Beyond the regret minimization barrier: an optimal algorithm for stochastic strongly-convex optimization

Elad Hazan
Technion - Israel Institute of Technology
ehazan@ie.technion.ac.il

Satyen Kale
Yahoo! Research
skale@yahoo-inc.com

Abstract

We give a novel algorithm for stochastic strongly-convex optimization in the gradient oracle model which returns an $O(\frac{1}{T})$-approximate solution after T gradient updates. 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 lower bound holds even in the full-information setting which reveals more information to the algorithm than just gradients. 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 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 was extensively discussed in Bottou and Bousquet (2007) and Shalev-Shwartz and Srebro (2008)). This makes first-order methods ideal for stochastic convex optimization: their error decreases as a polynomial in the number of iterations, usually 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 function on a convex, compact 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 gradient 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 $\mathbf{x}$.

An important special case is 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 training (see Shalev-Shwartz et al. (2009)).

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

Theorem 1 *Assume that F is λ-strongly convex and the gradient oracle is G-bounded. Then there exists an 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. (2010) 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[1] $f(\cdot, z_t)$. A simple analysis then shows that the cost of the average point, $\bar{x} = \frac{1}{T}\sum_{t=1}^{T} x_t$, 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 in 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*

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

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 gradient norms bounded by G.

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 $O(\frac{\log\log(T)}{T})$:

[1] Note that we are assuming that we have full access to the function $f(\cdot, z_t)$ here, rather than just gradient information.

	Previous 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	$\Omega(1)$ [trivial bound]	$\Omega\left(\frac{G^2 \log(T)}{\lambda}\right)$

Theorem 3 *Assume that F is λ-strongly convex, and the gradient oracle is 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*

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

1.1 Related work

For an in depth discussion of first-order methods, the reader is referred to Bertsekas (1999).

The study of lower bounds for stochastic convex optimization was undertaken in Nemirovski and Yudin (1983), and recently extended and refined in Agarwal et al. (2010).

Online convex optimization was introduced in Zinkevich (2003). Optimal lower bounds for the convex case, even in the stochastic setting, of $\Omega(\sqrt{T})$ are simple and given in Cesa-Bianchi and Lugosi (2006). For exp-concave cost functions, Ordentlich and Cover (1998) gave 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) gave 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) have recently published a technical report that has a very similar algorithm to ours, and also obtain an $O(\frac{1}{T})$ convergence rate. This work however was done independently and a preliminary version was published on arXiv (Hazan and Kale (2010)) before the technical report of Juditsky and Nesterov was available.

2 Setup and Background

2.1 Stochastic convex optimization

Consider the setting of stochastic convex optimization of a convex function F over a convex, compact set $\mathcal{K} \subseteq \mathbb{R}^n$. Let $\mathbf{x}^\star$ be a point in $\mathcal{K}$ where F is minimized. We make the following assumptions:

1. We assume that F is **λ-strongly convex**: i.e., for any two points $\mathbf{x}, \mathbf{y} \in \mathcal{K}$ and any $\alpha \in [0, 1]$, we have
$$F(\alpha\mathbf{x} + (1-\alpha)\mathbf{y}) \leq \alpha F(\mathbf{x}) + (1-\alpha)F(\mathbf{y}) - \frac{\lambda}{2}\alpha(1-\alpha)\|\mathbf{x} - \mathbf{y}\|^2.$$
F is λ-strongly-convex, for example, if $F(\mathbf{x}) = \mathbb{E}_Z[f(\mathbf{x}, Z)]$ and $f(\cdot, z)$ is λ-strongly-convex for every z in the support of Z.
This implies F satisfies the following inequality (to see this, set $\mathbf{y} = \mathbf{x}^\star$, divide by α, and take the limit as $\alpha \to 0^+$):
$$F(\mathbf{x}) - F(\mathbf{x}^\star) \geq \frac{\lambda}{2}\|\mathbf{x} - \mathbf{x}^\star\|^2 \tag{2}$$
This inequality holds even if $\mathbf{x}^\star$ is on the boundary of $\mathcal{K}$. In fact, (2) is the *only* requirement on the convexity of F for the analysis to work, we will simply assume that (2) holds.

