cdot\frac{\delta}{9}\cdot\frac{\mu+\lambda_T}{L+\lambda_T}} = \sqrt{\frac{2\delta}{L+\lambda_T}}.$$

Taking into account smoothness of L, we therefore deduce

$$\mathbb{P}[f^T(x_{T+1}) - \min f^T \le \delta \mid E_T] \ge 1-p,$$

thereby establishing (3.5). An application of Theorem 3.2 completes the proof. □

When using the proximal parameters $\lambda_i = \mu 2^i$, we obtain the following guarantee.

Corollary 5.3 (Efficiency of `BoostAlg` with geometric decay)**.** *Fix an arbitrary point $x_{\rm in} \in \mathbf{R}^d$ and let $\Delta_{\rm in}$ be any upper bound $\Delta_{\rm in} \ge f(x_{\rm in}) - \min f$. Fix a target accuracy $\delta' > 0$ and probability of failure $p \in (0,1)$, and set the algorithm parameters*

$$T = \lceil \log_2(\tau) \rceil, \qquad m = \left\lceil 18 \ln\left(\frac{2+T}{p}\right)\right\rceil, \qquad \delta = \frac{\delta'}{2+2T}, \qquad \lambda_i = \mu 2^i.$$

Then the point $x_{T+1} = \texttt{BoostAlg}(\delta, \Delta_{\rm in}, x_{\rm in}, T, m)$ satisfies

$$\mathbb{P}(f(x_{T+1}) - \min f \le \delta') \ge 1-p.$$

Moreover, the total number of calls to $\texttt{Alg}(\cdot)$ is

$$\left\lceil 18 \ln\left(\frac{\lceil 2 + \log_2(\tau)\rceil}{p}\right)\right\rceil \lceil 2 + \log_2(\tau)\rceil,$$

while the initialization errors satisfy

$$\max_{i=0,\ldots,T+1} \Delta_i \le \frac{\tau + 1 + 2\lceil \log_2(\tau)\rceil}{2 + 2\lceil \log_2(\tau)\rceil}\delta'.$$

5.2 General Setting

In this section, we assume that f is μ-strongly convex, but we do not assume it is smooth. In the previous section, we used smoothness to explicitly construct a sequence $\{\Delta_j\}_j$, which bounded the initial functional error of the proximal subproblems. In the general setting, explicit upper bounds may be unavailable. Thus, we instead show that the iterates with high probability will never leave a small ball centered at the minimizer $\bar{x}$ of f. The radius of this ball is

$$\bar{\epsilon} := \sqrt{\frac{2\delta}{\mu}} + \sqrt{\frac{2\delta}{\mu}\sum_{i=0}^{T}\frac{\lambda_i}{\mu+\lambda_{i-1}}}.$$

In the proof, we will show the initial functional errors of the proximal subproblems are bounded by any constant $M_{\bar{\epsilon}}$, which satisfies

$$M_{\bar{\epsilon}} \geq \sup_{x\in B_{\bar{\epsilon}}(\bar{x})\cap \mathrm{dom}\, f} \{f(x) - f^*\}. \tag{5.2}$$

For example, if the objective function f is ℓ-Lipschitz continuous on $B_{\bar{\epsilon}}(\bar{x}) \cap \mathrm{dom}\, f$, then we may simply set $M_{\bar{\epsilon}} = \ell\bar{\epsilon}$. We should note that once we specialize to stochastic gradient methods, the quantity $M_{\bar{\epsilon}}$ will appear only logarithmically in the sample complexity.

We use the bound (5.2) in the following algorithm, reminiscent of Algorithm (7). We purposefully, do not yet specify the procedure for obtaining x_{T+1} from x_T.

Algorithm 8: `GBoostAlg`$(\delta, \Delta, x, T, m)$

Input: accuracy $\delta > 0$, upper bound $\Delta > 0$, center $x \in \mathbf{R}^d$, iterations $m, T \in \mathbb{N}$
Set $\lambda_{-1} = 0$, $\Delta_{-1} = \Delta$, $x_{-1} = x$
Set $x_0 =$ `Alg-R`$(\delta/9, \lambda_{-1}, \Delta_{-1}, x_{-1}, m)$
Step $j = 1, \ldots, T$:
  $x_j =$ `Alg-R`$(\delta/9, \lambda_{j-1}, M_{\bar{\epsilon}}, x_{j-1}, m)$
Return Any point x_{T+1} satisfying (3.5).

