JMLR: Workshop and Conference Proceedings vol 49:1–6, 2016

Open Problem: Parameter-Free and Scale-Free Online Algorithms

Francesco Orabona FRANCESCO@ORABONA.COM
Dávid Pál DPAL@YAHOO-INC.COM
Yahoo Research, New York

Abstract

Existing vanilla algorithms for online linear optimization have $O((\eta R(u) + 1/\eta)\sqrt{T})$ regret with respect to any competitor u, where $R(u)$ is a 1-strongly convex regularizer and $\eta > 0$ is a tuning parameter of the algorithm. For certain decision sets and regularizers, the so-called *parameter-free* algorithms have $\widetilde{O}(\sqrt{R(u)T})$ regret with respect to any competitor u. Vanilla algorithm can achieve the same bound only for a fixed competitor u known ahead of time by setting $\eta = 1/\sqrt{R(u)}$. A drawback of both vanilla and parameter-free algorithms is that they assume that the norm of the loss vectors is bounded by a constant known to the algorithm. There exist *scale-free* algorithms that have $O((\eta R(u)+1/\eta)\sqrt{T}\max_{1 \le t \le T} \|\ell_t\|)$ regret with respect to any competitor u and for any sequence of loss vector $\ell_1, \ldots, \ell_T$. Parameter-free analogue of scale-free algorithms have never been designed. Is is possible to design algorithms that are simultaneously *parameter-free* and *scale-free*?

1. Introduction

We consider the standard Online Linear Optimization (OLO) (Cesa-Bianchi and Lugosi, 2006; Shalev-Shwartz, 2011) setting. In each round t, an algorithm chooses a point x_t from a convex *decision set* K and then receives a loss vector ℓ_t. Regret of the algorithm w.r.t. $u \in K$ is $\text{Regret}_T(u) = \sum_{t=1}^T \langle \ell_t, x_t \rangle - \sum_{t=1}^T \langle \ell_t, u \rangle$. The goal of the algorithm is to keep the regret small.

We focus on two particular sets, the N-dimensional probability simplex $\Delta_N = \{x \in \mathbb{R}^N : x \ge 0, \|x\|_1 = 1\}$ and the Hilbert space. OLO over Δ_N is referred to as the problem of Learning with Expert Advice (LEA).

Notation and preliminaries: We denote by $\mathbf{1}$ the vector $(1, 1, \ldots, 1) \in \mathbb{R}^N$. Shannon entropy $H(u) = -\sum_{i=1}^N u_i \ln u_i$ is defined for any $u \in \Delta_N$. The Kullback-Leibler divergence $D(u\|v) = \sum_{i=1}^N u_i \ln(u_i/v_i)$ is defined for any $u, v \in \Delta_N$. For any $p \in [1, \infty]$, $\|\cdot\|_p$ denotes p-norm in $\mathbb{R}^N$. We denote by $\|\cdot\|_*$ the dual norm of a norm $\|\cdot\|$. If $\mathcal{H}$ is a real Hilbert space, we denote by $\langle \cdot, \cdot \rangle$ its inner product, and by $\|\cdot\|$ the induced norm. The negative Shannon entropy, $-H(u)$, defined on Δ_N is 1-strongly convex with respect to $\|\cdot\|_1$. The dual norm of $\|\cdot\|_1$ is $\|\cdot\|_\infty$. The function $R(u) = \frac{1}{2}\|u\|^2$ defined on a Hilbert space with norm $\|\cdot\|$ is 1-strongly convex with respect to $\|\cdot\|$.

Follow The Regularized Leader (FTRL) algorithm with regularizer $R : K \to \mathbb{R}$ and *learning rate* $\eta > 0$ in round t chooses $x_t = \arg\min_{x \in K} \left(\frac{1}{\eta} R(x) + \sum_{s=1}^{t-1} \langle \ell_s, x \rangle \right)$. The following theorem is a slight modification of Shalev-Shwartz (2011, Theorem 2.11).

