A Short Note of
Smoothed Online Convex Optimization
(In Progress)

Yilin Yan
yilin.yan.tcs@gmail.com
Jilin University

June 6, 2021

1 Preliminaries

1.1 Online Convex Optimization

An instance of **Online Convex Optimization (OCO)** [Haz16] consists of an initial point $x_0 \in \mathbb{R}^d$, a sequence of hitting cost functions $f_1, \ldots, f_T$. The hitting cost are revealed online, one at a time. At each step t, the algorithm need to choose a point $x_t \in \mathbb{R}^d$ in response. The goal is to minimize the total cost defined by $\sum_{t=1}^{T} f_t(x_t)$.

1.2 Convex Bodies Chasing

An instance of **Convex Bodies Chasing (CBC)**, which was introduced by [FL93], consists of an initial point $x_0 \in \mathbb{R}^d$ and a sequence of convex sets $K_1, K_2, \ldots, K_T \subseteq \mathbb{R}^d$ called requests. The sets K_t are revealed online, one at a time. At each step t, the algorithm need to choose a point $x_t \in K_t$ and move from x_{t-1} to x_t [fig.1]. The goal is to minimize the total cost defined by $\sum_{t=1}^{T} \|x_t - x_{t-1}\|$, where $\|\cdot\|$ is a norm.

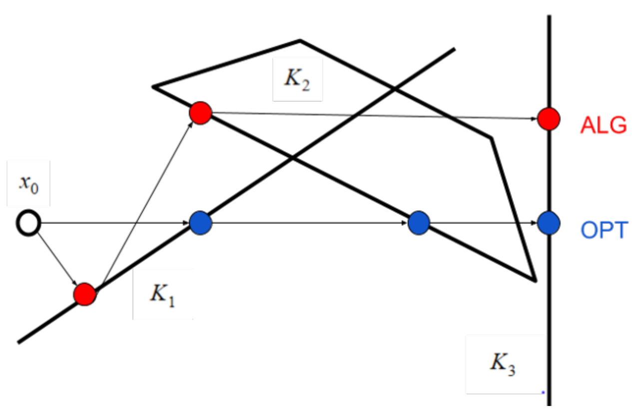


Figure 1: Convex Bodies Chasing (CBC)

The original paper [FL93] proved that the best lower bound of competitive ratio is at least $\sqrt{d}$ in Euclidean space and d in arbitrary normed spaces, and an algorithm with finite competitive ratio was given for the already non-trivial $d = 2$ case, and was conjectured to exist for larger d, until this conjecture was firstly resolved in [Bub+19] which proved an exponential $2^{O}(d)$ upper bound on the competitive ratio. On the other hand, due to a lack of progress on the upper bound, restricted cases such as chasing nested convex bodies were studied. Nested chasing was introduced in [Ban+19] and solved rather comprehensively in [Arg+19] and then [Bub+20] gave a nearly optimal algorithm for all l^p spaces, as well as a memoryless $\min(d, O(\sqrt{d}))$ competitive algorithm for Euclidean spaces based on the Steiner point of a convex body.

Convex Functions Chasing (CFC) introduced by [Ant+16] can be seen as an equivalent functional variant of CBC. In CFC, the request is a convex function f_n instead of a convex set. The cost function now consists of two additive parts: the movement cost $\sum_{t=1}^{T} \|x_t - x_{t-1}\|$ as CBC, and a service cost $\sum_{t=1}^{T} f_t(x_t)$. CFC subsumes CBC by considering for any body K, the function f_t which is 0 in the body and $+\infty$ outside the body. This functional perspective of CBC is equivalent to SOCO (described below) and thus opens more possibility for applications.

[Bub+19] shows that CFC in dimension d can be reduced to CBC in dimension $d+1$ and it turns out that CBC in dimension $d+1$ is enough to solve CFC in dimension d. Inspired by [Bub+20] chasing nested convex bodies based on the Steiner point of a convex body, and the equivalence of CFC and CBC, [Sel20] define a functional Steiner point of a convex function and apply it to the work function, then drastically improve the exponential upper bound proved by [Bub+19], and gives an algorithm achieving competitive ratio d for arbitrary normed spaces, which is exactly tight for l^∞, and nearly optimal competitive ratio $O(\sqrt{d})$, compared to a lower bound of $\sqrt{d}$ in Euclidean space.

