Surrogate Losses for Online Learning of Stepsizes in Stochastic Non-Convex Optimization

Zhenxun Zhuang [1] **Ashok Cutkosky** [2] **Francesco Orabona** [1 3]

Abstract

Stochastic Gradient Descent (SGD) has played a central role in machine learning. However, it requires a carefully hand-picked stepsize for fast convergence, which is notoriously tedious and time-consuming to tune. Over the last several years, a plethora of adaptive gradient-based algorithms have emerged to ameliorate this problem. In this paper, we propose new surrogate losses to cast the problem of learning the optimal stepsizes for the stochastic optimization of a non-convex smooth objective function onto an online convex optimization problem. This allows the use of no-regret online algorithms to compute optimal stepsizes on the fly. In turn, this results in a SGD algorithm with self-tuned stepsizes that guarantees convergence rates that are automatically adaptive to the level of noise.

1. Introduction

In recent years, Stochastic Gradient Descent (SGD) has became the tool of choice for fast optimization of convex and non-convex objective functions. Its simplicity of implementation allows for use in virtually any machine learning problem. In its basic version, it iteratively updates the solution to an optimization problem, moving in the negative direction of the gradient of the objective function at the current solution:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta_t \boldsymbol{g}(\boldsymbol{x}_t, \xi_t), \tag{1}$$

where $\boldsymbol{g}(\boldsymbol{x}_t, \xi_t)$ is a stochastic gradient of the objective function f at the current point $\boldsymbol{x}_t$ depending on the stochastic variable ξ_t. A critical component of the algorithm is the stepsize $\eta_t > 0$. In order to achieve a fast convergence, the stepsizes must be carefully selected, taking into account the objective function and characteristics of the noise. This task becomes particularly daunting because the noise might change over time due to a variety of factors such as, e.g., approaching the local minimum of the function, changing size of the minibatch, gradients calculated through a simulation.

For the above reasons, a number of variants of SGD have been proposed trying to "adapt" the stepsizes in more or less theoretically principled ways. Indeed, the idea of adapting stepsizes is an old one. A few famous examples are the Polyak's rule (Polyak, 1987), Stochastic Meta-Descent (Schraudolph, 1999), AdaGrad (Duchi et al., 2011). However, most of previous approaches to adapting the stepsizes are designed for convex functions or without a guaranteed strategy of converge to some optimal stepsize. In fact, often the definition itself of "optimal" stepsize is missing.

In this paper, we take a different and novel route. We study theoretically the setting of stochastic smooth non-convex optimization and we design *convex surrogate loss functions that upper bound the expected decrement of the objective function after an SGD update*. The first advantage of our approach is that the optimal stepsize can be now defined as the one minimizing the surrogate losses. Moreover, using a no-regret online learning algorithm (Cesa-Bianchi & Lugosi, 2006), we can adapt the stepsizes and guarantee that they will be close to the one of the a-posteriori optimal stepsize. Moreover, basing our approach on online learning methods, we gain the implicit robustness of these methods to adversarial conditions.

The rest of the paper is organized as follows. We begin by discussing related work (Section 2), and then introduce necessary definitions and assumptions (Section 3). Next, we introduce the surrogate loss functions (Section 4) and use them to design an algorithm that adapts global and coordinate-wise stepsizes (Sections 5 and 6). We also empirically validate our theoretical findings on a classification task (Section 7 and Appendix) showing that, differently from other adaptive methods, our method does not require fiddling with stepsizes to guarantee convergence in the stochastic setting. Finally, we draw some conclusions and describe the future work in Section 8.

[1] Department of Computer Science, Boston University, Boston, MA, US [2] Google, Mountain View, CA, US [3] Department of Electrical & Computer Engineering, Boston University, Boston, MA, US. Correspondence to: Zhenxun Zhuang <zxzhuang@bu.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

2. Related Work

Here we discuss the theoretical related work on adaptive stochastic optimization algorithms. First, the convergence of a random iterate of SGD for non-convex smooth functions has been proved by Ghadimi & Lan (2013). They also calculate how the optimal stepsize depends on the variance of the noise in the gradients and the smoothness of the objective function.

The optimal convergence rate was also obtained by Ward et al. (2018) using AdaGrad global stepsizes, without the need to tune parameters. Li & Orabona (2019) improves over their results by removing the assumption of bounded gradients. However, both analyses focus on the adaptivity of non-per-coordinate updates, and are somewhat complicated in order to deal with unbounded gradients or non-independence of the current stepsize from the current step gradient. In comparison, our technique is relatively simple, allowing us to easily show a nontrivial guarantee for per-coordinate updates. In addition, their results cannot recover linear rates of convergence assuming, for example, the Polyak-Łojasiewicz condition (Karimi et al., 2016).

The idea of tuning stepsizes with online learning has been explored in the online convex optimization literature (Koolen et al., 2014; van Erven & Koolen, 2016). There, the possible stepsizes are discretized and an expert algorithm is used to select the stepsize to use online. Instead, in our work the use of convex surrogate loss functions allows us to directly learn the optimal stepsize, without needing to discretize the range of stepsizes. This becomes very important when we consider the possibility of learning a stepsize for each coordinate (Section 6), as we avoid a computational overhead exponential in the dimension of the space d that a discretization would incur.

3. Definitions and Assumptions

We use bold lower-case letters to denote vectors, and bold upper-case letters for matrices, e.g., $\boldsymbol{u} \in \mathbb{R}^d$, $\boldsymbol{A} \in \mathbb{R}^{m\times n}$. The ith coordinate of a vector $\boldsymbol{u}$ is u_i. Throughout this paper, we study the Euclidean space $\mathbb{R}^d$ with the inner product $\langle \cdot, \cdot \rangle$. Unless explicitly noted, all norms are the Euclidean norm. The dual norm $\|\cdot\|_*$ is the norm defined by $\|\boldsymbol{v}\|_* = \sup_{\boldsymbol{u}}\{\langle \boldsymbol{u}, \boldsymbol{v}\rangle : \|\boldsymbol{u}\| \leq 1\}$. $\mathbb{E}[\boldsymbol{u}]$ means the expectation w.r.t. the underlying probability distribution of a random variable $\boldsymbol{u}$. The gradient of f at $\boldsymbol{x}$ is denoted by $\nabla f(\boldsymbol{x})$.

Now, we describe our first-order stochastic black-box oracle. In our setting, we will query the stochastic oracle two times on each round $t = 1, \dots, T$, obtaining the noisy gradients $\boldsymbol{g}(\boldsymbol{x}_t, \xi_t)$ and $\boldsymbol{g}(\boldsymbol{x}_t, \xi'_t)$. *Note that, when convenient, we will refer to $\boldsymbol{g}(\boldsymbol{x}_t, \xi_t)$ and $\boldsymbol{g}(\boldsymbol{x}_t, \xi'_t)$ as $\boldsymbol{g}_t$ and $\boldsymbol{g}'_t$ respectively.* We assume everywhere that our objective function is bounded from below and denote the infimum by $f^\star$, hence $f^\star > -\infty$. Also, we use $\mathbb{E}_t[\boldsymbol{u}]$ to denote the conditional expectation of $\boldsymbol{u}$ with respect to $\xi_1, \dots, \xi_{t-1}, \xi'_1, \dots, \xi'_{t-1}$. Further, we will make use of the following assumptions:

H1: The noisy gradients at time t are unbiased and independent given the past, that is

$$\mathbb{E}_t\left[\boldsymbol{g}(\boldsymbol{x}_t, \xi_t)\right] = \mathbb{E}_t\left[\boldsymbol{g}(\boldsymbol{x}_t, \xi'_t)\right] = \nabla f(\boldsymbol{x}_t),$$
$$\mathbb{E}_t\left[\langle \boldsymbol{g}(\boldsymbol{x}_t, \xi_t), \boldsymbol{g}(\boldsymbol{x}_t, \xi'_t)\rangle\right] = \|\nabla f(\boldsymbol{x}_t)\|^2 .$$

H2: The noisy gradients $\boldsymbol{g}_t$ have finite variance with respect to the L2 norm:

$$\mathbb{E}_t\left[\|\boldsymbol{g}(\boldsymbol{x}_t, \xi_t) - \nabla f(\boldsymbol{x}_t)\|^2\right] = \sigma_t^2 .$$

H3: The noisy gradients have bounded norm: $\|\boldsymbol{g}(\boldsymbol{x}_t, \xi_t)\| \leq L$, $\|\boldsymbol{g}(\boldsymbol{x}_t, \xi'_t)\| \leq L$.

