Anytime Guarantees under Heavy-Tailed Data

Matthew J. Holland

Osaka University
matthew-h@ar.sanken.osaka-u.ac.jp

Abstract

Under data distributions which may be heavy-tailed, many stochastic gradient-based learning algorithms are driven by feedback queried at points with almost no performance guarantees on their own. Here we explore a modified "anytime online-to-batch" mechanism which for smooth objectives admits high-probability error bounds while requiring only lower-order moment bounds on the stochastic gradients. Using this conversion, we can derive a wide variety of "anytime robust" procedures, for which the task of performance analysis can be effectively reduced to regret control, meaning that existing regret bounds (for the bounded gradient case) can be robustified and leveraged in a straightforward manner. As a direct takeaway, we obtain an easily implemented stochastic gradient-based algorithm for which all queried points formally enjoy sub-Gaussian error bounds, and in practice show noteworthy gains on real-world data applications.

Introduction

The ultimate goal of many learning tasks can be formulated as a minimization problem:

$$\min_{h} \mathrm{R}(h), \text{ s.t. } h \in \mathcal{H}. \tag{1}$$

What characterizes this as a learning problem is that R (henceforth called the *true objective*) is *unknown* to the learner, who must choose from the hypothesis class $\mathcal{H}$ a final candidate based only on incomplete and noisy (stochastic) feedback related to R (Haussler 1992; Vapnik 1999). One of the most ubiquitous and well-studied feedback mechanisms is the *stochastic gradient oracle* (Hazan 2016; Nemirovsky and Yudin 1983; Shalev-Shwartz 2012), in which the learner generates a sequence of candidates (h_t) based on a sequence of random sub-gradients (G_t), which are unbiased in the following sense:

$$\mathbf{E}\left[G_t \,|\, G_{[t-1]}\right] \in \partial \mathrm{R}(h_t), \text{ for all } t \geq 1. \tag{2}$$

Here $\partial \mathrm{R}(h)$ denotes the sub-differential of R evaluated at h, and we denote sub-sequences by $G_{[t]} \coloneqq (G_1, \ldots, G_t)$.[1]

[1]More strictly speaking, for each t, this inclusion holds almost surely over the random draw of $G_{[t]}$, and the conditional expectation is that of G_t conditioned on the sigma-algebra generated by $G_{[t-1]}$. See Ash and Doléans-Dade (2000, Ch. 5–6) for additional background on probabilistic foundations.

Our problem of interest is that of efficiently minimizing $\mathrm{R}(\cdot)$ over $\mathcal{H}$ when the noisy feedback is *potentially heavy-tailed*, i.e., for all steps t, it is unknown whether the distribution of G_t is congenial in the sub-Gaussian sense, or heavy-tailed in the sense of having infinite or undefined higher-order moments (Chen, Su, and Xu 2017). By "efficiently," we mean procedures with performance guarantees (high-probability error bounds) on par with the case in which the learner knows *a priori* that the feedback is sub-Gaussian (Devroye et al. 2016; Nazin et al. 2019).

Recently, notable progress has been made on this front, with a common theme of making principled modifications (e.g., truncation, data splitting + validation, etc.) to the the raw feedback (G_t) before passing it to a more traditional stochastic gradient-based update, to achieve sub-Gaussian bounds (with optimal dependence on the confidence level) while assuming just finite variance (Davis et al. 2019; Gorbunov, Danilova, and Gasnikov 2020; Nazin et al. 2019). Here we focus on two key limitations to the current state of the art: (a) many robust learning algorithms only have such guarantees when R is *strongly convex* (Chen, Su, and Xu 2017; Davis et al. 2019; Holland and Ikeda 2019); (b) without strong convexity, sub-Gaussian guarantees are unavailable for the iterates (h_t) being queried in (2), only for a running average of these iterates (Gorbunov, Danilova, and Gasnikov 2020; Nazin et al. 2019). While there exist general-purpose "anytime" online-to-batch conversions to ensure that the points being queried have guarantees (Cutkosky 2019), even the most refined conversions either require bounded gradients or are only in expectation (Joulani et al. 2020), meaning that under potentially heavy-tailed gradients, a direct anytime conversion based on existing results fails to achieve the desired guarantees.

In this paper, in order to address the issues described above, we introduce a modified mechanism for making the "anytime" conversion (Algorithm 1), which is both easy to implement and robust to the underlying data distribution. More concretely, assuming only that R is convex and smooth, and that raw gradients have finite variance, we obtain martingale concentration guarantees for truncated gradients queried at a moving average (Lemma 2), which lets us reduce the problem of obtaining error bounds to that of regret control (section), substantially broadening the domain to which the anytime conversion of Cutkosky (2019) can be

applied. Regret control for online learning algorithms (under *bounded* gradients) is a well-studied problem, and in section we show that existing well-known regret bounds can be readily modified to utilize the control offered by Lemma 2. In particular, we look at vanilla FTRL (Lemma 4), mirror descent (Lemma 5), and AO-FTRL (Theorem 8), giving us "anytime robust" analogues to results given in expectation by Joulani et al. (2020). As a natural takeaway, we obtain a stochastic gradient-based procedure (section) for which *all queried points* have sub-Gaussian error bounds (Corollary 7), a methodological improvement over the averaging scheme of Nazin et al. (2019), which we also empirically demonstrate has substantial practical benefits (section). Our results are stated with a high degree of generality (works on any reflexive Banach space), and taken together are suggestive of an appealing general-purpose learning strategy.

Outline of the paper In section we set out our basic notation, describe the basic principle underlying anytime conversions and highlight the impact it has on the excess risk. Our general-purpose robustified anytime algorithm design is described in detail in section , where we establish control over the gradient estimation error, and show how this leads to a robust analogue of existing regret-based excess error bounds, valid under potentially heavy-tailed gradients. We then put these general results to work in section , where we illustrate how it is straightforward to apply our techniques to many important classes of online learning algorithms to obtain variants which are both anytime and robust. In section , we carry out empirical investigations using real-world benchmark datasets to explore the practical utility of anytime feedback. Finally, we note that formal definitions of key concepts, support lemmas, and detailed proofs of all results in the main text are provided in the appendix (supplementary materials).

