Beyond the Regret Minimization Barrier: Optimal Algorithms for Stochastic Strongly-Convex Optimization

Elad Hazan*
Princeton University
Princeton, NJ 08540
ehazan@cs.princeton.edu

Satyen Kale
Yahoo! Labs New York City
New York, NY 10036
satyen@yahoo-inc.com

May 17, 2015

Abstract

We give novel algorithms for stochastic strongly-convex optimization in the gradient oracle model which return a $O(\frac{1}{T})$-approximate solution after T iterations. The first algorithm is deterministic, and achieves this rate via gradient updates and historical averaging. The second algorithm is randomized, and is based on pure gradient steps with a random step size.

This rate of convergence is optimal in the gradient oracle model. This improves upon the previously known best rate of $O(\frac{\log(T)}{T})$, which was obtained by applying an online strongly-convex optimization algorithm with regret $O(\log(T))$ to the batch setting.

We complement this result by proving that any algorithm has expected regret of $\Omega(\log(T))$ in the online stochastic strongly-convex optimization setting. This shows that any online-to-batch conversion is inherently suboptimal for stochastic strongly-convex optimization. This is the first formal evidence that online convex optimization is strictly more difficult than batch stochastic convex optimization.[1]

1 Introduction

Stochastic convex optimization has an inherently different flavor than standard convex optimization. In the stochastic case, a crucial resource is the number of data samples from the function to be optimized. This resource limits the precision of the output: given few samples there is simply not enough information to compute the optimum up to a certain precision. The error arising from this lack of information is called the *estimation error*.

The estimation error is independent of the choice of optimization algorithm, and it is reasonable to choose an optimization method whose precision is of the same order of magnitude as the sampling error: lesser precision is suboptimal, whereas much better precision is pointless. This issue is extensively discussed by Bottou and Bousquet [2007] and by Shalev-Shwartz and Srebro [2008]. This makes first-order methods ideal for stochastic convex optimization: their error decreases as a

*Supported by ISF Grant 810/11 and the Microsoft-Technion EC Center.

[1] An extended abstract of this work appeared in COLT 2011 [Hazan and Kale, 2011]. In this version we have included a new randomized algorithm which is based on pure gradient steps, and extended the results to strong convexity with respect to general norms.

polynomial in the number of iterations, usually make only one iteration per data point, and each iteration is extremely efficient.

In this paper we consider first-order methods for stochastic convex optimization. Formally, the problem of stochastic convex optimization is the minimization of a convex (possibly non-smooth) function on a convex domain $\mathcal{K}$:

$$\min_{\mathbf{x}\in\mathcal{K}} F(\mathbf{x}).$$

The stochasticity is in the access model: the only access to F is via a stochastic subgradient oracle, which given any point $\mathbf{x} \in \mathcal{K}$, produces a random vector $\hat{\mathbf{g}}$ whose expectation is a subgradient of F at the point $\mathbf{x}$, i.e., $\mathbb{E}[\hat{\mathbf{g}}] \in \partial F(\mathbf{x})$, where $\partial F(\mathbf{x})$ denotes the subdifferential set of F at $\mathbf{x}$.

We stress that F may be non-smooth. This is important for the special case when $F(\mathbf{x}) = \mathbb{E}_Z[f(\mathbf{x}, Z)]$ (the expectation being taken over a random variable Z), where for every fixed z, $f(\mathbf{x}, z)$ is a convex function of $\mathbf{x}$. The goal is to minimize F while given a sample $z_1, z_2, \ldots$ drawn independently from the unknown distribution of Z. A prominent example of this formulation is the problem of support vector machine (SVM) training [see Shalev-Shwartz et al., 2009]. For SVM training, the function F is convex but non-smooth.

An algorithm for stochastic convex optimization is allowed a budget of T calls to the gradient oracle. It sequentially queries the gradient oracle at consecutive points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$, and produces an approximate solution $\bar{\mathbf{x}}$. The *rate of convergence* of the algorithm is the expected excess cost of the point $\bar{\mathbf{x}}$ over the optimum, i.e. $\mathbb{E}[F(\bar{\mathbf{x}})] - \min_{\mathbf{x}\in\mathcal{K}} F(\mathbf{x})$, where the expectation is taken over the randomness in the gradient oracle and the internal random seed of the algorithm. The paramount parameter for measuring this rate is in terms of T, the number of gradient oracle calls.

Our first and main contribution is the first algorithm to attain the optimal rate of convergence in the case where F is λ-strongly convex, and the gradient oracle is G-bounded (see precise definitions in Section 2.1). After T gradient updates, the algorithm returns a solution which is $O(\frac{1}{T})$-close in cost to the optimum. Formally, we prove the following theorem.

Theorem 1. *Assume that F is λ-strongly convex and the gradient oracle is G-bounded. Then there exists a deterministic algorithm that after at most T gradient updates returns a vector $\bar{\mathbf{x}}$ such that for any $\mathbf{x}^\star \in \mathcal{K}$ we have*

$$\mathbb{E}[F(\bar{\mathbf{x}})] - F(\mathbf{x}^\star) \leq O\left(\frac{G^2}{\lambda T}\right).$$

This matches the lower bound of Agarwal et al. [2012] up to constant factors.

The previously best known rate was $O(\frac{\log(T)}{T})$, and follows by converting a more general online convex optimization algorithm of Hazan et al. [2007] to the batch setting. This standard online-to-batch reduction works as follows. In the online convex optimization setting, in each round $t = 1, 2, \ldots, T$, a decision maker (represented by an algorithm $\mathcal{A}$) chooses a point $\mathbf{x}_t$ in convex domain $\mathcal{K}$, and incurs a cost $f_t(\mathbf{x}_t)$ for an adversarially chosen convex cost function f_t. In this model performance is measured by the *regret*, defined as

$$\text{Regret}(\mathcal{A}) \;:=\; \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x}\in\mathcal{K}} \sum_{t=1}^{T} f_t(\mathbf{x}). \tag{1}$$

A regret minimizing algorithm is one that guarantees that the regret grows like $o(T)$. Given such an algorithm, one can perform batch stochastic convex optimization by setting f_t to be the function

$f(\cdot, z_t)$. A simple analysis then shows that the cost of the average point, $\bar{\mathbf{x}} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t$, converges to the optimum cost at the rate of the *average* regret, which converges to zero.

The best previously known convergence rates for stochastic convex optimization were obtained using this online-to-batch reduction, and thus these rates were equal to the average regret of the corresponding online convex optimization algorithm. While it is known that for general convex optimization, this online-to-batch reduction gives the optimal rate of convergence, such a result was not known for stochastic strongly-convex functions. In this paper we show that for stochastic strongly-convex functions, minimizing regret is strictly more difficult than batch stochastic strongly-convex optimization.

More specifically, the best known regret bound for λ-strongly-convex cost functions with gradients bounded in norm by G is $O(\frac{G^2\log(T)}{\lambda})$ [Hazan et al., 2007]. This regret bound holds even for adversarial, not just stochastic, strongly-convex cost functions. A matching lower bound was obtained by Takimoto and Warmuth [2000] for the adversarial setting.

Our second contribution in this paper is a matching lower bound for strongly-convex cost functions that holds *even in the stochastic setting*, i.e., if the cost functions are sampled i.i.d from an unknown distribution. Formally:

Theorem 2. *For any online decision-making algorithm $\mathcal{A}$, there is a distribution over λ-strongly-convex cost functions with norms of gradients bounded by G such that*

$$\mathbb{E}[\mathit{Regret}(\mathcal{A})] = \Omega\left(\frac{G^2\log(T)}{\lambda}\right).$$

Hence, our new rate of convergence of $O(\frac{G^2}{\lambda T})$ is the first to separate the complexity of stochastic and online strongly-convex optimization. The following table summarizes our contribution with respect to the previously known bounds. The setting is assumed to be stochastic λ-strongly-convex functions with expected subgradient norms bounded by G.

	Previously known bound	New bound here
Convergence rate	$O\left(\frac{G^2\log(T)}{\lambda T}\right)$ [Hazan et al., 2007]	$O\left(\frac{G^2}{\lambda T}\right)$
Regret	$\Omega\left(\frac{G^2}{\lambda}\right)$ (Trivial bound[2])	$\Omega\left(\frac{G^2\log(T)}{\lambda}\right)$

We also sharpen our results: Theorem 1 bounds the expected excess cost of the solution over the optimum by $O(\frac{1}{T})$. We can also show high probability bounds. In situations where it is possible to evaluate F at any given point efficiently, simply repeating the algorithm a number of times and taking the best point found bounds the excess cost by $O(\frac{G^2\log(\frac{1}{\delta})}{\lambda T})$ with probability at least $1-\delta$. In more realistic situations where it is not possible to evaluate F efficiently, we can still modify the algorithm so that with high probability, the actual excess cost of the solution is bounded by $O(\frac{\log\log(T)}{T})$:

[2]The lower bound follows from the work of Agarwal et al. [2012], but a simple lower bound example is the following. Consider an adversary that plays a fixed function, either $\frac{\lambda}{2}x^2$ or $\frac{\lambda}{2}(x-\frac{G}{\lambda})^2$, for all rounds, with $\mathcal{K} = [0, \frac{G}{\lambda}]$. On the first round, the loss of the algorithm's point x_1 for one of these two functions is at least $\frac{G^2}{8\lambda}$: this is because $\frac{\lambda}{2}x_1^2 + \frac{\lambda}{2}(x_1-\frac{G}{\lambda})^2 = \lambda(x_1-\frac{G}{2\lambda})^2 + \frac{G^2}{4\lambda} \geq \frac{G^2}{4\lambda}$. Clearly the best point in hindsight has 0 loss, so the regret of the algorithm is at least $\frac{G^2}{8\lambda}$ for one of the two functions.

Theorem 3. *Assume that F is λ-strongly convex, and the gradient oracle is strongly G-bounded. Then for any $\delta > 0$, there exists an algorithm that after at most T gradient updates returns a vector $\bar{\mathbf{x}}$ such that with probability at least $1 - \delta$, for any $\mathbf{x}^\star \in \mathcal{K}$ we have*

$$F(\bar{\mathbf{x}}) - F(\mathbf{x}^\star) \leq O\left(\frac{G^2(\log(\frac{1}{\delta}) + \log\log(T))}{\lambda T}\right).$$

The algorithm attaining the convergence rate claimed in Theorem 1 is deterministic, albeit not a pure gradient-step algorithm: it proceeds in epochs; each epoch performs gradient steps only. However, the initialization of any epoch is given by the *average* iterate of the previous epoch. A natural question that arises is whether there exists a pure gradient step algorithm, that performs only gradient steps with carefully controlled step size. We also give an algorithm achieving this (although using random step sizes).

