Online convex optimization in the bandit setting: gradient descent without a gradient

Abraham D. Flaxman * Adam Tauman Kalai † H. Brendan McMahan ‡

October 17, 2004

Abstract

We study a general online convex optimization problem. We have a convex set S and an unknown sequence of cost functions $c_1, c_2, \ldots$, and in each period, we choose a feasible point x_t in S, and learn the cost $c_t(x_t)$. If the function c_t is also revealed after each period then, as Zinkevich shows in [25], gradient descent can be used on these functions to get regret bounds of $O(\sqrt{n})$. That is, after n rounds, the total cost incurred will be $O(\sqrt{n})$ more than the cost of the best single feasible decision chosen with the benefit of hindsight, $\min_x \sum c_t(x)$.

We extend this to the "bandit" setting, where, in each period, only the cost $c_t(x_t)$ is revealed, and bound the expected regret as $O(n^{3/4})$.

Our approach uses a simple approximation of the gradient that is computed from evaluating c_t at a single (random) point. We show that this biased estimate is sufficient to approximate gradient descent on the sequence of functions. In other words, it is possible to use gradient descent without seeing anything more than the value of the functions at a single point. The guarantees hold even in the most general case: online against an adaptive adversary.

For the online linear optimization problem [15], algorithms with low regrets in the bandit setting have recently been given against oblivious [1] and adaptive adversaries [19]. In contrast to these algorithms, which distinguish between explicit *explore* and *exploit* periods, our algorithm can be interpreted as doing a small amount of exploration in each period.

1 Introduction

Consider three optimization settings where one would like to minimize a convex function (equivalently maximize a concave function). In all three settings, gradient descent is one of the most popular methods.

1. Offline: Minimize a fixed convex cost function c: $\mathbb{R}^d \to \mathbb{R}$. In this case, gradient descent is $x_{t+1} = x_t - \eta \nabla c(x_t)$.

2. Stochastic: Minimize a fixed convex cost function c given only "noisy" access to c. For example, at time $T = t$, we may only have access to $c_t(x) = c(x) + \epsilon_t(x)$, where $\epsilon_t(x)$ is a random sampling error. Here, stochastic gradient descent is $x_{t+1} = x_t - \eta \nabla c_t(x_t)$. (The intuition is that the expected gradient is correct; $\mathbf{E}[\nabla c_t(x)] = \nabla \mathbf{E}[c_t(x)] = \nabla c(x)$.) In non-convex cases, the additional randomness may actually help avoid local minima [3], in a manner similar to Simulated Annealing [13].

3. Online: Minimize an adversarially generated sequence of convex functions, $c_1, c_2, \ldots$. This requires that we choose a sequence $x_1, x_2, \ldots$ where each x_t is selected based only on $x_1, x_2, \ldots, x_{t-1}$ and $c_1, c_2, \ldots, c_{t-1}$. The goals is to have low *regret* $\sum c_t(x_t) - \min_{x \in S} \sum c_t(x)$ for not using the best single point, chosen with the benefit of hindsight. In this setting, Zinkevich analyzes the regret of gradient descent given by $x_{t+1} = x_t - \eta \nabla c_t(x_t)$.

We will focus on gradient descent in a "bandit" version of the online setting. As a motivating example, consider a company that has to decide, every week, how much to spend advertising on each of d different channels, represented as a vector $x_t \in \mathbb{R}^d$. At the end of each week, they calculate their total profit $p_t(x_t)$. In the offline case, one might assume that each week the function $p_1, p_2, \ldots$ are identical. In the stochastic case, one might assume that in different weeks the profit functions $p_t(x)$ will be noisy realizations of some underlying "true" profit function, for example $p_t(x) = p(x) + \epsilon_t(x)$, where $\epsilon_t(x)$ has mean 0. In the online case, *no assumptions* are made about a distribution over convex profit functions and instead they are modeled as the malicious choices of an adversary. This allows, for example, for more complicated time-dependent random noise or the effects of a bad economy, or even an environment that responds to the choices we make (an adaptive adversary).

*http://www.math.cmu.edu/~adf, Department of Mathematical Sciences, Carnegie Mellon University.

†http://people.cs.uchicago.edu/~kalai, Toyota Technical Institute at Chicago.

‡http://www.cs.cmu.edu/~mcmahan, Department of Computer Science, Carnegie Mellon University.

In the bandit setting, we only have black-box access to the function(s) and thus cannot access the gradient of c_t directly for gradient descent. (In the advertising example, the advertisers only find out the total profit of their chosen x_t, and not how much they would have profited from other values of x.) This type of optimization is sometimes referred to as *direct* or *gradient-free*.

In direct offline and stochastic optimization, a natural approach is to is to estimate the gradient at a point by evaluating the function at several nearby points point. (this is called *Finite Difference Stochastic Approximation*, see, for example, Chapter 6 of [23]). In the online setting, the functions change adversarially over time and we only can evaluate each function once. We use a one-point estimate of the gradient to circumvent this difficulty. Earlier one-point estimates of the gradient are due to by Granichin and Spall [8, 22].

Independently, R. Kleinberg has recently shown surprisingly similar guarantees for the same problem we consider, using a slightly different technique: $O(n^{3/4})$ regret. We discuss the differences in the related work section.

1.1 A one-point gradient estimate. Our one-point estimate of the gradient of a function $f : \mathbb{R}^d \to \mathbb{R}$, in d dimensions, is the following, for a random unit vector u and small $\delta > 0$:

$$\nabla f(x) \approx \mathbf{E}\left[d\frac{f(x+\delta u)}{\delta}u\right]. \tag{1.1}$$

We first present the intuition and then the theoretical justification. It would seem that in d dimensions one would require $d+1$ evaluations to estimate the gradient of a function. However, if one is satisfied with a random variable whose expected value is the gradient, one can in fact get by with a single evaluation.

For one dimensional $f\colon \mathbb{R} \to \mathbb{R}$, the approximation is the following: since $u = \pm 1$ with equal probability,

$$\mathbf{E}\left[\frac{f(x+\delta u)}{\delta}u\right] = \frac{f(x+\delta) - f(x-\delta)}{2\delta} \approx f'(x)$$

So, *in expectation*, $f(x+\delta u)u/\delta$ is close to the derivative of f for $u = \pm 1$.

Since the gradient of a d-dimensional function $f\colon \mathbb{R}^d \to \mathbb{R}$ can be expressed as d one-dimensional derivatives,

$$\nabla f(x) = \left(\frac{df}{dx_1}(x), \ldots, \frac{df}{dx_d}(x)\right),$$

it is natural to approximate the gradient by choosing a random (signed) unit coordinate vector u, i.e. random from $(\pm 1, 0, 0, \ldots), (0, \pm 1, 0, \ldots), \ldots$, and then using the approximation (1.1). The extra factor of d enters due to the fact that one only estimate a single coordinate axis.