Preliminaries

Terms and Notation

The underlying space For the underlying hypothesis class $\mathcal{H}$, we shall assume $\mathcal{H} \subset \mathcal{V}$, where $(\mathcal{V}, \|\cdot\|)$ is a normed linear space. For any normed space $(\mathcal{V}, \|\cdot\|)$, we will denote by $\mathcal{V}^*$ the usual dual of $\mathcal{V}$, namely the set of all continuous linear functionals on $\mathcal{V}$. As is traditional, the norm for the dual is defined $\|f\|_* \coloneqq \sup\{a : |f(v)| \leq a\|v\|, v \in \mathcal{V}\}$ for $f \in \mathcal{V}^*$. We denote the distance from a point v to a set $A \subset \mathcal{V}$ by $\operatorname{dist}(v; A) \coloneqq \inf\{\|v - u\| : u \in A\}$. We use the notation $\langle \cdot, \cdot \rangle$ to represent the coupling function between $\mathcal{V}$ and $\mathcal{V}^*$, namely $\langle v^*, v \rangle \coloneqq v^*(v)$ for each $v^* \in \mathcal{V}^*$ and all $v \in \mathcal{V}$; when $\mathcal{V}$ is a Hilbert space this coincides with the usual inner product. We denote the extended real line by $\overline{\mathbb{R}}$.

Convexity and smoothness We say that a function $f : \mathcal{V} \to \overline{\mathbb{R}}$ is *convex* if for all $0 < \alpha < 1$ and $u, v \in \mathcal{V}$, we have $f(\alpha u + (1-\alpha)v) \leq \alpha f(u) + (1-\alpha)f(v)$. The *effective domain* of f is defined $\operatorname{dom} f \coloneqq \{u \in \mathcal{V} : f(u) < \infty\}$. A convex function $f : \mathcal{V} \to \overline{\mathbb{R}}$ is said to be *proper* if $-\infty < f$ and $\operatorname{dom} f \neq \emptyset$. For any proper convex function $f : \mathcal{V} \to \overline{\mathbb{R}}$, the *sub-differential* of f at $h \in \mathcal{V}$ is $\partial f(h) \coloneqq \{v^* \in \mathcal{V}^* : f(u) - f(h) \geq \langle v^*, u - h \rangle, u \in \mathcal{V}\}$. For readability, we will sometimes make statements involving multi-valued functions; for example, the statement "$\langle \partial f(h), u \rangle = a$," is equivalent to the statement "$\langle v^*, u \rangle = a$ for all $v^* \in \partial f(h)$." When we say a certain point h^* is a *stationary point* of f on $\mathcal{H}$, we mean that $h^* \in \mathcal{H}$ and $0 \in \partial f(h^*)$. If the convex function f happens to be (Gateaux) differentiable at some $h \in \mathcal{H}$, then the sub-differential contains a unique element, $\partial f(h) = \{\nabla f(h)\}$, the *gradient* of f at h. When we say that f is λ-*smooth* on some open convex set $U \subset \mathcal{V}$, we mean that $\|\nabla f(h) - \nabla f(h')\|_* \leq \lambda\|h - h'\|$ for all $h, h' \in U$. For any sub-differentiable function f, we write $\mathrm{D}_f(u; v) \coloneqq f(u) - f(v) - \langle \partial f(v), u - v \rangle$; when f happens to be convex and differentiable, this becomes the usual *Bregman divergence* induced by f.

Miscellaneous notation For indexing purposes, we denote the set of all positive integers no greater than k by $[k] \coloneqq \{1, \ldots, k\}$. We denote $\alpha_{1:t} \coloneqq \sum_{i=1}^{t} \alpha_i$ for any integer $t \geq 1$, using the convention $\alpha_{1:0} \coloneqq 0$ as needed. We also denote sub-sequences in a similar fashion, with $\alpha_{[t]} \coloneqq (\alpha_1, \ldots, \alpha_t)$; this applies not only to (α_t), but also (h_t), (G_t) and other sequences used throughout the paper. Indicator functions (i.e., Bernoulli random variables) are typically denoted as $\mathrm{I}\{\texttt{event}\}$.

Anytime Conversions

As preparation, we start with almost no assumptions on the learning algorithm or feedback-generating process. Let (h_t) be an arbitrary sequence of candidates, henceforth referred to as the *ancillary iterates*. Letting (α_t) be a sequence of positive weights, we consider the corresponding *main iterates* $(\bar{h}_t)$, defined for all $t \geq 1$ as

$$\bar{h}_t = \texttt{Weighting}\left[h_t; h_{[t-1]}\right] \coloneqq \frac{\sum_{i=1}^{t} \alpha_i h_i}{\alpha_{1:t}}. \quad (3)$$

As a starting point, we note that the excess error of the weighted main iterates can be expressed in a convenient fashion.

Lemma 1 (Anytime lemma)**.** *Let $\mathcal{V}$ be a linear space, and let $\mathrm{R} : \mathcal{V} \to \overline{\mathbb{R}}$ be sub-differentiable. Let (h_t) be an arbitrary sequence of $h_t \in \operatorname{dom} \mathrm{R}$, and let $(\bar{h}_t)$ be generated via (3). Then we have*

$$\mathrm{R}(\bar{h}_T) - \mathrm{R}(h^*) = \frac{1}{\alpha_{1:T}} \left[\sum_{t=1}^{T} \alpha_t \left[\langle \partial\, \mathrm{R}(\bar{h}_t), h_t - h^* \rangle - \mathrm{D}_{\mathrm{R}}(h^*; \bar{h}_t) \right] - \sum_{t=1}^{T-1} \alpha_{1:t}\, \mathrm{D}_{\mathrm{R}}(\bar{h}_t; \bar{h}_{t+1}) \right]$$

for any reference point $h^* \in \operatorname{dom} \mathrm{R}$ and $T \geq 1$.