H4: The function $f : \mathbb{R}^d \to \mathbb{R}$ is M-smooth, that is f is differentiable and $\forall \boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^d$ we have $\|\nabla f(\boldsymbol{x}_1) - \nabla f(\boldsymbol{x}_2)\| \leq M\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|$. Note that (**H4**), for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, implies (Nesterov, 2003, Lemma 1.2.3)

$$|f(\boldsymbol{x}_2) - f(\boldsymbol{x}_1) - \langle \nabla f(\boldsymbol{x}_1), \boldsymbol{x}_2 - \boldsymbol{x}_1\rangle| \leq \frac{M}{2}\|\boldsymbol{x}_2 - \boldsymbol{x}_1\|^2.$$

We will also consider the Polyak-Łojasiewicz (PL) condition (Karimi et al., 2016), a much weaker version of *strong convexity*. The PL condition does not require convexity, but is still sufficient for showing a global linear convergence rate for gradient descent.

H5: A differentiable function f satisfies the PL condition if for some $\mu > 0$

$$\|\nabla f(\boldsymbol{x})\|^2 \geq 2\mu(f(\boldsymbol{x}) - f^\star), \quad \forall \boldsymbol{x} .$$

4. Surrogate Losses

Consider using SGD with non-convex M-smooth losses, using a fixed stepsize $0 < \eta \leq \frac{1}{M}$ and starting from an initial point $\boldsymbol{x}_1$. Assuming all the variances are bounded by σ^2, it is well-known that we obtain the following convergence rate (Ghadimi & Lan, 2013)

$$\mathbb{E}[\|\nabla f(\boldsymbol{x}_k)\|^2] \leq O\left((f(\boldsymbol{x}_1) - f^\star)/(\eta T) + \eta\sigma^2\right),$$

where k is a uniform random variable between 1 and T. From the above, it is immediate to see that we need a stepsize of the form $O(\min(\frac{\sqrt{f(\boldsymbol{x}_1) - f^\star}}{\sigma\sqrt{T}}, \frac{1}{M}))$ to have the best worst case convergence of $O(\frac{1}{T} + \frac{\sigma}{\sqrt{T}})$. In words, this means that we get a faster rate, $O(\frac{1}{T})$, when there is no noise, and a slower rate, $O(\frac{\sigma}{\sqrt{T}})$, in the presence of noise.

However, we usually do not know the variance of the noise σ, which makes the above optimal tuning of the stepsize difficult to achieve in practice. Even worse, the variance can change over time. For example, it may decrease over time if $f(\boldsymbol{x}) = \mathbb{E}[f_j(\boldsymbol{x})]$ and each f_j has zero gradient at the local optimum we are converging to. Moreover, even assuming the knowledge of the variance of the noise, the stepsizes proposed in Ghadimi & Lan (2013) assume the knowledge of the unknown quantity $f(\boldsymbol{x}_1) - f^\star$.

One solution would be to obtain an explicit estimate of the variances of the noise, for example by applying some concentration inequality to the sample variance, and use it to set the stepsizes. This approach is suboptimal because it does not directly optimize the convergence rates, relying instead on a loose worst-case analysis. Instead, we propose to directly estimate the stepsizes that achieve the best convergence rate using an online learning algorithm. Our approach is simple and efficient: we introduce new surrogate (strongly)-convex losses that make the problem of learning the stepsizes a simple one-dimensional online convex optimization problem.

Our strategy uses the smoothness of the objective function to transform the problem of optimizing a non-convex objective function to the problem of optimizing a series of convex loss functions, which we solve by an online learning algorithm. Specifically, at each time t define the surrogate loss $\ell_t : \mathbb{R} \to \mathbb{R}$ as

$$\ell_t(\eta) = \frac{M\eta^2}{2}\|\boldsymbol{g}(\boldsymbol{x}_t, \xi_t)\|^2 - \eta\langle \boldsymbol{g}(\boldsymbol{x}_t, \xi_t), \boldsymbol{g}(\boldsymbol{x}_t, \xi_t')\rangle, \quad (2)$$

where $\boldsymbol{g}(\boldsymbol{x}_t, \xi_t)$ and $\boldsymbol{g}(\boldsymbol{x}_t, \xi_t')$ are the noisy stochastic gradients received from the black-box oracle at time t. It is clear that ℓ_t is a convex function. Moreover, the following key result shows that these surrogate losses upper bound the expected decrease of the function value f.

Theorem 1. *Assume (**H4**) holds and η_t is independent from ξ_j and ξ_j' for $j \geq t$. Then, for the SGD update in (1), we have*

$$\mathbb{E}\left[f(\boldsymbol{x}_{t+1}) - f(\boldsymbol{x}_t)\right] \leq \mathbb{E}\left[\ell_t(\eta_t)\right] .$$

Proof. The M-smoothness of f gives us:

$$\begin{aligned}
&\mathbb{E}\left[f(\boldsymbol{x}_{t+1}) - f(\boldsymbol{x}_t)\right] \\
&\leq \mathbb{E}\left[\langle \nabla f(\boldsymbol{x}_t), \boldsymbol{x}_{t+1} - \boldsymbol{x}_t\rangle + \frac{M}{2}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_t\|^2\right] \\
&= \mathbb{E}\left[\langle \nabla f(\boldsymbol{x}_t), -\eta_t \boldsymbol{g}(\boldsymbol{x}_t, \xi_t)\rangle + \frac{M}{2}\eta_t^2\|\boldsymbol{g}(\boldsymbol{x}_t, \xi_t)\|^2\right] \\
&= \mathbb{E}\left[\langle \nabla f(\boldsymbol{x}_t), \mathbb{E}_t\left[-\eta_t \boldsymbol{g}(\boldsymbol{x}_t, \xi_t)\right]\rangle + \frac{M}{2}\eta_t^2\|\boldsymbol{g}(\boldsymbol{x}_t, \xi_t)\|^2\right].
\end{aligned}$$

Now observe that $\nabla f(\boldsymbol{x}_t) = \mathbb{E}_t\left[\boldsymbol{g}(\boldsymbol{x}_t, \xi_t')\right]$, so that

$$\begin{aligned}
&\mathbb{E}\left[\langle \mathbb{E}_t\left[\boldsymbol{g}(\boldsymbol{x}_t, \xi_t')\right], \mathbb{E}_t\left[-\eta_t \boldsymbol{g}(\boldsymbol{x}_t, \xi_t)\right]\rangle\right] \\
&\quad = \mathbb{E}\left[\mathbb{E}_t\left[\langle \boldsymbol{g}(\boldsymbol{x}_t, \xi_t'), -\eta_t \boldsymbol{g}(\boldsymbol{x}_t, \xi_t)\rangle\right]\right] \\
&\quad = \mathbb{E}\left[\langle \boldsymbol{g}(\boldsymbol{x}_t, \xi_t'), -\eta_t \boldsymbol{g}(\boldsymbol{x}_t, \xi_t)\rangle\right] .
\end{aligned}$$

Putting all together, we have the stated inequality. □

Algorithm 1 Stochastic Gradient Descent with Online Learning (SGDOL)

```
Input: $\boldsymbol{x}_1 \in \mathcal{X}$, $M$, an online learning algorithm $\mathcal{A}$
for $t = 1, 2, \dots, T$ do
    Compute $\eta_t$ by running $\mathcal{A}$ on $\ell_i, i = 1, \dots, t-1$, as defined in (2)
    Receive two independent unbiased estimates of $\nabla f(\boldsymbol{x}_t)$: $\boldsymbol{g}_t$, $\boldsymbol{g}_t'$
    Update $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta_t \boldsymbol{g}_t$
end for
Output: uniformly randomly choose a $\boldsymbol{x}_k$ from $\boldsymbol{x}_1, \dots, \boldsymbol{x}_T$.
```

Note that the assumption of the independence of η_t from the stochasticity ξ_j and ξ_j' of the current step is essential according to Li & Orabona (2019).