2. Assume we have oracle access to compute an unbiased estimator of a subgradient of F at any point $\mathbf{x}$, denoted $\hat{\mathbf{g}} \in \partial F(\mathbf{x})$, whose ℓ_2 norm bounded by some known value $\|\hat{\mathbf{g}}\| \leq G$. Such a gradient oracle is called **G-bounded**.

3. Assume that the domain $\mathcal{K}$ is endowed with an efficiently computable **projection operator** $\prod_{\mathcal{K}}(\mathbf{y}) = \arg\min_{\mathbf{x} \in \mathcal{K}} \|\mathbf{x} - \mathbf{y}\|$.

Assumptions 1 and 2 above imply the following lemma:

Lemma 4 *For all $\mathbf{x} \in \mathcal{K}$, we have $F(\mathbf{x}) - F(\mathbf{x}^\star) \leq \frac{2G^2}{\lambda}$.*

Proof: For any $\mathbf{x} \in \mathcal{K}$, let $\hat{\mathbf{g}} \in \partial F(\mathbf{x})$ be a subgradient of F at $\mathbf{x}$ such that $\|\hat{\mathbf{g}}\| \leq G$ (using assumption 2). Then by the convexity of F, we have $F(\mathbf{x}) - F(\mathbf{x}^\star) \leq \hat{\mathbf{g}} \cdot (\mathbf{x} - \mathbf{x}^\star)$, so that $F(\mathbf{x}) - F(\mathbf{x}^\star) \leq G\|\mathbf{x} - \mathbf{x}^\star\|$. But assumption 1 implies that $F(\mathbf{x}) - F(\mathbf{x}^\star) \geq \frac{\lambda}{2}\|\mathbf{x} - \mathbf{x}^\star\|^2$. Putting these together, we get that $\|\mathbf{x} - \mathbf{x}^\star\| \leq \frac{2G}{\lambda}$. Finally, we get $F(\mathbf{x}) - F(\mathbf{x}^\star) \leq G\|\mathbf{x} - \mathbf{x}^\star\| \leq \frac{2G^2}{\lambda}$. Since $\mathbf{x}^\star$ is the minimizer of F on $\mathcal{K}$, the lemma follows. ■

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

$$\text{AdversarialRegret} \;:=\; \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x}\in\mathcal{K}} \sum_{t=1}^{T} f_t(\mathbf{x}). \tag{3}$$

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, $F(\mathbf{x}) = \mathbb{E}_D[f_1(\mathbf{x})]$:

$$\text{StochasticRegret} \;:=\; \mathop{\mathbb{E}}_{D}\left[\sum_{t=1}^{T} F(\mathbf{x}_t)\right] - T \min_{\mathbf{x}\in\mathcal{K}} F(\mathbf{x}). \tag{4}$$

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.

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

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

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

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

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 5 *Set the parameters $T_1 = 2$ 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 $\mathbb{E}[F(\mathbf{x}_1^k)] - F(\mathbf{x}^\star) \leq \frac{8G^2}{\lambda T}$. The total number of gradient updates is at most T.*

The intra-epoch use of standard gradient decent is analyzed using the following Lemma from Zinkevich (2003), which we prove here for completeness:

Lemma 6 (Zinkevich (2003)) *Let $\|\hat{\mathbf{g}}_t\| \leq G$. Apply T iterations of the update $\mathbf{x}_{t+1} = \prod_{\mathcal{K}}(\mathbf{x}_t - \eta\hat{\mathbf{g}}_t)$. Then*

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

Proof: We use $\|\mathbf{x}_t - \mathbf{x}^\star\|^2$ as a potential function. Let $\mathbf{x}'_{t+1} = \mathbf{x}_t - \eta\hat{\mathbf{g}}_t$. We have

$$\|\mathbf{x}'_{t+1} - \mathbf{x}^\star\|^2 = \eta^2\|\hat{\mathbf{g}}_t\|^2 + \|\mathbf{x}_t - \mathbf{x}^\star\|^2 - 2\eta\hat{\mathbf{g}}_t \cdot (\mathbf{x}_t - \mathbf{x}^\star) \leq \eta^2 G^2 + \|\mathbf{x}_t - \mathbf{x}^\star\|^2 - 2\eta\hat{\mathbf{g}}_t \cdot (\mathbf{x}_t - \mathbf{x}^\star).$$