The above equality is a slight generalization of the anytime online-to-batch inequality introduced by Cutkosky (2019) and sharpened by Joulani et al. (2020); it follows by direct manipulations utilizing little more than the definition of D_{R}. The key point of Lemma 1 is that we can obtain control over the main iterates $\mathrm{R}(\bar{h}_t)$ using an ideal quantity that

Algorithm 1: Anytime robust online-to-batch conversion.

inputs: Weights (α_t), thresholds (c_t), algorithm $\mathcal{A}$, initial point h_1, max iterations T.
Initialize $\bar{h}_1 = h_1$.
for $t \in [T-1]$ **do**
  Obtain stochastic gradient G_t at $\bar{h}_t$, satisfying (4).
  Set $\overline{G}_t = \texttt{Process}[G_t; c_t]$ following (6)–(7).
  Ancillary update: $h_{t+1} = \mathcal{A}(h_t)$.
  Main update:
  $\bar{h}_{t+1} = \texttt{Weighting}[h_{t+1}; h_{[t]}]$, as in (3).
end for
return: $\bar{h}_T$.

depends directly on $\bar{h}_t$, rather than simply h_t, as is typical of traditional online-to-batch conversions (Cesa-Bianchi, Conconi, and Gentile 2004). This is important because it opens the door to new stochastic feedback processes, driven by the *main* iterates, rather than the ancillary ones. In other words, we want feedback that provides an estimate of some element of $\partial \mathrm{R}(\bar{h}_t)$, rather than $\partial \mathrm{R}(h_t)$. When R is convex, we have $\mathrm{D}_{\mathrm{R}} \geq 0$, and the subtracted terms can be utilized to sharpen our guarantees once we have regret bounds, as will be discussed in the technical appendices.

Anytime Robust Algorithm Design

In Algorithm 1, we give a summary of the modified online-to-batch conversion that we utilize throughout the rest of the paper. Essentially, we start with an arbitrary online learning algorithm $\mathcal{A}$, query the potentially heavy-tailed stochastic feedback after averaging the iterates, and process the raw gradients in a robust fashion before updating. In the following paragraphs, we describe the details of these steps.

Raw feedback process Let (G_t) denote a sequence of stochastic gradients $G_t \in \mathcal{V}^*$, which are conditionally unbiased in the sense that we have

$$\mathbf{E}_{[t-1]}\, G_t \coloneqq \mathbf{E}\left[G_t \,|\, G_{[t-1]}\right] \in \partial \mathrm{R}(\bar{h}_t) \tag{4}$$

for all $t \geq 1$, recalling our notation $G_{[t]} \coloneqq (G_1, \ldots, G_t)$. We emphasize to the reader that (4) differs from the traditional assumption (2) in terms of the points at which the sub-differential is being evaluated ($\bar{h}_t$ rather than h_t). As is traditional in the literature (Nazin et al. 2019; Nguyen et al. 2018), we shall also assume a uniform bound on the conditional variance, namely that for all $t \geq 1$, we have

$$\mathbf{E}_{[t-1]} \left\| G_t - \mathbf{E}_{[t-1]}\, G_t \right\|_*^2 \leq \sigma^2 < \infty. \tag{5}$$

We will not assume anything else about the underlying distribution of (G_t); as such, the gradients clearly may be unbounded or heavy-tailed in the sense of having infinite or undefined higher-order moments. In this setting, while one could naively use the raw sequence (G_t) as-is, since we have made extremely weak assumptions on the underlying distribution, it is always possible for heavy-tailed data to severely destabilize the learning process (Brownlees, Joly, and Lugosi 2015; Chen, Su, and Xu 2017; Lecué, Lerasle, and Mathieu 2018). As such, it is desirable to process the raw gradients in a statistically principled manner, such that the processed output provides useful feedback to be passed directly to $\mathcal{A}$.

Overall design for robust feedback A simple and popular approach to deal with heavy-tailed random vectors is to use norm-based truncation (Catoni and Giulini 2017; Nazin et al. 2019). As with Nazin et al. (2019), we process the raw gradients as follows:

$$\texttt{Process}\,[G; c] \coloneqq \begin{cases} \widetilde{g}, & \text{if } \|G - \widetilde{g}\|_* > c \\ G, & \text{else.} \end{cases} \tag{6}$$

Here $c > 0$ is a threshold, the point $\widetilde{g} \in \mathcal{V}^*$ used in this sub-routine is an "anchor" in the dual space, associated with some "primal anchor" $\widetilde{h} \in \mathcal{H}$ assumed to satisfy

$$\mathbf{P}\left\{ \mathrm{dist}(\widetilde{g}; \partial \mathrm{R}(\widetilde{h})) > \widetilde{\varepsilon}\sigma \right\} \leq \delta. \tag{7}$$

We discuss settings of the anchor points and the critical thresholds in the following paragraphs. To concisely summarize the robust feedback that we use, instead of naively using (G_t) as feedback for $\mathcal{A}$, we will pass $(\overline{G}_t)$, defined by $\overline{G}_t \coloneqq \texttt{Process}\,[G_t; c_t]$, based on a sequence of thresholds (c_t). The anchors $\widetilde{g}$ and $\widetilde{h}$ remain fixed throughout the learning process.

Determining the anchor points Let us first assume a primal anchor $\widetilde{h} \in \mathcal{H}$ is given; the key requirement we need to fulfill is the equation (7), which asks that the dual anchor $\widetilde{g}$ we choose be $\widetilde{\varepsilon}\sigma$-close to some sub-gradient of R with probability no less than $1 - \delta$. To achieve this under our weak assumptions on the stochastic gradient distribution is straightforward using modern robust mean estimation techniques. Perhaps the simplest example is generalized median-of-means, as follows. Under our assumption 4, we can obtain m unbiased stochastic gradients $G_1, \ldots, G_m$, such that $\mathbf{E}_{[i-1]}\, G_i \in \partial \mathrm{R}(\widetilde{h})$ for all $i \in [m]$. Assuming k divides m, partition these points into k equal-sized subsets, compute the empirical means $\overline{G}_1, \ldots, \overline{G}_k$ for each subset, and set the dual anchor as $\widetilde{g} = \texttt{GeoMed}\{\overline{G}_1, \ldots, \overline{G}_k\}$, where the sub-routine GeoMed refers to the geometric median of these points, a convex program for which many efficient algorithms are known (Vardi and Zhang 2000; Cohen et al. 2016). It is well known that under this $\widetilde{g}$ setting, if we set $k = \lceil 8 \log(\delta^{-1}) \rceil$, then (7) holds with

$$\widetilde{\varepsilon} \leq 4\sqrt{\frac{(1 + 8\log(\delta^{-1}))}{m}}.$$

See Lugosi and Mendelson (2019) for additional background on this and other robust vector mean estimators. Finally, we emphasize that the choice of primal anchor $\widetilde{h} \in \mathcal{H}$ is arbitrary, in that the theoretical guarantees to be discussed shortly hold regardless of which $\widetilde{h}$ we choose, given that the dual anchor is selected in the manner just described. In practice, the difference between "good" and "bad" primal anchors is in the bias incurred by the truncation (6). More concretely, considering the ideal case where $\widetilde{h}$ is a stationary (interior) point, we have $0 \in \partial \mathrm{R}(\widetilde{h})$ and thus thresholding

$\|G\|_*$ is sufficient. On the other hand, when $\widetilde{h}$ is far from a stationary point, $\widetilde{g}$ may have a large norm. If the learning process finds a good $\bar{h}_t$ such that G_t tends to be small, then $\|G_t - \widetilde{g}\|_*$ may be large, even though we probably want to use G_t as-is without truncation. This bias must be dealt with by proper threshold settings, which we describe in the next paragraph.