The theorem tells us that if we want to decrease the function f, we might instead try to minimize the convex surrogate losses ℓ_t. In the following, we build up on this intuition to design an online learning procedure that adapts the stepsizes of SGD to achieve the optimal convergence rate.

5. SGDOL: Adaptive Stepsizes with FTRL

The surrogate losses allow us to design an online convex optimization procedure to learn the optimal stepsizes. We call this procedure Stochastic Gradient Descent with Online Learning (SGDOL) and the pseudocode is in Algorithm 1. Remind that $\boldsymbol{g}(\boldsymbol{x}_t, \xi_t)$ and $\boldsymbol{g}(\boldsymbol{x}_t, \xi_t')$ are referred to as $\boldsymbol{g}_t$ and $\boldsymbol{g}_t'$ respectively for convenience. In each round, the stepsizes are chosen by an online learning algorithm $\mathcal{A}$ fed with the surrogate losses ℓ_t. The online learning algorithm aims at minimizing the regret: the difference between the cumulative sum of the losses incurred by the algorithm in each round, and the cumulative losses w.r.t any fixed point η (especially the one giving the smallest losses in hindsight). In formulas, for a 1-dimensional online convex optimization problem, the regret is defined as

$$\text{Regret}_T(\eta) = \sum_{t=1}^{T}(\ell_t(\eta_t) - \ell_t(\eta)) .$$

A small regret with respect to the optimal choice of η means that the losses of the algorithm are not too big compared to the best achievable losses from using a fixed point. In turn, this implies that the stepsizes chosen by the online algorithm will not be too far from the optimal (unknown) stepsize.

Employing SGDOL, we can prove the following Theorem.

Theorem 2. *Assume (**H1, H2, H4**) to hold. Then, for any*

$\eta > 0$, *SGDOL in Algorithm 1 satisfies*

$$\mathbb{E}\left[\left(\eta - \frac{M}{2}\eta^2\right)\sum_{t=1}^{T}\|\nabla f(\boldsymbol{x}_t)\|^2\right] \\ \leq f(\boldsymbol{x}_1) - f^\star + \mathbb{E}\left[\text{Regret}_T(\eta)\right] + \frac{M\eta^2}{2}\sum_{t=1}^{T}\mathbb{E}[\sigma_t^2]\ .$$

Proof. Summing the inequality in Theorem 1 from 1 to T:

$$\begin{aligned} f^\star - f(\boldsymbol{x}_1) &\leq \mathbb{E}[f(\boldsymbol{x}_{T+1})] - f(\boldsymbol{x}_1) \\ &= \sum_{t=1}^{T}\mathbb{E}\left[f(\boldsymbol{x}_{t+1}) - f(\boldsymbol{x}_t)\right] \\ &\leq \sum_{t=1}^{T}\mathbb{E}\left[\ell_t(\eta_t)\right] \\ &= \sum_{t=1}^{T}\mathbb{E}\left[\ell_t(\eta_t) - \ell_t(\eta)\right] + \sum_{t=1}^{T}\mathbb{E}\left[\ell_t(\eta)\right] \\ &\leq \mathbb{E}\left[\text{Regret}_T(\eta)\right] + \sum_{t=1}^{T}\mathbb{E}\left[\ell_t(\eta)\right]\ . \end{aligned}$$

Using the fact that

$$\mathbb{E}_t[\ell_t(\eta)] = \left(-\eta + \frac{M}{2}\eta^2\right)\|\nabla f(\boldsymbol{x}_t)\|^2 + \frac{M}{2}\eta^2\sigma_t^2,$$

we have the stated bound. □

The only remaining ingredient for SGDOL is to decide an online learning procedure. Given that the surrogate losses are strongly convex, we can use a Follow The Regularized Leader (FTRL) algorithm (Shalev-Shwartz, 2007; Abernethy et al., 2008; 2012; McMahan, 2017). Note that this is not the only possibility, e.g. we could even use an optimistic FTRL algorithm that achieves even smaller regret (Mohri & Yang, 2016). However, FTRL is enough to show the potential of our surrogate losses. In an online learning game in which we receive the convex losses ℓ_t, FTRL constructs the predictions $\boldsymbol{v}_t$ by solving the optimization problem

$$\boldsymbol{v}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{v}\in\mathbb{R}^d}\ r(\boldsymbol{v}) + \sum_{s=1}^{t}\ell_s(\boldsymbol{v}),$$

where $r : \mathbb{R}^d \to \mathbb{R}$ is a regularization function. We can upper bound the regret of FTRL with the following theorem.

Theorem 3. *(McMahan, 2017) Suppose r is chosen such that $h_t = r + \sum_{i=1}^{t}\ell_i$ is 1-strongly-convex w.r.t. some norm $\|\cdot\|_{(t)}$. Then, choosing any $\boldsymbol{g}_t \in \partial\ell_t(\boldsymbol{x}_t)$ on each round, for any $\boldsymbol{x}^\star \in \mathbb{R}^d$ and for any $T > 0$,*

$$\text{Regret}_T(\boldsymbol{x}^\star) \leq r(\boldsymbol{x}^\star) + \frac{1}{2}\sum_{t=1}^{T}\|\boldsymbol{g}_t\|_{(t),\star}^2,$$

where $\|\cdot\|_{(t),\star}$ is the dual norm of $\|\cdot\|_{(t)}$.

Algorithm 2 Follow the Regularized Leader (FTRL)

```
Parameters: $r(\boldsymbol{v}) \geq 0$
$\boldsymbol{v}_1 \leftarrow \arg\min_{\boldsymbol{v}\in\mathbb{R}^d}\ r(\boldsymbol{v})$
for $t = 1, 2, \ldots$ do
    Observe convex loss function $\ell_t : \mathbb{R}^d \to \mathbb{R}\cup\{\infty\}$
    Incur loss $\ell_t(\boldsymbol{v}_t)$
    Update $\boldsymbol{v}_{t+1} \leftarrow \arg\min_{\boldsymbol{v}\in\mathbb{R}^d}\ r(\boldsymbol{v}) + \sum_{s=1}^{t}\ell_s(\boldsymbol{v})$
end for
```

We can now put all together and obtain a convergence rate guarantee for SGDOL.

Theorem 4. *By choosing $r(\eta) = \frac{M\alpha}{2}\left(\eta - \frac{1}{M}\right)^2 + \mathcal{I}\left(\eta \in \left[0, \frac{2}{M}\right]\right)$ with $\alpha > 0$, assuming* (**H1 - H4**)*, and using FTRL, Algorithm 2, in Algorithm 1, for an uniformly randomly picked $\boldsymbol{x}_k$ from $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$, we have:*

$$\begin{aligned} &\mathbb{E}\left[\|\nabla f(\boldsymbol{x}_k)\|^2\right] \\ &\quad\leq \frac{2M}{T}\left(f(\boldsymbol{x}_1) - f^\star + \frac{5L^2}{M}\ln\left(1 + \frac{L^2T}{\alpha}\right)\right) \\ &\qquad + \frac{1}{T}\sqrt{2M\sum_{t=1}^{T}\mathbb{E}[\sigma_t^2]\left(f(\boldsymbol{x}_1) - f^\star + \frac{\alpha}{2M}\right)} \\ &\qquad + \frac{1}{T}\sqrt{10L^2\sum_{t=1}^{T}\mathbb{E}[\sigma_t^2]\ln\left(1 + \frac{L^2T}{\alpha}\right)}\ . \end{aligned}$$

Before proving this theorem, we make some observations.

The FTRL update gives us a very simple strategy to calculate the stepsizes η_t. In particular, the FTRL update has a closed form:

$$\eta_t = \max\left\{0, \min\left\{\frac{\alpha + \sum_{j=1}^{t-1}\langle\boldsymbol{g}_j, \boldsymbol{g}'_j\rangle}{M\left(\alpha + \sum_{j=1}^{t-1}\|\boldsymbol{g}_j\|^2\right)}, \frac{2}{M}\right\}\right\}\ .$$

Note that this update can be efficiently computed by keeping track of the quantities $\sum_{j=1}^{t-1}\langle\boldsymbol{g}_j, \boldsymbol{g}'_j\rangle$ and $\sum_{j=1}^{t-1}\|\boldsymbol{g}_j\|^2$.