However, the gradient in general does not depend on which orthonormal basis one chooses, and thus we use (1.1) with a uniformly random unit vector u. Moreover, this is an approximation has a very natural precise interpretation, as we show in Section 2. In particular, $(d/\delta)f(x+\delta u)u$ is an unbiased estimator the gradient of a *smoothed* version of f, where the value of f at x is replaced by the average value of f in a ball of radius δ centered at x. For a vector v selected uniformly at random from the unit ball $\|v\| \leq 1$, let

$$\hat{f}(x) = \mathbf{E}[f(x+\delta v)].$$

Then

$$\nabla \hat{f}(x) = \mathbf{E}[f(x+\delta u)u]d/\delta.$$

Interestingly, this does not require that f be differentiable.

Our method of obtaining a one-point estimate of the gradient is similar to a one-point estimates proposed independently by Granichin [8] and Spall [22].[1] Spall's estimate uses a perturbation vector p, in which each entry is a zero-mean independent random variable, to produce an estimate of the gradient $\hat{g}(x) = \frac{f(x+\delta p)}{\delta}\left[\frac{1}{p_1}, \frac{1}{p_2}, \ldots, \frac{1}{p_d}\right]^T$. This estimate is more of a direct attempt to estimate the gradient coordinatewise and is not rotationally invariant. Spall's analysis focuses on the stochastic setting and requires that the function is three-times differentiable. In [9], Granichin shows that a similar approximation is sufficient to perform gradient descent in a very general stochastic model.

Unlike [8, 9, 22], we work in an adversarial model, where instead of trying to make the restrictions on the randomness of nature as weak as possible, we pessimistically assume that nature is conspiring against us. In the online setting where the functions are not necessarily related, or even in the adversarial setting, a one-point estimate of the gradient is sufficient to make gradient descent work.

1.2 Guarantees and outline of analysis. We use the following online bandit version of Zinkevich's model. There is a fixed unknown sequence of convex functions $c_1, c_2, \ldots, c_n\colon S \to [-C, C]$, where $C > 0$ and $S \subseteq \mathbb{R}^d$ is a convex feasible set. The decision-maker sequentially chooses points $x_1, x_2, \ldots, x_n \in S$. After x_t is chosen, the value $c_t(x_t)$ is revealed, and

[1] We are grateful to Rakesh Vohra for pointing out these earlier works on one-point gradient estimates.

x_{t+1} must be chosen only based on $x_1, x_2, \ldots, x_t$ and $c_1(x_1), c_2(x_2), \ldots, c_t(x_t)$ (and private randomness).

Zinkevich shows that, when the gradient $\nabla c_t(x_t)$ is given to the decision-maker after each period, an online gradient descent algorithm guarantees,

$$\text{(1.2)} \quad \text{regret} = \sum_{t=1}^{n} c_t(x_t) - \min_{x \in S} \sum_{t=1}^{n} c_t(x) \leq DG\sqrt{n}.$$

Here D is the diameter of the feasible set, and G is an upper bound on the magnitudes of the gradients.

By elaborating on his technique, we present an update rule for computing a sequence of x_{t+1} in the absence of $\nabla c_t(x_t)$, that gives the following guarantee on expected regret. If we assume that the functions satisfy an L-Lipschitz condition (which is slightly less restrictive than a bounded gradient assumption), then,

$$\mathbf{E}\left[\sum_{t=1}^{n} c_t(x_t) - \min_{x \in S} \sum_{t=1}^{n} c_t(x)\right] \leq 6n^{3/4} d \left(\sqrt{CLD} + C\right).$$

Interestingly, the analysis can be performed with no Lipschitz or gradient assumption, yielding slightly worse bounds in the general case:

$$\mathbf{E}\left[\sum_{t=1}^{n} c_t(x_t) - \min_{x \in S} \sum_{t=1}^{n} c_t(x)\right] \leq 6n^{5/6} dC.$$

As expected, our guarantees in the bandit setting are worse than those of the full-information setting: $O(n^{3/4})$ or $O(n^{5/6})$ instead of $O(n^{1/2})$. Also notice that we require a bound C on the value of the functions, but our second guarantee has no dependence on the diameter D.

To prove these bounds, we have several pieces to put together. First, we show that Zinkevich's guarantee (1.2) holds unmodified for vectors that are unbiased estimates of the gradients. Here G becomes an upper bound on the magnitude of the estimates.

Now, the updates should roughly be of the form $x_{t+1} = x_t - \eta(d/\delta)\,\mathbf{E}[c_t(x_t + \delta u_t)u_t]$. Since we can only evaluate each function at one point, that point should be $x_t + \delta u_t$. However, our analysis applies to bound $\sum c_t(x_t)$ and not $\sum c_t(x_t + \delta u_t)$. Fortunately, these points are close together and thus these values should not be too different.

Another problem that arises is that the perturbations may move points outside the feasible set. To deal with these issues, we stay on a subset of the set such that the ball of radius δ around each point in the subset is contained in S. In order to do this, it is helpful to have bounds on the radii r, R of balls that are contained in S and that contain S, respectively. Then guarantees can be given in terms of R/r. Finally, we can use existing algorithms [18] to reshape the body so $R/r \leq d$ to get the final results.

We first present our results against an oblivious adversary and then extend them to an adaptive one in Section 3.3. Though the adaptive setting requires a more delicate definition, the adaptive analysis follows naturally from the fact that we use a single-point estimate of the gradient and fresh randomness.

1.3 Related work. For direct offline optimization, i.e. from an oracle that evaluates the function, in theory one can use the ellipsoid [11] or more recent random-walk based approaches [4]. In black-box optimization, practitioners often use Simulated Annealing [13] or finite difference/simulated perturbation stochastic approximation methods (see, for example, [23]). In the case that the functions may change dramatically over time, a single-point approximation to the gradient may be necessary. Granichin and Spall propose other single-point estimates of the gradient in [8, 22].

In addition to the appeal of an online model of convex optimization, Zinkevich's gradient descent analysis can be applied to several other online problems for which gradient descent and other special-purpose algorithms have been carefully analyzed, such as Universal Portfolios [6, 10, 14], online linear regression [16], and online shortest paths [24] (one convexifies to get an online shortest ow problem).

