Optimistic Bandit Convex Optimization

Authored by:

Mehryar Mohri
Scott Yang

Abstract

We introduce the general and powerful scheme of predicting information re-use in optimization algorithms. This allows us to devise a computationally efficient algorithm for bandit convex optimization with new state-of-the-art guarantees for both Lipschitz loss functions and loss functions with Lipschitz gradients. This is the first algorithm admitting both a polynomial time complexity and a regret that is polynomial in the dimension of the action space that improves upon the original regret bound for Lipschitz loss functions, achieving a regret of $widetilde O(T{11/16}d{3/8})$. Our algorithm further improves upon the best existing polynomial-in-dimension bound (both computationally and in terms of regret) for loss functions with Lipschitz gradients, achieving a regret of $widetilde O(T{8/13} d{5/3})$.

1 Paper Body

Bandit convex optimization (BCO) is a key framework for modeling learning problems with sequential data under partial feedback. In the BCO scenario, at each round, the learner selects a point (or action) in a bounded convex set and observes the value at that point of a convex loss function determined by an adversary. The feedback received is limited to that information: no gradient or any other higher order information about the function is provided to the learner. The learner?s objective is to minimize his regret, that is the difference between his cumulative loss over a finite number of rounds and that of the loss of the best fixed action in hindsight. The limited feedback makes the BCO setup relevant to a number of applications, including online advertising. On the other hand, it also makes the problem notoriously difficult and requires the learner to find a careful trade-off between exploration and exploitation. While it has been the subject of extensive study in recent years, the fundamental BCO problem remains one of the most challenging scenarios in machine learning where several questions concerning optimality guarantees remain open. e 5/6) is achievable for bounded The original work of Flaxman et al. [2005] showed that a regret of O(T e 3/4) for Lipschitz loss functions (the latter bound is also given in

[Kleinberg, loss functions and of O(T 2004]), both of which are still the best known results given by explicit algorithms. Agarwal et al. e 2/3) for loss functions that are both [2010] introduced an algorithm that maintains a regret of O(T Lipschitz and strongly convex, which is also still state-of-the-art. For functions that are Lipschitz and also admit Lipschitz gradients, Saha and Tewari [2011] designed an algorithm with a regret of e 2/3) regret, a result that was recently improved to O(T e 5/8) by Dekel et al. [2015]. O(T

Here, we further improve upon these bounds both in the Lipschitz and Lipschitz gradient settings. By incorporating the novel and powerful idea of predicting information re-use, we introduce an algorithm e T 11/16 for Lipschitz loss functions. Similarly, our algorithm also achieves with a regret bound of O the best regret guarantee among computationally tractable algorithms for loss functions with Lipschitz 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

e T 8/13 . Both bounds admit a relatively mild dependency on the dimension of the action gradients: O space.

We note that the recent remarkable work by [Bubeck et al., 2015, Bubeck and Eldan, 2015] has e 1/2), which matches the known proven the existence of algorithms that can attain a regret of O(T 1/2 lower bound ?(T) given by Dani et al.. Thus, the dependency of our bounds with respect to T is not optimal. Furthermore, two recent unpublished manuscripts, [Hazan and Li, 2016] and [Bubeck e 1/2). These results, once verified, would be et al., 2016], present algorithms achieving regret O(T ground-breaking contributions to the literature. However, unlike our algorithms, the regret bound for both of these algorithms admits a large dependency on the dimension d of the action space: exponential for [Hazan and Li, 2016], dO(9.5) for [Bubeck et al., 2016]. One hope is that the novel ideas introduced by Hazan and Li [2016] (the application of the ellipsoid method with a restart button and lower convex envelopes) or those by Bubeck et al. [2016] (which also make use of the restart idea but introduces a very original kernel method) could be combined with those presented in this paper to derive algorithms with the most favorable guarantees with respect to both T and d. We begin by formally introducing our notation and setup. We then highlight some of the essential ideas in previous work before introducing our new key insight. Next, we give a detailed description of our algorithm for which we prove theoretical guarantees in several settings.