While the computational complexity of calculating η_t by FTRL is negligible, SGDOL requires two unbiased gradients per step. This increases the computational complexity with respect to a plain SGD procedure by a factor of two.

The value of α affects how fast η_t deviates from its initial value $\frac{1}{M}$. Although Theorem 4 shows that a too small α would blow up the log factor, it also indicates setting α to be comparable with $M(f(x_1) - f(x^*))$ or smaller would suffice for not inducing a major influence on the convergence rate. In fact, preliminary experiments have shown that α has no notable influence on performance so long as it is comparable to M or smaller.

We can now prove the convergence rate in Theorem 4. For the proof, we need the following technical lemma.

Lemma 5. *Let $a_i \geq 0, \cdots, T$ and $h : [0, +\infty) \to [0, +\infty)$ nonincreasing function. Then*

$$\sum_{t=1}^{T} a_t h\left(a_0 + \sum_{i=1}^{t} a_i\right) \leq \int_{a_0}^{\sum_{t=0}^{T} a_t} h(x)dx \,.$$

Proof. Denote by $s_t = \sum_{i=0}^{t} a_i$.

$$a_i h(s_i) = \int_{s_{i-1}}^{s_i} h(s_i)dx \leq \int_{s_{i-1}}^{s_i} h(x)dx \,.$$

Summing over $i = 1, \cdots, T$, we have the stated bound. □

Proof of Theorem 4. As $\ell_t''(\eta) = M\|\boldsymbol{g}_t\|^2$, $h_t = r + \sum_{i=1}^{t} \ell_i$ is 1-strongly-convex w.r.t. the norm $\sqrt{M\left(\alpha + \sum_{s=1}^{t} \|\boldsymbol{g}_s\|^2\right)}\|\cdot\|$.

Applying Theorem 3, we get that, for any $\eta \in \left[0, \frac{2}{M}\right]$,

$$\begin{aligned}&\text{Regret}_T(\eta)\\ &\leq \frac{M\alpha}{2}\left(\eta - \frac{1}{M}\right)^2 + \frac{1}{2M}\sum_{t=1}^{T} \frac{\left(\ell_t'(\eta_t)\right)^2}{\alpha + \sum_{s=1}^{t} \|\boldsymbol{g}_s\|^2}.\end{aligned}$$

Now observe that

$$\begin{aligned}\left(\ell_t'(\eta_t)\right)^2 &= \left(-\langle \boldsymbol{g}_t, \boldsymbol{g}_t'\rangle + M\eta_t\|\boldsymbol{g}_t\|^2\right)^2\\ &\leq 2\langle \boldsymbol{g}_t, \boldsymbol{g}_t'\rangle^2 + 2M^2\eta_t^2\|\boldsymbol{g}_t\|^4\\ &\leq 2\|\boldsymbol{g}_t\|^2\|\boldsymbol{g}_t'\|^2 + 8\|\boldsymbol{g}_t\|^4 \leq 10L^2\|\boldsymbol{g}_t\|^2,\end{aligned}$$

where in the third line of which we used the Cauchy-Schwarz inequality and $\eta_t \leq \frac{2}{M}$. Hence, we get

$$\begin{aligned}&\frac{1}{2M}\sum_{t=1}^{T} \frac{\left(\ell_t'(\eta_t)\right)^2}{\alpha + \sum_{s=1}^{t} \|\boldsymbol{g}_s\|^2}\\ &\leq \frac{5L^2}{M}\sum_{t=1}^{T} \frac{\|\boldsymbol{g}_t\|^2}{\alpha + \sum_{s=1}^{t} \|\boldsymbol{g}_s\|^2}\\ &\leq \frac{5L^2}{M}\ln\left(\frac{\alpha + \sum_{t=1}^{T} \|\boldsymbol{g}_t\|^2}{\alpha}\right)\\ &\leq \frac{5L^2}{M}\ln\left(1 + \frac{L^2T}{\alpha}\right),\end{aligned}$$

where in the second inequality we used Lemma 5.

Put the last inequality above back into Theorem 2 yields

$$\begin{aligned}&\mathbb{E}\left[\left(\eta - \frac{M}{2}\eta^2\right)\sum_{t=1}^{T} \|\nabla f(\boldsymbol{x}_t)\|^2\right]\\ &\leq f(\boldsymbol{x}_1) - f^\star + \frac{M\alpha}{2}\left(\eta - \frac{1}{M}\right)^2\\ &\quad + \frac{5L^2}{M}\ln\left(1 + \frac{L^2T}{\alpha}\right) + \frac{M\eta^2}{2}\sum_{t=1}^{T}\mathbb{E}[\sigma_t^2] \,.\end{aligned}$$

Denote $A \triangleq \sum_{t=1}^{T} \mathbb{E}\left[\|\nabla f(\boldsymbol{x}_t)\|^2\right]$, we can transform the above into a quadratic inequality of η:

$$\begin{aligned}0 \leq\ &\frac{M}{2}\left(A + \alpha + \sum_{t=1}^{T}\mathbb{E}[\sigma_t^2]\right)\eta^2 - (A+\alpha)\eta\\ &+ f(\boldsymbol{x}_1) - f^\star + \frac{5L^2}{M}\ln\left(1 + \frac{L^2T}{\alpha}\right) + \frac{\alpha}{2M} \,.\end{aligned}$$

Choosing η as the minimizer of the right hand side: $\eta^* = \frac{\alpha + A}{M(\alpha+A) + M\sum_{t=1}^{T}\mathbb{E}[\sigma_t^2]}$ (which satisfies $\eta^* \leq \frac{2}{M}$) gives us

$$\begin{aligned}&\frac{(\alpha + A)^2}{2M\left(\alpha + A + \sum_{t=1}^{T}\mathbb{E}[\sigma_t^2]\right)}\\ &\leq f(\boldsymbol{x}_1) - f^\star + \frac{\alpha}{2M} + \frac{5L^2}{M}\ln\left(1 + \frac{L^2T}{\alpha}\right) \,.\end{aligned}$$

Solving this quadratic inequality of A yields

$$\begin{aligned}A \leq\ &2M\left(f(\boldsymbol{x}_1) - f^\star + \frac{5L^2}{M}\ln\left(1 + \frac{L^2T}{\alpha}\right)\right)\\ &+ \sqrt{2M\sum_{t=1}^{T}\mathbb{E}[\sigma_t^2]\left(f(\boldsymbol{x}_1) - f^\star + \frac{\alpha}{2M}\right)}\\ &+ \sqrt{10L^2\sum_{t=1}^{T}\mathbb{E}[\sigma_t^2]\ln\left(1 + \frac{L^2T}{\alpha}\right)} \,.\end{aligned}$$

By taking an $\boldsymbol{x}_k$ from $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$ randomly, we get:

$$\begin{aligned}\mathbb{E}\left[\|\nabla f(\boldsymbol{x}_k)\|^2\right] &= \mathbb{E}\left[\mathbb{E}\left[\|\nabla f(\boldsymbol{x}_k)\|^2 \middle| k\right]\right]\\ &= \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\|\nabla f(\boldsymbol{x}_t)\|^2\right],\end{aligned}$$

that completes the proof. □

Polyak-Łojasiewicz Condition. When we assume in addition that the objective function satisfies the Polyak-Łojasiewicz Condition (Karimi et al., 2016) (**H5**), we can get the linear rate in the noiseless case.