Threshold settings under smooth objectives Let us further assume that R is λ-smooth, still leaving $\mathcal{A}$ abstract. In this case, the sub-differential is simply $\partial \mathrm{R}(h) = \{\nabla \mathrm{R}(h)\}$, and so the error that we focus on is naturally that of the approximation $\overline{G}_t \approx \nabla \mathrm{R}(\bar{h}_t)$, for $t \in [T]$. With $(\overline{G}_t)$ generated as described in Algorithm 1, direct inspection shows us that

$$\overline{G}_t - \nabla \mathrm{R}(\bar{h}_t) = (G_t - \widetilde{g})(1 - \mathrm{I}_t) + \widetilde{g} - \nabla \mathrm{R}(\bar{h}_t) \quad (8)$$

where I_t is the Bernoulli random variable defined $\mathrm{I}_t \coloneqq \mathrm{I}\{\|G_t - \widetilde{g}\|_* > c_t\}$. The right-hand side of (8) has two terms we need to control. The first term is clearly bounded above by c_t, considering the truncation event. As for the second term, a smooth risk makes it easy to establish control in primal distance terms. More explicitly, we have

$$\begin{aligned}
&\|\widetilde{g} - \nabla \mathrm{R}(\bar{h}_t)\|_* \\
&\quad \le \|\widetilde{g} - \nabla \mathrm{R}(\widetilde{h})\|_* + \|\nabla \mathrm{R}(\widetilde{h}) - \nabla \mathrm{R}(\bar{h}_t)\|_* \\
&\quad \le \widetilde{\varepsilon}\sigma + \lambda \|\widetilde{h} - \bar{h}_t\| \qquad (9)
\end{aligned}$$

with probability no less than $1 - \delta$, where the latter inequality follows from λ-smoothness and the anchor property (7). Taking (8) and (9) together, we readily obtain

$$\begin{aligned}
\|\overline{G}_t - \nabla \mathrm{R}(\bar{h}_t)\|_* &\le \|G_t - \widetilde{g}\|_*(1 - \mathrm{I}_t) + \|\widetilde{g} - \nabla \mathrm{R}(\bar{h}_t)\|_* \\
&\le c_t + \widetilde{\varepsilon}\sigma + \lambda \|\widetilde{h} - \bar{h}_t\| \qquad (10)
\end{aligned}$$

on an event of probability at least $1 - \delta$. This inequality suggests an obvious choice for the threshold c_t that keeps the preceding upper bound tidy:

$$c_t = \widetilde{\varepsilon}\sigma + \lambda \|\widetilde{h} - \bar{h}_t\| + c_0, \quad t \in [T]. \qquad (11)$$

Here $c_0 > 0$ is positive parameter that is used to control the degree of bias incurred due to truncation.

Estimation error under smooth objectives Using the thresholding strategy described in the preceding paragraph, one can obtain sub-linear bounds on the weighted gradient error terms, as the next result shows.

Lemma 2. *Let* R *be convex and λ-smooth. Let $\mathcal{H} \subset \operatorname{dom} \mathrm{R}$ be convex with diameter $\Delta < \infty$. Given confidence parameter $0 < \delta < 1$ and iterations $T \ge \log(\delta^{-1})(\lceil \widetilde{\varepsilon}\sigma \rceil)^2$, running Algorithm 1 with thresholds (c_t) as in (11) with $c_0 = \max\{\lambda\Delta, \sigma\sqrt{T/\log(\delta^{-1})}\} + \widetilde{\varepsilon}\sigma$, and weights (α_t) such that $\mathbf{E}_{[t-1]}\, \alpha_t = \alpha_t$ almost surely, it follows that*

$$\begin{aligned}
&\sum_{t=1}^{T} \alpha_t \sup_{h,h' \in \mathcal{H}} \left[\langle \overline{G}_t - \nabla \mathrm{R}(\bar{h}_t), h - h' \rangle\right] \\
&\qquad\qquad \le \max\{q_\delta(T), r_\delta(T)\}
\end{aligned}$$

with probability no less than $1 - 2\delta$, where we have defined

$$q_\delta(T) \coloneqq 2\Delta\sigma\sqrt{2\log(\delta^{-1})}\left[\frac{\alpha_{1:T}}{\sqrt{T}} + \sqrt{\sum_{t=1}^{T}\alpha_t^2} + 2\left(\max_{t\in[T]}\alpha_t\right)\right]$$

$$r_\delta(T) \coloneqq 2\lambda\Delta^2\log(\delta^{-1})\left[\frac{\alpha_{1:T}}{T} + \sqrt{\frac{1}{T}\sum_{t=1}^{T}\alpha_t^2} + 2\sqrt{2}\left(\max_{t\in[T]}\alpha_t\right)\right].$$

The main benefit of this lemma is that it holds under very weak assumptions on the stochastic gradients. The main limitations are that the feasible set has a finite diameter, and prior knowledge of T and other factors are used for thresholding.