2 2.1

Preliminaries BCO scenario

The scenario of bandit convex optimization, which dates back to [Flaxman et al., 2005], is a sequential prediction problem on a convex compact domain K ? Rd . At each round t 2 [1, T], the learner selects a (possibly) randomized action xt 2 K and incurs the loss ft (xt) based on a convex function ft : K ! R chosen by the adversary. We assume that the adversary is oblivious, so that the loss functions are independent of the player?s actions. The objective of the learner is to minimize his regret with respect to the optimal static action in hindsight, that is, if we denote by A the learner?s randomized algorithm, the following quantity: " T # T X X RegT (A) = E ft (xt) min ft (x). (1) x2K

t=1
t=1

We will denote by D the diameter of the action space K in the Euclidean norm: D = supx,y2K kx yk2 . Throughout this paper, we will often use different induced norms. We will denote by k ? kA the normpinduced by a symmetric positive definite (SPD) matrix A 0, defined for all x 2 Rd by kxkA = x¿ Ax. Moreover, we will denote by k ? kA,? its dual norm, given by k ? kA 1 . To simplify the notation, we will write k ? kx instead of k ? kr2 R(x) , when the convex and twice differentiable function R : int(K) ! R is clear from the context. Here, int(K) is the set interior of K.

We will consider different levels of regularity for the functions ft selected by the adversary. We will always assume that they are uniformly bounded by some constant C ¿ 0, that is —ft (x)— ? C for all t 2 [1, T] and x 2 K, and, by shifting the loss functions upwards by at most C, we will also assume, without loss of generality, that they are non-negative: ft 0, for all t 2 [1, T]. Moreover, we will always assume that ft is Lipschitz on K (henceforth denoted C0,1 (K)): 8t 2 [1, T], 8x, y 2 K, —ft (x) ft (y)— ? Lkx yk2 . In some instances, we will further assume that the functions admit H-Lipschitz gradients on the interior of the domain (henceforth denoted C1,1 (int(K))): 9H ¿ 0 : 8t 2 [1, T], 8x, y 2 int(K), krft (x) rft (y)k2 ? Hkx yk2 . Since ft is convex, it admits a subgradient at any point in K. We denote by gt one element of the subgradient at the point xt 2 K selected by the learner at round t. When the losses are C1,1 , the only element of the subgradient is the gradient, and gt = rft (xt). We will use the shorthand Pt v1:t = s=1 vs to denote the sum of t vectors v1 , . . . , vt . In particular, g1:t will denote the sum of the subgradients gs for s 2 [1, t]. Lastly, we will denote by B1 (0) = x 2 Rd : kxk2 ? 1 ? Rd the d-dimensional Euclidean ball of radius one and by @B1 (0) the unit sphere. 2

2.2

Follow-the-regularized-leader template

A standard algorithm in online learning, both for the bandit and full-information setting is the follow-the-regularized-leader (FTRL) algorithm. At each round, the algorithm selects the action that minimizes the cumulative linearized loss augmented with a regularization term R : K ! R. Thus, the action xt+1 is defined as follows: ¿ xt+1 = argmin ?g1:t x + R(x), x2K

where ? ¿ 0 is a learning rate that determines the tradeoff between greedy optimization and regularization. If we had access to the subgradients at each round, then, FTRL with R(x) = kxk22 and ? = p1T p would yield a regret of O(dT), which is known to be optimal. But, since we only have access to the loss function values ft (xt) and since the loss functions change at each round, a more refined strategy is needed. 2.2.1

One-point gradient estimates and surrogate losses

One key insight into the bandit convex optimization problem, due to Flaxman et al. [2005], is that the subgradient of a smoothed version of the loss function can be estimated by sampling and rescaling around the point the algorithm originally intended to play. Lemma 1 ([Flaxman et al., 2005, Saha and Tewari, 2011]). Let f : K ! R be an arbitrary function (not necessarily differen-

tiable) and let U (@B1 (0)) denote the uniform distribution over the unit sphere. Then, for any ¿ 0 and any SPD matrix A 0, the function fb defined for all x 2 K b by f (x) = Eu?U (@B1 (0)) [f (x + Au)] is differentiable over int(K) and, for any x 2 int(K), gb = d f (x + Au)A 1 u is an unbiased estimate of rfb(x): ? d E f (x + Au)A 1 u = rfb(x). u?U (@B1 (0))