The following theorem is almost a direct consequence of Theorem 3.2. Following the notation of Algorithm 2, we define the events $E_j := \{x_i \in B_{\epsilon_{i-1}}(\bar{x}_i) \text{ for all } i \in [0, j]\}$, where $x_1, \ldots, x_T$ are the iterates generated by `GBoostAlg`.

Theorem 5.4 (Efficiency of `GBoostAlg`)**.** *Fix an arbitrary point $x_{\mathrm{in}} \in \mathbf{R}^d$ and let Δ_{in} be any upper bound $\Delta_{\mathrm{in}} \geq f(x_{\mathrm{in}}) - \min f$. Fix natural numbers $T, m \in \mathbb{N}$ and consider the iterates $x_0, \ldots, x_{T+1}$ generated by the algorithm* `GBoostAlg`$(\delta, \Delta_{\mathrm{in}}, x_{\mathrm{in}}, T, m)$*. Then in the event E_T, all the iterates $x_0, \ldots, x_T$ lie in $B_{\bar{\epsilon}}(\bar{x})$. Moreover, with probability at least $1 - (T+2)\exp\left(-\frac{m}{18}\right)$, the last iterate x_{T+1} satisfies*

$$f(x_{T+1}) - \min f \leq \delta\left(1 + \sum_{i=0}^{T}\frac{\lambda_i}{\mu+\lambda_{i-1}}\right).$$

Proof. As before, we verify that Algorithm 7 is an instantiation of Algorithm 2 with $p = \exp(-\frac{m}{18})$. More precisely, we prove by induction that for $j = 0, \ldots, T$, the iterate

x_j satisfies (3.4) and the inclusion $x_j \in B_{\bar{\epsilon}}(\bar{x})$ holds within the event E_j. To see the base case, observe that Lemma 2.2 guarantees that with probability $1-p$, the estimate holds:

$$\|x_0 - \bar{x}_0\| \leq 3\sqrt{\frac{2 \cdot \delta/9}{\mu}} = \epsilon_{-1}.$$

Moreover, in the event E_0, we have $\|x_0 - \bar{x}\| \leq \epsilon_{-1} \leq \bar{\epsilon}$. As an inductive hypothesis, suppose that (3.4) holds for the iterates $x_0, x_1, \ldots, x_{j-1}$ and that $x_0, x_1, \ldots, x_{j-1}$ lie in the ball $B_{\bar{\epsilon}}(\bar{x})$ in the event E_{j-1}. We first verify that x_j satisfies (3.4). To this end, suppose that the event E_{j-1} occurs. Then by the inductive assumption, the inclusion $x_{j-1} \in B_{\bar{\epsilon}}(\bar{x})$ holds. Therefore we deduce

$$f^{j-1}(x_{j-1}) - \min f^{j-1} \leq f(x_{j-1}) - f^* \leq M_{\bar{\epsilon}}.$$

Thus Lemma 2.2 guarantees that conditioned on E_{j-1} with probability $1-p$, the estimate holds:

$$\|x_j - \bar{x}_j\| \leq 3\sqrt{\frac{2 \cdot \delta/9}{\mu + \lambda_{j-1}}} = \epsilon_{j-1}.$$

We conclude that the estimate (3.4) holds for the iterate x_j, as needed.

Next, suppose that the event E_j holds. To prove the inclusion $x_j \in B_{\bar{\epsilon}}(\bar{x})$, observe that by strong convexity and the inequality (3.1), we have the estimate

$$\frac{\mu}{2}\|\bar{x}_j - \bar{x}\|^2 \leq f(\bar{x}_j) - f^* \leq f^{j-1}(\bar{x}_j) - f^* \leq \sum_{i=0}^{j-1} \frac{\lambda_i}{2}\|\bar{x}_i - x_i\|^2.$$

Rearranging and using the inductive assumption, we obtain the bound

$$\|\bar{x}_j - \bar{x}\| \leq \sqrt{\frac{2}{\mu}\sum_{i=0}^{j-1}\frac{\lambda_i}{2}\|\bar{x}_i - x_i\|^2} \leq \sqrt{\frac{2\delta}{\mu}\sum_{i=0}^{j-1}\frac{\lambda_i}{\mu + \lambda_{i-1}}}.$$

Consequently, the triangle inequality yields

$$\|x_j - \bar{x}\| \leq \|x_j - \bar{x}_j\| + \|\bar{x}_j - \bar{x}\| \leq \sqrt{\frac{2\delta}{\lambda_{j-1} + \mu}} + \sqrt{\frac{2\delta}{\mu}\sum_{i=0}^{j-1}\frac{\lambda_i}{\mu + \lambda_{i-1}}} \leq \bar{\epsilon},$$

as desired. This completes the induction. An application of Theorem 3.2 completes the proof. □

When using the proximal parameters $\lambda_i = \mu 2^i$, we obtain the following guarantee.