Independently, recently (less than a month apart), and single-handedly, R. Kleinberg [12] has given strikingly similar $O(n^{3/4})$ bounds for the same problem using an extremely similar approach. The main difference is that his algorithm, more like the finite difference methods, breaks time into phases of length $d + 1$, and evaluates $d + 1$ successive functions, each at a single nearby point (carefully incorporating randomness), and uses these values to construct an estimate of the gradient. In light of our analysis, one can interpret his algorithm as performing $d + 1$ random *one-point* gradient estimates, i.e., even if he only used a random one of the periods per phase, his algorithm should work. The analysis is more delicate since his randomness is not fresh each period, and the bounds are proven only against an oblivious adversary (which must choose the entire sequence of functions in advance, and may not adapting to the choices made by the algorithm). Under reasonable conditions (bounds on the function values, gradients, and Hessians) he proves regret bounds $O(d^3 n^{3/4})$. Otherwise, he faces similar issues and uses similar techniques such as rounding the body by putting it in isotropic position.

A similar line of research has developed for the

problem of online linear optimization [15, 1, 19]. Here, one wants to solve the related but incomparable problem of optimizing a sequence of linear functions, over a possibly non-convex feasible set, modeling problems such as online shortest paths and online binary search trees (which are difficult to convexify). Kalai and Vempala [15] show that, for such linear optimization problems in general, if the offline optimization problem is solvable efficiently, then regret can be bounded by $O(\sqrt{n})$ also by an efficient online algorithm, in the full-information model. Awerbuch and Kleinberg [1] generalize this to the bandit setting against an oblivious adversary. Blum and McMahan [19] give a simpler algorithm that applies to *adaptive* adversaries, that may choose their functions c_t depending on the previous points.

A few comparisons are interesting to make with the online linear optimization problem. First of all, for the bandit versions of the linear problems, there was a distinction between exploration periods and exploitation periods. During exploration, one action from a *barycentric spanner* [1] basis of d actions was chosen, for the sole purpose of estimating the linear objective function. In contrast, our algorithm does a little bit of exploration in each phase. Secondly, Blum and McMahan's algorithm [19], like ours, uses single-period exploration to compete against an adaptive adversary, with a careful Martingale analysis.

1.4 Notation. Let $\mathbb{B}$ and $\mathbb{S}$ be the unit ball and sphere centered around the origin in d dimensions, respectively,

$$\mathbb{B} = \{x \in \mathbb{R}^d \,|\, |x| \leq 1\}$$
$$\mathbb{S} = \{x \in \mathbb{R}^d \,|\, |x| = 1\}$$

The ball and sphere of radius a are $a\mathbb{B}$ and $a\mathbb{S}$, correspondingly.

Until Section 3.3, we fix the sequence of functions $c_1, c_2, \ldots c_n$: $S \to \mathbb{R}$ is in advance (meaning we are considering an oblivious adversary, not an adaptive one). The sequence of points we pick is denoted by $x_1, x_2, \ldots, x_n$. For the bandit setting, we need to use randomness, so we consider our *expected regret*:

$$\mathbf{E}\left[\sum_{t=1}^{n} c_t(x_t)\right] - \min_z \sum_{t=1}^{n} c_t(z).$$

Zinkevich assumes the existence of a projection oracle $\mathbf{P}_S(x)$, projecting the point x onto the nearest point in the convex set S,

$$\mathbf{P}_S(x) = \arg\min_{z \in S} |x - z|.$$

Projecting onto the set is an elegant way to handle the situation that a step along the gradient goes outside of the set, and is a common technique in the optimization literature. Note that computing $\mathbf{P}_S$ is "only" an offline convex optimization problem. While, for arbitrary feasible sets, this may be difficult in practice, for standard shapes, such as cube, ball, and simplex, the calculation is quite straightforward.

A function f is L-Lipschitz if

$$|f(x) - f(y)| \leq L|x - y|,$$

for all x, y in the domain of f.

We assume S contains the ball of radius r centered at the origin and is contained in the ball of radius R, i.e.,

$$r\mathbb{B} \subseteq S \subseteq R\mathbb{B}.$$

Technically, we must also assume that S is a closed set, so that the projection operation is well defined.

2 Approximating the gradient with a single sample

The main observation of this section is that we can estimate the gradient of a function f by taking a random unit vector u and scaling it by $f(x + \delta u)$, i.e. $\hat{g} = f(x + \delta u)u$. The approximation is correct in the sense that $\mathbf{E}[\hat{g}]$ is proportional to the gradient of a *smoothed* version of f. For any function f, for v selected uniformly at random from the unit ball, define

$$\hat{f}(x) = \mathbf{E}_{v \in \mathbb{B}}[f(x + \delta v)]. \tag{2.3}$$

LEMMA 2.1. *Fix $\delta > 0$, over random unit vectors u,*

$$\mathbf{E}_{u \in \mathbb{S}}[f(x + \delta u)u] = \frac{\delta}{d} \nabla \hat{f}(x).$$

Proof. If $d = 1$, then the fundamental theorem of calculus implies,

$$\frac{d}{dx} \int_{-\delta}^{\delta} f(x + v)dv = f(x + \delta) - f(x - \delta).$$

The d-dimensional generalization, which follows from Stoke's theorem, is,

$$\nabla \int_{\delta\mathbb{B}} f(x + v)dv = \int_{\delta\mathbb{S}} f(x + u)\frac{u}{\|u\|}du. \tag{2.4}$$

By definition,

$$\hat{f}(x) = \mathbf{E}[f(x + \delta v)] = \frac{\int_{\delta\mathbb{B}} f(x + v)dv}{\mathrm{vol}_d(\delta\mathbb{B})}. \tag{2.5}$$

Similarly,

$$\mathbf{E}[f(x + \delta u)u] = \frac{\int_{\delta\mathbb{S}} f(x + u) \cdot \frac{u}{\|u\|}du}{\mathrm{vol}_{d-1}(\delta\mathbb{S})}. \tag{2.6}$$

Combining Eq.'s (2.4), (2.5), and (2.6), and the fact that ratio of volume to surface area of a d-dimensional ball of radius δ is δ/d gives the lemma.

Notice that the function $\hat{f}$ is differentiable even when f is not.

3 Expected Gradient Descent

For direct optimization of a single function, it is fairly well-known that using vectors that have expectation equal to the gradient suffices to reach the minimum of a function, in the limit (see, for example, Chapter 10 of [5]). Here we present an analysis of expected gradient descent in the online setting.

For lack of a better choice, we use the starting point $x_1 = 0$, the center of a containing ball of radius $R \leq D$. First we state Zinkevich's theorem, which we prove in the appendix for completeness.