The result shows that if at each round t we sample ut ? U (@B1 (0)), define an SPD matrix At and play the point yt = xt + At u (assuming that yt 2 K), then gbt = d f (xt + At ut)At 1 ut is an unbiased estimate of the gradient of fb at the point xt originally intended: E[b gt] = rfb(xt). Thus, we ¿ can use FTRL with these smoothed gradient estimates: xt+1 = argminx2K ? gb1:t x + R(x), at the b cost of the approximation error from ft to ft . Furthermore, the norm of these estimate gradients can be bounded. Lemma 2. Let ¿ 0, ut 2 @B1 (0) and At 0, then the norm of gbt = d f (xt + At ut)At 1 ut can 2 be bounded as follows: kb gt k2A2 ? d2 C 2 . t

Proof. Since ft is bounded by C, we can write kb gt k2A2 ? t

d2 2

C 2 ut At 1 A2t At 1 ut ?

d2 2

C 2.

This gives us a bound on the Lipschitz constant of fbt in terms of d, , and C. 2.2.2

Self-concordant barrier as regularization

When sampling to derive a gradient estimate, we need to ensure that the point sampled lies within the feasible set K. A second key idea in the BCO problem, due to Abernethy et al. [2008], is to design ellipsoids that are always contained in the feasible sets. This is done by using tools from the theory of interior-point methods in convex optimization. Definition 1 (Definition 2.3.1 [Nesterov and Nemirovskii, 1994]). Let K ? Rd be closed convex, and let ? 0. A C 3 function R : int(K) ! R is a ?-self-concordant barrier for K if for any sequence d (zs)1 s=1 with zs ! @K, we have R(zs) ! 1, and if for all x 2 int(K), and y 2 R , the following inequalities hold: —r3 R(x)[y, y, y]— ? 2kyk3x , 3

—rR(x)¿ y— ? ? 1/2 kykx .

Since self-concordant barriers are preserved under translation, we will always assume for convenience that minx2K R(x) = 0. Nesterov and Nemirovskii [1994] show that any d-dimensional closed convex set admits an O(d)self-concordant barrier. This allows us to always choose a self-concordant barrier as regularization. We will use several other key properties of self-concordant barriers in this work, all of which are stated precisely in Appendix 7.1.

3

Previous work

The original paper by Flaxman et al. [2005] sampled indiscriminately around spheres and projected e T 3/4 for C0,1 loss functions. back onto the feasible set at each round. This yielded a regret of O 1,1 The follow-up work of Saha and Tewari [2011] showed that for C loss functions, one can run FTRL with a self-concordant barrier as regularization and sample around the Dikin ellipsoid to attain an e T 2/3 . improved regret bound of O

More recently, Dekel et al. [2015] showed that by averaging the smoothed gradient estimates e T 5/8 . and still using the self-concordant barrier as regularization, one can achieve a regret of O P k 1 Specifically, denote by g?t = k+1 bt i the average of the past k + 1 incurred gradients, where i=0 g gbt i = 0 for t i ? 0. Then we can play FTRL on these averaged smoothed gradient estimates: xt+1 = argmin2K ?? gt¿ x + R(x), to attain the better guarantee.

Abernethy and Rakhlin [2009] derive a generic estimate for FTRL algorithms with self-concordant barriers as regularization: Lemma 3 ([Abernethy and Rakhlin, 2009]-Theorem 2.2-2.3). Let K be a closed convex set in Rd and let R be a ?-self-concordant barrier for K. Let {gt }Tt=1 ? Rd and ? ¿ 0 be such that ¿ ?kgt kxt ,? ? 1/4 for all t 2 [1, T]. Then, the FTRL update xt+1 = argminx2K g1:t x + R(x) admits the following guarantees: kxt

xt+1 kxt ? 2?kgt kxt ,? ,

8x 2 K,

T X t=1

gt¿ (xt

x) ? 2?

T X

1 kgt k2xt ,? + R(x). ? t=1

By Lemma 2, if we use FTRL with smoothed gradients, then the upper bound in this lemma can be further bounded by T X 1 C 2 d2 1 2? kb gt k2xt ,? + R(x) ? 2?T 2 + R(x). ? ? t=1

Furthermore, the regret is then bounded by the sum of this upper bound and the cost of approximating ft with fbt . On the other hand, Dekel et al. [2015] showed that if we used FTRL with averaged smoothed gradients instead, then the upper bound in this lemma can be bounded as ? ? T X 1 32C 2 d2 1 2 2 2 2? k? gt kxt ,? + R(x) ? 2?T + 2D L + R(x). 2 (k + 1) ? ? t=1

The extra factor (k + 1) in the denominator, at the cost of now approximating ft with f?t , is what contributes to their improved regret result.