Theorem 6. *Choosing $r(\eta) = \frac{M\alpha}{2}\left(\eta - \frac{1}{M}\right)^2 + \mathcal{I}\left(\eta \in \left[0, \frac{2}{M}\right]\right)$ with $\alpha > 0$, assume* (**H1 - H5**)*, and that $\boldsymbol{g}(\boldsymbol{x}_t, \xi_t) = \nabla f(\boldsymbol{x}_t)$ for all t (i.e. there is no noise). Then feeding Algorithm 2 into Algorithm 1, yields:*

$$\mathbb{E}\left[f(\boldsymbol{x}_{T+1}) - f^\star\right] \leq \left(1 - \frac{\mu}{M}\right)^T \left[f(\boldsymbol{x}_1) - f^\star\right] .$$

Proof. From the update rule of η_t, we have that when there is no noise, $\eta_t = \frac{1}{M}$ all the time, thus:

$$\begin{aligned}
&\mathbb{E}[f(\boldsymbol{x}_{T+1}) - f^\star] \\
&\quad \leq \mathbb{E}\left[f(\boldsymbol{x}_T) - f^\star + \langle \nabla f(\boldsymbol{x}_T), \boldsymbol{x}_{T+1} - \boldsymbol{x}_T\rangle\right] \\
&\qquad + \mathbb{E}\left[\frac{M}{2}\|\boldsymbol{x}_{T+1} - \boldsymbol{x}_T\|^2\right] \\
&\quad = \mathbb{E}\left[f(\boldsymbol{x}_T) - f^\star + \left(-\eta_T + \frac{M\eta_T^2}{2}\right)\|\nabla f(\boldsymbol{x}_T)\|^2\right] \\
&\quad = \mathbb{E}\left[f(\boldsymbol{x}_T) - f^\star - \frac{1}{2M}\|\nabla f(\boldsymbol{x}_T)\|^2\right] \\
&\quad \leq \mathbb{E}\left[\left(1 - \frac{\mu}{M}\right)(f(\boldsymbol{x}_T) - f^\star)\right] \leq \cdots \leq \\
&\quad \leq \left(1 - \frac{\mu}{M}\right)^T [f(\boldsymbol{x}_1) - f^\star] . \qquad \square
\end{aligned}$$

6. Adapting Per-coordinate Stepsizes

In the previous Section, we have shown how to use the surrogate loss functions to adapt a stepsize. Another common strategy in practice is to use a *per-coordinate* stepsize. This kind of scheme is easily incorporated into our framework and we show that it can provide improved adaptivity to per-coordinate variances.

Specifically, we consider $\boldsymbol{\eta}_t$ now to be a vector in $\mathbb{R}^d$, $\boldsymbol{\eta}_t = (\eta_{t,1}, \dots, \eta_{t,d})$ and use the update $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \boldsymbol{\eta}_t \boldsymbol{g}_t$ where $\boldsymbol{\eta}_t \boldsymbol{g}_t$ now indicates coordinate-wise product $(\eta_{t,1} g_{t,1}, \dots, \eta_{t,d} g_{t,d})$. Then we define

$$\begin{aligned}
\ell_t(\boldsymbol{\eta}) &= \frac{M}{2}\|\boldsymbol{\eta}\boldsymbol{g}(\boldsymbol{x}_t, \xi_t)\|^2 - \langle \boldsymbol{\eta}\boldsymbol{g}(\boldsymbol{x}_t, \xi_t), \boldsymbol{g}(\boldsymbol{x}_t, \xi_t')\rangle \\
&= \sum_{i=1}^d \left[\frac{M}{2}\eta_i^2 g(\boldsymbol{x}_t, \xi_t)_i^2 - \eta_i g(\boldsymbol{x}_t, \xi_t)_i g(\boldsymbol{x}_t, \xi_t')_i\right] .
\end{aligned}$$

To take advantage of this scenario, we need more detail about the variance, which we encapsulate in the following assumption:

H2': The noisy gradients $\boldsymbol{g}_t$ have finite variance in each coordinate:

$$\mathbb{E}_t\left[(g(\boldsymbol{x}_t, \xi_t)_i - \nabla f(\boldsymbol{x}_t)_i)^2\right] = \sigma_{t,i}^2 .$$

Note that this assumption is not actually stronger than (**H2**) because we can define $\sigma_t^2 = \sum_{i=1}^d \sigma_{t,i}^2$. This merely provides finer-grained variable names.

Also, we make the assumption:

H3': The noisy gradients have bounded coordinate values:

$$|g(\boldsymbol{x}_t, \xi_t)_i| \leq L_i, \ |g(\boldsymbol{x}_t, \xi_t')_i| \leq L_i .$$

Now the exact same argument as for Theorem 2 yields:

Theorem 7. *Assume (**H4**) and the two noisy gradients in each round t to satisfy (**H1**) and (**H2'**). Then, for any $\eta \in \mathbb{R}^d$ with $\eta_i > 0$ for all i, the per-coordinate variant of Algorithm 1 obtains*

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^T \sum_{i=1}^d \left(\eta_i - \frac{M}{2}\eta_i^2\right)\nabla f(\boldsymbol{x}_t)_i^2\right] \\
&\quad \leq f(\boldsymbol{x}_1) - f^\star + \mathbb{E}\left[\mathrm{Regret}_T(\boldsymbol{\eta})\right] + \frac{M}{2}\sum_{i=1}^d \sum_{t=1}^T \eta_i \mathbb{E}[\sigma_{t,i}^2] .
\end{aligned}$$

With this Theorem in hand, once again all that remains is to choose the online learning algorithm. To this end, observe that we can write $\ell_t(\eta) = \sum_{i=1}^d \ell_{t,i}(\eta_i)$ where

$$\ell_{t,i}(\eta_i) = \frac{M}{2}\eta_i^2 g(x_t, \xi_t)_i^2 - \eta_i g(x_t, \xi_t)_i g(x_t, \xi_t')_i .$$

Thus, we can take our online learning algorithm to be a per-coordinate instantiation of Algorithm 2, and the total regret is simply the sum of the per-coordinate regrets. Each per-coordinate regret can be analyzed in exactly the same way as Algorithm 2, leading to

$$\begin{aligned}
&\mathrm{Regret}_T(\boldsymbol{\eta}) = \sum_{i=1}^d \mathrm{Regret}_{T,i}(\eta_i), \\
&\mathrm{Regret}_{T,i}(\eta_i) \leq \frac{M\alpha}{2}\left(\eta_i - \frac{1}{M}\right)^2 + \frac{5L_i^2}{M}\ln\left(1 + \frac{L_i^2 T}{\alpha}\right) .
\end{aligned}$$

From these inequalities we can make a per-coordinate bound on the gradient magnitudes. In words, the coordinates which have smaller variances $\sigma_{t,i}^2$ achieve smaller gradient values faster than coordinates with larger variances. Further, we preserve adaptivity to the full variance $\sum_{t=1}^T \mathbb{E}[\sigma_t^2]$ in the rate of decrease of $\|\nabla f(x)\|^2$.

Theorem 8. *Assume* (**H1**, **H2'**, **H3'**, **H4**). *Suppose we run a per-coordinate variant of Algorithm 1, with regularizer $r(\eta_i) = \frac{M\alpha}{2}\left(\eta_i - \frac{1}{M}\right)^2 + \mathcal{I}\left(\eta_i \in \left[0, \frac{2}{M}\right]\right)$ in each coordinate with $\alpha > 0$. Then, for each $i \in \{1, \dots, d\}$, we have*

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^T \nabla f(\boldsymbol{x}_t)_i^2\right] \\
&\quad \leq 2M\left(f(\boldsymbol{x}_1) - f^\star + \sum_{i=1}^d \frac{5L_i^2}{M}\ln\left(1 + \frac{L_i^2 T}{\alpha}\right)\right) \\
&\qquad + \sqrt{2M\sum_{t=1}^T \mathbb{E}[\sigma_{t,i}^2]\left(f(\boldsymbol{x}_1) - f^\star + \frac{d\alpha}{2M}\right)} \\
&\qquad + \sqrt{10\sum_{t=1}^T \mathbb{E}[\sigma_{t,i}^2]\sum_{i=1}^d L_i^2 \ln\left(1 + \frac{L_i^2 T}{\alpha}\right)} \\
&\qquad + (d-1)\alpha .
\end{aligned}$$

Further, with $\sigma_t = \sum_{t=1}^{T} \sigma_{t,i}^2$ it also holds

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T} \|\nabla f(\boldsymbol{x}_t)\|^2\right] \\
&\quad \leq 2M\left(f(\boldsymbol{x}_1) - f^\star + \sum_{i=1}^{d} \frac{5L_i^2}{M} \ln\left(1 + \frac{L_i^2 T}{\alpha}\right)\right) \\
&\quad\quad + \sqrt{2M \sum_{t=1}^{T} \mathbb{E}[\sigma_t^2] \left(f(\boldsymbol{x}_1) - f^\star + \frac{d\alpha}{2M}\right)} \\
&\quad\quad + \sqrt{10 \sum_{t=1}^{T} \mathbb{E}[\sigma_t^2] \sum_{i=1}^{d} L_i^2 \ln\left(1 + \frac{L_i^2 T}{\alpha}\right)} .
\end{aligned}$$