1.1 Related Work

For an in depth discussion of first-order methods, the reader is referred to the book by Bertsekas [1999].

The study of lower bounds for stochastic convex optimization was undertaken by Nemirovski and Yudin [1983], and recently extended and refined by Agarwal et al. [2012].

Online convex optimization was introduced by Zinkevich [2003]. Optimal lower bounds for the convex case, even in the stochastic setting, of $\Omega(\sqrt{T})$ are simple and given in the book by Cesa-Bianchi and Lugosi [2006]. For exp-concave cost functions, Ordentlich and Cover [1998] give a $\Omega(\log T)$ lower bound on the regret, even when the cost functions are sampled according to a known distribution. For strongly convex functions, no non-trivial stochastic lower bound was known. Takimoto and Warmuth [2000] give a $\Omega(\log T)$ lower bound in the regret for adaptive adversaries. Abernethy et al. [2009] put this lower bound in a general framework for min-max regret minimization.

It has been brought to our attention that Juditsky and Nesterov [2010] and Ghadimi and Lan [2010] have recently published technical reports that have very similar results to ours, and also obtain an $O(\frac{1}{T})$ convergence rate. Our work was done independently and a preliminary version was published on arXiv [Hazan and Kale, 2010] before the technical reports of Juditsky and Nesterov [2010] and Ghadimi and Lan [2010] were made available. Note that the high probability bound in this paper has better dependence on T than the result of Ghadimi and Lan [2010]: we lose an additional $\log\log T$ factor vs. the $\log^2 \log T$ factor lost in the paper of Ghadimi and Lan [2010]). Our lower bound on the regret for stochastic online strongly-convex optimization is entirely new.

Following our work, a number of other works have appeared which obtain the optimal $O(\frac{1}{T})$ convergence rate using other methods. Rakhlin et al. [2012] show that for strongly convex cost functions that are also smooth, a $O(\frac{1}{T})$ rate is attainable by vanilla stochastic gradient descent (SGD), and further that SGD with special averaging of the last iterates recovers this optimal rate even in the non-smooth case. They also show that empirically, our algorithm indeed performs better than vanilla averaged SGD; though it is slightly worse than the suffix-averaging variant of SGD in their paper. Shamir and Zhang [2013] later considered the last iterate of vanilla SGD, for which they show $O(\frac{\log T}{T})$ convergence rate in the strongly convex case. This complements the bound of $O(\frac{1}{T})$ on the suboptimality of a random iterate from the random SGD variant we give in this paper.

2 Setup and Background

In this section we give basic definitions and describe the optimization framework for our results.

2.1 Stochastic Convex Optimization

We work in a Euclidean space[3] $\mathcal{H}$ with norm $\|\cdot\|$ with the dual norm $\|\cdot\|_\star$. For $\mathbf{x}, \mathbf{w} \in \mathcal{H}$, let $\mathbf{w} \cdot \mathbf{x}$ denote their inner product. For a convex and differentiable function f, we denote by ∇f its gradient at a given point. Consider the setting of stochastic convex optimization of a convex (possibly non-smooth) function F over a convex (possibly non-compact) set $\mathcal{K} \subseteq \mathcal{H}$. Let $\mathbf{x}^\star$ be a point in $\mathcal{K}$ where F is minimized. We make the following assumptions:

1. We assume that we have a convex and differentiable function $\mathcal{R} : \mathcal{H} \to \mathbb{R} \cup \{-\infty, +\infty\}$ with its corresponding Bregman divergence defined as:

   $$B_{\mathcal{R}}(\mathbf{y}, \mathbf{x}) \;:=\; \mathcal{R}(\mathbf{y}) - \mathcal{R}(\mathbf{x}) - \nabla\mathcal{R}(\mathbf{x}) \cdot (\mathbf{y} - \mathbf{x}).$$

   By direct substitution, this definition implies that for any vectors $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathcal{H}$,

   $$(\nabla\mathcal{R}(\mathbf{z}) - \nabla\mathcal{R}(\mathbf{y})) \cdot (\mathbf{x} - \mathbf{y}) = B_{\mathcal{R}}(\mathbf{x}, \mathbf{y}) - B_{\mathcal{R}}(\mathbf{x}, \mathbf{z}) + B_{\mathcal{R}}(\mathbf{y}, \mathbf{z}). \tag{2}$$

   We assume further that $\mathcal{R}$ is strongly-convex w.r.t. the norm $\|\cdot\|$, i.e., for any two points $\mathbf{x}, \mathbf{y} \in \mathcal{H}$, we have

   $$B_{\mathcal{R}}(\mathbf{y}, \mathbf{x}) \;\geq\; \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|^2.$$

2. We assume that F is **λ-strongly convex w.r.t.** $B_{\mathcal{R}}$: i.e., for any two points $\mathbf{x}, \mathbf{y} \in \mathcal{K}$ and any $\alpha \in [0, 1]$, we have

   $$F(\alpha\mathbf{x} + (1-\alpha)\mathbf{y}) \;\leq\; \alpha F(\mathbf{x}) + (1-\alpha)F(\mathbf{y}) - \lambda\alpha(1-\alpha)B_{\mathcal{R}}(\mathbf{y}, \mathbf{x}).$$

   A sufficient condition for F to be λ-strongly-convex w.r.t. $B_{\mathcal{R}}$ is if $F(\mathbf{x}) = \mathbb{E}_Z[f(\mathbf{x}, Z)]$ and $f(\cdot, z)$ is λ-strongly-convex w.r.t. $B_{\mathcal{R}}$ for every z in the support of Z.

   This implies F satisfies the following inequality:

   $$F(\mathbf{x}) - F(\mathbf{x}^\star) \;\geq\; \lambda B_{\mathcal{R}}(\mathbf{x}^\star, \mathbf{x}). \tag{3}$$

   This follows by setting $\mathbf{y} = \mathbf{x}^\star$, dividing by α, taking the limit as $\alpha \to 0^+$, and using the fact that $\mathbf{x}^\star$ is the minimizer of F. This inequality holds even if $\mathbf{x}^\star$ is on the boundary of $\mathcal{K}$. In fact, (3) is the *only* requirement on the strong convexity of F for the analysis to work; we will simply assume that (3) holds.

3. Assume that we have a stochastic subgradient oracle for F, i.e., we have black-box access to an algorithm that computes an unbiased estimator $\hat{\mathbf{g}}$ of some subgradient of F at any point $\mathbf{x}$, i.e., $\mathbb{E}[\hat{\mathbf{g}}] \in \partial F(\mathbf{x})$. We assume that each call to the oracle uses randomness that is independent of all previously made calls. Further, we assume that at any point $\mathbf{x} \in \mathcal{K}$, the stochastic subgradient $\hat{\mathbf{g}}$ output by the oracle satisfies one of the assumptions below:

[3] In this paper, we work in a Euclidean space for simplicity. Our results extend without change to any real Banach space $\mathcal{B}$ with norm $\|\cdot\|$ with the dual space $\mathcal{B}^\star$ and the dual norm $\|\cdot\|_\star$, with the additional assumption that $\mathcal{K}$ is compact.

(a) $\mathbb{E}[\|\hat{\mathbf{g}}\|_\star^2] \leq G^2$.

(b) $\mathbb{E}\left[\exp\left(\frac{\|\hat{\mathbf{g}}\|_\star^2}{G^2}\right)\right] \leq \exp(1)$.

It is easy to see that assumption 3b implies assumption 3a by Jensen's inequality. We will need the stronger assumption 3b to prove high probability bounds. We call an oracle satisfying the weaker assumption 3a G-**bounded**, and an oracle satisfying the stronger assumption 3b **strongly** G-**bounded**. For a G-bounded oracle, note that by Jensen's inequality, we also have that $\|\mathbb{E}[\hat{\mathbf{g}}]\|_\star^2 \leq G^2$, so in particular, at all points $\mathbf{x} \in \mathcal{K}$, there is a subgradient of F with $\|\cdot\|_\star$ norm bounded by G.

For example, in the important special case $F(\mathbf{x}) = \mathbb{E}_Z[f(\mathbf{x}, Z)]$ where $f(\cdot, z)$ is convex for every z in the support of Z, we can obtain such a stochastic subgradient oracle simply by taking a subgradient of $f(\cdot, z)$.

4. The Fenchel conjugate of $\mathcal{R}$ is the function $\mathcal{R}^\star : \mathcal{H} \to \mathbb{R} \cup \{-\infty, +\infty\}$

$$\mathcal{R}^\star(\mathbf{w}) \;:=\; \sup_{\mathbf{x}} \mathbf{w} \cdot \mathbf{x} - \mathcal{R}(\mathbf{x}).$$

By the properties of Fenchel conjugacy [see Borwein and Lewis, 2006, for more details], we have that $\nabla\mathcal{R}^\star = \nabla\mathcal{R}^{-1}$. We assume that the following "Bregman update and projection" operations can be carried out efficiently over the domain $\mathcal{K}$, for any $\mathbf{x}, \mathbf{g} \in \mathcal{H}$:

$$\mathbf{y} \;=\; \nabla\mathcal{R}^\star(\nabla\mathcal{R}(\mathbf{x}) - \eta\mathbf{g}).$$

$$\mathbf{x}' \;=\; \arg\min_{\mathbf{z}\in\mathcal{K}} \left\{B_\mathcal{R}(\mathbf{z}, \mathbf{y})\right\}.$$

In general this is a convex optimization problem and can be solved efficiently; however the method described in this paper is really useful when this operation can be carried very efficiently (say linear time).

