Smoothed Online Convex Optimization via Online Balanced Descent

Gautam Goel Niangjun Chen Adam Wierman *

ABSTRACT

We study *smoothed online convex optimization*, a version of online convex optimization where the learner incurs a penalty for changing her actions between rounds. Given a $\Omega(\sqrt{d})$ lower bound on the competitive ratio of any online algorithm, where d is the dimension of the action space, we ask under what conditions this bound can be beaten. We introduce a novel algorithmic framework for this problem, Online Balanced Descent (OBD), which works by iteratively projecting the previous point onto a carefully chosen level set of the current cost function so as to balance the switching costs and hitting costs. We demonstrate the generality of the OBD framework by showing how, with different choices of "balance," OBD can improve upon state-of-the-art performance guarantees for both competitive ratio and regret; in particular, OBD is the first algorithm to achieve a dimension-free competitive ratio, $3 + O(1/\alpha)$, for locally polyhedral costs, where α measures the "steepness" of the costs. We also prove bounds on the dynamic regret of OBD when the balance is performed in the dual space that are dimension-free and imply that OBD has sublinear static regret.

1. INTRODUCTION

In this paper we develop a new algorithmic framework, Online Balanced Descent (OBD), for online convex optimization problems with switching costs, a class of problems termed smoothed online convex optimization (SOCO). Specifically, we consider a setting where a learner plays a series of rounds $1, 2, \ldots, T$. In each round, the learner observes a convex cost function f_t, picks a point x_t from a convex set $\mathcal{X}$, and then incurs a *hitting cost* $f_t(x_t)$. Additionally, she incurs a *switching cost* for changing her actions between successive rounds, $\|x_t - x_{t-1}\|$, where $\|\cdot\|$ is a norm.

This setting generalizes classical Online Convex Optimization (OCO), and has received considerable attention in recent years as a result of the recognition that switching costs play a crucial role in many learning, algorithms, control, and networking problems. In particular, many applications have, in reality, some cost associated with a change of action that motivates the learner to adopt "smooth" sequences of actions. For example, switching costs have received considerable attention in the k-armed bandit setting [1, 13, 17] and the core of the Metrical Task Systems (MTS) literature is determining how to manage switching costs, e.g., the k-server problem [6, 7].

Outside of learning, SOCO has received considerable attention in the networking and control communities. In these problems there is typically a measurable cost to changing an action. For example, one of the initial applications where SOCO was adopted is the dynamic management of service capacity in data centers [20, 21], where the wear-and-tear costs of switching servers into and out of deep power-saving states is considerable. Other applications where SOCO has seen real-world deployment are the dynamic management of routing between data centers [19, 23], management of electrical vehicle charging [15], video streaming [14], speech animation [16], multi-timescale control [12], power generation planning [4], and the thermal management of System-on-Chip (SoC) circuits [24, 25].

An important aspect of nearly all the problems mentioned above is that they are *high-dimensional*, i.e., the dimension d of the action space is large. For example, in the case of dynamic management of data centers the dimension grows with the heterogeneity of the storage and compute nodes in the cluster, as well as the heterogeneity of the incoming workloads. However, the design of algorithms for high-dimensional SOCO problems has proven challenging, with fundamental lower bounds blocking progress.

Initial results on SOCO focused on finding competitive algorithms in the low-dimensional settings. Specifically, [20] introduced the problem in the one-dimensional case and gave a 3-competitive algorithm. A few years later, [5] gave a 2-competitive algorithm, still for the one-dimensional case. Following these papers, [3] claimed that SOCO is equivalent to the classical problem of Convex Body Chasing [11], in the sense that a competitive algorithm for one problem implies the existence of a competitive problem for the other. Using this connection, they claimed to show the existence of a constant competitive algorithm for two-dimensional SOCO. However, their analysis turned out to have a bug and has been retracted [22].