Proof. The proof is nearly identical to that of Theorem 4. We have

$$\begin{aligned}
&\mathbb{E}\left[\sum_{i=1}^{d} \left(\eta_i - \frac{M}{2}\eta_i^2\right) \sum_{t=1}^{T} \nabla f(\boldsymbol{x}_t)_i^2\right] \\
&\quad \leq f(\boldsymbol{x}_1) - f^\star + \frac{M\alpha}{2} \sum_{i=1}^{d} \left(\eta_i - \frac{1}{M}\right)^2 \\
&\quad + \sum_{i=1}^{d} \frac{5L_i^2}{M} \ln\left(1 + \frac{L_i^2 T}{\alpha}\right) + \sum_{i=1}^{d} \frac{M\eta_i^2}{2} \sum_{t=1}^{T} \mathbb{E}[\sigma_{t,i}^2] .
\end{aligned}$$

Let $A_i \triangleq \mathbb{E}[\sum_{t=1}^{T} \nabla f(\boldsymbol{x}_t)_i^2]$ and choosing η_i by the same strategy in Theorem 4 as $\frac{\alpha + A_i}{M(\alpha + A_i + \sum_{t=1}^{T} \mathbb{E}[\sigma_{t,i}^2])}$ to obtain

$$\begin{aligned}
&\sum_{i=1}^{d} \frac{(\alpha + A_i)^2}{2M\left(\alpha + A_i + \sum_{t=1}^{T} \mathbb{E}[\sigma_{t,i}^2]\right)} \\
&\quad \leq f(\boldsymbol{x}_1) - f^\star + \frac{d\alpha}{2M} + \sum_{i=1}^{d} \frac{5L_i^2}{M} \ln\left(1 + \frac{L_i^2 T}{\alpha}\right) .
\end{aligned}$$

Now, the first statement of the Theorem follows by observing that each term on the LHS is non-negative so that the sum can be lower-bounded by any individual term. For the second statement, define

$$\begin{aligned}
Q_i &= \frac{(\alpha + A_i)^2}{2M\left(\alpha + A_i + \sum_{t=1}^{T} \mathbb{E}[\sigma_{t,i}^2]\right)} \\
Q &= f(\boldsymbol{x}_1) - f^\star + \frac{d\alpha}{2M} + \sum_{i=1}^{d} \frac{5L_i^2}{M} \ln\left(1 + \frac{L_i^2 T}{\alpha}\right),
\end{aligned}$$

so that $\sum_{i=1}^{d} Q_i \leq Q$. By the quadratic formula and definition of Q_i, we have

$$A_i = 2MQ_i + \sqrt{2MQ_i \sum_{t=1}^{T} \mathbb{E}[\sigma_{t,i}^2]} - \alpha .$$

Thus,

$$\begin{aligned}
\sum_{i=1}^{d} A_i &= 2MQ - d\alpha + \sum_{i=1}^{d} \sqrt{2MQ_i \sum_{t=1}^{T} \mathbb{E}[\sigma_{t,i}^2]} \\
&\leq 2MQ - d\alpha + \sqrt{2M} \sqrt{\sum_{i=1}^{d} Q_i} \sqrt{\sum_{i=1}^{d} \sum_{t=1}^{T} \mathbb{E}[\sigma_{t,i}^2]} \\
&= 2MQ - d\alpha + \sqrt{2MQ \sum_{t=1}^{T} \mathbb{E}[\sigma_t^2]} .
\end{aligned}$$

From which the second statement follows. □

7. Experiments

SGD is widely known to enjoy good empirical properties, but our learning rate schedule is very unique, so to validate our theoretical findings, we experiment on fitting a classification model on the adult (a9a) dataset from the LibSVM website (Chang & Lin, 2001). The objective function is

$$f(\boldsymbol{x}) := \frac{1}{m} \sum_{i=1}^{m} \phi(\boldsymbol{a}_i^\top \boldsymbol{x} - y_i),$$

where $\phi(\theta) = \frac{\theta^2}{1+\theta^2}$, and $(\boldsymbol{a}_i, y_i)$ are the couples feature vector/label. The loss function ϕ is non-convex, 1-Lipschitz and 2-smooth w.r.t. the ℓ_2 norm.

We consider the minimization problem with respect to all training samples. Also, as the dataset is imbalanced towards the group with annual income less than 50K, we subsample that group to balance the dataset, which results in 15682 samples with 123 features each. In addition, we append a constant element to each sample feature vector to introduce a constant bias. $\boldsymbol{x}_1$ is initialized to be all zeros. For each setting, we repeat the experiment independently but with the same initialization for 5 times, and plot the average of the relevant quantities. In this setting, the noise on the gradient is generated by the use of minibatches.

The learning rates of AdaGrad Global (a single learning rate is used for all dimensions), SGD, and Adam are selected as the ones giving the best convergence rate when the full batch scheme, namely zero noise, is employed, and are shown in the legend. We take the reciprocal of SGD's best learning rate as the parameter M for SGDOL Global, and we set $\alpha = 10$ without any tuning based on our discussion on the influence of α in Section 5.

We compare these four algorithms on three different minibatch sizes, namely different noise scales: using all samples, 50 i.i.d. samples, or 1 random sample for evaluating the gradient at a point. We report the results in Figure 1. In each column, the top plot shows $\mathbb{E}[\|\nabla f(\boldsymbol{x}_k)\|^2]$ vs. number of

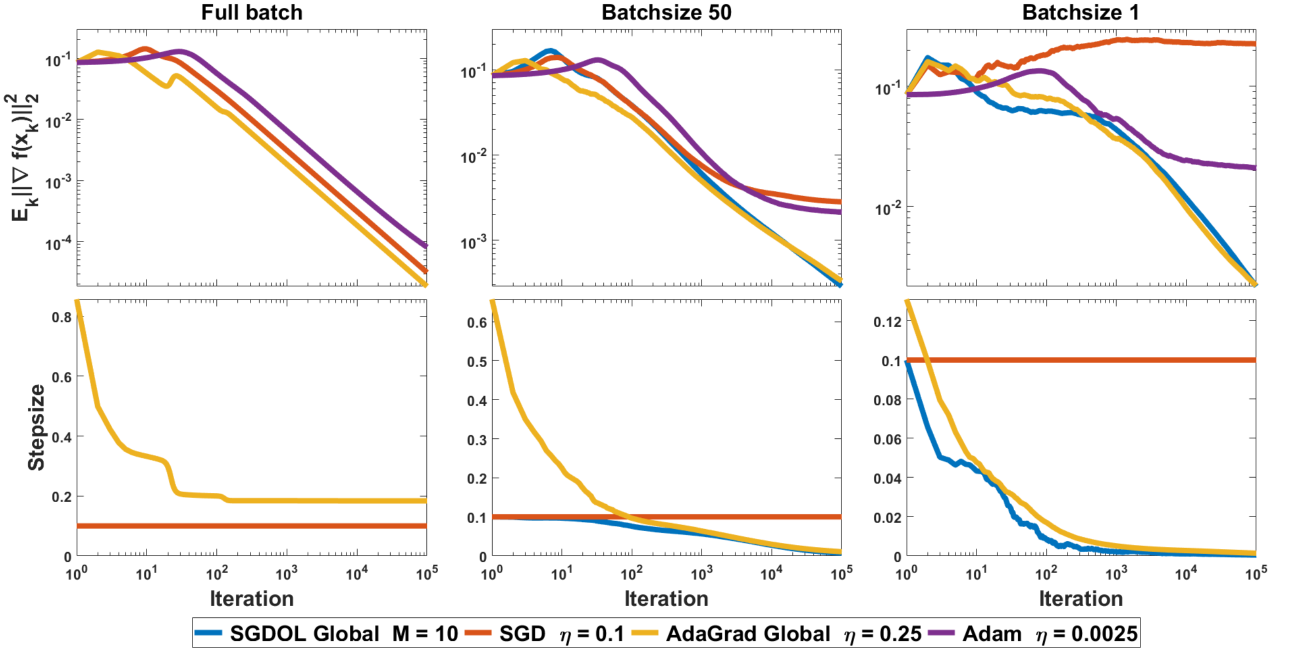

Figure 1. Comparison of performance of SGDOL Global, SGD, AdaGrad Global, Adam on a non-linear classification model with different minibatch sizes.

iterations, whereas the bottom one is the per-round stepsizes on each case. Note that there is no per-round stepsize for Adam. The x-axis in all figures, and the y-axis in the top three are logarithmic.