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

$$\mathbf{x}_{t+1}^k = \prod_{\mathcal{K}} (\mathbf{x}_t^k - \eta_k \hat{\mathbf{g}}_t)$$

  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 $\mathbf{x}_1^k$.

Since $\mathbf{x}_{t+1} = \prod_{\mathcal{K}}(\mathbf{x}'_{t+1})$ and $\mathbf{x}^\star \in \mathcal{K}$, the fact that projections of an external point on a convex set reduces the distance to any point inside it (see Zinkevich (2003)) implies that $\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2 \leq \|\mathbf{x}'_{t+1} - \mathbf{x}^\star\|^2$. Putting these together, and rearranging, we get

$$\hat{\mathbf{g}}_t \cdot (\mathbf{x}_t - \mathbf{x}^\star) \leq \frac{\eta G^2}{2} + \frac{\|\mathbf{x}_t - \mathbf{x}^\star\|^2 - \|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2}{2\eta}.$$

Summing up over all $t = 1, 2, \ldots, T$, we get the stated bound. ■

Lemma 7 *Apply T iterations of the update $\mathbf{x}_{t+1} = \prod_{\mathcal{K}}(\mathbf{x}_t - \eta \hat{\mathbf{g}}_t)$, where $\hat{\mathbf{g}}_t$ is an unbiased estimator for a subgradient $\mathbf{g}_t$ of F at $\mathbf{x}_t$ satisfying $\|\hat{\mathbf{g}}_t\| \leq G$. Then*

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

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 applying Lemma 6, we get the required bound. ■

Define $V_k = \frac{G^2}{2^{k-2}\lambda}$, and notice that the algorithm sets $T_k = \frac{4G^2}{\lambda V_k}$ and $\eta_k = \frac{V_k}{2G^2}$. Define $\Delta_k = F(\mathbf{x}_1^k) - F(\mathbf{x}^\star)$. Using Lemma 7 we prove the following key lemma:

Lemma 8 *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 4. 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 7 we have

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

since $\Delta_k = F(\mathbf{x}_1^k) - F(\mathbf{x}^\star) \geq \frac{\lambda}{2}\|\mathbf{x}_1^k - \mathbf{x}^\star\|^2$ by λ-strong convexity of F. Hence, we get

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

as required. The second inequality uses the induction hypothesis, and the last two equalities use the definition of V_k and the values $\eta_k = \frac{V_k}{2G^2}$ and $T_k = \frac{4G^2}{\lambda V_k}$. ■

We can now prove our main theorem:

Proof:[Theorem 5.] The number of epochs made are given by the largest value of k satisfying $\sum_{i=1}^k T_i \leq T$, i.e.

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

This value is $k^\dagger = \lfloor \log_2(\frac{T}{2} + 1) \rfloor$. The final point output by the algorithm is $\mathbf{x}_1^{k^\dagger+1}$. Applying Lemma 8 to $k^\dagger + 1$ we get

$$\mathbb{E}[F(\mathbf{x}_1^{k^\dagger+1})] - F(\mathbf{x}^\star) \;=\; \mathbb{E}[\Delta_{k^\dagger+1}] \;\leq\; V_{k^\dagger+1} \;=\; \frac{G^2}{2^{k^\dagger-1}\lambda} \;\leq\; \frac{8G^2}{\lambda T},$$

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

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. The only difference in the new algorithm, dubbed Epoch-GD-Proj, is that the update in line 7 requires a projection onto a smaller set, and becomes

$$\text{Update} \;\; \mathbf{x}_{t+1}^k = \prod_{\mathcal{K} \cap B(\mathbf{x}_1^k, \sqrt{2V_k/\lambda})} (\mathbf{x}_t^k - \eta_k \hat{\mathbf{g}}_t) \tag{5}$$

Here $B(\mathbf{x}, r)$ denotes the ℓ_2 ball of radius r around the point x, and $V_k = \frac{G^2}{2^{k-2}\lambda}$ as defined earlier. Since the intersection of two convex sets is also a convex set, the above projection can be computed via a convex program.