THEOREM 3.1. *[25] Let $c_1, c_2, \ldots, c_n \colon S \to \mathbb{R}$ be an arbitrary sequence of convex, differentiable functions. Let $x_1, x_2, x_3, \ldots, x_n \in S$ be defined by $x_1 = 0$ and $x_{t+1} = \mathbf{P}_S(x_t - \eta \nabla c_t(x_t))$. Then, for $G = \max_t \|\nabla c_t(x_t)\|$,*

$$\sum_{t=1}^{n} c_t(x_t) - \min_{x \in S} \sum_{t=1}^{n} c_t(x) \leq \frac{R^2}{\eta} + n\frac{\eta G^2}{2}.$$

Plugging in $\eta = \frac{R}{G\sqrt{n}}$ gives a regret bound of $RG\sqrt{n}$. Now we are ready to give the randomized analysis. In this section, we show it only for any fixed sequence of functions. In Section 3.3, we prove it for an adaptive adversary that may choose c_{t+1} with knowledge of x_t and the previous points.

In the randomized version, each period t we get a random vector g_t with expectation equal to the gradient.

LEMMA 3.1. *Let $c_1, c_2, \ldots, c_n \colon S \to \mathbb{R}$ be a fixed sequence of convex, differentiable functions. Let $z_1, z_2, \ldots, z_n \in S$ be defined by $z_1 = 0$ and $z_{t+1} = \mathbf{P}_S(z_t - \eta g_t)$, where $\eta > 0$ and $g_1, \ldots, g_n$ are vector-valued random variables with (a) $\mathbf{E}[g_t \,|\, z_t] = \nabla c_t(z_t)$, (b) $\|g_t\| \leq G$, for some $G > 0$ (this also implies $\|\nabla c_t(x)\| \leq G$), and (c) $S \subseteq R\mathbb{B}$. Then, for $\eta = \frac{R}{G\sqrt{n}}$,*

$$\mathbf{E}\Big[\sum_{t=1}^{n} c_t(z_t)\Big] - \min_{x \in S} \sum_{t=1}^{n} c_t(x) \leq RG\sqrt{n}.$$

Proof. Let $x_\star$ be a point in S minimizing $\sum_{t=1}^{n} c_t(x)$. Define the functions $h_t : S \to \mathbb{R}$ by,

$$h_t(x) = c_t(x) + x \cdot \xi_t, \text{where } \xi_t = g_t - \nabla c_t(z_t).$$

The first thing to note is that,

$$\nabla h_t(z_t) = \nabla c_t(z_t) + \xi_t = g_t.$$

Therefore doing deterministic gradient descent on the random functions h_t is equivalent to doing expected gradient descent on the fixed functions c_t. Thus, by Theorem 3.1,

$$\begin{aligned} RG\sqrt{n} &\geq \sum_{t=1}^{n} h_t(z_t) - \min_{x \in S} \sum_{t=1}^{n} h_t(x) \\ &\geq \sum_{t=1}^{n} h_t(z_t) - \sum_{t=1}^{n} h_t(x_\star) \end{aligned} \tag{3.7}$$

Secondly, note that,

$$\mathbf{E}[h_t(z_t)] = \mathbf{E}[c_t(z_t)] + \mathbf{E}[\xi_t \cdot z_t] = \mathbf{E}[c_t(z_t)].$$

This follows from (a), because $\mathbf{E}[\xi_t | z_t] = 0$ for any z_t, thus $\mathbf{E}[\xi_t \cdot z_t | z_t] = 0$ and thus $\mathbf{E}[\xi_t \cdot z_t] = 0$. Similarly, for any fixed $x \in S$, $\mathbf{E}[h_t(x)] = c_t(x)$. In particular, $\mathbf{E}[h_t(x_\star)] = c_t(x_\star)$. Thus, taking expectations of Eq. (3.7),

$$RG\sqrt{n} \geq \sum_{t=1}^{n} \mathbf{E}[c_t(z_t)] - \sum_{t=1}^{n} c_t(x_\star)$$

Using $\sum c_t(x_\star) = \min_{x \in S} \sum c_t(x)$, and plugging in $\eta = R/G\sqrt{n}$ gives the lemma.

3.1 Algorithm and analysis.

In this section, we analyze the algorithm given in Figure 1.

BGD(α, δ, ν)

- $y_1 = 0$
- At each period t:
  - select unit vector u_t uniformly at random
  - $x_t := y_t + \delta u_t$
  - play x_t, and observe cost incurred $c_t(x_t)$
  - $y_{t+1} := \mathbf{P}_{(1-\alpha)S}(y_t - \nu c_t(x_t) u_t)$

Figure 1: Bandit gradient descent algorithm

We begin with a few observations.

OBSERVATION 3.1. *The optimum in $(1-\alpha)S$ is near the optimum in S,*

$$\min_{x \in (1-\alpha)S} \sum_{t=1}^{n} c_t(x) \leq 2\alpha C n + \min_{x \in S} \sum_{t=1}^{n} c_t(x).$$

Proof. For any $x \in S$, we have $(1-\alpha)x \in (1-\alpha)S$ and,

$$\begin{aligned} c_t\big((1-\alpha)x\big) &\leq (1-\alpha)c_t(x) + \alpha c_t(0) \\ &\leq c_t(x) + 2\alpha C \end{aligned}$$

We have used the fact that $|c_t(x)|, |c_t(0)| \leq C$. Summing over n periods,

$$\sum_{t=1}^{n} c_t\big((1-\alpha)x\big) \leq 2\alpha Cn + \sum_{t=1}^{n} c_t(x).$$

In particular, this holds for $x_\star = \arg\min_{x\in S} \sum c_t(x)$.

OBSERVATION 3.2. *For any x contained in $(1-\alpha)S$ the ball of radius αr centered at x is contained in S.*

Proof. Since $r\mathbb{B} \subseteq S$ and S is convex, we have

$$(1-\alpha)S + \alpha r\mathbb{B} \subseteq (1-\alpha)S + \alpha S = S.$$

The next observation establishes a bound on the maximum the function can change in $(1-\alpha)S$, an effective Lipschitz condition.