Corollary 5.5 (Efficiency of `GBoostAlg` with geometric decay)**.** *Fix an arbitrary point $x_{\text{in}} \in \mathbf{R}^d$ and let Δ_{in} be any upper bound $\Delta_{\text{in}} \geq f(x_{\text{in}}) - \min f$. Fix a target accuracy $\delta' > 0$ and probability of failure $p \in (0,1)$, and set the algorithm parameters*

$$T = \lceil \log_2(\tau) \rceil, \qquad m = \left\lceil 18 \ln\left(\frac{2+T}{p}\right)\right\rceil, \qquad \delta = \frac{\delta'}{2+2T}, \qquad \lambda_i = \mu 2^i.$$

Consider the iterates $x_0, \ldots, x_{T+1}$ generated by the algorithm $\texttt{GBoostAlg}(\delta, \Delta_{\rm in}, x_{\rm in}, T, m)$. *Then in the event E_T, all the iterates $x_1, x_2, \ldots, x_T$ lie in the ball $B_{\sqrt{2\delta'/\mu}}(\bar{x})$. Moreover, the last iterate x_{T+1} satisfies*

$$\mathbb{P}(f(x_{T+1}) - \min f \leq \delta') \geq 1 - p.$$

The total number of calls to $\texttt{Alg}(\cdot)$ *is*

$$\left\lceil 18 \ln\left(\frac{\lceil 2 + \log_2(\tau)\rceil}{p}\right)\right\rceil \lceil 2 + \log_2(\tau)\rceil.$$

To illustrate Corollaries 5.3 and (5.5), we now specialize the result to the setting when $\texttt{Alg}(\cdot)$ is the stochastic proximal gradient method and its accelerated variant. This application is only meant to be an illustration; indeed, `proxBoost` can be applied to other streaming algorithms, as well. For example, by exactly the same reasoning, one can couple `proxBoost` with variance reduced methods for minimizing finite sums of expectations [25].

Illustration: robust (accelerated) stochastic gradient methods

Consider the optimization problem

$$\min_{x \in \mathcal{X}}\ \varphi(x) + \psi(x)$$

where $\mathcal{X}$ is a closed convex set, the loss $\varphi\colon \mathbf{R}^d \to \mathbf{R}$ is μ-strongly convex and L-smooth, and the regularizer $\psi\colon \mathbf{R}^d \to \mathbf{R}$ is continuous and convex. We denote the condition number of this problem by $\tau = L/\mu$. We also define the function $f = \varphi + \psi(x) + \iota_{\mathcal{X}}$, where $\iota_{\mathcal{X}}$ is a function that is 0 on $\mathcal{X}$ and $+\infty$ off of it. Following the standard literature on streaming algorithms, we suppose that the only access to φ is through a stochastic gradient oracle. Namely, fix a probability space $(\Omega, \mathcal{F}, P)$ and let $G\colon \mathbf{R}^d \times \Omega \to \mathbf{R}$ be a measurable map satisfying

$$\mathbb{E}_z G(x, z) = \nabla\varphi(x) \qquad \text{and} \qquad \mathbb{E}_z\|G(x, z) - \nabla\varphi(x)\|^2 \leq \sigma^2.$$

We suppose that for any point x, we may sample $z \sim P$ and compute the vector $G(x, z)$, which serves as an unbiased estimator of the gradient $\nabla\varphi(x)$. The performance of standard numerical methods within this model of computation is judged by their sample complexity—the number of stochastic gradient evaluations $G(x, z)$ with $z \sim P$ required by the algorithm to produce an approximate minimizer of the problem.