However, the connection to Convex Body Chasing also highlights a fundamental limitation. In particular, it is not possible to design a constant competitive algorithm for high-dimensional SOCO without making restrictions on the cost

*This is an extended abstract of [10]. Readers can refer to the full version for more details and proofs. Gautam Goel and Adam Wierman are with the Department of Computing and Mathematical Sciences, California Institute of Technology. Niangjun Chen is with Institute for High Performance Computing. Email:{ggoel, adamw}@caltech.edu, chennj@ihpc.a-star.edu.sg

functions considered, due to the following proposition:

Proposition 1. *For general convex cost functions and ℓ_2 switching costs, the competitive ratio of any algorithm is $\Omega(\sqrt{d})$.*

Given the importance of high-dimensional SOCO problems in practical applications, this lower bound motivated the exploration of "beyond worst-case" analysis for SOCO as a way of breaking through the $\sqrt{d}$ barrier. To this end, [2,4, 8,9,19] all explored the value of predictions in SOCO, highlighting that it is possible to provide constant-competitive algorithms for high-dimensional SOCO problems using algorithms that have predictions of future cost functions, e.g., [19] gave an algorithm based on receding horizon control that is $1+O(1/w)$-competitive when given w-step lookahead. Recently, this was revisited in the case quadratic switching costs by [18], which gives an algorithm that combines receding horizon control with gradient descent to achieve a competitive ratio that decays exponentially in w.

The prior discussion highlights the challenges associated with designing algorithms for high-dimensional SOCO problems, both in terms of competitive ratio and dynamic regret. In this paper, we introduce a new, general algorithmic framework, Online Balanced Descent (OBD), that yields an algorithm with a dimension-free, constant competitive ratio for locally polyhedral cost functions and ℓ_2 switching costs. OBD achieves these results without relying on predictions of future cost functions.

The key idea behind OBD is to move using a projection onto a carefully chosen level set at each step chosen to "balance" the switching and hitting costs incurred. The resulting OBD algorithm is efficient to implement, even in high dimensions. They are also *memoryless*, i.e., do not use any information about previous cost functions.

The technical results of the paper bound the competitive ratio of OBD. In both cases we obtain results that improve the state-of-the-art. In the case of *competitive ratio*, we obtain the first results that break through the $\sqrt{d}$ barrier without the use of predictions. In particular, we show that OBD with ℓ_2 switching costs yields a constant, dimension-free competitive ratio for locally polyhedral cost functions, i.e. functions which grow at least linearly away from their minimizer. Specifically, in Theorem 3 we show that OBD has a competitive ratio of $3 + O(1/\alpha)$, where α bounds the "steepness" of the costs. Note that [5] shows that no memoryless algorithm can achieve a competitive ratio better than 3 for locally polyhedral functions. By equivalence of norms in finite dimensional space, our algorithm is also competitive when the switching costs are arbitrary norms (though the exact competitive ratio may depend on d).

2. A COMPETITIVE ALGORITHM

In this section, we use the OBD framework to give the first algorithm with a dimension-free, constant competitive ratio for online convex optimization with switching costs in general Euclidean spaces, under mild assumptions on the structure of the cost functions. Recall that, for the most general case, where no constraints other than convexity are applied to the cost functions, the competitive ratio of any online algorithm must be $\Omega(\sqrt{d})$ for ℓ_2 switching costs, i.e., must grow with the dimension d of the decision space. Our goal in this section is to understand when a dimension-free, constant competitive ratio can be obtained. Thus, we are naturally led to restrict the type of cost functions we consider.

Our main result in this section is a new online algorithm whose competitive ratio is constant with respect to dimension when the cost functions are *locally polyhedral*, a class that includes the form of cost functions used in many applications of online convex optimization, e.g, tracking problems and penalized estimation problems. Roughly speaking, locally polyhedral functions are those that grow at least linearly as one moves away from the minimizer, at least in a small neighborhood.

Definition 2. *A function f_t with minimizer v_t is **locally α-polyhedral** with respect to the norm $\|\cdot\|$ if there exists some $\epsilon > 0$, such that for all $x \in \mathcal{X}$ with $\|x - v_t\| \leq \epsilon$, $f_t(x) - f_t(v_t) \geq \alpha \|x - v_t\|$.*