A general strategy Let us define the regret incurred by Algorithm 1 after T steps by

$$\mathrm{Regret}(T; \mathcal{A}) \coloneqq \sum_{t=1}^{T} \alpha_t \langle \overline{G}_t, h_t - h^* \rangle, \qquad (12)$$

where the reference point $h^* \in \operatorname{dom} \mathrm{R}$ is left implicit in the notation. This weighted linear regret is somewhat special since the losses (i.e., $h \mapsto \alpha_t\langle \overline{G}_t, h\rangle$) are *evaluated* on the ancillary sequence (h_t), but they are *defined* in terms of potentially biased stochastic gradients which depend on the main sequence $(\bar{h}_t)$. With this notion of regret in hand, note that from Lemma 1, we immediately have the following expression:

$$\begin{aligned}
&\mathrm{R}(\bar{h}_T) - \mathrm{R}(h^*) = \\
&\quad \frac{1}{\alpha_{1:T}}\left[\mathrm{Regret}(T; \mathcal{A}) + \sum_{t=1}^{T} \alpha_t \langle \overline{G}_t - \nabla \mathrm{R}(\bar{h}_t), h^* - h_t \rangle \right. \\
&\qquad \left. - \sum_{t=1}^{T} \alpha_t \, \mathrm{D}_{\mathrm{R}}(h^*; \bar{h}_t) - \sum_{t=1}^{T-1} \alpha_{1:t} \, \mathrm{D}_{\mathrm{R}}(\bar{h}_t; \bar{h}_{t+1})\right]. \qquad (13)
\end{aligned}$$

This inequality offers us a nice starting point for analyzing a wide class of "anytime robust algorithms," since the second sum can clearly be controlled using Lemma 2. It just remains to seek out regret bounds for different choices of the underlying algorithm $\mathcal{A}$ which are sub-linear, up to error terms that are amenable to Lemma 2. We give several concrete examples in the next section. To close this section, by combining our notion of regret with the preceding lemma, we can obtain a "robust" analogue of Cutkosky (2019, Thm. 1), which is valid under unbounded, heavy-tailed stochastic gradients.

Corollary 3. *Under the assumptions of Lemma 2, for any reference point $h^* \in \mathcal{H}$, we have*

$$\begin{aligned}
&\mathrm{R}(\bar{h}_T) - \mathrm{R}(h^*) \\
&\quad \le \frac{1}{\alpha_{1:T}}\left[\mathrm{Regret}(T; \mathcal{A}) + \max\{q_\delta(T), r_\delta(T)\} - B_T\right]
\end{aligned}$$

with probability no less than $1 - 2\delta$, where $B_T \ge 0$ denotes the sum of all the Bregman divergence terms given in (13).

Anytime Robust Learning Algorithms

Thus far, the underlying algorithm object $\mathcal{A}$ used in Algorithm 1 has been left abstract. In this section, we illustrate how (13) can be utilized for important classes of algorithms, by obtaining regret bounds that are sub-linear up to error terms that can be controlled using Lemma 2. Our running assumptions are that $(\mathcal{V}, \|\cdot\|)$ is a reflexive Banach space, $\mathcal{H} \subseteq \mathcal{V}$ is convex and closed, R is sub-differentiable, and the sequence $(\overline{G}_t)$ driven by $(\bar{h}_t)$ is precisely as in Algorithm 1.

Anytime Robust FTRL

Here we consider the setting in which $\mathcal{A}$ is implemented using a form of follow-the-regularized-leader (FTRL). Letting (ψ_t) be a sequence of regularizer functions $\psi_t : \mathcal{V} \to \mathbb{R}$, we are interested in the ancillary sequence (h_t) generated by

$$h_{t+1} = \mathcal{A}(h_t) \in \operatorname*{arg\,min}_{h \in \mathcal{H}} \left[\psi_{t+1}(h) + \sum_{i=1}^{t} \alpha_i \langle \overline{G}_i, h \rangle \right]. \tag{14}$$

The initial value is set using an extra regularizer ψ_1, with $h_1 \in \arg\min_{h \in \mathcal{H}} \psi_1(h)$. We proceed assuming that the sequence (h_t) exists, but we do not require the minimizer in (14) to be unique.

Lemma 4. *Let $\mathcal{A}$ be implemented as in (14), assuming that for each step $t \geq 1$, the regularizer ψ_t is κ_t-strongly convex. Then, for any reference point $h^* \in \mathcal{H}$, we have*

$$\begin{aligned} &\mathrm{Regret}(T; \mathcal{A}) \leq \\ &\psi_T(h^*) - \psi_1(h_1) + \sum_{t=1}^{T} \left[\psi_t(h_{t+1}) - \psi_{t+1}(h_{t+1}) \right] \\ &+ \sum_{t=1}^{T} \left[\frac{\|\partial \mathrm{R}(\bar{h}_t)\|_*^2}{2\kappa_t} + \alpha_t \langle \partial \mathrm{R}(\bar{h}_t) - \overline{G}_t, h_{t+1} - h_t \rangle \right]. \end{aligned} \tag{15}$$

This lemma is a natural anytime robust analogue of standard FTRL regret bounds (Orabona 2020, Sec. 7.8). While the above bound holds as long as R is sub-differentiable, in the special case where R is smooth, the final sum on the right-hand side of (15) is amenable to direct application of Lemma 2, as desired. Combining this with (13), one can immediately derive excess risk bounds for the output of Algorithm 1 under this FTRL-type of implementation, for a wide variety of regularization strategies.

Anytime Robust SMD

Next we consider the closely related setting in which $\mathcal{A}$ is implemented using a form of stochastic mirror descent (SMD). Assuming $\mathcal{H} \subset \mathcal{V}$ is bounded, closed, and convex, let $\Phi : \mathcal{V} \to \mathbb{R}$ be a differentiable and strictly convex function. Let $\mathcal{A}$ generate (h_t) based on the update

$$h_{t+1} = \mathcal{A}(h_t) = \operatorname*{arg\,min}_{h \in \mathcal{H}} \left[\langle \overline{G}_t, h \rangle + \frac{1}{\beta_t} \mathrm{D}_\Phi(h; h_t) \right]. \tag{16}$$

The function D_Φ is the Bregman divergence induced by Φ; see the appendix for more detailed background. The step sizes (β_t) are assumed positive, but can be set freely.

Lemma 5. *Let $\mathcal{A}$ be implemented as in (16), with Φ chosen to be κ-strongly convex on $\mathcal{H}$. Then for any reference point $h^* \in \mathcal{H}$, we have*

$$\begin{aligned} &\langle \overline{G}_t, h_t - h^* \rangle \\ &\leq \frac{\mathrm{D}_\Phi(h^*; h_t) - \mathrm{D}_\Phi(h^*; h_{t+1})}{\beta_t} + \frac{\beta_t}{2\kappa} \|\partial \mathrm{R}(\bar{h}_t)\|_*^2 \\ &\quad + \langle \partial \mathrm{R}(\bar{h}_t) - \overline{G}_t, h_{t+1} - h_t \rangle \end{aligned}$$

for all $t \geq 1$.