Fix an initial point x_{in} and let $\Delta_{\rm in} > 0$ satisfy $\Delta_{\rm in} \geq f(x_0) - f^*$. It is well known that an appropriately modified proximal stochastic gradient method can generate a point x satisfying $\mathbb{E}f(x) - f^* \leq \epsilon$ with sample complexity

$$\mathcal{O}\left(\tau \log\left(\frac{\Delta_{\rm in}}{\epsilon}\right) + \frac{\sigma^2}{\mu\epsilon}\right). \tag{5.3}$$

The accelerated stochastic gradient method of [17, Multi-stage AC-SA, Proposition 7] and the simplified optimal algorithm of [24, Restarted Algorithm C, Corollary 9] have the substantially better sample complexity

$$\mathcal{O}\left(\sqrt{\tau}\log\left(\frac{\Delta_{\text{in}}}{\epsilon}\right)+\frac{\sigma^2}{\mu\epsilon}\right).$$

Clearly, we may use either of these two procedures as `Alg`$(\cdot)$ within the `proxBoost` framework.

Smooth setting. Suppose that $\psi \equiv 0$ and $\mathcal{X} = \mathbb{R}^d$. Then using Corollary 5.3, we deduce that the two resulting algorithms will find a point x satisfying

$$\mathbb{P}[f(x) - f^* \leq \epsilon] \geq 1 - p$$

with sample complexity

$$\mathcal{O}\left(\ln(\tau)\ln\left(\frac{\ln\tau}{p}\right)\cdot\left(\tau\ln\left(\frac{\Delta_{in}\ln(\tau)}{\epsilon}\vee\tau\right)+\frac{\sigma^2\ln(\tau)}{\mu\epsilon}\right)\right),$$

and

$$\mathcal{O}\left(\ln(\tau)\ln\left(\frac{\ln\tau}{p}\right)\cdot\left(\sqrt{\tau}\ln\left(\frac{\Delta_{in}\ln(\tau)}{\epsilon}\vee\tau\right)+\frac{\sigma^2\ln(\tau)}{\mu\epsilon}\right)\right),$$

for the unaccelerated and accelerated methods, respectively. Thus, `proxBoost` endows the stochastic gradient method and its accelerated variant with high confidence guarantees at an overhead cost that is only polylogarithmic in τ and logarithmic in $1/p$.

General setting. Let us return to the general case. Unlike the smooth setting, we have not yet specified how to perform the last step of `GBoostAlg`, that is how to obtain the point x_{T+1}. For this purpose, we will apply the Robust Stochastic Mirror Descent (RSMD) algorithm of [22, Section 6] to the problem of minimizing f^T. Roughly speaking, this algorithm will find an ϵ-optimal point in function value with probability $1-p$ using a similar number of samples, up to multiplication by $\log(1/p)$, as the unaccelerated stochastic gradient method. Recall that the smooth part of f^T, namely $\varphi + \frac{\mu 2^T}{2}\|\cdot - x_T\|^2$ is smooth and strongly convex with parameters $2L$ and $\mu + L$, respectively. Hence the sample complexity of this last step is dominated by $\frac{\sigma^2}{\epsilon L}\log(1/p)$, which is negligible compared to the overall cost of the previous iterations of `GBoostAlg`.

We now explain the application of RSMD more formally. Set δ, δ', λ_i, m, and T according to Corollary 5.5. Unlike the previous algorithms of this paper, RSMD requires an estimate on the distance of the initial iterate x_T to the minimizer $\bar{x}_{T+1}$. Using Corollary 5.5, we have such an estimate. Namely, within event E_T, we compute:

$$\frac{\mu+\lambda_T}{2}\|x_T - \bar{x}_{T+1}\|^2 \leq f^T(x_T) - f^T(x^*) \leq f^T(x_T) - f^* \leq M_{\sqrt{2\delta'/\mu}}.$$

Rearranging, we deduce

$$\|x_T - \bar{x}_{T+1}\| \leq r_0 := \sqrt{\frac{M_{\sqrt{2\delta'/\mu}}}{\mu + \lambda_T}}.$$

We will now apply [22, Theorem 3] with appropriate parameter settings. Fix a real number $\gamma > 1/2$ and define N to be smallest integer satisfying

$$N \geq \max\left\{4\log_2\left(\frac{2Lr_0}{C\delta}\right), \frac{2\gamma}{C'}\ln\left(\frac{2Lr_0^2}{C\delta}\right), \frac{\sigma^2\gamma}{CL\delta}\right\},$$

where the constants C and C' are the same as the ones appearing in [22, Theorem 3]. Without loss of generality we may suppose that the last term in the maximum is dominant and that $N = \frac{\sigma^2\gamma}{CL\delta}$. Then [22, Theorem 3] guarantees that the procedure will output a point y satisfying

$$\mathbb{P}[f^T(y) - \min f^T \leq \delta] \geq 1 - C\log\left(\frac{4LC'r_0^2}{C\delta}\right)\exp(-\gamma),$$