1.3 Performance metrics

The two CS communities with perhaps the largest literatures on OCO-type problems are the machine learning and online algorithms communities, and the formulations studied in each community are quite different.

1. Competive ratio: In the online algorithms literature, the goal is typically to minimize the competitive ratio, which is the maximum ratio between the cost of the algorithm and the cost of the offline optimal (dynamic) solution.

2. Dynamic regret: In the machine learning literature, the goal is typically to minimize the regret, which is the difference between the cost of the algorithm and the cost of the offline optimal static solution.

2 Smoothed Online Convex Optimization

Smoothed Online Convex Optimization (SOCO) is also known as OCO with switching costs. Compared with general OCO, the cost function now consists of two additive parts in SOCO: hitting cost functions $f_1, \ldots, f_T$ from before, and a swiching cost, a.k.a., movement cost $\|x_t - x_{t-1}\|$. The objective is to minimize the cost. The total cost incurred is thus:

$$\text{cost(ALG)} = \sum_{t=1}^{T} f_t(x_t) + \|x_t - x_{t-1}\|$$

SOCO was first introduced by [Lin+12] with applications on resource management in data centers. Initial results on SOCO focused on finding competitive algorithms in the low-dimensional settings. Specifically, [Lin+12] introduced the problem in the one-dimensional case and gave a 3-competitive algorithm. A few years later, still for the one-dimensional case, [Ban+15] gave a 2-competitive algorithm. It is easy to see that SOCO is equivalent to CFC, and thus to CBC. To be more specific, [Ant+16] claimed that SOCO is equivalent to CBC in the sense that a competitive algorithm for one problem implies the existence of a competitive problem for the other. However, their analysis turned out to have a bug and until a few years later, a formal proof of this equivalence is provided by [LGW20] again: if there exists a $C-$competitive algorithm that can solve the CBC problem in $(d+1)-$dimensional space, then, there exists a $4C-$competitive algorithm that can solve the SOCO problem in $d-$dimensional space.

However, the connection to CBC also highlights a fundamental limitation. The dimension of SOCO 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. Following from connections to CBC, [CGW18] proves the lower bound that for general convex cost functions and l_2 switching costs, the competitive ratio of any online algorithm for SOCO is $\Omega(\sqrt{d})$. Furthermore, [LGW20] proves that CBC violates some conditions, thus any online algorithm for CBC with a finite prediction window has competitive ratio lower bounded by $\Omega(\sqrt{d})$ too. To be specific, up untill now, even the best algorithms for general CBC proposed by [Sel20] has a upper bound of $O(\sqrt{d})$ for Euclidean space, which grows with both dimension d and time T. These evidences show that, even with prediction, it is not possible to design a constant competitive algorithm for high-dimensional SOCO without making restrictions on the cost functions considered. Given the importance of high-dimensional SOCO problems in practical applications, this lower bound motivated the exploration of beyond worst-case analysis [Rou20] for SOCO as a way of breaking through the barrier, to achieve a dimension-free, constant competitive ratio.

3 The Power of Prediction

A lot of previous works try to explored the value of predictions in SOCO [Lin+12; Che+15; Che+16; Com+19], highlighting that it is possible to provide constant-competitive algorithms for high-dimensional SOCO problems using algorithms that have predictions of future cost functions. However, these works using predictions optimistically assume simple prediction errors such as a finite prediction window that contains perfect predictions [Lin+12], or with probabilistic prediction models which provides theoretical bounds on exp ected competitive performance [Che+15; Che+16], which both avoid accurately quantifying the impact of prediction errors when incorporating realistic predictions into the analysis of online algorithms [Com+19].