This lemma can be interpreted easily as an anytime robust analogue of traditional regret bounds for SMD (e.g., (Orabona 2020, Lem. 6.7)). It can be combined with (13) and Lemma 2 to obtain the following guarantee.

Theorem 6 (Anytime robust mirror descent)**.** *Under the setting of Lemmas 2 and 5, denote the diameter of $\mathcal{H}$ with respect to D_Φ as $\Delta_\Phi := \sup_{h,h' \in \mathcal{H}} \mathrm{D}_\Phi(h; h') < \infty$. Setting the weight sequences such that $\alpha_t/\alpha_{t-1} \geq \beta_t/\beta_{t-1}$ and $\beta_t \leq \kappa/\lambda$, we have that for any h^* which is a stationary point of R on $\mathcal{H}$, the inequality*

$$\mathrm{R}(\bar{h}_T) - \mathrm{R}(h^*) \leq \frac{1}{\alpha_{1:T}} \left[\frac{\alpha_T}{\beta_T} \Delta_\Phi + \max\{q_\delta(T), r_\delta(T)\} \right]$$

holds with probability no less than $1 - 2\delta$.

In contrast with Nazin et al. (2019) who query at the ancillary iterates, the preceding high-probability error bounds effectively give us sub-Gaussian guarantees for all points used to query stochastic gradients. As an important special case, consider the setting where $\mathcal{V}$ is Euclidean space, and the underlying norm used is the ℓ_2 norm $\|\cdot\|_2$. In this case, it is easy to verify that setting $\Phi(u) = \|u\|_2^2/2$, with $\kappa = 1$ the update (16) amounts to

$$h_{t+1} = \mathcal{A}(h_t) = \Pi_{\mathcal{H}} \left[h_t - \beta_t \overline{G}_t \right] \tag{17}$$

where $\Pi_{\mathcal{H}}[\cdot]$ denotes projection onto $\mathcal{H}$. That is, *anytime robust stochastic gradient descent*. These settings lead us to the following corollary.

Corollary 7 (Anytime robust SGD)**.** *Consider $\mathcal{A}$ implemented using (17), with weights $\alpha_t = 1$ and $\beta_t \leq 1/\lambda$ for all $t \in [T]$. Then we have*

$$\begin{aligned} &\mathrm{R}(\bar{h}_T) - \mathrm{R}(h^*) \leq \\ &\frac{2\Delta^2}{T\beta_T} + \max\left\{ 8\Delta\sigma \sqrt{\frac{2\log(\delta^{-1})}{T}}, \frac{12\lambda\Delta^2 \log(\delta^{-1})}{T} \right\} \end{aligned}$$

with probability no less than $1 - 2\delta$.

Anytime Robust AO-FTRL

In this sub-section, we consider the case that $\mathcal{A}$ is implemented using an adaptive optimistic follow-the-leader (AO-FTRL) procedure, namely updating as

$$\begin{aligned} &h_{t+1} = \mathcal{A}(h_t) \\ &\in \operatorname*{arg\,min}_{h \in \mathcal{H}} \left[\alpha_{t+1} \langle \widetilde{G}_t, h \rangle + \sum_{i=1}^{t} \alpha_i \langle \overline{G}_i, h \rangle + \sum_{i=0}^{t} \varphi_i(h) \right]. \end{aligned} \tag{18}$$

Here (φ_t) is a sequence of regularizers that is now summed over for later notational convenience. Recalling the FTRL update (14), then clearly the AO-FTRL update is almost the same, save for the presence of $\widetilde{G}_t$ at each step t, with the interpretation is that it provides a prediction of the loss that will be incurred in the following step, i.e., $\widetilde{G}_t \approx \overline{G}_{t+1}$.

Theorem 8. *Let Algorithm 1 be run under the assumptions of Lemma 2, with $\mathcal{A}$ implemented as in (18), setting $\widetilde{G}_t = \overline{G}_{t-1}$ for each $t > 1$. In addition, let each $\varphi_t(\cdot)$ be convex and non-negative, and denoting the regularizer partial sums as $\psi_t(\cdot) := \sum_{i=0}^{t-1} \varphi_i(\cdot)$, let each ψ_t be κ_t-strongly convex, with weights set such that $(\lambda/\kappa_t)\alpha_t^2 \leq \alpha_{1:(t-1)}$ for $t > 1$. Then, for any $h^* \in \mathcal{H}$ we have*

$$\begin{aligned}
&\mathrm{R}(\bar{h}_T) - \mathrm{R}(h^*) \\
&\leq \frac{1}{\alpha_{1:T}} \left[\frac{\alpha_1^2}{2\kappa_1} \|\nabla \mathrm{R}(\bar{h}_1) - \widetilde{G}_1\|_*^2 \right. \\
&\quad \left. + \sum_{t=1}^{T} \left[\varphi_{t-1}(h^*) - \varphi_{t-1}(h_t)\right] + 2\max\left\{q_\delta(T), r_\delta(T)\right\} \right]
\end{aligned}$$

with probability no less than $1 - 4\delta$.

This theorem can be considered a robust, high-probability analogue of the results in expectation given by Joulani et al. (2020, Thm. 3). As such, it can be readily combined with existing regularization techniques (Joulani et al. 2020, Sec. 4) to achieve the same rates (in T) under potentially heavy-tailed noise, with minimal computational overhead.

Empirical Analysis

In this section we complement the preceding theoretical analysis with an application of the proposed learning strategy to real-world benchmark datasets. The practical utility of various gradient truncation mechanisms has already been well-studied in the literature (Chen, Su, and Xu 2017; Prasad et al. 2018; Lecué, Lerasle, and Mathieu 2018; Holland and Ikeda 2019), and thus our chief point of interest here is if and when the feedback scheme utilized in Algorithm 1 can outperform the traditional feedback mechanism given by (2), under a convex, differentiable true objective. Put more succinctly, the key question is: *is there a practical benefit to querying at points with guarantees?*

Experimental setup Considering the context of key related work (Gorbunov, Danilova, and Gasnikov 2020; Nazin et al. 2019), we focus on averaged SGD as our baseline, and consider several real-world classification datasets of varying size, using standard multi-class logistic regression as our model.[2] We test three different learning procedures: averaged SGD using traditional feedback (2) (denoted `SGD-Ave`), anytime robust SGD precisely as in Algorithm 1 and Corollary 7 (denoted `Anytime-Robust-SGD`), and finally anytime SGD without the robustification sub-routine `Process` (denoted `Anytime-SGD`).