with the number of samples that scales as $\frac{\sigma^2\gamma}{L\delta}\cdot\log\left(\frac{4L^2C'r_0^2N}{\sigma^2\gamma}\right)$. Therefore, it follows that we may obtain a point y satisfying

$$\mathbb{P}[f^T(y) - \min f^T \leq \delta] \geq 1 - p,$$

using on the order of

$$\frac{\sigma^2\ln(\tau)}{L\delta'}\log\left(\frac{Lr_0^2\ln(\tau)}{\delta'}\right)\log\left(\frac{\log(\frac{Lr_0^2\ln(\tau)}{\delta'})}{p}\right)$$

samples. Therefore, appealing to Corollary 5.5 we deduce the following. If within $\texttt{GBoostAlg}(\delta, \Delta_{\text{in}}, x_{\text{in}}, T, m)$, we use the proximal accelerated stochastic gradient method as $\texttt{Alg}(\cdot)$ to generate $x_0, x_1, \ldots, x_T$ and use RSMD to generate x_{T+1}, then we can be sure that the returned point x_{T+1} satisfies $f(x_{T+1}) - f^* \leq \delta'$ with probability $1-p$. Moreover the number of samples used is on the order of

$$\mathcal{O}\Bigg(\ln(\tau)\ln\Big(\frac{\ln\tau}{p}\Big)\Bigg(\sqrt{\tau}\ln\left(\frac{\Delta_{in}\ln(\tau)}{\delta'} \vee \frac{M_{\sqrt{2\delta'/\mu}}\ln(\tau)}{\delta'}\right) + \frac{\sigma^2\ln(\tau)}{\mu\delta'}\Bigg) + \\ + \frac{\sigma^2\ln(\tau)}{L\delta'}\log\left(\frac{M_{\sqrt{2\delta'/\mu}}\ln(\tau)}{\delta'}\right)\log\left(\frac{\log\left(\frac{M_{\sqrt{2\delta'/\mu}}\ln(\tau)}{\delta'}\right)}{p}\right)\Bigg).$$

Thus, GBoostAlg combined with RSMD in the last step endows the accelerated stochastic proximal gradient method with high confidence guarantees at an overhead cost that is only polylogarithmic in τ, $1/\delta'$, $M_{\sqrt{2\delta'/\mu}}$, and logarithmic in $1/p$.

References

[1] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. *J. Comput. System Sci.*, 58(1, part 2):137–147, 1999. Twenty-eighth Annual ACM Symposium on the Theory of Computing (Philadelphia, PA, 1996).

[2] H. Asi and J.C. Duchi. Stochastic (approximate) proximal point methods: Convergence, optimality, and adaptivity. *arXiv:1810.05633*, 2018.

[3] H. Asi and J.C. Duchi. The importance of better models in stochastic optimization. *arXiv:1903.08619*, 2019.

[4] N.S. Aybat, A. Fallah, M. Gurbuzbalaban, and A. Ozdaglar. A universally optimal multistage accelerated stochastic gradient method. *arXiv:1901.08022*, 2019.

[5] P.L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[6] A. Beck. *First-order methods in optimization*, volume 25. SIAM, 2017.

[7] S. Bubeck, N. Cesa-Bianchi, and G. Lugosi. Bandits with heavy tail. *IEEE Transactions on Information Theory*, 59(11):7711–7717, 2013.

[8] O. Catoni. Challenging the empirical mean and empirical variance: a deviation study. *Ann. Inst. Henri Poincaré Probab. Stat.*, 48(4):1148–1185, 2012.

[9] Y. Chen, L. Su, and J. Xu. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 1(2):44, 2017.

[10] D. Davis and D. Drusvyatskiy. Stochastic model-based minimization of weakly convex functions. *SIAM Journal on Optimization*, 29(1):207–239, 2019.

[11] D. Davis and B. Grimmer. Proximally guided stochastic method for nonsmooth, nonconvex problems. *arXiv:1707.03505*, 2017.

[12] J.C. Duchi and F. Ruan. Stochastic methods for composite and weakly convex optimization problems. *SIAM Journal on Optimization*, 28(4):3229–3259, 2018.

[13] R. Frostig, R. Ge, S.M. Kakade, and A. Sidford. Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015.

[14] R. Ge, S.M. Kakade, R. Kidambi, and P. Netrapalli. The step decay schedule: A near optimal, geometrically decaying learning rate procedure. *arXiv:1904.12838*, 2019.