OBSERVATION 3.3. *For any x contained in $(1-\alpha)S$ and any y contained in S we have*

$$|c_t(x) - c_t(y)| \leq \frac{2C}{\alpha r}|x-y|.$$

Proof. Let $y = x + \Delta$. If $|\Delta| > \alpha r$, the observation follows from $|c_t| < C$. Otherwise, let $z = x + \alpha r \frac{\Delta}{|\Delta|}$, the point at distance αr from x in the direction Δ. By the previous observation, we know $z \in S$. Also, $y = \frac{|\Delta|}{\alpha r} z + \left(1 - \frac{|\Delta|}{\alpha r}\right) x$, so,

$$\begin{aligned} c_t(y) &\leq \frac{|\Delta|}{\alpha r} c_t(z) + \left(1 - \frac{|\Delta|}{\alpha r}\right) c_t(x) \\ &= c_t(x) + \frac{c_t(z) - c_t(x)}{\alpha r}|\Delta| \\ &\leq c_t(x) + \frac{2C}{\alpha r}|\Delta|. \end{aligned}$$

Now we are ready to select the parameters.

THEOREM 3.2. *For any $n \geq \left(\frac{3Rd}{2r}\right)^2$ and for $\nu = \frac{R}{C\sqrt{n}}$, $\delta = \sqrt[3]{\frac{rR^2d^2}{12n}}$, and $\alpha = \sqrt[3]{\frac{3Rd}{2r\sqrt{n}}}$, the expected regret of* BGD(ν, δ, α) *is upper bounded by*

$$\mathbf{E}\left[\sum_{t=1}^{n} c_t(x_t)\right] - \min_{x\in S} \sum_{t=1}^{n} c_t(x) \leq 3Cn^{5/6}\sqrt[3]{dR/r}.$$

Proof. We begin by showing that the points $x_t \in S$. Since $y_t \in (1-\alpha)S$, Observation 3.2 implies this fact as long as $\frac{\delta}{r} \leq \alpha < 1$, which is the case for $n \geq (3Rd/2r)^2$.

Now, we show that Lemma 3.1 applies to the feasible set $(1-\alpha)S$ with cost vectors $\hat{c}_t$ and points $y_1, \ldots, y_n \in (1-\alpha)S$. This bounds the performance of the y_t's against the costs $\hat{c}_t$ relative to the optimum in $(1-\alpha)S$. It is then sufficient to show that our actual cost $\sum c_t(x_t)$ is close to $\sum \hat{c}_t(y_t)$, and the actual minimum (of $\sum c_t$ in S) is close to the minimum of $\sum \hat{c}_t$ in $(1-\alpha)S$.

Suppose we wanted to run the gradient descent algorithm on the functions $\hat{c}_t$ defined by (2.3), and the set $(1-\alpha)S$. If we let

$$g_t = \frac{d}{\delta} c_t(y_t + \delta u_t) u_t$$

then (since u_t is selected uniformly at random from $\mathbb{S}$) Lemma 2.1 says $\mathbf{E}[g_t \,|\, y_t] = \nabla \hat{c}_t(y_t)$. So Lemma 3.1 applies with the update rule:

$$\begin{aligned} y_{t+1} &= \mathbf{P}_{(1-\alpha)S}(y_t - \eta g_t) \\ &= \mathbf{P}_{(1-\alpha)S}(y_t - \eta \frac{d}{\delta} c_t(y_t + \delta u_t) u_t), \end{aligned}$$

which is exactly the update rule we are using to obtain y_t (called z_t in Lemma 2.1) in BGD, with $\eta = \nu\delta/d$. Since

$$\|g_t\| = \left\| \frac{d}{\delta} c_t(y_t + \delta u_t) u_t \right\| \leq dC/\delta,$$

we can apply Lemma 3.1 with $G = dC/\delta$. By our choice of ν, we have $\eta = R/G\sqrt{n}$, and so the expected regret is upper bounded by

$$\mathbf{E}\left[\sum_{t=1}^{n} \hat{c}_t(y_t)\right] - \min_{x\in(1-\alpha)S} \sum_{t=1}^{n} \hat{c}_t(x) \leq \frac{RdC\sqrt{n}}{\delta}.$$

Let $L = \frac{2C}{\alpha r}$, which will act as an "effective Lipschitz constant". Notice that for $x \in (1-\alpha)S$, since $\hat{c}_t$ is an average over inputs within δ of x, Observation 3.3 shows that $|\hat{c}_t(x) - c_t(x)| \leq \delta L$. Since $\|y_t - x_t\| = \delta$, Observation 3.3 also shows that

$$|\hat{c}_t(y_t) - c_t(x_t)| \leq |\hat{c}_t(y_t) - c_t(y_t)| + |c_t(y_t) - c_t(x_t)| \leq 2\delta L.$$

These with the above imply,

$$\begin{aligned} \mathbf{E}\left[\sum_{t=1}^{n} \big(c_t(x_t) - 2\delta L\big)\right] - \min_{x\in(1-\alpha)S} \sum_{t=1}^{n} \big(c_t(x) + \delta L\big) \\ \leq \frac{RdC\sqrt{n}}{\delta}, \end{aligned}$$

so rearranging terms and using Observation 3.1 gives

$$\begin{aligned} (3.8)\quad \mathbf{E}\left[\sum_{t=1}^{n} c_t(x_t)\right] - \min_{x\in S} \sum_{t=1}^{n} c_t(x) \\ \leq \frac{RdC\sqrt{n}}{\delta} + 3\delta Ln + 2\alpha Cn. \end{aligned}$$

Plugging in $L = \frac{2C}{\alpha r}$ gives,

$$\mathbf{E}\left[\sum_{t=1}^{n} c_t(x_t)\right] - \min_{x\in S}\sum_{t=1}^{n} c_t(x) \leq \frac{RdC\sqrt{n}}{\delta} + \frac{\delta}{\alpha}\frac{6Cn}{r} + \alpha 2Cn.$$

This expression is of the form $\frac{a}{\delta} + b\frac{\delta}{\alpha} + c\alpha$. Setting $\delta = \sqrt[3]{\frac{a^2}{bc}}$ and $\alpha = \sqrt[3]{\frac{ab}{c^2}}$ gives a value of $3\sqrt[3]{abc}$. The lemma is achieved for $a = RdC\sqrt{n}$, $b = 6Cn/r$ and $c = 2Cn$.

THEOREM 3.3. *If each c_t is L-Lipschitz, then for n su ciently large and $\nu = \frac{R}{C\sqrt{n}}$, $\alpha = \frac{\delta}{r}$, and $\delta = n^{-.25}\sqrt{\frac{RdCr}{3(Lr+C)}}$,*

$$\mathbf{E}\left[\sum_{t=1}^{n} c_t(x_t)\right] - \min_{x\in S}\sum_{t=1}^{n} c_t(x) \leq 2n^{3/4}\sqrt{3RdC(L + C/r)}.$$

Proof. The proof is quite similar to the proof of Theorem 3.2. Again we check that the points $x_t \in S$, which it is for n is sufficiently large. We now have a direct Lipschitz constant, so we can use it directly in Eq. (3.8). Plugging this in with chosen values of α and δ gives the lemma.