As can be seen, the stepsize of SGDOL is the same as SGD at first, but gradually decreases automatically. Also, the larger the noise, the sooner the decreasing phase starts. The decrease of the learning rate makes the convergence of SGDOL possible. In particular, SGDOL recovers the performance of SGD in the noiseless case, while it allows convergence in the noisy cases through an automatic decrease of the stepsizes. AdaGrad also enjoys nice convergence, and is comparable to ours. In contrast, when noise exists, after reaching a proximity of a stationary point, SGD and Adam oscillates thereafter without converging, and the value it oscillates around depends on the variance of the noise. This underlines the superiority of the surrogate losses, rather than choosing a stepsize based on a worst-case convergence rate bound.

More experiments can be found in the Appendix.

8. Conclusions and Future Work

We have presented a novel way to cast the problem of adaptive stepsize selection for the stochastic optimization of smooth (non-convex) functions as an online convex optimization problem with a simple quadratic convex surrogate. The reduction goes through the use of novel surrogate convex losses. This framework allows us to import the rich literature of no-regret online algorithms to learn stepsizes on the fly. We exemplified the power of this method with the SGDOL algorithm which enjoys an optimal convergence guarantee for any level of noise, without the need to estimate the noise nor tune the stepsizes. Moreover, we recover linear convergence rates under the PL-condition. The overall price to pay is a factor of 2 in the computation of the gradients. We also presented a per-coordinate version of SGDOL that achieves faster convergence on the coordinates with less noise.

We feel that we have barely scratched the surface of what might be possible with these surrogate losses. Hence, future work will focus on extending their use to other scenarios. For example, we plan to use it in locally private SGD algorithms where additional noise is added on the gradients to ensure privacy of the data (Song et al., 2013). We are also interested in investigating whether adding convexity would give us better results to recover SGD's performance. Another potential direction is to eliminate the need of knowing M, e.g. by automatically adapting to it on the fly.

9. Acknowledgment

This material is based upon work supported by the National Science Foundation under grant no. 1740762 "Collaborative Research: TRIPODS Institute for Optimization and Learning".

References

Abernethy, J. D., Hazan, E., and Rakhlin, A. Competing in the dark: An efficient algorithm for bandit linear optimization. In Servedio, R. A. and Zhang, T. (eds.), *Proc. of the 21st Annual Conference on Learning Theory, COLT*, pp. 263–274. Omnipress, 2008.

Abernethy, J. D., Hazan, E., and Rakhlin, A. Interior-point methods for full-information and bandit online learning. *IEEE Trans. Information Theory*, 58(7):4164–4175, 2012.

Baydin, A. G., Cornish, R., Rubio, D. M., Schmidt, M., and Wood, F. Online learning rate adaptation with hypergradient descent. In *Sixth International Conference on Learning Representations (ICLR), Vancouver, Canada, April 30 – May 3, 2018*, 2018.

Cesa-Bianchi, N. and Lugosi, G. *Prediction, learning, and games*. Cambridge University Press, 2006.

Chang, C.-C. and Lin, C.-J. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. Technical Report 2010-24, UC Berkeley Electrical Engineering and Computer Science, 2010. Available at http://cs.berkeley.edu/~jduchi/projects/DuchiHaSi10.pdf.

Duchi, J. C., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

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

Karimi, H., Nutini, J., and Schmidt, M. Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 795–811. Springer, 2016.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Koolen, W. M., van Erven, T., and Grünwald, P. Learning the learning rate for prediction with expert advice. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 27*, pp. 2294–2302. Curran Associates, Inc., 2014.

Li, X. and Orabona, F. On the convergence of stochastic gradient descent with adaptive stepsizes. In *Proc. of the 22nd International Conference on Artificial Intelligence and Statistics, AISTATS*, 2019.

McMahan, H. B. A survey of algorithms and analysis for adaptive online learning. *The Journal of Machine Learning Research*, 18(1):3117–3166, 2017.

Mohri, M. and Yang, S. Accelerating online convex optimization via adaptive prediction. In Gretton, A. and Robert, C. C. (eds.), *Proc. of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS*, volume 51 of *Proceedings of Machine Learning Research*, pp. 848–856, Cadiz, Spain, 09–11 May 2016. PMLR.

Nesterov, Y. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer, 2003.

Polyak, B. T. *Introduction to Optimization*. Optimization Software Inc, New York, 1987.

Reddi, S. J., Kale, S., and Kumar, S. On the convergence of Adam and beyond. In *International Conference on Learning Representations*, 2018.

Rosenbrock, H. H. An automatic method for finding the greatest or least value of a function. *The Computer Journal*, 3(3):175–184, 1960.

Schraudolph, N. N. Local gain adaptation in stochastic gradient descent. In *In Proc. Intl. Conf. Artificial Neural Networks*, pp. 569–574. IEE, London, 1999.

Shalev-Shwartz, S. Online learning: Theory, algorithms, and applications. Technical report, The Hebrew University, 2007. PhD thesis.

Song, S., Chaudhuri, K., and Sarwate, A. D. Stochastic gradient descent with differentially private updates. In *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*, pp. 245–248. IEEE, 2013.

Tieleman, T. and Hinton, G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 2012.

van Erven, T. and Koolen, W. M. MetaGrad: Multiple learning rates in online learning. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29*, pp. 3666–3674. Curran Associates, Inc., 2016.

Ward, R., Wu, X., and Bottou, L. AdaGrad stepsizes: Sharp convergence over nonconvex landscapes, from any initialization. *arXiv preprint arXiv:1806.01811*, 2018.

Zeiler, M. D. ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

10. Appendix

10.1. 2D Rosenbrock Function

The popular 2-D Rosenbrock benchmark (Rosenbrock, 1960) takes the form:

$$f(x,y) := (1-x)^2 + 100(y-x^2)^2 \ .$$

It is non-convex and has one global minimum at $x = y = 1$.

To add stochasticity, we apply additive white Gaussian noise to each gradient. To have a robust estimate of the performance, we repeat each experiment independently with the same parameters for 40 times and take the average.

We compare the performance of SGDOL Global with a bunch of popular adaptive optimization algorithms on the Rosenbrock function with 3 levels of added noise: zero noise, small noise ($\sigma = 0.2$), and large noise ($\sigma = 5$). The competitors are: SGD, AdaGrad Global (Duchi et al., 2010) (with one learning rate for all dimensions), AdaGrad Coordinate (Duchi et al., 2010) (with one learning rate for each dimension), Adam (Kingma & Ba, 2015), RMSProp (Tieleman & Hinton, 2012), AdaDelta (Zeiler, 2012), AMSGrad (Reddi et al., 2018), and Hypergradient (Baydin et al., 2018). Also, we test the performance of the stepsize proposed by Ghadimi & Lan (2013), denoted by SGD GL, given that in this synthetic experiment we know all the relevant quantities. We stress the fact that in the real-world setting this kind of stepsize cannot be used. We select the stepsize of all optimization algorithms except for SGDOL Global and SGD GL to be the one giving best convergence rate when running on the objective function with zero noise added. We choose M of SGDOL to be the reciprocal of SGD's best learning rate which happens to be very close to the smoothness at the optimal point, 1002. We set $x_1 = y_1 = 0$.

In Figure 2, the top plots show $\mathbb{E}[\|\nabla f(\boldsymbol{x}_k)\|^2]$ vs. number of iterations, the middle ones reflect the curve of the optimality gap $f(\boldsymbol{x}_t) - f^*$ at each round t since we know $f^* = 0$, whereas the bottom ones are the per-round stepsizes on each case. Note that there is no per-round stepsize for Adam. In Figure 3, the top plots show $\mathbb{E}[\|\nabla f(\boldsymbol{x}_k)\|^2]$ vs. number of iterations, and the bottom ones reflect the curve of the optimality gap $f(\boldsymbol{x}_t) - f^*$ at each round t. The x-axis in all figures are logarithmic, and the y-axis in all figures except for those showing stepsizes are logarithmic.