Note that all strictly convex functions f_t which are locally α-polyhedral automatically satisfy $f_t(x) - f_t(v_t) \geq \alpha \|x - v_t\|$ for all x, not just those x which are ϵ close to the minimizer v_t. In this setting, local polyhedrality is analogous to strong convexity; instead of requiring that the cost functions grow at least quadratically as one moves away from the minimizer, the definition requires that cost functions grow at least linearly. The following examples illustrate the breadth of this class of functions. One important class of examples are functions of the form $\|x - v_t\|_a$ where $\|\cdot\|_a$ is an arbitrary norm; it follows from the equivalence of norms that such functions are locally polyhedral. Intuitively, such functions represent "tracking" problems, where we seek to get close to the point v_t. Another important example is the class $f(x_t) = g(x_t) + h(x_t)$ where g is locally polyhedral and h is an arbitrary non-negative convex function whose minimizer coincides with that of g; since $f(x_t) - f(v_t) \geq g(x_t) - g(v_t)$, f is also locally polyhedral. This lets us handle interesting functions such as $f(x_t) = \|x_t\|_1 + x_t' Q x_t$ where Q is a positive semidefinite matrix, or even $f(X_t) = 2\|X_t\|_\infty - \log\det(I + X_t)$ where the decision variable X_t is a positive semidefinite matrix.

Let us now informally describe how the Online Balanced Descent works. Online Balanced Descent is *lazy*: instead of moving directly towards the minimizer v_t, it moves to the closest point which results in a suitably large decrease in the hitting cost. This can be interpreted as projecting onto a sublevel set of the current cost function. The trick is to make sure that not too much switching cost is incurred in the process. This is accomplished by carefully picking the sublevel set so that the hitting costs and switching costs are balanced. A formal description is given Algorithm 1. Note that the memoryless algorithm proposed in [5] can be seen as a special case of Algorithm 1 when the decision variables are scalar.

Theorem 3. *For every $\alpha > 0$, there exists a choice of β such that Algorithm 1 has competitive ratio at most $3 + O(1/\alpha)$ when run on locally α-polyhedral cost functions with ℓ_2 switching costs. More generally, let $\|\cdot\|$ be an arbitrary norm. There exists a choice of β such that Algorithm 1 has competitive ratio at most $\frac{\max\{k_2,1\}}{\min\{k_1,1\}}(3 + O(1/\alpha))$ when run on locally α-polyhedral cost functions with switching cost $\|\cdot\|$. Here k_1 and k_2 are constants such that $k_1\|x\| \leq \|x\|_2 \leq k_2\|x\|$.*

We note that in the ℓ_2 setting Theorem 3 has a form which is connected to the best known lower bound on the

Algorithm 1 (Primal) Online Balanced Descent

for $t = 1, \dots, T$ **do**
  Observe cost function f_t, set $v_t = \operatorname{argmin}_x f_t(x)$.
  if $\|x_{t-1} - v_t\| < \beta f_t(v_t)$ **then**
    Set $x_t = v_t$
  else
    Let $x(l) = \Pi^{\Phi}_{K_t^l}(x_{t-1})$, increase l until $\|x(l) - x_{t-1}\| = \beta l$. Here K_t^l is the l-sublevel set of f_t, i.e., $K_t^l = \{x \mid f_t(x) \leq l\}$, and Π^{Φ} denotes projection under the mirror map Φ.
    $x_t = x(l)$.
  end if
end for

competitive ratio of memoryless algorithms. In particular, [5] use a 1-dimensional example with locally polyhedral cost functions to prove the following bound.

Proposition 4. *No memoryless algorithm can have a competitive ratio less than 3.*

3. REFERENCES

[1] R. Agrawal, M. Hegde, and D. Teneketzis. Multi-armed bandit problems with multiple plays and switching cost. *Stochastics and Stochastic Reports*, 29(4):437–459, 1990.

[2] L. Andrew, S. Barman, K. Ligett, M. Lin, A. Meyerson, A. Roytman, and A. Wierman. A tale of two metrics: Simultaneous bounds on competitiveness and regret. In *Conference on Learning Theory*, pages 741–763, 2013.

[3] A. Antoniadis, N. Barcelo, M. Nugent, K. Pruhs, K. Schewior, and M. Scquizzato. Chasing convex bodies and functions. In *Latin American Symposium on Theoretical Informatics*, pages 68–81. Springer, 2016.