On the other hand, in the case without the use of predictions, for nearly a decade there were no algorithms for SOCO that worked beyond one dimension ($d = 1$) until Online Balanced Descent (OBD) proposed by [CGW18] obtains the first results that break through the $\Omega(\sqrt{d})$ barrier, in the special case where the cost functions are polyhedral. Following this work, [Goe+19] shows that OBD also provides a dimension-free competitive ratio when the hitting costs are strongly convex and that a variant of OBD called Regularized OBD achieves the optimal competitive ratio when hitting costs are strongly convex. However, they pessimistically assume that no trustworthy predictions are available [Com+19].

While there has been considerable progress on designing algorithms for SOCO both with and without predictions, to this point the results all rely on specific structural assumptions about the costs. Besides, up until now, the algorithms that are designed to be competitive without predictions are not able to take advantage of predictions when they are available, while the algorithms that are designed to be competitive with predictions are not able to be competitive without the use of predictions [LGW20].

However, [Com+19] introduces a simple prediction error model that connects prediction errors to the potential performance loss and analyze dynamic regret of OBD in the presence of prediction errors, which inspires us to ask: Is it possible to achieve a dimension-free, constant competitive ratio for high-dimensional SOCO problems under a unified analytical framework combining structural assumptions and predictions?

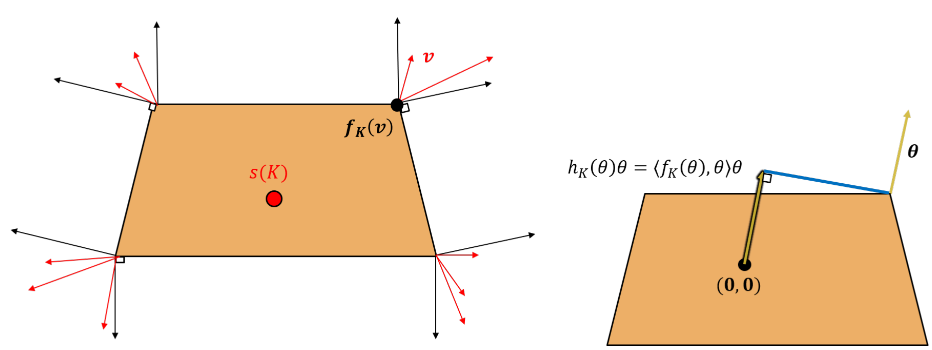
Two popular paradigms for dealing with uncertainty are online algorithms that are designed to work without knowing the input to the problem in advance, and machine learning that makes future predictions by fitting a model to prior data. Recent work has begun to incorporate machine learned predictions into the design of online algorithms to improve their performance [LV18; PSK18; Mit18], a.k.a. Learning-Augmented Algorithms. The goal is to incorporate the ML predictions in a manner that improves the performance of the online algorithm if the predictions are accurate (a design goal called consistency), but not degrade it significantly if the predictions are inaccurate (a design goal called robustness). Note that these properties are ensured by the algorithm without any knowledge of the quality of the predictions. [LV18; PSK18] introduce three properties when evaluating the algorithms with prediction:

1. Independence: The algorithm should be independent of the predictor, and make no assumptions about the predictor's error types and distribution. This way, by improving the predictor or the algorithm independently, we can have an improvement in performance. Also, the same algorithm can be used in different settings, which may have different distributions and error types.

2. Consistency: The performance of the algorithm should improve with better predictions. Essentially, the algorithm should actually use the predictor.

3. Robustness: The performance of the algorithm should degrade gracefully with bad predictions. More specifically, the algorithm performance should be bounded even with a bad predictor.

4 Functional Steiner Point and Work Function



(a) Primal Definition (b) Dual Definition

Figure 2: Steiner Point

[Sel20] gives an algorithm achieving competitive ratio d for arbitrary normed spaces, which is exactly tight for l^∞, and nearly optimal competitive ratio $O(\sqrt{d})$, compared to a lower bound of $\sqrt{d}$ in Euclidean space, which drastically improve the exponential upper bound proved by [Bub+19]. However, this general bound is not particularly useful, not only because of constant and dimension-free competitive ratio motivated by high-dimensional applications, but also because that many real-world problems that can be modeled as CBC and SOCO do have nice structures and properties, which makes it possible to design online algorithms with constant competive ratio, independet of the dimension d and time T. What we are striving to figure out is the possibility to achive constant and dimension-free competitive ratio considering the algorithms prposed by [Sel20], under as few as possible structural assumptions.