The behavior of both the curve of $\mathbb{E}[\|\nabla f(\boldsymbol{x}_k)\|^2]$ vs. number of iterations and the curve of stepsizes are similar to Figure 1. And the behavior of the curve of optimality gap is similar to that of the curve of $\mathbb{E}[\|\nabla f(\boldsymbol{x}_k)\|^2]$ vs. number of iterations.

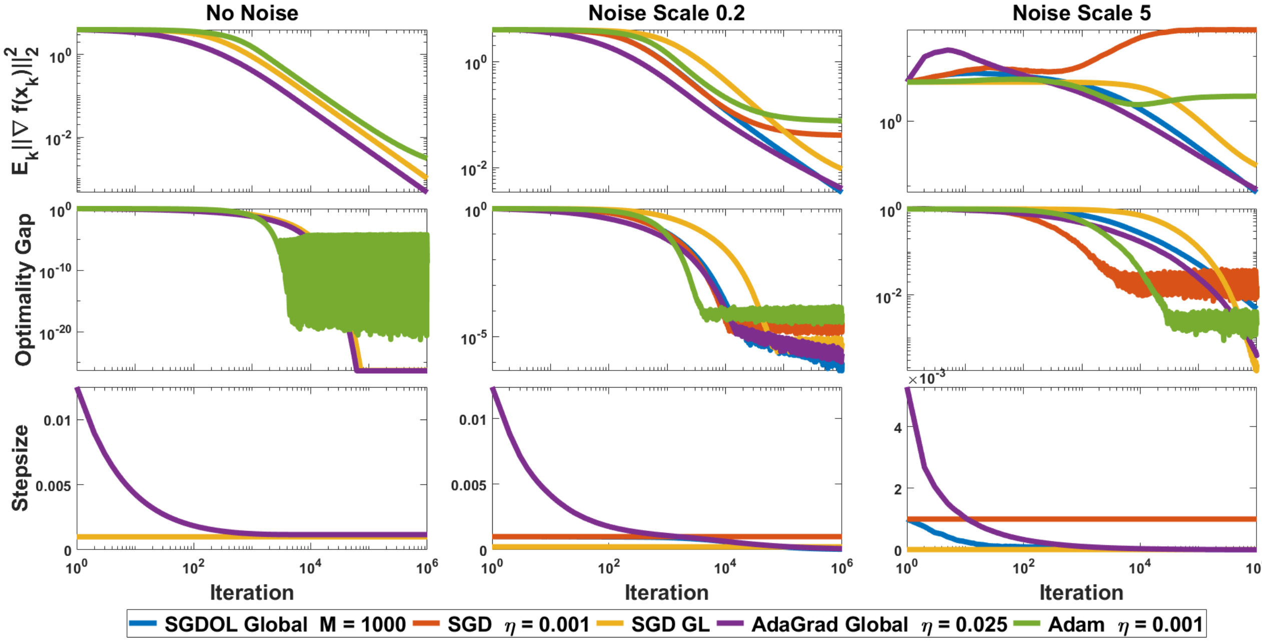

Figure 2. Comparison of SGDOL Global, SGD, SGD GL, AdaGrad Global, and Adam on the 2D Rosenbrock function with different levels of noise.

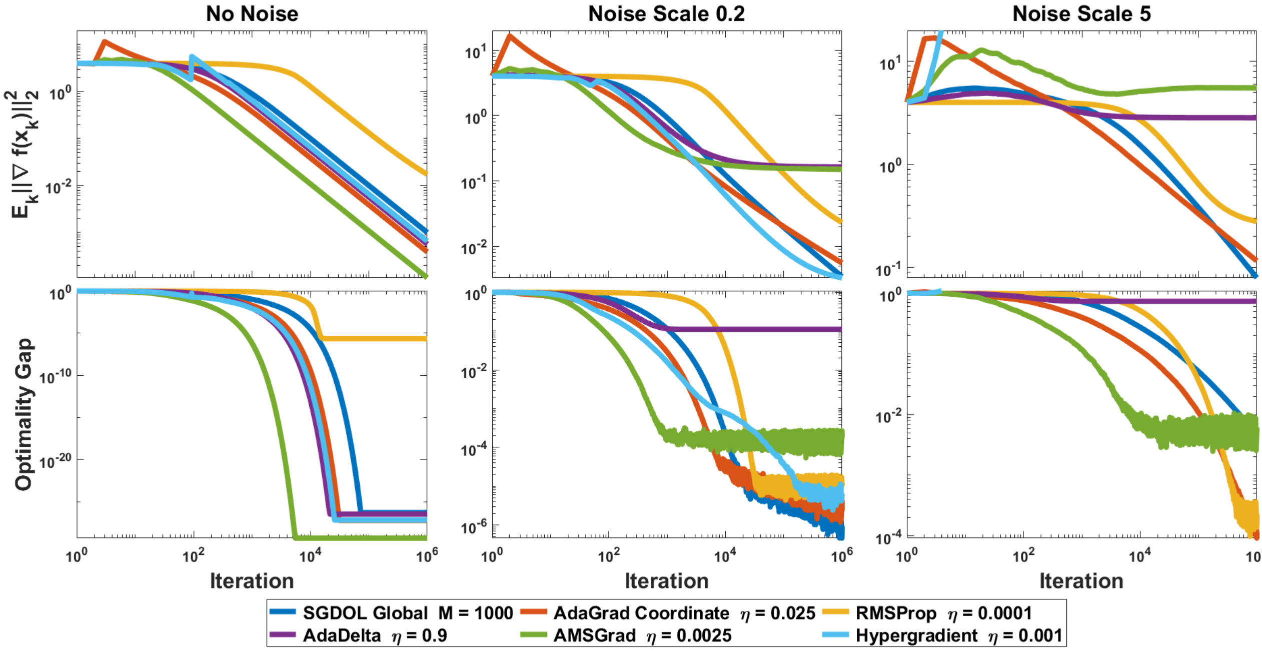

Figure 3. Comparison of SGDOL Global, AdaGrad Coordinate, RMSProp, AdaDelta, AMSGrad, and Hypergradient on the 2D Rosenbrock function with different levels of noise.

10.2. Other Results for Fitting a Non-Linear Classification Model

Here we show the results for comparison between SGDOL Global and other optimization algorithms listed in the above subsection but applied to the classification task introduced in Section 7. Note that here we don't know f^* so we don't report the curve of the optimality gap. The comparison shown in Figure 4 is similar to what is reported in Figure 1.

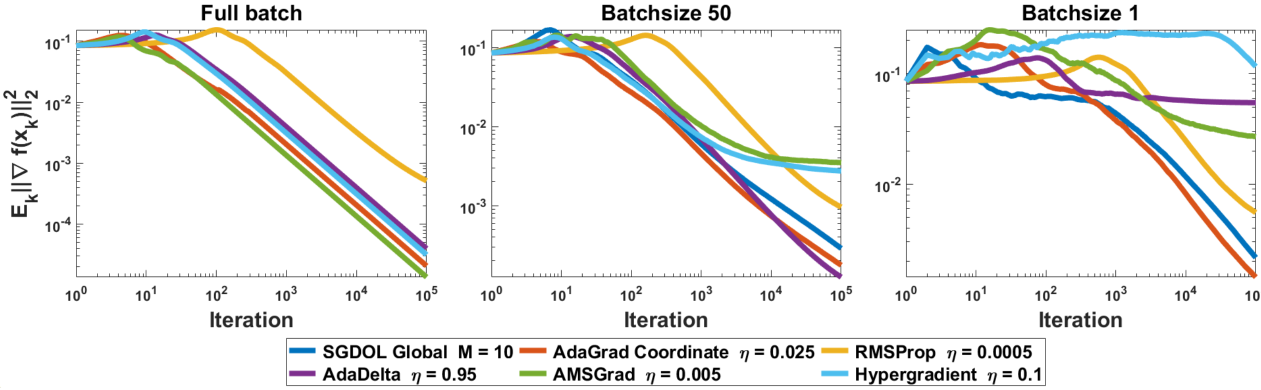

Figure 4. Comparison of performance of SGDOL Global, AdaGrad Coordinate, RMSProp, AdaDelta, AMSGrad, and Hypergradient on a non-linear classification model with different minibatch sizes.