In general, finding surrogate losses that can both be approximated accurately and admit only a mild variance is a delicate task, and it is not clear how the constructions presented above can be improved.

4 4.1

Algorithm Predicting the predictable

Rather than designing a newer and better surrogate loss, our strategy will be to exploit the structure of the current state-of-the-art method. Specifically, we draw upon the technique of predictable sequences from [Rakhlin and Sridharan, 2013]. The idea here is to allow the learner to preemptively ?guess? the 4

gradient at the next step and optimize for this in the FTRL update. Specifically, if get+1 is an estimate of the time t + 1 gradient gt+1 based on information up to time t, then the learner should play: xt+1 = argmin(g1:t + get+1)¿ x + R(x). x2K

This optimistic FTRL algorithm admits the following guarantee: Lemma 4 (Lemma 1 [Rakhlin and Sridharan, 2013]). Let K be a closed convex set in Rd , and let R be a ?-self-concordant barrier for K. Let {gt }Tt=1 ? Rd and ? ¿ 0 such that ?kgt get kxt ,? ? 1/4 8t 2 [1, T]. Then the FTRL update xt+1 =

argminx2K (g1:t + get+1)¿ x + R(x) admits the following guarantee: T T X X 1 8x 2 K, gt¿ (xt x) ? 2? kgt get k2xt ,? + R(x). ? t=1 t=1 In general, it is not clear what would be a good prediction candidate. Indeed, this is why Rakhlin and Sridharan [2013] called this algorithm an ?optimistic? FTRL. However, notice that if we elect to play the averaged smoothed losses as in [Dekel et al., 2015], then the update at each time is Pk Pk 1 1 g?t = k+1 bt i . This implies that the time t + 1 gradient is g?t+1 = k+1 bt+1 i , which i=0 g i=0 g includes the smoothed gradients from time t + 1 down to time t (k 1). The key insight here is that at time t, all but the (t + 1)-th gradient are known!

This means that if we predict k

get+1 =

1 X gbt+1 k + 1 i=0

k

1 1 X gbt+1 = gbt+1 i , k+1 k + 1 i=1

i

then the first term in the bound of Lemma 4 will be in terms of k

gt

get =

1 X gbt k + 1 i=0

k

1 X gbt k + 1 i=1

i

i

=

1 gbt . k+1

In other words, all but the time t smoothed gradient will cancel out. Essentially, we are predicting the predictable portion of the averaged gradient and guaranteeing that the optimism will pay off. 1 Moreover, where we gained a factor of k+1 in the averaged loss case, we should expect to gain a 1 factor of (k+1)2 by using this optimistic prediction. Note that this technique of optimistically predicting the variance reduction is widely applicable. As alluded to with the reference to [Schmidt et al., 2013], many variance reduction-type techniques, particularly in stochastic optimization, use historical information in their estimates (e.g. SVRG [Johnson and Zhang, 2013], SAGA [Defazio et al., 2014]). In these cases, it is possible to ?predict? the information re-use and improve the convergence rates of each algorithm. 4.2

Description and pseudocode

Here, we give a detailed description of our algorithm, O PTIMISTIC BCO. At each round t, the algorithm uses a sample ut from the uniform distribution over the unit sphere to define an unbiased estimate of the gradient of fbt , a smoothed version of the loss function ft , as described in Section 2.2.1: d gbt ft (yt)(r2 R(xt)) 1/2 ut . Next, the trailing average of these unbiased estimates over a fixed Pk 1 window of length k + 1 is computed: g?t = k+1 bt i . The remaining steps executed at each i=0 g round coincide with the Follow-the-Regularized-Leader update with a self-concordant barrier used as a regularizer, augmented with an optimistic prediction of the next round?s trailing average.