For example, if $\mathcal{R}(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_2^2$, where $\|\cdot\|_2$ is the usual Euclidean ℓ_2 norm, then $B_\mathcal{R}(\mathbf{x}, \mathbf{y}) = \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2^2$, and the Bregman update and projection operations reduce to the usual projected gradient algorithm:

$$\mathbf{x}' \;=\; \arg\min_{\mathbf{z}\in\mathcal{K}} \frac{1}{2}\|(\mathbf{x} - \eta\mathbf{g}) - \mathbf{z}\|_2^2.$$

The above assumptions imply the following lemma:

Lemma 1. *For all $\mathbf{x} \in \mathcal{K}$, and $\mathbf{x}^\star$ the minimizer of F, we have $F(\mathbf{x}) - F(\mathbf{x}^\star) \leq \frac{2G^2}{\lambda}$.*

Proof. For any $\mathbf{x} \in \mathcal{K}$, let $\mathbf{g} \in \partial F(\mathbf{x})$ be a subgradient of F at $\mathbf{x}$ such that $\|\mathbf{g}\|_\star \leq G$ (the existence of $\mathbf{g}$ is guaranteed by assumption 3a). Then by the convexity of F, we have $F(\mathbf{x}) - F(\mathbf{x}^\star) \leq \mathbf{g} \cdot (\mathbf{x} - \mathbf{x}^\star)$, so that by the Cauchy-Schwarz inequality, we have $F(\mathbf{x}) - F(\mathbf{x}^\star) \leq G\|\mathbf{x} - \mathbf{x}^\star\|$. But assumption 1 and 2 imply that

$$F(\mathbf{x}) - F(\mathbf{x}^\star) \;\geq\; \lambda B_\mathcal{R}(\mathbf{x}^\star, \mathbf{x}) \;\geq\; \frac{\lambda}{2}\|\mathbf{x}^\star - \mathbf{x}\|^2.$$

Putting these together, we get that $\|\mathbf{x} - \mathbf{x}^\star\| \leq \frac{2G}{\lambda}$. Finally, we have

$$F(\mathbf{x}) - F(\mathbf{x}^\star) \;\leq\; G\|\mathbf{x} - \mathbf{x}^\star\| \;\leq\; \frac{2G^2}{\lambda}.$$

□

2.2 Online Convex Optimization and Regret

Recall the setting of online convex optimization given in the introduction. In each round $t = 1, 2, \ldots, T$, a decision-maker needs to choose a point $\mathbf{x}_t \in \mathcal{K}$, a convex set. Then nature provides a convex cost function $f_t : \mathcal{K} \to \mathbb{R}$, and the decision-maker incurs the cost $f_t(\mathbf{x}_t)$. The (adversarial) regret of the decision-maker is defined to be

$$\text{AdversarialRegret} \;:=\; \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x}\in\mathcal{K}} \sum_{t=1}^{T} f_t(\mathbf{x}). \tag{4}$$

When the cost functions f_t are drawn i.i.d. from some unknown distribution D, (stochastic) regret is traditionally defined measured with respect to the expected cost function, $F(\mathbf{x}) = \mathbb{E}_D[f_1(\mathbf{x})]$:

$$\text{StochasticRegret} \;:=\; \mathop{\mathbb{E}}_{D}\left[\sum_{t=1}^{T} F(\mathbf{x}_t)\right] - T \min_{\mathbf{x}\in\mathcal{K}} F(\mathbf{x}). \tag{5}$$

In either case, if the decision-making algorithm is randomized, then we measure the performance by the expectation of the regret taken over the random seed of the algorithm in addition to any other randomness.

When cost functions are drawn i.i.d. from an unknown distribution D, it is easy to check that

$$\mathop{\mathbb{E}}_{D}\left[\min_{\mathbf{x}\in\mathcal{K}} \sum_{t=1}^{T} f_t(\mathbf{x})\right] \;\le\; \min_{\mathbf{x}\in\mathcal{K}} \mathop{\mathbb{E}}_{D}\left[\sum_{t=1}^{T} f_t(\mathbf{x})\right],$$

by considering the point $\mathbf{x}^\star = \arg\min_{\mathbf{x}\in\mathcal{K}} \mathbb{E}_D\left[\sum_{t=1}^{T} f_t(\mathbf{x})\right]$. So

$$\mathop{\mathbb{E}}_{D}[\text{AdversarialRegret}] \;\ge\; \text{StochasticRegret}.$$

Thus, for the purpose of proving lower bounds on the regret (expected regret in the case of randomized algorithms), it suffices to prove such bounds for StochasticRegret. We prove such lower bounds in Section 5. For notational convenience, henceforth the term "regret" refers to StochasticRegret.

3 The Optimal Algorithm and its Analysis

Our algorithm is an extension of stochastic gradient descent. The new feature is the introduction of "epochs" inside of which standard stochastic gradient descent is used, but in each consecutive epoch the learning rate decreases exponentially.

Our main result is the following theorem, which immediately implies Theorem 1.

Theorem 4. *Set the parameters $T_1 = 4$ and $\eta_1 = \frac{1}{\lambda}$ in the* Epoch-GD *algorithm. The final point $\mathbf{x}_1^k$ returned by the algorithm has the property that*

$$\mathbb{E}[F(\mathbf{x}_1^k)] - F(\mathbf{x}^\star) \;\le\; \frac{16G^2}{\lambda T}.$$

The total number of gradient updates is at most T.

Algorithm 1 EPOCH-GD

Input: parameters η_1, T_1 and total time T.
Initialize $\mathbf{x}_1^1 \in \mathcal{K}$ arbitrarily, and set $k = 1$.
while $\sum_{i=1}^{k} T_i \leq T$ **do**
  // Start epoch k
  for $t = 1$ to T_k **do**
    Query the gradient oracle at $\mathbf{x}_t^k$ to obtain $\hat{\mathbf{g}}_t$
    Update

$$\mathbf{y}_{t+1}^k = \nabla\mathcal{R}^\star(\nabla\mathcal{R}(\mathbf{x}_t^k) - \eta_k\hat{\mathbf{g}}_t),$$
$$\mathbf{x}_{t+1}^k = \arg\min_{\mathbf{x}\in\mathcal{K}} \left\{B_\mathcal{R}(\mathbf{x}, \mathbf{y}_{t+1}^k)\right\}.$$

  end for
  Set $\mathbf{x}_1^{k+1} = \frac{1}{T_k}\sum_{t=1}^{T_k} \mathbf{x}_t^k$
  Set $T_{k+1} \leftarrow 2T_k$ and $\eta_{k+1} \leftarrow \eta_k/2$.
  Set $k \leftarrow k + 1$
end while
return $\mathbf{x}_1^k$.

The intra-epoch use of online mirror decent is analyzed using the following lemma, which follows the ideas of Zinkevich [2003], Bartlett et al. [2007], and given here for completeness:

Lemma 2. *Starting from an arbitrary point $\mathbf{x}_1 \in \mathcal{K}$, apply T iterations of the update*

$$\mathbf{y}_{t+1} = \nabla\mathcal{R}^\star(\nabla\mathcal{R}(\mathbf{x}_t) - \eta\hat{\mathbf{g}}_t),$$
$$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}\in\mathcal{K}} \left\{B_\mathcal{R}(\mathbf{x}, \mathbf{y}_{t+1})\right\}.$$

Then for any point $\mathbf{x}^\star \in \mathcal{K}$, we have

$$\sum_{t=1}^{T} \hat{\mathbf{g}}_t \cdot (\mathbf{x}_t - \mathbf{x}^\star) \leq \frac{\eta}{2}\sum_{t=1}^{T} \|\hat{\mathbf{g}}_t\|_\star^2 + \frac{B_\mathcal{R}(\mathbf{x}^\star, \mathbf{x}_1)}{\eta}.$$

Proof. Since $\nabla\mathcal{R}^\star = \nabla\mathcal{R}^{-1}$, we have $\nabla\mathcal{R}(\mathbf{y}_{t+1}) = \nabla\mathcal{R}(\mathbf{x}_t) - \eta\hat{\mathbf{g}}_t$. Thus, we have

$$\begin{aligned}
\hat{\mathbf{g}}_t \cdot (\mathbf{x}_t - \mathbf{x}^\star) &= \frac{1}{\eta}(\nabla\mathcal{R}(\mathbf{y}_{t+1}) - \nabla\mathcal{R}(\mathbf{x}_t)) \cdot (\mathbf{x}^\star - \mathbf{x}_t) \\
&= \frac{1}{\eta}[B_\mathcal{R}(\mathbf{x}^\star, \mathbf{x}_t) - B_\mathcal{R}(\mathbf{x}^\star, \mathbf{y}_{t+1}) + B_\mathcal{R}(\mathbf{x}_t, \mathbf{y}_{t+1})] \qquad \text{via (2)} \\
&\leq \frac{1}{\eta}[B_\mathcal{R}(\mathbf{x}^\star, \mathbf{x}_t) - B_\mathcal{R}(\mathbf{x}^\star, \mathbf{x}_{t+1}) + B_\mathcal{R}(\mathbf{x}_t, \mathbf{y}_{t+1})],
\end{aligned}$$

where the last inequality follows from the Pythagorean Theorem for Bregman divergences [see Bregman, 1967]: since $\mathbf{x}_{t+1}$ is the Bregman projection of $\mathbf{y}_{t+1}$ on the convex set $\mathcal{K}$, and $\mathbf{x}^\star \in K$, we have $B_\mathcal{R}(\mathbf{x}^\star, \mathbf{x}_{t+1}) \leq B_\mathcal{R}(\mathbf{x}^\star, \mathbf{y}_{t+1})$. Summing over all iterations, and using the non-negativity

of the Bregman divergence, we get

$$\begin{aligned}\sum_{t=1}^{T} \hat{\mathbf{g}}_t \cdot (\mathbf{x}_t - \mathbf{x}^\star) &\leq \frac{1}{\eta}[B_{\mathcal{R}}(\mathbf{x}^\star, \mathbf{x}_1) - B_{\mathcal{R}}(\mathbf{x}^\star, \mathbf{x}_{T+1})] + \frac{1}{\eta}\sum_{t=1}^{T} B_{\mathcal{R}}(\mathbf{x}_t, \mathbf{y}_{t+1})] \\ &\leq \frac{1}{\eta} B_{\mathcal{R}}(\mathbf{x}^\star, \mathbf{x}_1) + \frac{1}{\eta}\sum_{t=1}^{T} B_{\mathcal{R}}(\mathbf{x}_t, \mathbf{y}_{t+1}). \end{aligned} \tag{6}$$

We proceed to bound $B_{\mathcal{R}}(\mathbf{x}_t, \mathbf{y}_{t+1})$. By the definition of Bregman divergence, we get

$$\begin{aligned} B_{\mathcal{R}}(\mathbf{x}_t, \mathbf{y}_{t+1}) + B_{\mathcal{R}}(\mathbf{y}_{t+1}, \mathbf{x}_t) &= (\nabla\mathcal{R}(\mathbf{x}_t) - \nabla\mathcal{R}(\mathbf{y}_{t+1})) \cdot (\mathbf{x}_t - \mathbf{y}_{t+1}) \\ &= \eta\hat{\mathbf{g}}_t \cdot (\mathbf{x}_t - \mathbf{y}_{t+1}) \\ &\leq \frac{1}{2}\eta^2\|\hat{\mathbf{g}}_t\|_\star^2 + \frac{1}{2}\|\mathbf{x}_t - \mathbf{y}_{t+1}\|^2. \end{aligned}$$

The last inequality uses the fact that since $\|\cdot\|$ and $\|\cdot\|_\star$ are dual norms, we have

$$\mathbf{w} \cdot \mathbf{v} \leq \|\mathbf{w}\|_\star\|\mathbf{v}\| \leq \frac{1}{2}\|\mathbf{w}\|_\star^2 + \frac{1}{2}\|\mathbf{v}\|^2.$$