We prove the following high probability result, which in turn directly implies Theorem 3.

Theorem 9 *Given $\delta > 0$ for success probability $1 - \delta$, set $\tilde{\delta} = \frac{\delta}{k^\dagger}$ for $k^\dagger = \lfloor \log_2(\frac{T}{300} + 1) \rfloor$. Set the parameters $T_1 = 300 \log(1/\tilde{\delta})$ and $\eta_1 = \frac{1}{3\lambda}$ 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*

$$F(\mathbf{x}_1^k) - F(\mathbf{x}^\star) \leq \frac{1200G^2 \log(1/\tilde{\delta})}{\lambda T}.$$

The total number of gradient updates is at most T.

The following lemma is analogous to Lemma 7, but provides a high probability guarantee.

Lemma 10 *Let D be an upper bound on $\|\mathbf{x}_1 - \mathbf{x}^\star\|$. Apply T iterations of the update $\mathbf{x}_{t+1} = \prod_{\mathcal{K}\cap B(\mathbf{x}_1,D)}(\mathbf{x}_t - \eta\hat{\mathbf{g}}_t)$, where $\hat{\mathbf{g}}_t$ is an unbiased estimator for the subgradient of F at $\mathbf{x}_t$ satisfying $\|\hat{\mathbf{g}}_t\| \le G$. Then for any $\delta \in (0,1)$, with probability at least $1-\delta$ we have*

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

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

Proof: Using the same notation as in the proof of Lemma 7, let $\mathbb{E}_{t-1}[\hat{\mathbf{g}}_t] = \mathbf{g}_t$, a subgradient of F at $\mathbf{x}_t$. Since as before, $\mathbb{E}_{t-1}[\hat{\mathbf{g}}_t \cdot (\mathbf{x}_t - \mathbf{x}^\star)] = \mathbf{g}_t \cdot (\mathbf{x}_t - \mathbf{x}^\star)$, the following defines as 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 $\|\mathbf{g}_t\| = \|\,\mathbb{E}_{t-1}[\hat{\mathbf{g}}_t]\| \le \mathbb{E}_{t-1}[\|\hat{\mathbf{g}}_t\|] \le G$, and so we can bound $|X_t|$ as follows:

$$|X_t| \le \|\mathbf{g}_t\|\|\mathbf{x}_t - \mathbf{x}^\star\| + \|\hat{\mathbf{g}}_t\|\|\mathbf{x}_t - \mathbf{x}^\star\| \;\le\; 4GD,$$

where the last inequality uses the fact that $\mathbf{x}^\star, \mathbf{x}_t \in B(\mathbf{x}_1, D)$, and hence by the triangle inequality $\|\mathbf{x}_t - \mathbf{x}^\star\| \le \|\mathbf{x}_t - \mathbf{x}_1\| + \|\mathbf{x}_1 - \mathbf{x}^\star\| \le 2D$.

By Azuma's inequality (see Lemma 12), with probability at least $1-\delta$, the following holds:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbf{g}_t \cdot (\mathbf{x}_t - \mathbf{x}^\star) - \frac{1}{T}\sum_{t=1}^{T}\hat{\mathbf{g}}_t \cdot (\mathbf{x}_t - \mathbf{x}^\star) \;\le\; \frac{4GD\sqrt{2\log(1/\delta)}}{\sqrt{T}}. \tag{6}$$

By the convexity of F, we have $F(\mathbf{x}_t) - F(\mathbf{x}^\star) \le \mathbf{g}_t \cdot (\mathbf{x}_t - \mathbf{x}^\star)$. Then, by using Lemma 6 and inequality (6), we get the claimed bound. ∎

We now prove the analogue of Lemma 8. In this case, the result holds with high probability. As before, define $V_k = \frac{G^2}{2^{k-2}\lambda}$, and notice that the algorithm sets $T_k = \frac{600G^2\log(1/\tilde{\delta})}{\lambda V_k}$ and $\eta_k = \frac{V_k}{6G^2}$.