Theorem 1 (Regret of FTRL) *If $R : K \to \mathbb{R}$ is 1-strongly convex function with respect a norm $\|\cdot\|$ then for any sequence $\{\ell_t\}_{t=1}^\infty$ such that $\|\ell_t\|_* \le 1$, FTRL with learning rate η satisfies for all $T \ge 0$ and all $u \in K$,* $\text{Regret}_T(u) \le \frac{R(u) - \inf_{v \in K} R(v)}{\eta} + \frac{\eta T}{2}$.

2. Learning with Expert Advice

Hedge algorithm (Freund and Schapire, 1997) for LEA satisfies

$$\forall u \in \Delta_N \qquad \text{Regret}_T(u) \le \sqrt{2T \ln N} \tag{1}$$

This bound is known to be optimal in the worst-case sense (Cesa-Bianchi and Lugosi, 2006, Section 3.7). However, (1) has two drawbacks. First, the right-hand side of (1) is independent of u, that is, the algorithm does not adapt to u. Second, Hedge satisfies (1) only if $\ell_1, \ell_2, \dots, \ell_T \in [-1,1]^N$. We would like to allow loss vectors that are arbitrary vectors in $\mathbb{R}^N$.

Hedge is identical to FTRL with regularizer $R(u) = -H(u)$. Theorem 1 implies that Hedge with learning rate η, satisfies

$$\text{Regret}_T(u) \le \frac{\ln N - H(u)}{\eta} + \frac{\eta T}{2} = \frac{D\left(u \middle\| \frac{1}{N}\mathbf{1}\right)}{\eta} + \frac{\eta T}{2} \,. \tag{2}$$

Let $p \in [0, \ln N)$. If we choose $\eta = \sqrt{\frac{\ln(N)-p}{T}}$, we get that

$$\forall u \in \Delta_N, \qquad H(u) \ge p \quad \Longrightarrow \quad \text{Regret}_T(u) \le \sqrt{2T(\ln(N) - p)} \,. \tag{3}$$

The bound (1) corresponds to choice $p = 0$ and $\eta = \sqrt{\frac{\ln N}{T}}$.

Instead of the family of algorithms parametrized by $p \in [0, \ln N)$ that satisfy bound (3), one *would like to have* a single algorithm (without any tuning parameters) satisfying

$$\forall u \in \Delta_N \qquad \text{Regret}_T(u) \le \sqrt{2T(\ln N - H(u))} = \sqrt{2T \cdot D\left(u \middle\| \tfrac{1}{N}\mathbf{1}\right)} \,. \tag{4}$$

Note that (4) is stronger than (3) in following the sense: A single algorithm satisfying (4) implies (3) for all p simultaneously. However, a family of algorithms $\{A_p \,:\, p \in [0, \ln N)\}$ parametrized by p where A_p satisfies (3), does not yield a single algorithm satisfying (4).

Bounds of the form (4) were not considered until 2009. Since then, however, there have been a lot of work (Chaudhuri et al., 2009; Chernov and Vovk, 2010; Koolen and van Erven, 2015; Luo and Schapire, 2014, 2015; Foster et al., 2015; Orabona and Pál, 2016a) on algorithms that satisfy slightly looser[1] versions of (4)

$$\forall u \in \Delta_N \qquad \text{Regret}_T(u) \le \widetilde{O}(\sqrt{T(1 + \ln N - H(u))}) = \widetilde{O}\left(\sqrt{T\left(1 + D\left(u \middle\| \tfrac{1}{N}\mathbf{1}\right)\right)}\right) \,. \tag{5}$$

Algorithms of this type are called *parameter-free* since, in contrast to Hedge, they do not need to know p. The regret in (4) is called a *quantile bound* because one can bound the regret with respect to (ϵN)-th best expert for any $\epsilon \in (0,1)$. Indeed, regret with respect to (ϵN)-th best expert is upper bounded by the regret with respect to the average of the top ϵN experts, which can be expressed as the regret with respect to a competitor u that, up to permutation of coordinates, has the form $u = (1/(\epsilon N), \dots, 1/(\epsilon N), 0, \dots, 0)$. Such competitor satisfies $H(u) = \ln(\epsilon N)$ and the regret with respect to any such u is $\widetilde{O}(\sqrt{T(1 + \ln(1/\epsilon)})$. Namely, the last bound does not depend on the number of experts N, only on the quantile ϵ. These algorithms remove the first drawback of Hedge.