[2]Additional details for all the datasets used are provided in the appendix.

At a high level, for each dataset of interest, we run multiple independent randomized trials, and for each trial, we run the methods of interest for multiple "epochs" (i.e., multiple passes over the data), recording the on-sample (training) and off-sample (testing) performance at the end of each epoch. As a simple and lucid example that implies a convex objective, we use multi-class logistic loss under a linear model; for a dataset with k distinct classes, each predictor returns precisely k scores which are computed as a linear combination of the input features. Thus with k classes and d_{in} input features, the total dimensionality is $d = kd_{\text{in}}$. For these experiments we run 10 independent trials. Everything is implemented by hand in Python (ver. 3.8), making significant use of the `numpy` library (ver. 1.20).[3] For each method and each trial, the dataset is randomly shuffled before being split into training and testing subsets. If n is the size of any given dataset, then the training set is of size $n_{\text{tr}} := \lfloor 0.8n \rfloor$, and the test set is of size $n - n_{\text{tr}}$. Within each trial, for each epoch, the training data is also randomly shuffled. For all methods, the step size in update (17) is fixed at $\beta_t = 2/\sqrt{n_{\text{tr}}}$, for all steps t; this setting is appropriate for `Anytime-`$\star$ methods due to Corollary 7, and also for `SGD-Ave` based on standard results such as Nemirovski et al. (2009, Sec. 2.3). The (G_t) are obtained by direct computation of the logistic loss gradients, averaged over a mini-batch of size 8; this size was set arbitrarily for speed and stability, and no other mini-batch values were tested. Furthermore, for each method and each trial, the initial value h_1 is randomly generated in a dimension-wise fashion from the uniform distribution on the interval $[-0.05, 0.05]$. All raw input features are normalized to the unit interval $[0, 1]$ in a per-feature fashion. We do not do any regularization, for any method being tested. We test three different learning procedures: averaged SGD using traditional feedback (2) (denoted `SGD-Ave`), anytime robust SGD precisely as in Algorithm 1 and Corollary 7 (denoted `Anytime-Robust-SGD`), and finally anytime SGD without the robustification sub-routine `Process` (denoted `Anytime-SGD`).

Details for `Anytime-Robust-SGD` First, as a simple choice of anchors $\widetilde{h}$ and $\widetilde{g}$, we set $\widetilde{h} = h_1$ and estimate $\widetilde{g}$ using the empirical mean on the training data set; this is meant to provide a transparent baseline for what is possible without any fine-tuning. As discussed in section , a natural refinement is to set aside a small subset of data and dedicate a subset to mean estimation (i.e., computation of the dual anchor); see Lugosi and Mendelson (2019) for several examples of well-known practical robust high-dimensional mean estimators. As for the thresholds (c_t) used in the `Process` sub-routine, we set $c_t = \sqrt{n_{\text{tr}}/\log(\delta^{-1})}$ for all t, with a confidence level of $\delta = 0.05$ fixed throughout.

Results and discussion Our results are summarized in Figure 1, which plots the average training and test losses. For each trial, losses are averaged over datasets, and these average losses are themselves averaged over all trials to obtain the values plotted here. The impact of using feedback with

[3]A public repository including all experimental code has been published: https://github.com/feedbackward/anytime

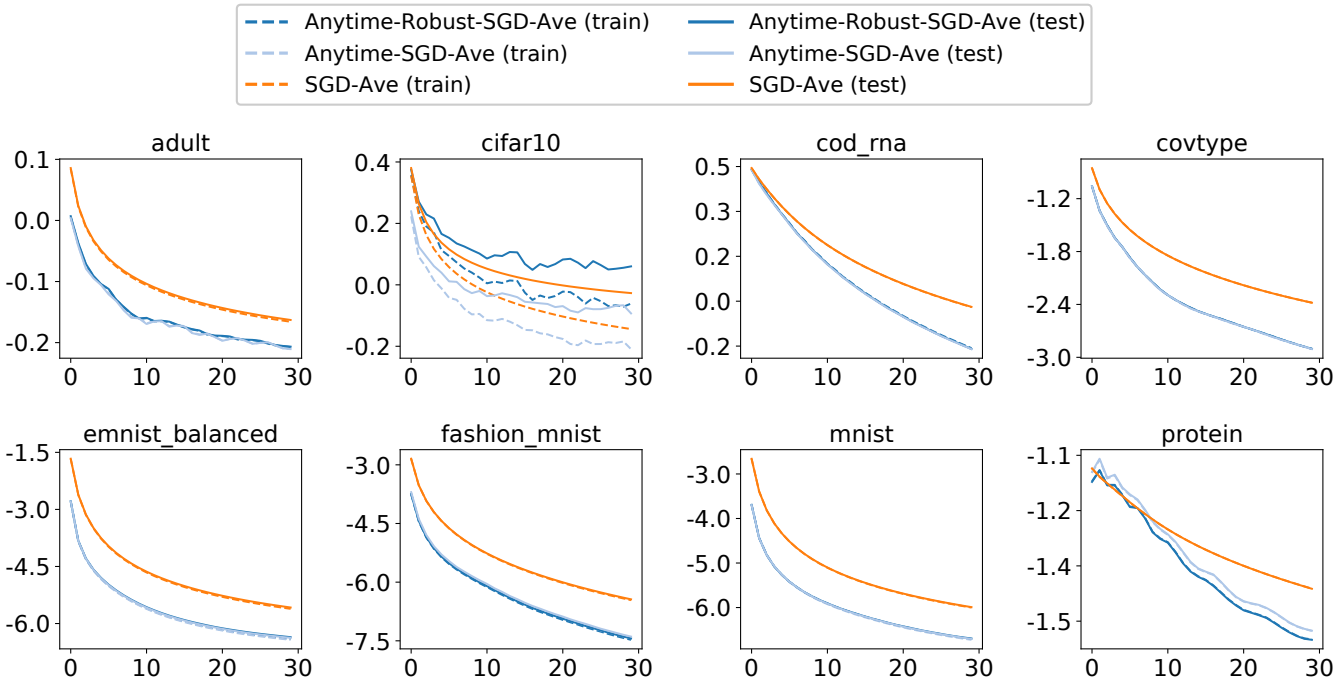

Figure 1: Training and test loss versus epoch number, averaged over all trials, for each method. The eight individual plot titles correspond to dataset names.