3.2 Reshaping. The above regret bound depends on R/r, which can be very large. To remove this dependence (or at least the dependence on $1/r$), we can reshape the body to make it more "round."

The set S, with $r\mathbb{B} \subseteq S \subseteq R\mathbb{B}$ can be put in *isotropic position* [21]. Essentially, this amounts to estimating the covariance of random samples from the body and applying an affine transformation T so that the new covariance matrix is the identity matrix.

A body $T(S) \subseteq \mathbb{R}^d$ in isotropic position has several nice properties, including $\mathbb{B} \subseteq T(S) \subseteq d\mathbb{B}$. So, we first apply the preprocessing step to find T which puts the body in isotropic position. This gives us a new $R' = d$ and $r' = 1$. The following observation shows that we can use $L' = LR$.

OBSERVATION 3.4. *Let $c'_t(u) = c_t(T^{-1}(u))$. Then c'_t is LR-Lipschitz.*

Proof. Let $x_1, x_2 \in S$ and $u_1 = T(x_1)$, $u_2 = T(x_2)$. Observe that,

$$|c'_t(u_1) - c'_t(u_2)| = |c_t(x_1) - c_t(x_2)| \leq L\|x_1 - x_2\|.$$

To make this a LR-Lipschitz condition on c'_t, it suffices to show that $\|x_1 - x_2\| \leq R\|u_1 - u_2\|$. Suppose not, i.e. $\|x_1 - x_2\| > R\|u_1 - u_2\|$. Define $v_1 = \frac{u_1-u_2}{\|u_1-u_2\|}$ and $v_2 = -v_1$. Observe that $\|v_2 - v_1\| = 2$, and since $T(S)$ contains the ball of radius 1, $v_1, v_2 \in T(S)$. Thus, $y_1 = T^{-1}(v_1)$ and $y_2 = T^{-1}(v_2)$ are in S. Then, since T is affine,

$$\begin{aligned}\|y_1 - y_2\| &= \frac{1}{\|u_1 - u_2\|}\|T^{-1}(u_1 - u_2) - T^{-1}(u_2 - u_1)\| \\ &= \frac{2}{\|u_1 - u_2\|}\|T^{-1}(u_1) - T^{-1}(u_2)\| \\ &= \frac{2}{\|u_1 - u_2\|}\|x_1 - x_2\| > 2R,\end{aligned}$$

where the last line uses the assumption $\|x_1 - x_2\| > R\|u_1 - u_2\|$. The inequality $\|y_1 - y_2\| > 2R$ contradicts the assumption that S is contained in a sphere of radius R.

Many common shapes such as balls, cubes, etc., are already nicely shaped, but there exist MCMC algorithms for putting any body into isotropic position from a membership oracle [17, 18]. (Note that the projection oracle we assume is a stronger oracle than a membership oracle.) The latest (and greatest) algorithm for putting a body into isotropic position, due to Lovasz and Vempala [18], runs in time $O(d^4)$poly-log$(d, \frac{R}{r})$. This algorithm puts the body into nearly isotropic position, which means that $\mathbb{B} \subseteq T(S) \subseteq 1.01d\mathbb{B}$. After such preprocessing we would have $r' = 1, R' = 1.01d, L' = LR$, and $C' = C$. This gives,

COROLLARY 3.1. *For a set S of diameter D, and c_t L-Lipschitz, after putting S into near- isotropic position, the BGD algorithm has expected regret,*

$$\mathbf{E}\left[\sum_{t=1}^{n} c_t(x_t)\right] - \min_{x\in S}\sum_{t=1}^{n} c_t(x) \leq 6n^{3/4}d\left(\sqrt{CLR} + C\right).$$

Without the L-Lipschitz condition,

$$\mathbf{E}\left[\sum_{t=1}^{n} c_t(x_t)\right] - \min_{x\in S}\sum_{t=1}^{n} c_t(x) \leq 6n^{5/6}dC$$

Proof. Using $r' = 1, R' = 1.01d, L' = LR$, and $C' = C$, In the first case, we get an expected regret of at most $2n^{3/4}\sqrt{6(1.01d)dC(LR + C)}$. In the second case, we get an expected regret of at most $3Cn^{5/6}\sqrt{2(1.01d)d}$.

3.3 Adaptive adversary. Until now, we have analyzed the algorithm in the case that there is a fixed but unknown sequence of functions $c_1, c_2, \ldots$. In other words, an *oblivious adversary* must fix the entire sequence of functions in advance, with knowledge of our

algorithm but without knowledge of any of its random decisions.

However, in many situations the function c_t may actually depend on the sequence of previous observed choices of the algorithm, $x_1, x_2, \ldots, x_{t-1}$. That is, we can think of it as a game between the decision maker and the adversary. Each period, the decision maker chooses an x_t with knowledge of only x_1, $c_1(x_1)$, x_2, $c_2(x_2)$, ..., x_{t-1}, $c_{t-1}(x_{t-1})$, while the adversary chooses c_t with knowledge of x_1, c_1, x_2, c_2, ..., x_{t-1}, c_{t-1}.

In this section, we sketch why the theorems we have shown all hold against an adaptive adversary, up to changes in the multiplicative constants of at most a factor of 3. Now each function c_t is itself a random variable which may depend on our choices of $x_1, x_2, \ldots, x_{t-1}$. The bounds relating the costs $c_t(x_t)$, $c_t(y_t)$, and $\hat{c}_t(y_t)$ were all worst-case bounds, i.e., they hold for arbitrary c_t, regardless of whether the c_t are adaptively chosen or not. Thus it suffices to bound

$$\mathbf{E}\left[\sum \hat{c}_t(y_t) - \min_{y \in s} \sum \hat{c}_t(y)\right].$$

So we may as well pretend that the adversary's only goal is to maximize the above regret. In this case, from the adversary's knowledge of our sequence $x_1, x_2, \ldots, x_{t-1}$ and our algorithm, it can deduce our values $y_1, y_2, \ldots, y_t$. Thus, it remains to show that expected gradient descent works in the adversarial setting. (In other words, we will show that the adversary's extra knowledge of the sequence of x_t's cannot help for the purpose of maximizing the above regret.)

To do this, we generalize Lemma 3.1 to the adaptive setting. By analogy to our use of Lemma 3.1 in the proof of Theorem 3.2, the result for adaptive adversaries follows from applying Lemma 3.2 to the $\hat{c}_t$'s and points y_t in feasible set $(1-\alpha)S$ from BGD.