1. Earlier papers have extra logarithmic factors. Foster et al. (2015); Orabona and Pál (2016a) have a fixed multiplicative constant hidden in $\widetilde{O}(\cdot)$

The second drawback of Hedge is removed by the AdaHedge algorithm due to de Rooij et al. (2014); see also Orabona and Pál (2016a). AdaHedge lifts the assumption $\ell_t \in [-1,1]^N$. Namely, for any sequence of loss vectors $\{\ell_t\}_{t=1}^{\infty}$, $\ell_t \in \mathbb{R}^N$, any $T \geq 0$ and any $u \in \Delta_N$, AdaHedge satisfies $\text{Regret}_T(u) \leq 5.3\sqrt{\ln N \sum_{t=1}^{T} \|\ell_t\|_\infty^2}$. AdaHedge is *scale-free* which means that its predictions x_t are the same for $\{\ell_t\}_{t=1}^{\infty}$ and $\{c\ell_t\}_{t=1}^{\infty}$, where c is any positive constant.

Our first open problem is to design an algorithm that combines the advantages of parameter-free and scale-free algorithms. More formally: *Does there exist a universal constant $C > 0$ and for each $N \geq 2$ and each $\pi \in \Delta_N$, is there an algorithm (without any tuning parameters) that runs in $O(N)$ time per round and satisfies, for any sequence of loss vectors $\{\ell_t\}_{t=1}^{\infty}$, $\ell_t \in \mathbb{R}^N$,*

$$\forall T \geq 0 \quad \forall u \in \Delta_N \qquad \text{Regret}_T(u) \leq C\sqrt{(1 + D(u\|\pi)) \sum_{t=1}^{T} \|\ell_t\|_\infty^2} \ ? \tag{6}$$

Attempts to solve the open problem include all sorts of doubling tricks. For example, the algorithm could keep an upper bound B_t on $\max_{1 \leq i \leq t} \|\ell_i\|_\infty$, for example, of the form $B_t = \|\ell_1\| 2^k$ doubling whenever larger $\|\ell_t\|_\infty$ is observed. However, resulting regret bounds might depend on $\log_2\left(\frac{\max_{1\leq t\leq T}\|\ell_t\|_\infty}{\|\ell_1\|_\infty}\right)$ which can be $\Omega(T)$ in the worst-case.

A more promising approach is to use infinitely many copies of AdaHedge with learning rates 2^{-i}, $i = 1, 2, \ldots$, and combine their predictions using a master AdaHedge with a certain non-uniform prior distribution. However, the resulting algorithm runs in $\Theta(N \max_{u \in \Delta_N} \log D(u\|\pi))$ time per round.

3. OLO over Hilbert Spaces

The situation with algorithms for OLO over a Hilbert space $\mathcal{H}$ is very similar to that of LEA. FTRL with regularizer $\frac{1}{2}\|u\|^2$ and learning rate η satisfies (cf. Theorem 1)

$$\forall u \in \mathcal{H} \qquad \text{Regret}_T(u) \leq \frac{\|u\|^2}{2\eta} + \frac{\eta T}{2}, \tag{7}$$

assuming that $\|\ell_1\|, \|\ell_2\|, \ldots, \|\ell_T\| \leq 1$. Bound (7) is a direct analogue of (2) for LEA. A simple choice $\eta = 1/\sqrt{T}$ leads to an algorithm that satisfies

$$\text{Regret}_T(u) \leq \frac{1}{2}\left(1 + \|u\|^2\right)\sqrt{T}\ . \tag{8}$$

However, this algorithm and the bound (8) have two drawbacks. First, the dependency on $\|u\|$ is suboptimal. As we will see shortly, the quadratic dependency can be replaced by an (almost) linear dependency. Second, the bound holds only for sequences of loss vectors with $\|\ell_t\| \leq 1$, $t = 1, 2, \ldots, T$. A robust algorithm should be able to handle any sequence of loss vectors in $\mathcal{H}$.