As described in Section 4.1, all but one of the terms in the trailing average are known and we predict their occurence: k

get+1 =

1 X gbt+1 i , k + 1 i=1

¿

xt+1 = argmin ? (? g1:t + get+1) x + R(x). x2K

Note that Theorem 3 implies that the actual point we play, yt , is always a feasible point in K. Figure 1 presents the pseudocode of the algorithm. 5

O PTIMISTIC BCO(R, , ?, k, x1) 1 for t 1 to T do 2 ut S AMPLE(U (@B1 (0))) 1 3 yt xt + (r2 R(xt)) 2 ut 4 P LAY(yt) 5 ft (yt) R ECEIVE L OSS(yt) 1 d 6 gbt ft (yt)(r2 R(xt)) 2 ut P k 1 7 g?t bt i i=0 g k+1 P k 1 8 get+1 bt+1 i i=1 g k+1 ¿ 9 xt+1 argminx2K ? (? g1:t + get+1) x + R(x) PT 10 return t=1 ft (yt) Figure 1: Pseudocode of O PTIMISTIC BCO, with R : int(K) ! R, x1 2 K.

5

2 (0, 1], ? ¿ 0, k 2 Z, and

Regret guarantees

In this section, we state our main results, which are regret guarantees for O PTIMISTIC BCO in the C0,1 and C1,1 cases. We also highlight the analysis and proofs for each regime. 5.1

Main results

The following is our main result for the C0,1 case. Theorem 1 (C0,1 Regret). Let K ? Rd be a convex set with diameter D and (ft)Tt=1 a sequence of loss functions with each ft : K ! R+ C-bounded and L-Lipschitz. Let R be a ?-self-concordant barrier for K. Then, for ?k ? 12Cd , the regret of O PTIMISTIC BCO can be bounded as follows: Ck 2Cd2 ?T 1 RegT (O PTIMISTIC BCO) ? ?LT + L DT + + 2 + log(1/?) 2 2 (k + 1) ? ” p # p p 1/2 p 48d k + LT 2?D 3L + 2DLk + .

In particular, for ? = T 11/16 d for the regret of the algorithm:

3/8

, =T

5/16 3/8

d

, k = T 1/8 d1/4 , the following guarantee holds

? ? e T 11/16 d3/8 . RegT (O PTIMISTIC BCO) = O

The above result is the first improvement on the regret of Lipschitz losses in terms of T since the original algorithm of Flaxman et al. [2005] that is realizable from a concrete algorithm as well as polynomial in both dimension and time (both computationally and in terms of regret). Theorem 2 (C1,1 Bound). Let K ? Rd be a convex set with diameter D and (ft)Tt=1 a sequence of loss functions with each ft : K ! R+ C-bounded, L-Lipschitz and H-smooth. Let R be a ?-selfconcordant barrier for K. Then, for ?k ? 12d , the regret of O PTIMISTIC BCO can be bounded as follows: RegT (O PTIMISTIC BCO) ? ?LT + H 2 D2 T ”p # p 3L1/2 p 48d 1 + (T L + DHT)2?kD + 2DL + p + log(1/?) + Ck + ? k ? k In particular, for ? = T 8/13 d for the regret of the algorithm:

5/6

, =T
5/26 1/3
d
, k = T 1/13 d5/3 , the following guarantee holds
? ? e T 8/13 d5/3 . RegT (O PTIMISTIC BCO) = O 6
d2 T . 2 (k + 1)2

This result is currently the best polynomial-in-time regret bound that is also polynomial in the dimension of the action space (both computationally and in terms of regret). It improves upon the work of Saha and Tewari [2011] and Dekel et al. [2015]. We now explain the analysis of both results, starting with Theorem 1 for C0,1 losses. 5.2

C0,1 analysis

Our analysis proceeds in two steps. We first modularize the cost of approximating the original losses ft (yt) incurred with the averaged smoothed losses that we treat as surrogate losses. Then we show that the algorithm minimizes the regret against the surrogate losses effectively. The proofs of all lemmas in this section are presented in Appendix 7.2. Lemma 5 (C0,1 Structural bound on true losses in terms of smoothed losses). Let (ft)Tt=1 be a sequence of loss functions, and assume that ft : K ! R+ is C-bounded and L-Lipschitz, where K ? Rd . Denote d fbt (x) = E [ft (x + At u)], gbt = ft (yt)At 1 ut , yt = xt + At ut u?U (@B1 (0))

PT for arbitrary At , , and ut . Let x? = argminx2K t=1 ft (x), and let x?? 2 argminy2K,dist(y,@K)¿? ky x? k. Assume that we play yt at every round. Then the following structural estimate holds: RegT (A) = E[

T X
ft (yt)
t=1
ft (x?)] ? ?LT + 2L DT +
T X t=1
E[fbt (xt)
fbt (x??)].