LEMMA 3.2. *Let $c_1, c_2, \ldots, c_n : S \to \mathbb{R}$ be a sequence of convex, differentiable functions (c_{t+1} possibly depending on $z_1, z_2, \ldots, z_t$) and $z_1, z_2, \ldots, z_n \in S$ be defined by $z_1 = 0$ and $z_{t+1} = \mathbf{P}_S(z_t - \eta g_t)$, where $\eta > 0$ and $g_1, \ldots, g_n$ are vector-valued random variables with (a) $\mathbf{E}[g_t \mid z_1, c_1, z_2, c_2, \ldots, z_t, c_t] = \nabla c_t(z_t)$, (b) $\|g_t\| \leq G$, for some $G > 0$ (this also implies $\|\nabla c_t(x)\| \leq G$), and (c) $S \subseteq R\mathbb{B}$. Then, for $\eta = \frac{R}{G\sqrt{n}}$,*

$$\mathbf{E}\left[\sum_{t=1}^{n} c_t(z_t) - \min_{x \in S} \sum_{t=1}^{n} c_t(x)\right] \leq 3RG\sqrt{n}.$$

Proof. Again, we define the functions $h_t : S \to \mathbb{R}$ by,

$$h_t(x) = c_t(x) + x \cdot \xi_t, \text{where } \xi_t = g_t - \nabla c_t(z_t).$$

We no longer use $x_\star = \arg\min_{x \in S} \sum c_t(x)$, because $x_\star$ is not fixed. Instead, the properties we will use are that, $\|\xi_t\| \leq 2G$ (following from $\|g_t\|, \|\nabla c_t(z_t)\| \leq G$), that, as before, $\mathbf{E}[\xi_t | z_t] = 0$, and that $\mathbf{E}[\xi_t \cdot \xi_s] = 0$ for $1 \leq s < t \leq n$. The latter two follow from condition (a) of Lemma 3.2.

Following the proof of Lemma 3.1, we note that $\nabla h_t(z_t) = \nabla c_t(z_t) + \xi_t = g_t$. So, in hindsight, it is exactly as if we have done deterministic gradient descent on the sequence of h_t's. Thus, by Theorem 3.1,

$$RG\sqrt{n} \geq \sum_{t=1}^{n} h_t(z_t) - \min_{x \in S} \sum_{t=1}^{n} h_t(x) \tag{3.9}$$

We still have that $\mathbf{E}[h_t(z_t)] = \mathbf{E}[c_t(z_t)] + \mathbf{E}[\xi_t \cdot z_t] = \mathbf{E}[c_t(z_t)]$. So, after taking expectations of Eq. (3.9), it suffices to show that,

$$\mathbf{E}\left[\min_{x \in S} \sum_{t=1}^{n} h_t(x)\right] \leq \mathbf{E}\left[\min_{x \in S} \sum_{t=1}^{n} c_t(x)\right] + 2RG\sqrt{n}$$

To see this, observe that,

$$\begin{aligned}\left|\sum_{t=1}^{n} \big(h_t(x) - c_t(x)\big)\right| &= \left|x \cdot \sum \xi_t\right| \\ &\leq \|x\| \left\|\sum \xi_t\right\| \\ &\leq R \left\|\sum \xi_t\right\|\end{aligned}$$

The above is an absolute statement for any realization of h_t's and c_t's and any $x \in S$. Therefore, it is a bound on the difference between the minima of $\sum h_t(x)$ and $\sum c_t(x)$. It now suffices to show that $E[\|\sum \xi_t\|] \leq 2G\sqrt{n}$, because then the expected difference in minima between $\sum h_t(x)$ and $\sum c_t(x)$ would be at most $2RG\sqrt{n}$, as required.

$$\begin{aligned}\left(\mathbf{E}\left[\left\|\sum \xi_t\right\|\right]\right)^2 &\leq \mathbf{E}\left[\left\|\sum \xi_t\right\|^2\right] \\ &= \sum_{1 \leq t \leq n} \mathbf{E}[\|\xi_t\|^2] + 2 \sum_{1 \leq s < t \leq n} \mathbf{E}[\xi_s \cdot \xi_t] \\ &= \sum_{1 \leq t \leq n} \mathbf{E}[\|\xi_t\|^2] \\ &\leq 4nG^2\end{aligned}$$

In the above we have used $\mathbf{E}[\xi_s \cdot \xi_t] = 0$ and that $\|\xi_t\| \leq 2G$ as mentioned.

4 Conclusions

We have given algorithms for bandit online optimization of convex functions. Our approach is to extend Zinkevich's gradient descent analysis to a situation where we do not have access to the gradient. We give a simple trick for approximating the gradient of a function by a

single sample, and we give a simple understanding of this approximation as being the gradient of a smoothed function. This is similar to an approximation proposed in [22]. The simplicity of our approximation make it straightforward to analyze this algorithm in an online setting, with few assumptions.

Interestingly, the earlier bandit analysis of online linear optimization of McMahan and Blum [19] similarly uses single samples whose expectation correctly reconstruct the linear function.

It is also worth noting that the online shortest paths problem [24] can be convexified, where one chooses ows rather than paths. Given a ow, it is possible to choose a random path whose expected time is equal to the time of the ow. Using this approach, one can apply our algorithm to the bandit online shortest paths problem [1, 19] to get guarantees against an adaptive adversary.

Zinkevich presents a few nice variations on the model and algorithms. He shows that an adaptive step size $\eta_t = O(1/\sqrt{t})$ can be used with similar guarantees. It is likely that a similar adaptive step size could be used here.

He also proves that gradient descent can be compared, to an extent, with a non-stationary adversary. He shows that relative to any sequence $z_1, z_2, \ldots, z_n$, it achieves,

$$\sum_{t=1}^{n} c_t(x_t) - \min_{z_1,z_2,\ldots,z_n \in S} \sum c_t(z_t) \leq O\left(GD\sqrt{n(1+\sum \|z_t - z_{t-1}\|)}\right).$$

Thus, compared to an adversary that moves a total distance $o(n)$, he has regret $o(n)$. These types of guarantees may be extended to the bandit setting.

The algorithm has potentially wide application as it can be applied to minimizing any function(s) over a convex set. If the range of function values were unknown, or for other practical reasons, it would make sense to use the update $y_{t+1} := y_t - \nu(c_t(x_t) - c_{t-1}(x_{t-1}))u_t$. This has the same expected value as the update we suggested, but its magnitude may be smaller.

Acknowledgements. We would like to thank David McAllester and Rakesh Vohra for helpful discussions and especially Martin Zinkevich for advice on the adaptive proof.