Thus, by our assumption $B_{\mathcal{R}}(\mathbf{x}, \mathbf{y}) \geq \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|^2$, we have

$$B_{\mathcal{R}}(\mathbf{x}_t, \mathbf{y}_{t+1}) \leq \frac{1}{2}\eta^2\|\hat{\mathbf{g}}_t\|_\star^2 + \frac{1}{2}\|\mathbf{x}_t - \mathbf{y}_{t+1}\|^2 - B_{\mathcal{R}}(\mathbf{y}_{t+1}, \mathbf{x}_t) \leq \frac{\eta^2}{2}\|\hat{\mathbf{g}}_t\|_\star^2.$$

Plugging this bound into (6), we get the required bound. □

Lemma 3. *Starting from an arbitrary point $\mathbf{x}_1 \in \mathcal{K}$, apply T iterations of the update*

$$\begin{aligned}\mathbf{y}_{t+1} &= \nabla\mathcal{R}^\star(\nabla\mathcal{R}(\mathbf{x}_t) - \eta\hat{\mathbf{g}}_t), \\ \mathbf{x}_{t+1} &= \arg\min_{\mathbf{x}\in\mathcal{K}} \{B_{\mathcal{R}}(\mathbf{x}, \mathbf{y}_{t+1})\}, \end{aligned}$$

where $\hat{\mathbf{g}}_t$ is an unbiased estimator for a subgradient $\mathbf{g}_t$ of F at $\mathbf{x}_t$ satisfying assumption 3a. Then for any point $\mathbf{x}^\star \in \mathcal{K}$, we have

$$\frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T} F(\mathbf{x}_t)\right] - F(\mathbf{x}^\star) \leq \frac{\eta G^2}{2} + \frac{B_{\mathcal{R}}(\mathbf{x}^\star, \mathbf{x}_1)}{\eta T}.$$

By convexity of F, we have the same bound for $\mathbb{E}[F(\bar{\mathbf{x}})] - F(\mathbf{x}^\star)$, where $\bar{\mathbf{x}} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{x}_t$.

Proof. For a random variable X measurable w.r.t. the randomness until round t, let $\mathbb{E}_{t-1}[X]$ denote its expectation conditioned on the randomness until round $t-1$. By the convexity of F, we get

$$F(\mathbf{x}_t) - F(\mathbf{x}^\star) \leq \mathbf{g}_t \cdot (\mathbf{x}_t - \mathbf{x}^\star) = \mathop{\mathbb{E}}_{t-1}[\hat{\mathbf{g}}_t \cdot (\mathbf{x}_t - \mathbf{x}^\star)],$$

since $\mathbb{E}_{t-1}[\hat{\mathbf{g}}_t] = \mathbf{g}_t$ and $\mathbb{E}_{t-1}[\mathbf{x}_t] = \mathbf{x}_t$. Taking expectations of the inequality, we get that

$$\mathbb{E}[F(\mathbf{x}_t)] - F(\mathbf{x}^\star) \leq \mathbb{E}[\hat{\mathbf{g}}_t \cdot (\mathbf{x}_t - \mathbf{x}^\star)].$$

Summing up over all $t = 1, 2, \ldots, T$, and taking the expectation on both sides of the inequality in Lemma 2, we get the required bound. □

Define $V_k = \frac{G^2}{2^{k-2}\lambda}$ and $\Delta_k = F(\mathbf{x}_1^k) - F(\mathbf{x}^\star)$. The choice of initial parameters $T_1 = 4$ and $\eta_1 = \frac{1}{\lambda}$ was specified in Theorem 4, and by definition $T_k = T_1 2^{k-1}$ and $\eta_k = \eta_1 2^{-(k-1)}$. Using Lemma 3 we prove the following key lemma:

Lemma 4. *For any k, we have $\mathbb{E}[\Delta_k] \leq V_k$.*

Proof. We prove this by induction on k. The claim is true for $k = 1$ since $\Delta_k \leq \frac{2G^2}{\lambda}$ by Lemma 1. Assume that $\mathbb{E}[\Delta_k] \leq V_k$ for some $k \geq 1$ and now we prove it for $k+1$. For a random variable X measurable w.r.t. the randomness defined up to epoch $k+1$, let $\mathbb{E}_k[X]$ denote its expectation conditioned on all the randomness up to epoch k. By Lemma 3 we have

$$\begin{aligned} \mathop{\mathbb{E}}_k[F(\mathbf{x}_1^{k+1})] - F(\mathbf{x}^\star) &\leq \frac{\eta_k G^2}{2} + \frac{B_{\mathcal{R}}(\mathbf{x}^\star, \mathbf{x}_1^k)}{\eta_k T_k} \\ &\leq \frac{\eta_k G^2}{2} + \frac{\Delta_k}{\eta_k T_k \lambda}, \end{aligned}$$

since $\Delta_k = F(\mathbf{x}_1^k) - F(\mathbf{x}^\star) \geq \lambda B_{\mathcal{R}}(\mathbf{x}^\star, \mathbf{x}_1^k)$ by λ-strong convexity of F with respect to $B_{\mathcal{R}}$. Hence, we get

$$\mathbb{E}[\Delta_{k+1}] \leq \frac{\eta_k G^2}{2} + \frac{\mathbb{E}[\Delta_k]}{\eta_k T_k \lambda} \leq \frac{\eta_k G^2}{2} + \frac{V_k}{\eta_k T_k \lambda} = \frac{\eta_1 G^2}{2^k} + \frac{V_k}{\eta_1 T_1 \lambda} = V_{k+1},$$

as required. The second inequality uses the induction hypothesis, and the last two equalities use the definition of V_k, the equalities $T_k = T_1 2^{k-1}$ and $\eta_k = \eta_1 2^{-(k-1)}$, and the initial values $T_1 = 4$ and $\eta_1 = \frac{1}{\lambda}$. □

We can now prove our main theorem:

Proof of Theorem 4. The number of epochs made are given by the largest value of k satisfying $\sum_{i=1}^k T_i \leq T$, i.e.,

$$\sum_{i=1}^{k} 2^{i-1} T_1 = (2^k - 1) T_1 \leq T.$$

This value is $k^\dagger = \lfloor \log_2(\frac{T}{T_1} + 1) \rfloor$. The final point output by the algorithm is $\mathbf{x}_1^{k^\dagger+1}$. Applying Lemma 4 to $k^\dagger + 1$ we get

$$\mathbb{E}[F(\mathbf{x}_1^{k^\dagger+1})] - F(\mathbf{x}^\star) = \mathbb{E}[\Delta_{k^\dagger+1}] \leq V_{k^\dagger+1} = \frac{G^2}{2^{k^\dagger-1}\lambda} \leq \frac{4T_1 G^2}{\lambda T} = \frac{16G^2}{\lambda T},$$

as claimed. The while loop in the algorithm ensures that the total number of gradient updates is naturally bounded by T. □

3.1 A Randomized Stopping Variant

In this section we describe a pure stochastic gradient descent algorithm with random step sizes that has the same (expected) rate of convergence.

Our main theorem of this section is:

Algorithm 2 RANDOM-STEP-GD

Input: parameters η_1, T_1 and total time T.
Initialize $\mathbf{x}_1 \in \mathcal{K}$ arbitrarily, and set $k = 1$, $B_1 = 1$, $B_2 \in \{1, 2, \ldots, T_1\}$ uniformly at random.
for $t = 1, 2, \ldots$ **do**
  if $t = B_{k+1}$ **then**
    Set $k \leftarrow k + 1$.
    Set $T_k \leftarrow 2T_{k-1}$ and $\eta_k \leftarrow \eta_{k-1}/2$.
    Set $B_{k+1} \in \{B_k, B_k + 1, \ldots, B_k + T_k - 1\}$ uniformly at random.
    if $B_{k+1} > T$ **then**
      Break **for** loop.
    end if
  end if
  Query the gradient oracle at $\mathbf{x}_t$ to obtain $\hat{\mathbf{g}}_t$.
  Update

$$\begin{aligned} \mathbf{y}_{t+1} &= \nabla\mathcal{R}^\star(\nabla\mathcal{R}(\mathbf{x}_t) - \eta_k\hat{\mathbf{g}}_t) \\ \mathbf{x}_{t+1} &= \arg\min_{\mathbf{x}\in\mathcal{K}} \{B_\mathcal{R}(\mathbf{x}, \mathbf{y}_{t+1})\} \end{aligned}$$

end for
return $\mathbf{x}_t$.

Theorem 5. *Set the parameters $T_1 = 4$ and $\eta_1 = \frac{1}{\lambda}$ in the* RANDOM-STEP-GD *algorithm. The final point $\mathbf{x}_t$ returned by the algorithm has the property that*

$$\mathbb{E}[F(\mathbf{x}_t)] - F(\mathbf{x}^\star) \leq \frac{16G^2}{\lambda T}$$

where the expectation is taken over the gradient estimates as well as the internal randomization of the algorithm.

Proof. The proof of this theorem is on the same lines as before. In particular, we divide up the entire time period into (possibly overlapping) epochs. For $k = 1, 2, \ldots$, epoch k consists of the following sequence of T_k rounds: $\{B_k, B_k + 1, \ldots, B_k + T_k - 1\}$. Note that B_{k+1} is a uniformly random time in the above sequence. The behavior of the algorithm in rounds $B_k, B_k + 1, \ldots, B_{k+1} - 1$ can be simulated by the following thought-experiment: starting with $\mathbf{x}_{B_k}$, run T_k iterations of stochastic mirror descent, i.e.,

$$\begin{aligned} \nabla\mathcal{R}(\mathbf{y}_{t+1}) &= \nabla\mathcal{R}(\mathbf{x}_t) - \eta_k\hat{\mathbf{g}}_t, \\ \mathbf{x}_{t+1} &= \arg\min_{\mathbf{x}\in\mathcal{K}} \{B_\mathcal{R}(\mathbf{x}, \mathbf{y}_{t+1})\}, \end{aligned}$$

for $t = B_k, \ldots, B_k + T_k - 1$, and return $\mathbf{x}_{B_{k+1}}$. Conditioning on $\mathbf{x}_{B_{k-1}}$, and taking expectations, since B_{k+1} was chosen uniformly at random from a sequence of T_k rounds, we get

$$\mathbb{E}[F(\mathbf{x}_{B_{k+1}})] = \frac{1}{T_k} \sum_{t=B_k}^{B_k+T_k-1} \mathbb{E}[F(\mathbf{x}_t)].$$

Now, by Lemma 3, we conclude that

$$\mathbb{E}[F(\mathbf{x}_{B_{k+1}})] - F(\mathbf{x}^\star) \leq \frac{\eta_k G^2}{2} + \frac{B_\mathcal{R}(\mathbf{x}^\star, \mathbf{x}_{B_k})}{\eta_k T_k}. \tag{7}$$