Thus, at the price of ?LT + 2L DT , it suffices to look at the performance of the averaged losses for the algorithm. Notice that the only assumptions we have made so far are that we play points sampled on an ellipsoid around the desired point scaled by and that the loss functions are Lipschitz. Lemma 6 (C0,1 Structural bound on smoothed losses in terms of averaged losses). Let (ft)Tt=1 be a sequence of loss functions, and assume that ft : K ! R+ is C-bounded and L-Lipschitz, where K ? Rd . Denote d fbt (x) = E [ft (x + At u)], gbt = ft (yt)At 1 ut , yt = xt + At ut u?U (@B1 (0))

for arbitrary At , , and ut . Let x? = argminx2K ? argminy2K,dist(y,@K)¿? ky x k. Furthermore, denote k

f?t (x) =
1 Xb ft i (x), k + 1 i=0
PT
t=1
ft (x), and let x??

2
k
g?t =
1 X gbt i . k + 1 i=0

Assume that we play yt at every round. Then we have the structural estimate: ? T T X X ? Ck E fbt (xt) fbt (x??) ? + LT sup E[kxt i xt k2] + E g?t¿ (xt 2 t2[1,T],i2[0,kt] t=1 t=1
? x??) .

While we use averaged smoothed losses as in [Dekel et al., 2015], the analysis in this lemma is actually somewhat different. Because Dekel et al. [2015] always assume that the loss functions are in C1,1 , they elect to use the following decomposition: fbt (xt)
f?t (xt) + f?t (xt) f?t (x??) + f?t (x??) fbt (x??). Pk Pk 1 1 b b This is because they can relate rf?t (x) = k+1 ?t = k+1 i=0 rft i (x?) to g i=0 rft i (xt i) 0,1 using the fact that the gradients are Lipschitz. Since the gradients of C functions are not Lipschitz, we cannot use the same analysis. Instead, we use the decomposition fbt (xt)
fbt (x??) = fbt (xt)
fbt (x??) = fbt (xt)
fbt i (xt i) + fbt i (xt i)

The next lemma affirms that we do indeed get the improved predictable component of the average gradient. 7
f?t (x??) + f?t (x??) 1 (k+1)2
fbt (x??).
factor from predicting the

Lemma 7 (C0,1 Algorithmic bound on the averaged losses). Let (ft)Tt=1 be a sequence of loss functions, and assume that ft : K ! R+ is C-bounded and L-Lipschitz, where K ? Rd . Let PT x? = argminx2K t=1 ft (x), and let x?? 2 argminy2K,dist(y,@K)¿? ky x? k. Assume that we play according to the algorithm with ?k ? 12Cd . Then we maintain the following guarantee: T X t=1
? E g?t¿ (xt
? 2Cd2 ?T 1 x??) ? 2 + R(x??). 2 (k + 1) ?

So far, we have demonstrated a bound on the regret of the form: Ck RegT (A) ? ?LT + 2L DT + + LT sup E[kxt i 2 t2[T],i2[kt]
xt k2] +
2Cd2 ?T 1 + R(x?). 2 (k + 1)2 ?

Thus, it remains to find a tight bound on supt2[1,T],i2[0,kt] E[kxt i xt k2], which measures the stability of the actions across the history that we average over. This result is similar to that of Dekel et al. [2015], except that we additionally need to account for the optimistic gradient prediction used. Lemma 8 (C0,1 Algorithmic bound on the stability of actions). Let (ft)Tt=1 be a sequence of loss functions, and assume that ft : K ! R+ is C-bounded and L-Lipschitz, where K ? Rd . Assume that we play according to the algorithm with ?k ? 12Cd . Then the following estimate holds: ! p 1/2 p p 3L 48Cd E[kxt i xt k2] ? 2?kD + 2DL + p . k k Proof. [of Theorem 1] Putting all the pieces

together from Lemmas 5, 6, 7, 8, shows that p p ? p 1/2 p Ck 2Cd2 ?T 1 48Cd k RegT (A)??LT +L DT + + 2 + R(x?)+LT 2?D 3L + 2DLk+ . 2 (k + 1)2 ? Since x? is at least ? away from the boundary, it follows from [Abernethy and Rakhlin, 2009] that R(x?) ? ? log(1/?). Plugging in the stated quantities for ?, k, and yields the result. 5.3