References

[Haz16] Elad Hazan. "Introduction to Online Convex Optimization". In: *Foundations and Trends® in Optimization* 2.3-4 (2016), pp. 157–325. ISSN: 2167-3888. DOI: 10.1561/2400000013. URL: http://dx.doi.org/10.1561/2400000013 (cit. on p. 1).

[FL93] Joel Friedman and Nathan Linial. "On convex body chasing". In: *Discrete & Computational Geometry* 9.3 (1993), pp. 293–321 (cit. on p. 1).

[Bub+19] Sébastien Bubeck et al. "Competitively chasing convex bodies". In: *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*. 2019, pp. 861–868 (cit. on pp. 2, 4).

[Ban+19] Nikhil Bansal et al. "Nested convex bodies are chaseable". In: *Algorithmica* (2019), pp. 1–14 (cit. on p. 2).

[Arg+19] CJ Argue et al. "A nearly-linear bound for chasing nested convex bodies". In: *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM. 2019, pp. 117–122 (cit. on p. 2).

[Bub+20] Sébastien Bubeck et al. "Chasing nested convex bodies nearly optimally". In: *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM. 2020, pp. 1496–1508 (cit. on p. 2).

[Ant+16] Antonios Antoniadis et al. "Chasing convex bodies and functions". In: *LATIN 2016: Theoretical Informatics*. Springer. 2016, pp. 68–81 (cit. on p. 2).

[Sel20] Mark Sellke. "Chasing convex bodies optimally". In: *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM. 2020, pp. 1509–1518 (cit. on pp. 2–4).

[Lin+12] Minghong Lin et al. "Dynamic right-sizing for power-proportional data centers". In: *IEEE/ACM Transactions on Networking* 21.5 (2012), pp. 1378–1391 (cit. on pp. 2, 3).

[Ban+15] Nikhil Bansal et al. "A 2-competitive algorithm for online convex optimization with switching costs". In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2015)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik. 2015 (cit. on p. 2).

[LGW20] Yiheng Lin, Gautam Goel, and Adam Wierman. "Online optimization with predictions and non-convex losses". In: *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 4.1 (2020), pp. 1–32 (cit. on p. 3).

[CGW18] Niangjun Chen, Gautam Goel, and Adam Wierman. "Smoothed online convex optimization in high dimensions via online balanced descent". In: *Conference On Learning Theory*. PMLR. 2018, pp. 1574–1594 (cit. on p. 3).

[Rou20] Tim Roughgarden. *Beyond the Worst-Case Analysis of Algorithms*. Cambridge University Press, 2020 (cit. on p. 3).

[Che+15] Niangjun Chen et al. "Online convex optimization using predictions". In: *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*. 2015, pp. 191–204 (cit. on p. 3).

[Che+16] Niangjun Chen et al. "Using predictions in online optimization: Looking forward with an eye on the past". In: *ACM SIGMETRICS Performance Evaluation Review* 44.1 (2016), pp. 193–206 (cit. on p. 3).

[Com+19] Joshua Comden et al. "Online optimization in cloud resource provisioning: Predictions, regrets, and algorithms". In: *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 3.1 (2019), pp. 1–30 (cit. on p. 3).

[Goe+19] Gautam Goel et al. "Beyond online balanced descent: An optimal algorithm for smoothed online optimization". In: *Advances in Neural Information Processing Systems* 32 (2019), pp. 1875–1885 (cit. on p. 3).

[LV18] Thodoris Lykouris and Sergei Vassilvtiskii. "Competitive caching with machine learned advice". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 3296–3305 (cit. on p. 4).

[PSK18] Manish Purohit, Zoya Svitkina, and Ravi Kumar. "Improving online algorithms via ml predictions". In: *Advances in Neural Information Processing Systems*. 2018, pp. 9661–9670 (cit. on p. 4).

[Mit18] Michael Mitzenmacher. "A model for learned bloom filters, and optimizing by sandwiching". In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. 2018, pp. 462–471 (cit. on p. 4).