Starting from (7), if we choose learning rate $\eta = D/\sqrt{T}$, we get a family of algorithms parametrized by $D \in [0, \infty)$ that satisfy an analogue of the bound (3):

$$\forall u \in \mathcal{H} \qquad \|u\| \leq D \quad \Longrightarrow \quad \text{Regret}_T(u) \leq D\sqrt{T}\ . \tag{9}$$

Instead of family of algorithms parametrized by $D \in [0, \infty)$ satisfying bound (9), one *would like to have* a single algorithm (without any tuning parameters) satisfying

$$\forall u \in \mathcal{H} \qquad \text{Regret}_T(u) \le \|u\| \sqrt{T} \,. \tag{10}$$

Bound (10) is an analogue of (4) for LEA. Similar to LEA, (10) is stronger than (9) in the following sense: A single algorithm satisfying (10) implies (9) for all values of $D \in [0, \infty)$. However, a family of algorithms $\{A_D : D \in [0, \infty)\}$ parametrized by D where A_D satisfies (9), does not yield a single algorithm that satisfies (10). Finally, note that (10) has better dependency on $\|u\|$ than (8).

Similar to LEA, there have been a lot of work on algorithms (Streeter and McMahan, 2012; Orabona, 2013; McMahan and Abernethy, 2013; McMahan and Orabona, 2014; Orabona, 2014) that satisfy a slightly weaker version of (10)

$$\forall u \in \mathcal{H} \qquad \text{Regret}_T(u) \le \big(O(1) + \text{polylog}(1 + \|u\|) \, \|u\| \,\big) \sqrt{T} \,, \tag{11}$$

where $\text{polylog}(1 + \|u\|)$ represents a function that is upper bounded by a polynomial in $\log(1 + \|u\|)$.[2] Algorithms satisfying (11) are called *parameter-free*, since they do not need to know D. Moreover, the bound (11) has much better dependency on $\|u\|$ than (8).

Analogue of AdaHedge for OLO over Hilbert space is FTRL with adaptive learning rate $\eta_t = 1/\sqrt{\sum_{i=1}^{t-1} \|\ell_i\|^2}$. Orabona and Pál (2015) (see also Orabona and Pál (2016b)) showed that the resulting algorithm is scale-free and, for any sequence of loss vectors $\{\ell_t\}_{t=1}^{\infty}$, $\ell_t \in \mathcal{H}$, it satisfies

$$\forall T \ge 0 \quad \forall u \in \mathcal{H} \qquad \text{Regret}_T(u) \le \left(6.25 + \frac{1}{2} \|u\|^2 \right) \sqrt{T} \max_{1 \le t \le T} \|\ell_t\| \,.$$

We stress that the algorithm does *not* need to know $\max_{1 \le t \le T} \|\ell_t\|$.

Our second open problem is to design an algorithm that combines the advantages of parameter-free and scale-free algorithms for OLO over a Hilbert space $\mathcal{H}$. Formally: *Construct an algorithm (without any tuning parameters) that makes $O(1)$ vector operations in $\mathcal{H}$ and $O(1)$ other operations per round, and satisfies for any sequence of loss vectors $\{\ell_t\}_{t=1}^{\infty}$, $\ell_t \in \mathcal{H}$,*

$$\forall T \ge 0 \;\; \forall u \in \mathcal{H} \;\; \text{Regret}_T(u) \le \big(O(1) + \text{polylog}(1 + \|u\|) \, \|u\| \,\big) \sqrt{\sum_{t=1}^{T} \|\ell_t\|^2}.$$

We offer \$100 USD for a positive solution of this problem. Here $\text{polylog}(1 + \|u\|)$ represents a function that is upper bounded by a polynomial in $\log(1 + \|u\|)$ and $O(1)$ represents a universal constant.