[4] M. Badiei, N. Li, and A. Wierman. Online convex optimization with ramp constraints. In *IEEE Conference on Decision and Control*, pages 6730–6736, 2015.

[5] N. Bansal, A. Gupta, R. Krishnaswamy, K. Pruhs, K. Schewior, and C. Stein. A 2-competitive algorithm for online convex optimization with switching costs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 96–109, 2015.

[6] A. Borodin and R. El-Yaniv. *Online computation and competitive analysis.* Cambridge University Press, 2005.

[7] A. Borodin, N. Linial, and M. E. Saks. An optimal on-line algorithm for metrical task system. *J. ACM*, 39(4):745–763, Oct. 1992.

[8] N. Chen, A. Agarwal, A. Wierman, S. Barman, and L. L. Andrew. Online convex optimization using predictions. In *ACM SIGMETRICS Performance Evaluation Review*, volume 43, pages 191–204. ACM, 2015.

[9] N. Chen, J. Comden, Z. Liu, A. Gandhi, and A. Wierman. Using predictions in online optimization: Looking forward with an eye on the past. *SIGMETRICS Perform. Eval. Rev.*, 44(1):193–206, June 2016.

[10] N. Chen, G. Goel, and A. Wierman. Smoothed online convex optimization in high dimensions via online balanced descent. *CoRR*, abs/1803.10366, 2018.

[11] J. Friedman and N. Linial. On convex body chasing. *Discrete & Computational Geometry*, 9(3):293–321, Mar 1993.

[12] G. Goel, N. Chen, and A. Wierman. Thinking fast and slow: Optimization decomposition across timescales. *CoRR*, abs/1704.07785, 2017.

[13] S. Guha and K. Munagala. Multi-armed bandits with metric switching costs. In *International Colloquium on Automata, Languages, and Programming*, pages 496–507. Springer, 2009.

[14] V. Joseph and G. de Veciana. Jointly optimizing multi-user rate adaptation for video transport over wireless systems: Mean-fairness-variability tradeoffs. In *IEEE INFOCOM*, pages 567–575, 2012.

[15] S.-J. Kim and G. B. Giannakis. Real-time electricity pricing for demand response using online convex optimization. In *IEEE Innovative Smart Grid Tech.*, pages 1–5, 2014.

[16] T. Kim, Y. Yue, S. Taylor, and I. Matthews. A decision tree framework for spatiotemporal sequence prediction. In *ACM International Conference on Knowledge Discovery and Data Mining*, pages 577–586, 2015.

[17] T. Koren, R. Livni, and Y. Mansour. Multi-armed bandits with metric movement costs. In *Advances in Neural Information Processing Systems*, pages 4122–4131, 2017.

[18] Y. Li, G. Qu, and N. Li. Online optimization with predictions and switching costs: Fast algorithms and the fundamental limit. *CoRR*, abs/1801.07780, 2018.

[19] M. Lin, Z. Liu, A. Wierman, and L. L. Andrew. Online algorithms for geographical load balancing. In *IEEE Green Computing Conference*, pages 1–10, 2012.

[20] M. Lin, A. Wierman, L. L. H. Andrew, and T. Eno. Dynamic right-sizing for power-proportional data centers. In *IEEE INFOCOM*, pages 1098–1106, 2011.

[21] T. Lu, M. Chen, and L. L. Andrew. Simple and effective dynamic provisioning for power-proportional data centers. *IEEE Transactions on Parallel and Distributed Systems*, 24(6):1161–1171, 2013.

[22] K. Pruhs. Errata. `http://people.cs.pitt.edu/~kirk/Errata.html`.

[23] H. Wang, J. Huang, X. Lin, and H. Mohsenian-Rad. Exploring smart grid and data center interactions for electric power load balancing. *ACM SIGMETRICS Performance Evaluation Review*, 41(3):89–94, 2014.

[24] F. Zanini, D. Atienza, L. Benini, and G. De Micheli. Multicore thermal management with model predictive control. In *IEEE. European Conf. Circuit Theory and Design*, pages 711–714, 2009.

[25] F. Zanini, D. Atienza, G. De Micheli, and S. P. Boyd. Online convex optimization-based algorithm for thermal management of MPSoCs. In *The Great lakes symposium on VLSI*, pages 203–208, 2010.