guarantees is immediate; in all cases, we see a notable boost in learning efficiency. This positive effect holds essentially uniformly across the datasets used, with no hyperparameter tuning. For CIFAR-10, we observe that the robustified version performs worse than than vanilla anytime averaged SGD; this looks to be due to the simple $\widetilde{h} = h_1$ setting, and can be readily mitigated by updating $\widetilde{h}$ after one pass over the data. It is reasonable to conjecture that if we were to shift to more complex non-linear models, from the resulting lack of convexity in the objective, there might emerge a tradeoff between the stability encouraged by Algorithm 1, and the benefits of parameter space exploration that are incidental to the noisier gradients arising under (2).

Future Directions

From a technical perspective, the most salient direction moving forward is strengthening the robust estimation subroutines to reduce the amount of prior knowledge required, and to potentially extend the methodology to cover non-smooth R. The requirement of a bounded domain can be removed (in the non-anytime setting) by using a more sophisticated update procedure (Gorbunov, Danilova, and Gasnikov 2020), and extending insights of this nature to refine or modify the procedure used to obtain Lemma 2 is of natural interest. In most learning tasks of interest, the variance of the underlying feedback distribution may change significantly, and an adaptive strategy for setting (c_t) is of interest both for strengthening formal guarantees and improving efficiency and stability in practice.

Acknowledgements

This work was supported by the JSPS KAKENHI Grant Number 19K20342, and by JST ACT-X Grant Number JPMJAX200O.

References

Ash, R. B.; and Doléans-Dade, C. A. 2000. *Probability and Measure Theory*. Academic Press, 2nd edition.

Brownlees, C.; Joly, E.; and Lugosi, G. 2015. Empirical risk minimization for heavy-tailed losses. *Annals of Statistics*, 43(6): 2507–2536.

Catoni, O.; and Giulini, I. 2017. Dimension-free PAC-Bayesian bounds for matrices, vectors, and linear least squares regression. *arXiv preprint arXiv:1712.02747*.

Cesa-Bianchi, N.; Conconi, A.; and Gentile, C. 2004. On the Generalization Ability of On-Line Learning Algorithms. *IEEE Transactions on Information Theory*, 50(9): 2050–2057.

Chen, Y.; Su, L.; and Xu, J. 2017. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. In *Proceedings of the ACM on Measurement and Analysis of Computing Systems*. ACM.

Cohen, M. B.; Lee, Y. T.; Miller, G.; Pachocki, J.; and Sidford, A. 2016. Geometric median in nearly linear time. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, 9–21.

Cutkosky, A. 2019. Anytime online-to-batch, optimism and acceleration. In *36th International Conference on Machine*

Learning (ICML), volume 97 of *Proceedings of Machine Learning Research*, 1446–1454.

Davis, D.; Drusvyatskiy, D.; Xiao, L.; and Zhang, J. 2019. Robust stochastic optimization with the proximal point method. *arXiv preprint arXiv:1907.13307v3*.

Devroye, L.; Lerasle, M.; Lugosi, G.; and Oliveira, R. I. 2016. Sub-Gaussian mean estimators. *Annals of Statistics*, 44(6): 2695–2725.

Gorbunov, E.; Danilova, M.; and Gasnikov, A. 2020. Stochastic optimization with heavy-tailed noise via accelerated gradient clipping. *arXiv preprint arXiv:2005.10785*.

Haussler, D. 1992. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100(1): 78–150.

Hazan, E. 2016. Introduction to Online Convex Optimization. *Foundations and Trends® in Optimization*, 2(3-4): 157–325.

Holland, M. J.; and Ikeda, K. 2019. Better generalization with less data using robust gradient descent. In *36th International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*.

Joulani, P.; Raj, A.; György, A.; and Szepesvári, C. 2020. A Simpler Approach to Accelerated Stochastic Optimization: Iterative Averaging Meets Optimism. In *37th International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, 4984–4993.

Lecué, G.; Lerasle, M.; and Mathieu, T. 2018. Robust classification via MOM minimization. *arXiv preprint arXiv:1808.03106v1*.

Lugosi, G.; and Mendelson, S. 2019. Robust multivariate mean estimation: the optimality of trimmed mean. *arXiv preprint arXiv:1907.11391v1*.

Nazin, A. V.; Nemirovsky, A. S.; Tsybakov, A. B.; and Juditsky, A. B. 2019. Algorithms of robust stochastic optimization based on mirror descent method. *Automation and Remote Control*, 80(9): 1607–1627.

Nemirovski, A.; Juditsky, A.; Lan, G.; and Shapiro, A. 2009. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4): 1574–1609.

Nemirovsky, A. S.; and Yudin, D. B. 1983. *Problem complexity and method efficiency in optimization*. Wiley-Interscience.

Nguyen, L. M.; Nguyen, P. H.; van Dijk, M.; Richtárik, P.; Scheinberg, K.; and Takáč, M. 2018. SGD and Hogwild! convergence without the bounded gradients assumption. *arXiv preprint arXiv:1802.03801v2*.

Orabona, F. 2020. A Modern Introduction to Online Learning. *arXiv preprint arXiv:1912.13213v3*.

Prasad, A.; Suggala, A. S.; Balakrishnan, S.; and Ravikumar, P. 2018. Robust estimation via robust gradient estimation. *arXiv preprint arXiv:1802.06485*.

Shalev-Shwartz, S. 2012. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2): 107–194.

Vapnik, V. N. 1999. *The Nature of Statistical Learning Theory*. Statistics for Engineering and Information Science. Springer, 2nd edition.

Vardi, Y.; and Zhang, C.-H. 2000. The multivariate L_1-median and associated data depth. *Proceedings of the National Academy of Sciences*, 97(4): 1423–1426.