[15] S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization I: A generic algorithmic framework. *SIAM J. Optim.*, 22(4):1469–1492, 2012.

[16] S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, ii: shrinking procedures and optimal algorithms. *SIAM Journal on Optimization*, 23(4):2061–2089, 2013.

[17] S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, II: Shrinking procedures and optimal algorithms. *SIAM J. Optim.*, 23(4):2061–2089, 2013.

[18] J.L. Goffin. On convergence rates of subgradient optimization methods. *Math. Programming*, 13(3):329–347, 1977.

[19] D. Hsu and S. Sabato. Loss minimization and parameter estimation with heavy tails. *The Journal of Machine Learning Research*, 17(1):543–582, 2016.

[20] M.R. Jerrum, L.G. Valiant, and V.V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoret. Comput. Sci.*, 43(2-3):169–188, 1986.

[21] E. Joly, G. Lugosi, and R.I. Oliveira. On the estimation of the mean of a random vector. *Electronic Journal of Statistics*, 11(1):440–451, 2017.

[22] A. Juditsky, A. Nazin, A. Nemirovsky, and A. Tsybakov. Algorithms of robust stochastic optimization based on mirror descent method. *arXiv:1907.02707*, 2019.

[23] A. Juditsky and Y. Nesterov. Deterministic and stochastic primal-dual subgradient algorithms for uniformly convex minimization. *Stoch. Syst.*, 4(1):44–80, 2014.

[24] A. Kulunchakov and J. Mairal. Estimate sequences for stochastic composite optimization: Variance reduction, acceleration, and robustness to noise. *arXiv:1901.08788*, 2019.

[25] A. Kulunchakov and J. Mairal. A generic acceleration framework for stochastic composite optimization. *arXiv:1906.01164*, 2019.

[26] G. Lan and Y. Zhou. An optimal randomized incremental gradient method. *Math. Program.*, 171(1-2, Ser. A):167–215, 2018.

[27] H. Lin, J. Mairal, and Z. Harchaoui. A universal catalyst for first-order optimization. In *Advances in Neural Information Processing Systems*, pages 3366–3374, 2015.

[28] G. Lugosi and S. Mendelson. Risk minimization by median-of-means tournaments. *arXiv:1608.00757*, 2016.

[29] G. Lugosi and S. Mendelson. Sub-gaussian estimators of the mean of a random vector. *The Annals of Statistics*, 47(2):783–794, 2019.

[30] B. Martinet. Régularisation d'inéquations variationnelles par approximations successives. *Rev. Française Informat. Rech. Opérationnelle*, 4(Sér. R-3):154–158, 1970.

[31] B. Martinet. Détermination approchée d'un point fixe d'une application pseudo-contractante. Cas de l'application prox. *C. R. Acad. Sci. Paris Sér. A-B*, 274:A163–A165, 1972.

[32] S. Minsker. Geometric median and robust estimation in Banach spaces. *Bernoulli*, 21(4):2308–2335, 2015.

[33] J.-J. Moreau. Proximité et dualité dans un espace hilbertien. *Bull. Soc. Math. France*, 93:273–299, 1965.

[34] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM J. Optim.*, 19(4):1574–1609, 2008.

[35] A.S. Nemirovsky and D.B. Yudin. *Problem complexity and method efficiency in optimization*. A Wiley-Interscience Publication. John Wiley & Sons, Inc., New York, 1983. Translated from the Russian and with a preface by E. R. Dawson, Wiley-Interscience Series in Discrete Mathematics.

[36] Y. Nesterov. *Lectures on convex optimization*, volume 137 of *Springer Optimization and Its Applications*. Springer, Cham, 2018. Second edition of [MR2142598].

[37] R.T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM J. Control Optimization*, 14(5):877–898, 1976.

[38] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[39] S. Shalev-Shwartz, O. Shamir, N. Srebro, and K. Sridharan. Stochastic convex optimization. In *COLT*, 2009.

[40] S. Shalev-Shwartz and T. Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. *Math. Program.*, 155(1-2, Ser. A):105–145, 2016.

[41] Y. Xu, Q. Lin, and T. Yang. Accelerated stochastic subgradient methods under local error bound condition. *arXiv:1607.01027*, 2016.

[42] T. Yang, Y. Yan, Z. Yuan, and R. Jin. Why does stagewise training accelerate convergence of testing error over SGD? *arXiv:1812.03934*, 2018.

[43] D. Yin, Y. Chen, R. Kannan, and P. Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5650–5659, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.