Lemma 11 *For any k, with probability $(1-\tilde{\delta})^{k-1}$ we have $\Delta_k \le V_k$.*

Proof: We prove this by induction on k. The claim is true for $k=1$ since $\Delta_k \le \frac{2G^2}{\lambda}$ by Lemma 4. Assume that $\Delta_k \le V_k$ for some $k \ge 1$ with probability at least $(1-\tilde{\delta})^{k-1}$ and now we prove it for $k+1$. We condition on the event that $\Delta_k \le V_k$. Since $\Delta_k \ge \frac{\lambda}{2}\|\mathbf{x}_1^k - \mathbf{x}^\star\|^2$ by λ-strong convexity, this conditioning implies that $\|\mathbf{x}_1^k - \mathbf{x}^\star\| \le \sqrt{2V_k/\lambda}$, which explains the specification (see (5) for the radius of the ball for the projection in line 7 of Epoch-GD-Proj). So Lemma 10 applies with $D = \sqrt{2V_k/\lambda}$ and hence we have with probability at least $1-\tilde{\delta}$,

$$\begin{aligned}
\Delta_{k+1} \;&=\; F(\mathbf{x}_1^{k+1}) - F(\mathbf{x}^\star) \\
&\le\; \frac{\eta_k G^2}{2} + \frac{\|\mathbf{x}_1^k - \mathbf{x}^\star\|^2}{2\eta_k T_k} + \frac{8G\sqrt{V_k}\sqrt{\log(1/\tilde{\delta})}}{\sqrt{\lambda T_k}} \qquad \text{(by Lemma 10)} \\
&\le\; \frac{\eta_k G^2}{2} + \frac{V_k}{\eta_k T_k \lambda} + \frac{8G\sqrt{V_k}\sqrt{\log(1/\tilde{\delta})}}{\sqrt{\lambda T_k}},
\end{aligned}$$

since $V_k \ge \Delta_k \ge \frac{\lambda}{2}\|\mathbf{x}_1^k - \mathbf{x}^\star\|^2$ as above. For $T_k = \frac{600G^2\log(1/\tilde{\delta})}{\lambda V_k}$ and $\eta_k = \frac{V_k}{6G^2}$ we get

$$F(\mathbf{x}_1^{k+1}) - F(\mathbf{x}^\star) \;\le\; \frac{V_k}{12} + \frac{V_k}{100\log(1/\tilde{\delta})} + \frac{V_k}{3} \;\le\; \frac{V_k}{2} \;=\; V_{k+1}.$$

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

We can now prove our high probability theorem:

Proof:[Theorem 9] As in the proof of Theorem 1, we get that final epoch is $k^\dagger = \lfloor \log_2(\frac{T}{300} + 1)\rfloor$. The final point output is $\mathbf{x}_1^{k^\dagger+1}$.

By Lemma 11, we have with probability at least $(1-\tilde{\delta})^{k^\dagger}$ that

$$F(\mathbf{x}_1^{k^\dagger+1}) - F(\mathbf{x}^\star) \;=\; \Delta_{k^\dagger+1} \;\leq\; V_{k^\dagger+1} \;=\; \frac{G^2}{2^{k^\dagger-1}\lambda} \;\leq\; \frac{1200G^2\log(1/\tilde{\delta})}{\lambda T},$$

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. ■

For completeness we state Azuma's inequality for martingales used in the proof above:

Lemma 12 (Azuma's inequality) *Let $X_1, \ldots, X_T$ be a martingale difference sequence. Suppose that $|X_t| \leq b$. Then, for $\delta > 0$, we have*

$$\Pr\left[\sum_{t=1}^{T} X_t \geq b\sqrt{2T\ln(1/\delta)}\right] \;\leq\; \delta.$$