Now, just as before, we define $V_k = \frac{G^2}{2^{k-2}\lambda}$ and $\Delta_k = F(\mathbf{x}_{B_k}) - F(\mathbf{x}^\star)$. Recall the choice of initial parameters $T_1 = 4$ and $\eta_1 = \frac{1}{\lambda}$ as specified in Theorem 5. Now, arguing exactly as in Lemma 4

Lemma 5. *For any k, we have $\mathbb{E}[\Delta_k] \leq V_k$.*

Proof. We prove this by induction on k. The claim is true for $k = 1$ since $\Delta_k \leq \frac{2G^2}{\lambda}$ by Lemma 1. Assume that $\mathbb{E}[\Delta_k] \leq V_k$ for some $k \geq 1$ and now we prove it for $k+1$. For a random variable X measurable w.r.t. the randomness defined up to epoch $k+1$, let $\mathbb{E}_k[X]$ denote its expectation conditioned on all the randomness up to epoch k. By Lemma 3 we have

$$\begin{aligned}\mathop{\mathbb{E}}_k[F(\mathbf{x}_1^{k+1})] - F(\mathbf{x}^\star) &\leq \frac{\eta_k G^2}{2} + \frac{B_{\mathcal{R}}(\mathbf{x}^\star, \mathbf{x}_1^k)}{\eta_k T_k} \\ &\leq \frac{\eta_k G^2}{2} + \frac{\Delta_k}{\eta_k T_k \lambda},\end{aligned}$$

since $\Delta_k = F(\mathbf{x}_1^k) - F(\mathbf{x}^\star) \geq \lambda B_{\mathcal{R}}(\mathbf{x}^\star, \mathbf{x}_1^k)$ by λ-strong convexity of F with respect to $\mathcal{R}$. Hence, we get

$$\mathbb{E}[\Delta_{k+1}] \;\leq\; \frac{\eta_k G^2}{2} + \frac{\mathbb{E}[\Delta_k]}{\eta_k T_k \lambda} \;\leq\; \frac{\eta_k G^2}{2} + \frac{V_k}{\eta_k T_k \lambda} \;=\; \frac{\eta_1 G^2}{2^k} + \frac{V_k}{\eta_1 T_1 \lambda} \;=\; V_{k+1},$$

as required. As before, the second inequality above uses the induction hypothesis, and the last two equalities use the definition of V_k, the equalities $T_k = T_1 2^{k-1}$ and $\eta_k = \eta_1 2^{-(k-1)}$, and the initial values $T_1 = 4$ and $\eta_1 = \frac{1}{\lambda}$. □

Now just as in the proof of Theorem 4, since we output $\mathbf{x}_t = \mathbf{x}_{B_{k^\dagger+1}}$, where $k^\dagger$, the number of epochs, is at least[4] $\lfloor \log_2(\frac{T}{T_1} + 1) \rfloor$, we conclude that $\mathbb{E}[F(\mathbf{x}_t)] - F(\mathbf{x}^\star) \leq \frac{16G^2}{\lambda T}$ as required. □

4 High Probability Bounds

While Epoch-GD algorithm has a $O(\frac{1}{T})$ rate of convergence, this bound is only on the expected excess cost of the final solution. In applications we usually need the rate of convergence to hold with high probability. Markov's inequality immediately implies that with probability $1 - \delta$, the actual excess cost is at most a factor of $\frac{1}{\delta}$ times the stated bound. While this guarantee might be acceptable for not too small values of δ, it becomes useless when δ gets really small.

There are two ways of remedying this. The easy way applies if it is possible to evaluate F efficiently at any given point. Then we can divide the budget of T gradient updates into $\ell = \log_2(1/\delta)$ consecutive intervals of $\frac{T}{\ell}$ rounds each, and run independent copies of Epoch-GD in each. Finally, we take the ℓ solutions obtained, and output the best one (i.e., the one with the minimum F value). Applying Markov's inequality to every run of Epoch-GD, with probability at least $1/2$, we obtain a point with excess cost at most $\frac{64G^2\ell}{\lambda T} = \frac{64G^2 \log_2(1/\delta)}{\lambda T}$, and so with probability at least $1 - 2^{-\ell} = 1 - \delta$, the best point has excess cost at most $\frac{64G^2 \log_2(1/\delta)}{\lambda T}$. This finishes the description of the easy way to obtain high probability bounds.

The easy way fails if it is not possible to evaluate F efficiently at any given point. For this situation, we now describe how using essentially the same algorithm with slightly different parameters, we can get a high probability guarantee on the quality of the solution. To prove the high

[4]Here we have an inequality rather than an equality as in the previous algorithm since we may have more epochs due to the random early stopping of epochs.

probability bound, we need to make the stronger assumption 3b, i.e., for all points $\mathbf{x} \in \mathcal{K}$, the stochastic subgradient $\hat{\mathbf{g}}$ output by the oracle satisfies $\mathbb{E}[\exp(\frac{\|\hat{\mathbf{g}}\|_\star^2}{G^2})] \le e$.

The only differences in the new algorithm, dubbed Epoch-GD-Proj, are as follows. The algorithm takes a new parameter, D_1. The update in line 7 requires a projection onto a smaller set, and becomes

$$\begin{aligned} \mathbf{y}_{t+1}^k &= \nabla\mathcal{R}^\star(\nabla\mathcal{R}(\mathbf{x}_t^k) - \eta_k\hat{\mathbf{g}}_t), \\ \mathbf{x}_{t+1}^k &= \operatorname*{arg\,min}_{\mathbf{x}\in\mathcal{K}\cap\mathcal{B}(\mathbf{x}_1^k, D_k)} \left\{ B_{\mathcal{R}}(\mathbf{x}, \mathbf{y}_{t+1}^k) \right\}. \end{aligned} \tag{8}$$

Here $\mathcal{B}(\mathbf{x}, D) = \{\mathbf{y} : \|\mathbf{y} - \mathbf{x}\| \le D\}$ denotes the ball of radius D around the point $\mathbf{x}$, and D_k is computed in the algorithm. The update in line 10 now becomes:

$$\text{Set } T_{k+1} \leftarrow 2T_k,\ \eta_{k+1} \leftarrow \eta_k/2, \text{ and } D_{k+1} \leftarrow D_k/\sqrt{2}.$$

Since the intersection of two convex sets is also a convex set, the above projection can be computed via a convex program.[5] A completely analogous version of Random-Step-GD is an easy extension; it enjoys the same high probability bound as given below. We prove the following high probability result, which in turn directly implies Theorem 3.

Theorem 6. *Given $\delta > 0$ for success probability $1-\delta$, set $\tilde{\delta} = \frac{\delta}{k^\dagger}$ for $k^\dagger = \lfloor \log_2(\frac{T}{450} + 1)\rfloor$. Set the parameters $T_1 = 450$, $\eta_1 = \frac{1}{3\lambda}$, and $D_1 = 2G\sqrt{\frac{\log(2/\tilde{\delta})}{\lambda}}$ in the* Epoch-GD-Proj *algorithm. The final point $\mathbf{x}_1^k$ returned by the algorithm has the property that with probability at least $1-\delta$, we have*

$$F(\mathbf{x}_1^k) - F(\mathbf{x}^\star) \le \frac{1800G^2\log(2/\tilde{\delta})}{\lambda T}.$$

The total number of gradient updates is at most T.

The following lemma is analogous to Lemma 3, but provides a high probability guarantee.

Lemma 6. *For any given $\mathbf{x}^\star \in \mathcal{K}$, let D be an upper bound on $\|\mathbf{x}_1 - \mathbf{x}^\star\|$. Apply T iterations of the update*

$$\begin{aligned} \mathbf{y}_{t+1} &= \nabla\mathcal{R}^\star(\nabla\mathcal{R}(\mathbf{x}_t) - \eta\hat{\mathbf{g}}_t), \\ \mathbf{x}_{t+1} &= \operatorname*{arg\,min}_{\mathbf{x}\in\mathcal{K}\cap\mathcal{B}(\mathbf{x}_1, D)} \{B_{\mathcal{R}}(\mathbf{x}, \mathbf{y}_{t+1})\}. \end{aligned}$$

where $\hat{\mathbf{g}}_t$ is an unbiased estimator for the subgradient of F at $\mathbf{x}_t$ satisfying assumption 3b. Then for any $\delta \in (0,1)$, with probability at least $1-\delta$ we have

$$\frac{1}{T}\sum_{t=1}^T F(\mathbf{x}_t) - F(\mathbf{x}^\star) \le \frac{\eta G^2\log(2/\delta)}{2} + \frac{B_{\mathcal{R}}(\mathbf{x}^\star, \mathbf{x}_1)}{\eta T} + \frac{4GD\sqrt{3\log(2/\delta)}}{\sqrt{T}}.$$

By the convexity of F, the same bound also holds for $F(\bar{\mathbf{x}}) - F(\mathbf{x}^\star)$, where $\bar{\mathbf{x}} = \frac{1}{T}\sum_{t=1}^T \mathbf{x}_t$.

[5] It was suggested to us by a referee that in practice, computing x_{t+1}^k by taking a Bregman projection on $\mathcal{K} \cap B'(\mathbf{x}_1^k, D_k)$, where $B'(\mathbf{x}, r) = \{\mathbf{y} : B_{\mathcal{R}}(\mathbf{y}, \mathbf{x}) \le D^2/2\}$ is the "Bregman ball of radius D around the point $\mathbf{x}$", might be more efficient than a projection on $\mathcal{K} \cap \mathcal{B}(\mathbf{x}_1^k, D_k)$. This depends on the application, but it is easy to see that all the proofs (and thus the high-probability guarantees) go through simply because the Bregman balls are a subset of the norm $\|\cdot\|$ balls, i.e., $B'(\mathbf{x}, D) \subseteq \mathcal{B}(\mathbf{x}, D)$, by the strong-convexity of $\mathcal{R}$ w.r.t. the norm $\|\cdot\|$. We prefer to leave the update in terms of the norm $\|\cdot\|$ balls since generally speaking projections on larger sets are easier; the specific choice can be tailored to the application.