References

[1] B. Awerbuch and R. Kleinberg. Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches. In *Proceedings of the 36th ACM Symposium on Theory of Computing*, 2004.

[2] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. Gambling in a rigged casino: the adversarial multi-armed bandit problem. In *Proc. of the 36th Annual Symposium on Foundations of Computer Science*, pp. 322–331, 1995.

[3] Dimitri P. Bertsekas and John N. Tsitsiklis. *Neuro-Dynamic Programming.* Athena Scientific, Belmont, MA, 1996.

[4] D. Bertsimas and S. Vempala. Solving convex programs by random walks. In *Proc. of the 34th ACM Symposium on the Theory of Computing*, 2002.

[5] J. Birge and F. Loveaux. *Introduction to Stochastic Programming.* Springer Verlag, New York, 1997.

[6] Thomas Cover. Universal Portfolios. In *Mathematical Finance* **1**, 1-29, 1991.

[7] A. Frieze and R. Kannan. Log-Sobolev inequalities and sampling from log-concave distributions. *Annals of Applied Probability*, 9 (1999), 14–26.

[8] O. N. Granichin. Stochastic approximation with input perturbation for dependent observation disturbances. *Vestn. Leningrad. Gos Univ.*, 1989, Ser. 1, no. 4, 27–31.

[9] O. N. Granichin. Randomized algorithms for stochastic approximation under arbitrary disturbances. *Automation and Remote Control*, 63: 2 (2002) 209–219.

[10] David P. Helmbold, Robert E. Schapire, Yoram Singer, and Manfred K. Warmuth. On-line portfolio selection using multiplicative updates. In *Mathematical Finance*, 8(4):325-347, 1998.

[11] L. Khachiyan. A polynomial algorithm in linear programming (in Russian). *Doklady Akedamii Nauk SSSR*, 244, 1093-1096, 1979.

[12] R. Kleinberg. Nearly Tight Bounds for the Continuum-Armed Bandit Problem. To appear in *Proc. 18th Annual Conf. on Neural Information Processing Systems*, 2004.

[13] S. Kirkpatrick, C. D. Gelatt Jr., M. P. Vecchi. Optimization by Simulated Annealing. *Science*, 220, 4598, 671-680, 1983.

[14] A. Kalai and S. Vempala. Efficient algorithms for universal portfolios. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, 2000.

[15] A. Kalai and S. Vempala. Efficient algorithms for the online decision problem. In *Proceedings of the 16th Conference on Computational Learning Theory,* 2003.

[16] Jyrki Kivinen and Manfred K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1-63, 10 January 1997.

[17] R. Kannan and L. Lovasz and M. Simonovits. Random walks and an $O^*(n^5)$ volume algorithm for convex bodies. *Random Struct. Algorithms*, 11(1), pp. 1–50, 1997.

[18] L. Lovasz and S. Vempala. Simulated Annealing in Convex Bodies and an $0^*(n^4)$ Volume Algorithm. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, p. 650, 2003.

[19] H. Brendan McMahan and Avrim Blum. Online Geo-

metric Optimization in the Bandit Setting Against an Adaptive Adversary. In *Proceedings of the 17th Annual Conference on Learning Theory*, COLT 2004.

[20] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller. Equation of State Calculations by Fast Computing Machines. *J. Chem. Phys.*, 21, 6, 1087-1092, 1953.

[21] V. D. Milman and A. Pajor. Isotropic position and inertia ellipsoids and zonoids of the unit ball of a normed n-dimensional space. *Lecture Notes in Mathematics* 1376, Springer, Berlin (1989), 64-104.

[22] J. C. Spall. A One-Measurement Form of Simultaneous Perturbation Stochastic Approximation. *Automatica*, 33, pp. 332-341.

[23] J. C. Spall. *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control.* John Wiley and Sons, Hoboken, NJ, 2003.

[24] E. Takimoto and M. Warmuth. Path Kernels and Multiplicative Updates. In *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory*, pp. 74-89, 2002.

[25] M. Zinkevich. Online Convex Programming and Generalized Infinitesimal Gradient Ascent. In *Proceedings of the Twentieth International Conference on Machine Learning*, pp. 928-936, 2003.

A Proof of Zinkevich's theorem

For completeness, we give a proof of Theorem 3.1.

Proof. Let $x_\star$ be a point in S minimizing $\sum_{t=1}^{n} c_t(x)$.

Following Zinkevich's analysis, we use $\|x_t - x_\star\|^2/2\eta$ as a potential function. We will show,

(1.10)
$$c_t(x_t) - c_t(x_\star) \leq \frac{\|x_t - x_\star\|^2 - \|x_{t+1} - x_\star\|^2}{2\eta} + \frac{\eta^2 G^2}{2}.$$

Summing over t periods telescopes to give the theorem,

$$\begin{aligned}\sum_{t=1}^{n} c_t(x_t) - c_t(x_\star) &\leq \frac{\|x_1 - x_\star\|^2}{2\eta} + n\frac{\eta^2 G^2}{2} \\ &\leq \frac{R^2}{2\eta} + n\frac{\eta^2 G^2}{2}\end{aligned}$$

The last step follows because we chose $x_1 = 0$ and $S \subseteq R\mathbb{B}$.

Since c_t is convex and differentiable, we can bound the difference between $c_t(x_t)$ and $c_t(x_\star)$ in terms of the gradient. For the rest of this proof, define $g_t = \nabla c_t(x_t)$.

(1.11)
$$c_t(x_t) - c_t(x_\star) \leq \nabla g_t \cdot (x_t - x_\star)$$

Since S is convex, we have $\|\mathbf{P}_S(x) - x_\star\| \leq \|x - x_\star\|$ for any $x \in \mathbb{R}^d$. So,

$$\begin{aligned}\|x_{t+1} - x_\star\|^2 &= \|\mathbf{P}_S(x_t - \eta g_t) - x_\star\|^2 \\ &\leq \|x_t - \eta g_t - x_\star\|^2 \\ &= \|x_t - x_\star\|^2 + \eta^2\|g_t\|^2 - 2\eta(x_t - x_\star)\cdot g_t \\ &\leq \|x_t - x_\star\|^2 + \eta^2 G^2 - 2\eta(x_t - x_\star)\cdot g_t.\end{aligned}$$

After rearranging terms, we have

$$g_t \cdot (x_t - x_\star) \leq \frac{\|x_t - x_\star\|^2 - \|x_{t+1} - x_\star\|^2 + \eta^2 G^2}{2\eta}.$$

Combining this with Eq. (1.11) gives Eq. (1.10) as required.