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 and the gradient oracle is 1-bounded, and fine tune these parameters in the next subsection via an easy 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 13 *Let $p, p' \in [\frac{1}{4}, \frac{3}{4}]$ such that $|p'-p| \leq 1/8$. Then*

$$d_{TV}(B_p^n, B_{p'}^n) \leq \frac{1}{2}\sqrt{(p'-p)^2 n}.$$

Proof: Pinsker's inequality says that $d_{TV}(B_p^n, B_{p'}^n) \leq \sqrt{\frac{1}{2}\mathrm{RE}(B_p^n \| B_{p'}^n)}$, 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

$$\mathrm{RE}(B_p^n \| B_{p'}^n) = n\mathrm{RE}(B_p \| B_{p'}) = n\left[p\log\left(\frac{p}{p'}\right) + (1-p)\log\left(\frac{1-p}{1-p'}\right)\right], \tag{7}$$

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

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

for $\varepsilon \leq 1/8$. Plugging this (7) 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[2] 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 14 *Fix a round t. Let $\varepsilon \leq \frac{1}{8\sqrt{t}}$ 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*

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

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

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

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,

$$\Pr_p[\mathcal{A}(X) < p + \varepsilon] \;\geq\; 3/4. \tag{8}$$

Similarly, we can show that

$$\Pr_{p'}[\mathcal{A}(X') > p + \varepsilon] \;\geq\; 3/4, \tag{9}$$

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

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

Now (8) implies that $\Pr_p(A) < 1/4$ and (9) implies that $\Pr_{p'}(A) \geq 3/4$. But then by Lemma 13 we have

$$\frac{1}{2} \;<\; |\Pr_p(A) - \Pr_{p'}(A)| \;\leq\; d_{TV}(B_p^{t-1}, B_{p'}^{t-1}) \;\leq\; \frac{1}{2}\sqrt{(p'-p)^2(t-1)} \;\leq\; \frac{1}{2}\sqrt{16\varepsilon^2(t-1)} \leq \frac{1}{4},$$

a contradiction. ∎

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

Corollary 15 *The bound of Lemma 14 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.

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

Proof: Fixing the random seed R of $\mathcal{A}$, we get a deterministic algorithm, and then Lemma 14 gives the following bound on the sum of the conditional expected regrets:

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

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 16 *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 show the following claim now:

Claim 17 *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 claim, 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 $\Omega(\log(T))$. So now we prove the claim.

Proof: 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 14. Fix any $p' \in R$ and $t \in F$. First, note that $\varepsilon = 4^{-(k+4)} \leq \frac{1}{8\sqrt{t}}$, 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 14 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 18 *For any algorithm $\mathcal{A}$, there is distribution over λ-strongly convex cost functions with gradients bounded in norm by G such that the expected regret of $\mathcal{A}$ is $\Omega\Big(\frac{G^2 \log(T)}{\lambda}\Big)$.*

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 expectation of the cost when playing $x \in \mathcal{K}$ is given by

$$\mathbb{E}[f_t(x)] = \mathop{\mathbb{E}}_{X \sim B_p}\left[\frac{\lambda}{2}\left(x - \frac{G}{\lambda}X_t\right)^2\right] = \frac{G^2}{2\lambda}\,\mathbb{E}[(x' - X_t)^2] \tag{10}$$

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 (10), 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 16, 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 algorithm itself 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 hopefully 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 artefact 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 to give similar lower bounds?

Acknowledgements

We thank an anonymous referee for several useful suggestions.

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 Ravikumar, and Martin J. Wainwright. Information-theoretic lower bounds on the oracle complexity of convex optimization. In *arXiv:1009.0571v1*, 2010.

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

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

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

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

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`.

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. *Math. Oper. Res.*, 23:960–982, November 1998.

Shai Shalev-Shwartz and Nathan Srebro. SVM optimization: inverse dependence on training set size. In *ICML*, pages 928–935, 2008.

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

Eiji Takimoto and Manfred K. Warmuth. The minimax strategy for gaussian density estimation. In *COLT*, pages 100–106, 2000.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, pages 928–936, 2003.