Proof. First, note that since the oracle uses independent randomness in every call to it, we conclude that that for all t, $\hat{\mathbf{g}}_t$ is independent of $\hat{\mathbf{g}}_1, \ldots, \hat{\mathbf{g}}_{t-1}$ given $\mathbf{x}_t$, and thus by assumption 3b we have

$$\mathop{\mathbb{E}}_{t}\left[\exp\left(\frac{\|\hat{\mathbf{g}}_t\|_\star^2}{G^2}\right)\right] \;=\; \mathbb{E}\left[\exp\left(\frac{\|\hat{\mathbf{g}}_t\|_\star^2}{G^2}\right)\middle|\,\mathbf{x}_t\right] \;\le\; \exp(1). \tag{9}$$

The proof proceeds on similar lines as that of Lemma 3, except that we use high-probability bounds rather than expected bounds. Using the same notation as in the proof of Lemma 3, let $\mathbb{E}_{t-1}[\hat{\mathbf{g}}_t] = \mathbf{g}_t$, a subgradient of F at $\mathbf{x}_t$. We now need to bound $\sum_{t=1}^T \hat{\mathbf{g}}_t \cdot (\mathbf{x}_t - \mathbf{x}^\star)$ in terms of $\sum_{t=1}^T \mathbf{g}_t \cdot (\mathbf{x}_t - \mathbf{x}^\star)$, and $\sum_{t=1}^T \|\hat{\mathbf{g}}_t\|_\star^2$ in terms of G^2T.

As before, $\mathbb{E}_{t-1}[\hat{\mathbf{g}}_t \cdot (\mathbf{x}_t - \mathbf{x}^\star)] = \mathbf{g}_t \cdot (\mathbf{x}_t - \mathbf{x}^\star)$, and thus the following defines a martingale difference sequence:

$$X_t \;:=\; \mathbf{g}_t \cdot (\mathbf{x}_t - \mathbf{x}^\star) - \hat{\mathbf{g}}_t \cdot (\mathbf{x}_t - \mathbf{x}^\star).$$

Note that $\|\mathbf{g}_t\|_\star = \|\,\mathbb{E}_{t-1}[\hat{\mathbf{g}}_t]\|_\star \le \mathbb{E}_{t-1}[\|\hat{\mathbf{g}}_t\|_\star] \le G$, and so we can bound $|X_t|$ as follows:

$$|X_t| \le \|\mathbf{g}_t\|_\star \|\mathbf{x}_t - \mathbf{x}^\star\| + \|\hat{\mathbf{g}}_t\|_\star \|\mathbf{x}_t - \mathbf{x}^\star\| \;\le\; 2GD + 2D\|\hat{\mathbf{g}}_t\|_\star,$$

where the last inequality uses the fact that since $\mathbf{x}^\star, \mathbf{x}_t \in \mathcal{B}(\mathbf{x}_1, D)$, we have $\|\mathbf{x}_t - \mathbf{x}^\star\| \le \|\mathbf{x}_t - \mathbf{x}_1\| + \|\mathbf{x}_1 - \mathbf{x}^\star\| \le 2D$. This implies that

$$\mathop{\mathbb{E}}_{t}\left[\exp\left(\frac{X_t^2}{16G^2D^2}\right)\right] \;\le\; \mathop{\mathbb{E}}_{t}\left[\exp\left(\frac{4D^2(2G^2+2\|\hat{\mathbf{g}}_t\|_\star^2)}{16G^2D^2}\right)\right] \;\le\; \exp(\tfrac{1}{2})\sqrt{\mathop{\mathbb{E}}_{t}\left[\exp\left(\frac{\|\hat{\mathbf{g}}_t\|_\star^2}{G^2}\right)\right]} \;\le\; \exp(1),$$

where the second inequality follows by Jensen's inequality and the inequality $(a+b)^2 \le 2a^2 + 2b^2$, and the last by (9).

By Lemma 8, with probability at least $1 - \delta/2$, we have $\sum_{t=1}^T X_t \le 4GD\sqrt{3\log(2/\delta)T}$, which implies that

$$\frac{1}{T}\sum_{t=1}^T F(\mathbf{x}_t) - F(\mathbf{x}^\star) \;\le\; \frac{1}{T}\sum_{t=1}^T \mathbf{g}_t \cdot (\mathbf{x}_t - \mathbf{x}^\star) - \frac{1}{T}\sum_{t=1}^T \hat{\mathbf{g}}_t \cdot (\mathbf{x}_t - \mathbf{x}^\star) \;\le\; \frac{4GD\sqrt{3\log(2/\delta)}}{\sqrt{T}}, \tag{10}$$

where the first inequality follows by convexity of F.

Next, consider $\mathbb{E}[\exp(\frac{\sum_{t=1}^T \|\hat{\mathbf{g}}_t\|_\star^2}{G^2})]$. We can upper bound this as follows:

$$\begin{aligned}
\mathbb{E}\left[\exp\left(\frac{\sum_{t=1}^T \|\hat{\mathbf{g}}_t\|_\star^2}{G^2}\right)\right] \;&=\; \mathbb{E}\left[\mathop{\mathbb{E}}_{T}\left[\exp\left(\frac{\sum_{t=1}^T \|\hat{\mathbf{g}}_t\|_\star^2}{G^2}\right)\right]\right] \\
&=\; \mathbb{E}\left[\exp\left(\frac{\sum_{t=1}^{T-1} \|\hat{\mathbf{g}}_t\|_\star^2}{G^2}\right)\mathop{\mathbb{E}}_{T}\left[\exp\left(\frac{\|\hat{\mathbf{g}}_T\|_\star^2}{G^2}\right)\right]\right] \\
&\le\; \mathbb{E}\left[\exp\left(\frac{\sum_{t=1}^{T-1} \|\hat{\mathbf{g}}_t\|_\star^2}{G^2}\right)\cdot \exp(1)\right]
\end{aligned}$$

by (9). Continuing inductively, we conclude that $\mathbb{E}[\exp(\frac{\sum_{t=1}^T \|\hat{\mathbf{g}}_t\|_\star^2}{G^2})] \le \exp(T)$, which implies (via Markov's inequality) that with probability at least $1 - \delta/2$, we have

$$\sum_{t=1}^T \|\hat{\mathbf{g}}_t\|_\star^2 \;\le\; G^2T\log(2/\delta). \tag{11}$$

Then, by using Lemma 2 and inequalities (10) and (11), we get the claimed bound. □

We now prove the analogue of Lemma 4. In this case, the result holds with high probability. As before, define $V_k = \frac{G^2}{2^{k-2}\lambda}$ and $\Delta_k = F(\mathbf{x}_1^k) - F(\mathbf{x}^\star)$. Recall the choice of initial parameters $T_1 = 450$ and $\eta_1 = \frac{1}{3\lambda}$ as specified in Theorem 3.

Lemma 7. *For any k, with probability $(1-\tilde{\delta})^{k-1}$ we have $\Delta_k \leq V_k \log(2/\tilde{\delta})$.*

Proof. For notational convenience, in the following we define:

$$L \;:=\; \log(2/\tilde{\delta}).$$

We prove the lemma by induction on k. The claim is true for $k=1$ since $\Delta_k \leq \frac{2G^2L}{\lambda}$ by Lemma 1. Assume that $\Delta_k \leq V_k L$ for some $k \geq 1$ with probability at least $(1-\tilde{\delta})^{k-1}$ and now we prove the corresponding statement for $k+1$. We condition on the event that $\Delta_k \leq V_k L$. Since $\Delta_k \geq \frac{\lambda}{2}\|\mathbf{x}_1^k - \mathbf{x}^\star\|^2$ by λ-strong convexity, this conditioning implies that $\|\mathbf{x}_1^k - \mathbf{x}^\star\| \leq \sqrt{2V_kL/\lambda} = D_k$. So Lemma 6 applies with $D = D_k$ and hence we have with probability at least $1-\tilde{\delta}$,

$$\begin{aligned}
\Delta_{k+1} &= F(\mathbf{x}_1^{k+1}) - F(\mathbf{x}^\star) \\
&\leq \frac{\eta_k G^2 L}{2} + \frac{B_{\mathcal{R}}(\mathbf{x}^\star, \mathbf{x}_1^k)}{\eta_k T_k} + 10G\sqrt{\frac{V_k L}{\lambda T_k}} && \text{(by Lemma 6)} \\
&\leq \frac{\eta_k G^2 L}{2} + \frac{\Delta_k}{\eta_k T_k} + 10G\sqrt{\frac{V_k L}{\lambda T_k}} && \text{(by } \lambda\text{-strong convexity of } F) \\
&\leq \frac{\eta_k G^2 L}{2} + \frac{V_k L}{\eta_k T_k \lambda} + 10G\sqrt{\frac{V_k L}{\lambda T_k}} && \text{(by induction hypothesis)} \\
&= \frac{\eta_1 G^2 L}{2^k} + \frac{V_k L}{\eta_1 T_1 \lambda} + 10G\sqrt{\frac{V_k L}{\lambda T_1 2^{k-1}}} && \text{(by definition of } T_k, \eta_k) \\
&= \frac{V_k L}{12} + \frac{V_k L}{150} + \frac{V_k\sqrt{L}}{3} && \text{(using values of } T_1, \eta_1, V_k) \\
&\leq \frac{V_k L}{2} \;=\; V_{k+1} L.
\end{aligned}$$

Factoring in the conditioned event, which happens with probability at least $(1-\tilde{\delta})^{k-1}$, overall, we get that $\Delta_{k+1} \leq V_{k+1}$ with probability at least $(1-\tilde{\delta})^k$. □

We can now prove our high probability theorem:

Theorem 6. Proceeding exactly as in the proof of Theorem 1, we get that final epoch is $k^\dagger = \lfloor \log_2(\frac{T}{T_1}+1) \rfloor$. The final point output is $\mathbf{x}_1^{k^\dagger+1}$. By Lemma 7, we have with probability at least $(1-\tilde{\delta})^{k^\dagger}$ that

$$\begin{aligned}
F(\mathbf{x}_1^{k^\dagger+1}) - F(\mathbf{x}^\star) \;=\; \Delta_{k^\dagger+1} \;&\leq\; V_{k^\dagger+1}\log(2/\tilde{\delta}) \\
&= \frac{G^2\log(2/\tilde{\delta})}{2^{k^\dagger-1}\lambda} \;\leq\; \frac{4T_1G^2\log(2/\tilde{\delta})}{\lambda T} \;=\; \frac{1800G^2\log(2/\tilde{\delta})}{\lambda T},
\end{aligned}$$

as claimed. Since $\tilde{\delta} = \frac{\delta}{k^\dagger}$, and hence $(1-\tilde{\delta})^{k^\dagger} \geq 1-\delta$ as needed. The while loop in the algorithm ensures that the total number of gradient updates is bounded by T. □

In the analysis, we used the following well-known martingale inequality, a restatement of Lemma 2 of Lan et al. [2012]. Here, $\mathbb{E}_t[\cdot]$ denotes the expectation at time t conditioned on all the randomness till time $t-1$.