C1,1 analysis

The analysis of the C1,1 regret bound is similar to the C0,1 case. The only difference is that we leverage the higher regularity of the losses to provide a more refined estimate on the cost of approximating ft with f?t . Apart from that, we will reuse the bounds derived in Lemmas 6, 7, and 8. The proof of the following lemma, along with that of Theorem 2, is provided in Appendix 7.3. Lemma 9 (C1,1 Structural bound on true losses in terms of smoothed losses). Let (ft)Tt=1 be a sequence of loss functions, and assume that ft : K ! R+ is C-bounded, L-Lipschitz, and H-smooth, where K ? Rd . Denote d fbt (x) = E [ft (x + At u)], gbt = ft (yt)A 1 ut , yt = xt + At ut t

u?U (@B1 (0))

PT for arbitrary At , , and ut . Let x? = argminx2K t=1 ft (x), and let x?? 2 argminy2K,dist(y,@K)¿? ky x? k. Assume that we play yt at every round. Then the following structural estimate holds: T T X X RegT (A) = E[ft (yt) ft (x?)] ? ?LT + 2H 2 D2 T + E[fbt (xt) fbt (x??)]. t=1

t=1

Conclusion

We designed a computationally efficient algorithm for bandit convex optimization admitting stateof-the-art guarantees for C0,1 and C1,1 loss functions. This was achieved using the general and powerful technique of predicting predictable information re-use. The ideas we describe here are directly applicable to other areas of optimization, in particular stochastic optimization. Acknowledgements This work was partly funded by NSF CCF-1535987 and IIS-1618662 and NSF GRFP DGE-1342536. 8

2 References

J. Abernethy and A. Rakhlin. Beating the adaptive bandit with high probability. In COLT, 2009. J. Abernethy, E. Hazan, and A. Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In COLT, pages 263?274, 2008. A. Agarwal, O. Dekel, and L. Xiao. Optimal algorithms for online convex optimization with multi-point bandit feedback. In COLT, pages 28?40, 2010. S. Bubeck and R. Eldan. Multi-scale exploration of convex functions and bandit convex optimization. CoRR, abs/1507.06580, 2015. S. Bubeck, O. Dekel, T. Koren, and Y. Peres. Bandit convex optimization: sqrt {T} regret in one dimension. CoRR, abs/1502.06398, 2015. S. Bubeck, R. Eldan, and Y. T. Lee. Kernel-based methods for bandit convex optimization. CoRR, abs/1607.03084, 2016. V. Dani, T. P. Hayes, and S. M. Kakade. Stochastic linear optimization under bandit feedback. A. Defazio, F. Bach, and

S. Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In NIPS, pages 1646?1654, 2014. O. Dekel, R. Eldan, and T. Koren. Bandit smooth convex optimization: Improving the bias-variance tradeoff. In NIPS, pages 2908?2916, 2015. A. D. Flaxman, A. T. Kalai, and H. B. McMahan. Online convex optimization in the bandit setting: Gradient descent without a gradient. In SODA, pages 385?394, 2005. E. Hazan and Y. Li. An optimal algorithm for bandit convex optimization. CoRR, abs/1603.04350, 2016. R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In NIPS, pages 315?323, 2013. R. D. Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In Advances in Neural Information Processing Systems, pages 697?704, 2004. Y. Nesterov. Introductory Lectures on Convex Optimization: A Basic Course. Springer, New York, NY, USA, 2004. Y. Nesterov and A. Nemirovskii. Interior-point Polynomial Algorithms in Convex Programming. Studies in Applied Mathematics. Society for Industrial and Applied Mathematics, 1994. ISBN 9781611970791. A. Rakhlin and K. Sridharan. Online learning with predictable sequences. In COLT, pages 993?1019, 2013. A. Saha and A. Tewari. Improved regret guarantees for online smooth convex optimization with bandit feedback. In AISTATS, pages 636?642, 2011. M. W. Schmidt, N. L. Roux, and F. R. Bach. Minimizing finite sums with the stochastic average gradient. CoRR, abs/1309.2388, 2013.

9