Lemma 8. *Let $X_1, \ldots, X_T$ be a martingale difference sequence, i.e., $\mathbb{E}_t[X_t] = 0$ for all t. Suppose that for some values σ_t, for $t = 1, 2, \ldots, T$, we have $\mathbb{E}_t[\exp(\frac{X_t^2}{\sigma_t^2})] \leq \exp(1)$. Then with probability at least $1-\delta$, we have*

$$\sum_{t=1}^{T} X_t \;\leq\; \sqrt{3\log(1/\delta)\sum_{t=1}^{T}\sigma_t^2}.$$

5 Lower Bounds on Stochastic Strongly Convex Optimization

In this section we prove Theorem 2 and show that any algorithm (deterministic or randomized) for online stochastic strongly-convex optimization must have $\Omega(\log(T))$ regret on some distribution. We start by proving a $\Omega(\log T)$ lower bound for the case when the cost functions are 1-strongly convex with respect to the Euclidean norm and the gradient oracle is 1-bounded, and fine tune these parameters in the next subsection by way of reduction.

In our analysis, we need the following standard lemma, which we reprove here for completeness. Here, for two distributions P, P' defined on the same probability space, $d_{TV}(P, P')$ is the total variation distance, i.e.

$$d_{TV}(P, P') = \sup_A |P(A) - P'(A)|$$

where the supremum ranges over all events A in the probability space.

Let B_p be the Bernoulli distribution on $\{0, 1\}$ with probability of obtaining 1 equal to p. Let B_p^n denote the product measure on $\{0, 1\}^n$ induced by taking n independent Bernoulli trials according to B_p (thus, $B_p^1 = B_p$).

Lemma 9. *Let $p, p' \in [\frac{1}{4}, \frac{3}{4}]$ such that $|p' - p| \leq 1/8$. Then*

$$d_{TV}(B_p^n, B_{p'}^n) \leq 2\sqrt{(p'-p)^2 n}.$$

Proof. By Pinsker's inequality, we have $d_{TV}(B_p^n, B_{p'}^n) \leq \sqrt{\frac{1}{2}\mathrm{RE}(B_p^n \| B_{p'}^n)}$, where $\mathrm{RE}(B_p^n \| B_{p'}^n) = \mathbb{E}_{X \sim B_p^n}[\ln \frac{B_p^n(X)}{B_{p'}^n(X)}]$ is the relative entropy between B_p^n and $B_{p'}^n$. To bound $\mathrm{RE}(B_p^n \| B_{p'}^n)$, note that the additivity of the relative entropy for product measures implies that

$$\mathrm{RE}(B_p^n \| B_{p'}^n) = n\mathrm{RE}(B_p \| B_{p'}) = n\left[p\log\left(\frac{p}{p'}\right) + (1-p)\log\left(\frac{1-p}{1-p'}\right)\right]. \tag{12}$$

Without loss of generality, assume that $p' \geq p$, and let $p' = p + \varepsilon$, where $0 \leq \varepsilon \leq 1/8$. Using the Taylor series expansion of $\log(1+x)$, we get the following bound

$$p\log\left(\frac{p}{p'}\right) + (1-p)\log\left(\frac{1-p}{1-p'}\right) = \sum_{i=1}^{\infty}\left[\frac{(-1)^i}{p^{i-1}} + \frac{1}{(1-p)^{i-1}}\right]\varepsilon^i \leq \sum_{i=2}^{\infty} 4^{i-1}\varepsilon^i \leq 8\varepsilon^2,$$

for $\varepsilon \leq 1/8$. Plugging this (12) and using Pinsker's inequality, we get the stated bound. □

We now turn to showing our lower bound on expected regret. We consider the following online stochastic strongly-convex optimization setting: the domain is $\mathcal{K} = [0,1]$. For every $p \in [\frac{1}{4}, \frac{3}{4}]$, define a distribution over strongly-convex cost functions parameterized by p as follows: choose $X \in \{0,1\}$ from B_p, and return the cost function

$$f(x) = (x - X)^2.$$

With some abuse of notation, we use B_p to denote this distribution over cost functions.

Under distribution B_p, the expected cost function F is

$$F(x) := \mathbf{E}[f(x)] = p(x-1)^2 + (1-p)x^2 = x^2 + 2px + p = (x-p)^2 + c_p,$$

where $c_p = p - p^2$. The optimal point is therefore $x^\star = p$, with expected cost c_p. The regret for playing a point x (i.e., excess cost over the minimal expected cost) is

$$F(x) - F(x^\star) = (x-p)^2 + c_p - c_p = (x-p)^2.$$

Now let $\mathcal{A}$ be a deterministic[6] algorithm for online stochastic strongly-convex optimization. Since the cost functions until time t are specified by a bit string $X \in \{0,1\}^{t-1}$ (i.e., the cost function at time t is $(x - X_t)^2$), we can interpret the algorithm as a function that takes a variable length bit string, and produces a point in $[0,1]$, i.e., with some abuse of notation,

$$\mathcal{A} : \{0,1\}^{\leq T} \longrightarrow [0,1],$$

where $\{0,1\}^{\leq T}$ is the set of all bit strings of length up to T.

Now suppose the cost functions are drawn from B_p. Fix a round t. Let X be the $t-1$ bit string specifying the cost functions so far. Note that X has distribution B_p^{t-1}. For notational convenience, denote by $\Pr_p[\cdot]$ and $\mathbb{E}_p[\cdot]$ the probability of an event and the expectation of a random variable when the cost functions are drawn from B_p, and since these are defined by the bit string X, they are computed over the product measure B_p^{t-1}.

Let the point played by $\mathcal{A}$ at time t be $x_t = \mathcal{A}(X)$. The regret (conditioned on the choice of X) in round t is then

$$\text{regret}_t \;:=\; (\mathcal{A}(X) - p)^2,$$

and thus the expected (over the choice of X) regret of $\mathcal{A}$ in round t is $\mathbb{E}_p[\text{regret}_t] = \mathbb{E}_p[(\mathcal{A}(X)-p)^2]$.

We now show that for any round t, for two distributions over cost functions B_p and $B_{p'}$ that are close (in terms of $|p-p'|$), but not too close, the regret of $\mathcal{A}$ on at least one of the two distributions must be large.

Lemma 10. *Fix a round t. Let $\varepsilon \leq \frac{1}{16\sqrt{t}}$ be a parameter. Let $p, p' \in [\frac{1}{4}, \frac{3}{4}]$ such that $2\varepsilon \leq |p-p'| \leq 4\varepsilon$. Then we have*

$$\mathop{\mathbb{E}}_{p}[\textit{regret}_t] + \mathop{\mathbb{E}}_{p'}[\textit{regret}_t] \;\geq\; \frac{1}{4}\varepsilon^2.$$

Proof. Assume without loss of generality that $p' \geq p + 2\varepsilon$. Let X and X' be $(t-1)$-bit vectors parameterizing the cost functions drawn from B_p^{t-1} and $B_{p'}^{t-1}$ respectively. Then

$$\mathop{\mathbb{E}}_{p}[\text{regret}_t] + \mathop{\mathbb{E}}_{p'}[\text{regret}_t] \;=\; \mathop{\mathbb{E}}_{p}[(\mathcal{A}(X) - p)^2] + \mathop{\mathbb{E}}_{p'}[(\mathcal{A}(X') - p')^2].$$

[6] We will remove the deterministic requirement shortly and allow randomized algorithms.

Now suppose the stated bound does not hold. Then by Markov's inequality, we have

$$\Pr_p[(\mathcal{A}(X) - p)^2 < \varepsilon^2] \;\geq\; 3/4,$$

or in other words,

$$\Pr_p[\mathcal{A}(X) < p + \varepsilon] \;\geq\; 3/4. \tag{13}$$

Similarly, we can show that

$$\Pr_{p'}[\mathcal{A}(X') > p + \varepsilon] \;\geq\; 3/4, \tag{14}$$

since $p' \geq p + 2\varepsilon$. Now define the event

$$A \;:=\; \{Y \in \{0,1\}^{t-1} :\; \mathcal{A}(Y) > p + \varepsilon\}.$$

Now (13) implies that $\Pr_p(A) < 1/4$ and (14) implies that $\Pr_{p'}(A) \geq 3/4$. But then by Lemma 9 we have

$$\frac{1}{2} \;<\; |\Pr_p(A) - \Pr_{p'}(A)| \;\leq\; d_{TV}(B_p^{t-1}, B_{p'}^{t-1}) \;\leq\; 2\sqrt{(p'-p)^2(t-1)} \;\leq\; 2\sqrt{16\varepsilon^2(t-1)} \;\leq\; \frac{1}{2},$$

a contradiction. □

We now show how to remove the deterministic requirement on $\mathcal{A}$:

Corollary 1. *The bound of Lemma 10 holds even if $\mathcal{A}$ is randomized:*

$$\mathop{\mathbb{E}}_{p,R}[\mathit{regret}_t] + \mathop{\mathbb{E}}_{p',R}[\mathit{regret}_t] \;\geq\; \frac{1}{4}\varepsilon^2,$$

where $\mathbb{E}_{p,R}[\cdot]$ denotes the expectation computed over the random seed R of the algorithm as well as the randomness in the cost functions.

Proof. Fixing the random seed R of $\mathcal{A}$, we get a deterministic algorithm, and then Lemma 10 gives the following bound on the sum of the conditional expected regrets:

$$\mathop{\mathbb{E}}_{p}[\mathrm{regret}_t|R] + \mathop{\mathbb{E}}_{p'}[\mathrm{regret}_t|R] \;\geq\; \frac{1}{4}\varepsilon^2.$$

Now taking expectations over the random seed R, we get the desired bound. □

Thus, from now on we allow $\mathcal{A}$ to be randomized. We now show the desired lower bound on the expected regret:

Theorem 7. *The expected regret for algorithm $\mathcal{A}$ is at least $\Omega(\log(T))$.*

Proof. We prove this by showing that there is one value of $p \in [\frac{1}{4}, \frac{3}{4}]$ such that regret of $\mathcal{A}$ when cost functions are drawn from B_p is at least $\Omega(\log(T))$.

We assume that T is of the form $16+16^2+\cdots 16^k = \frac{1}{15}(16^{k+1}-16)$ for some integer k: if it isn't, we ignore all rounds $t > T'$, where $T' = \frac{1}{15}(16^{k^\star+1} - 16)$ for $k^\star = \lfloor \log_{16}(15T + 16) - 1 \rfloor$, and show that in the first T' rounds the algorithm can be made to have $\Omega(\log(T))$ regret. We now divide the time periods $t = 1, 2, \ldots, T'$ into consecutive epochs of length $16, 16^2, \ldots, 16^{k^\star}$. Thus, epoch k, denoted E_k, has length 16^k, and consists of the time periods $t = \frac{1}{15}(16^k - 16) + 1, \ldots, \frac{1}{15}(16^{k+1} - 16)$. We prove the following lemma momentarily:

Lemma 11. *There exists a collection of nested intervals, $[\frac{1}{4}, \frac{3}{4}] \supseteq I_1 \supseteq I_2 \supseteq I_3 \supseteq \cdots$, such that interval I_k corresponds to epoch k, with the property that I_k has length $4^{-(k+3)}$, and for every $p \in I_k$, for at least half the rounds t in epoch k, algorithm $\mathcal{A}$ has $\mathbb{E}_{p,R}[regret_t] \geq \frac{1}{8} \cdot 16^{-(k+3)}$.*

As a consequence of this lemma, we get that there is a value of $p \in \bigcap_k I_k$ such that in every epoch k, the total regret is

$$\sum_{t \in E_k} \frac{1}{8} \cdot 16^{-(k+3)} \geq \frac{1}{2} 16^k \cdot \frac{1}{8} \cdot 16^{-(k+3)} = \frac{1}{16^4}.$$

Thus, the regret in every epoch is $\Omega(1)$. Since there are $k^\star = \Theta(\log(T))$ epochs total, the regret of the algorithm is at least $\Omega(\log(T))$. □

We now turn to prove Lemma 11.

Lemma 11. We build the nested collection of intervals iteratively as follows. For notational convenience, define I_0 to be some arbitrary interval of length 4^{-3} inside $[\frac{1}{4}, \frac{3}{4}]$. Suppose for some $k \geq 0$ we have found the interval $I_k = [a, a + 4^{-(k+3)}]$. We want to find the interval I_{k+1} now. For this, divide up I_k into 4 equal quarters of length $\varepsilon = 4^{-(k+4)}$, and consider the first and fourth quarters, viz. $L = [a, a + 4^{-(k+4)}]$ and $R = [a + 3 \cdot 4^{-(k+4)}, a + 4^{-(k+3)}]$. We now show that one of L or R is a valid choice for I_{k+1}, and so the construction can proceed.

Suppose L is not a valid choice for I_{k+1}, because there is some point $p \in L$ such that for more than half the rounds t in E_{k+1}, we have $\mathbb{E}_{p,R}[\text{regret}_t] < 16^{-(k+1)}$. Then we show that R is a valid choice for I_{k+1} as follows. Let $H = \{t \in E_{k+1} : \mathbb{E}_{p,R}[\text{regret}_t] < \frac{1}{8} \cdot 16^{-(k+4)}\}$. Now, we claim that for all $p' \in R$, and all $t \in H$, we must have $\mathbb{E}_{p',R}[\text{regret}_t] > \frac{1}{8} \cdot 16^{-(k+4)}$, which would imply that R is a valid choice for I_{k+1}, since by assumption, $|H| \geq \frac{1}{2}|E_{k+1}|$.

To show this we apply Lemma 10. Fix any $p' \in R$ and $t \in H$. First, note that $\varepsilon = 4^{-(k+4)} \leq \frac{1}{16\sqrt{t}}$, since $t \leq 16^{k+2}$. Next, we have $p' - p \geq 2\varepsilon$ (since we excluded the middle two quarters of I_k), and $|p - p'| \leq 4\varepsilon$ (since I_k has length $4^{-(k+3)}$). Then Lemma 10 implies that

$$\mathop{\mathbb{E}}_{p,R}[\text{regret}_t] + \mathop{\mathbb{E}}_{p',R}[\text{regret}_t] \geq \frac{1}{4} \cdot 16^{-(k+4)},$$

which implies that $\mathbb{E}_{p',R}[\text{regret}_t] \geq \frac{1}{8} \cdot 16^{-(k+4)}$ since $\mathbb{E}_{p,R}[\text{regret}_t] < \frac{1}{8} \cdot 16^{-(k+4)}$, as required. □

5.1 Dependence on the Gradient Bound and on Strong Convexity

A simple corollary of the previous proof gives us tight lower bounds in terms of the natural parameters of the problem: the strong-convexity parameter λ and the upper bound on the norm of the subgradients G. The following Corollary implies Theorem 2.

Corollary 2. *For any algorithm $\mathcal{A}$, there is distribution over λ-strongly convex cost functions over a bounded domain $\mathcal{K} \subset \mathbb{R}$ with gradients bounded in norm by G such that the expected regret of $\mathcal{A}$ is $\Omega\left(\frac{G^2 \log(T)}{\lambda}\right)$.*

Proof. The online convex optimization setting we design is very similar: let $\lambda, G \geq 0$ be given parameters. The domain is $\mathcal{K} = [0, \frac{G}{\lambda}]$. In round t, we choose $X_t \in \{0, 1\}$ from B_p, and return

$$f_t(x) = \frac{\lambda}{2}\left(x - \frac{G}{\lambda}X_t\right)^2$$

as the cost function. Notice that the cost functions are always λ-strongly convex, and in addition, for any $x \in \mathcal{K}$, the gradient of the cost function at x is bounded in norm by G.

Denote $x' = \frac{\lambda x}{G}$ to be the scaled decision x, mapping it from $\mathcal{K}$ to $[0, 1]$. The expected cost when playing $x \in \mathcal{K}$ is given by

$$\mathbb{E}[f_t(x)] = \mathop{\mathbb{E}}_{X \sim B_p}\left[\frac{\lambda}{2}\left(x - \frac{G}{\lambda}X_t\right)^2\right] = \frac{G^2}{2\lambda}\,\mathbb{E}[(x' - X_t)^2]. \tag{15}$$

Given an algorithm $\mathcal{A}$ for this online convex optimization instance, we derive another algorithm, $\mathcal{A}'$, which plays points $x' \in \mathcal{K}' = [0, 1]$ and receives the cost function $(x' - X_t)^2$ in round t (i.e., the setting considered in Section 5). When $\mathcal{A}$ plays x_t in round t and obtains cost function $\frac{\lambda}{2}\left(x - \frac{G}{\lambda}X_t\right)^2$, the algorithm $\mathcal{A}'$ plays the point $x'_t = \frac{\lambda}{G}x_t$ and receives the cost function $(x' - X_t)^2$.

The optimum point for the setting of $\mathcal{A}$ is $\frac{G}{\lambda}p$, with expected cost $\frac{G^2}{2\lambda}$ times the expected cost for the optimum point p for the setting of $\mathcal{A}'$. By equation (15), the cost of $\mathcal{A}$ is $\frac{G^2}{2\lambda}$ times that of $\mathcal{A}'$. Hence, the regret of $\mathcal{A}$ is $\frac{G^2}{2\lambda}$ times that of $\mathcal{A}'$.

By Theorem 7, there is a value of p such that the expected regret of $\mathcal{A}'$ is $\Omega(\log T)$, and hence the expected regret of $\mathcal{A}$ is $\Omega\left(\frac{G^2 \log(T)}{\lambda}\right)$, as required. □

6 Conclusions

We have given an algorithm for stochastic strongly-convex optimization with an optimal rate of convergence $O(\frac{1}{T})$. The Epoch-GD algorithm has an appealing feature of returning the average of the most recent points (rather than all points visited by the algorithm as in previous approaches). This is an intuitive feature which, as demonstrated by Rakhlin et al. [2012], works well in practice for important applications such as support vector machine training.

Our analysis deviates from the common template of designing a regret minimization algorithm and then using online-to-batch conversion. In fact, we show that the latter approach is inherently suboptimal by our new lower bound on the regret of online algorithms for stochastic cost functions. This combination of results formally shows that the batch stochastic setting is strictly easier than its online counterpart, giving us tighter bounds.

A few questions remain open. The high-probability bound algorithm Epoch-GD-Proj has an extra factor of $O(\log\log(T))$ in its convergence rate. Is it possible to devise an algorithm that has $O(\frac{1}{T})$ convergence rate with high probability? We believe the answer is yes; the $O(\log\log(T))$ is just an artifact of the analysis. In fact, as we mention in Section 4, if it is possible to evaluate F efficiently at any given point, then this dependence can be removed. Also, our lower bound proof is somewhat involved. Are there easier information theoretic arguments that give similar lower bounds?

Acknowledgements

We gratefully acknowledge the help of two anonymous reviewers that gave insightful feedback and significantly helped shape the final version of this manuscript.

References

Jacob Abernethy, Alekh Agarwal, Peter L. Bartlett, and Alexander Rakhlin. A stochastic view of optimal regret through minimax duality. In *COLT*, 2009.

Alekh Agarwal, Peter L. Bartlett, Pradeep D. Ravikumar, and Martin J. Wainwright. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. *IEEE Transactions on Information Theory*, 58(5):3235–3249, 2012.

Peter L. Bartlett, Elad Hazan, and Alexander Rakhlin. Adaptive online gradient descent. In *NIPS*, 2007.

Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2nd edition, September 1999. ISBN 1886529000.

Jonathan M. Borwein and Adrian S. Lewis. *Convex Analysis and Nonlinear Optimization*. Springer, 2006.

Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. In *NIPS*, 2007.

Lev M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7:200–217, 1967.

Nicolò Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

Saeed Ghadimi and Guanghui Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization. In *Optimization Online*, 2010. URL `http://www.optimization-online.org/DB_HTML/2010/07/2669.html`.

Elad Hazan and Satyen Kale. An optimal algorithm for stochastic strongly-convex optimization. In *arXiv:1006.2425v1*, June 2010. URL `http://arxiv.org/abs/1006.2425`.

Elad Hazan and Satyen Kale. Beyond the regret minimization barrier: an optimal algorithm for stochastic strongly-convex optimization. In *COLT*, 2011.

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

Anatoli Juditsky and Yuri Nesterov. Primal-dual subgradient methods for minimizing uniformly convex functions. August 2010. URL `http://hal.archives-ouvertes.fr/docs/00/50/89/33/PDF/Strong-hal.pdf`.

Guanghui Lan, Arkadi Nemirovski, and Alexander Shapiro. Validation analysis of mirror descent stochastic approximation method. *Math. Program.*, 134(2):425–458, 2012.

Arkadi S. Nemirovski and David B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. John Wiley UK/USA, 1983.

Erik Ordentlich and Thomas M. Cover. The cost of achieving the best portfolio in hindsight. *Mathematics of Operations Research*, 23:960–982, November 1998.

Alexander Rakhlin, Ohad Shamir, and Karthik Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. In *ICML*, 2012.

Shai Shalev-Shwartz and Nathan Srebro. SVM optimization: inverse dependence on training set size. In *ICML*, 2008.

Shai Shalev-Shwartz, Ohad Shamir, Karthik Sridharan, and Nati Srebro. Stochastic convex optimization. In *COLT*, 2009.

Ohad Shamir and Tong Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *ICML*, 2013.

Eiji Takimoto and Manfred K. Warmuth. The minimax strategy for gaussian density estimation. In *COLT*, 2000.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, 2003.