The problem is interesting even for one-dimensional Hilbert space $\mathcal{H} = \mathbb{R}$. Also, the reductions in Orabona and Pál (2016a) implies that solving the one-dimensional problem might solve both the general Hilbert space and the LEA cases.

Similar to LEA, attempts to solve the open problem included a doubling trick on the maximum norm $\|\ell_t\|$ seen so far. This approach fails the same way as for LEA. Differently from the LEA case, running infinitely many copies of FTRL and combining their predictions using AdaHedge with a non-uniform prior is problematic because the predictions of the copies are not bounded. The open problem for Hilbert space seems to be harder than the open problem than LEA, and constructing even a computationally inefficient algorithm is an interesting open problem.

2. It can be shown that for OLO over Hilbert space the extra poly-logarithmic factor is necessary (McMahan and Abernethy, 2013; Orabona, 2013).

References

N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.

K. Chaudhuri, Y. Freund, and D. Hsu. A parameter-free hedging algorithm. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22 (NIPS 2009), December 7-12, Vancouver, BC, Canada*, pages 297–305, 2009.

A. Chernov and V. Vovk. Prediction with advice of unknown number of experts. In Peter Grünwald and Peter Spirtes, editors, *Proc. of the 26th Conf. on Uncertainty in Artificial Intelligence (UAI 2010), July 8-11, Catalina Island, California, USA*. AUAI Press, 2010.

S. de Rooij, T. van Erven, P. D. Grünwald, and W. M. Koolen. Follow the leader if you can, hedge if you must. *The Journal of Machine Learning Research*, 15(1):1281–1316, 2014.

D. J. Foster, A. Rakhlin, and K. Sridharan. Adaptive online learning. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3375–3383. Curran Associates, Inc., 2015.

Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

W. M. Koolen and T. van Erven. Second-order quantile methods for experts and combinatorial games. In Peter Grünwald, Elad Hazan, and Satyen Kale, editors, *Proc. of the 28th Conf. on Learning Theory (COLT 2015), July 3-6, Paris, France*, pages 1155–1175, 2015.

H. Luo and R. E. Schapire. A drifting-games analysis for online learning and applications to boosting. In Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 1368–1376. Curran Associates, Inc., 2014.

H. Luo and R. E. Schapire. Achieving all with no parameters: AdaNormalHedge. In Peter Grünwald, Elad Hazan, and Satyen Kale, editors, *Proc. of the 28th Conf. on Learning Theory (COLT 2015), July 3-6, Paris, France*, pages 1286–1304, 2015.

H. B. McMahan and J. Abernethy. Minimax optimal algorithms for unconstrained linear optimization. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26 (NIPS 2013)*, pages 2724–2732, 2013.

H. B. McMahan and F. Orabona. Unconstrained online linear learning in Hilbert spaces: Minimax algorithms and normal approximations. In Vitaly Feldman, Csaba Szepesvri, and Maria Florina Balcan, editors, *Proc. of The 27th Conf. on Learning Theory (COLT 2014), June 13-15, Barcelona, Spain*, pages 1020–1039, 2014.

F. Orabona. Dimension-free exponentiated gradient. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 1806–1814. Curran Associates, Inc., 2013.

F. Orabona. Simultaneous model selection and optimization through parameter-free stochastic learning. In *Advances in Neural Information Processing Systems 27*, 2014.

F. Orabona and D. Pál. Scale-free algorithms for online linear optimization. In Kamalika Chaudhuri and Claudio Gentile, editors, *Proc. of Algorithmic Learning Theory (ALT) 2015, October 4-6, Banff, AB, Canada*, pages 287–301. Springer, 2015.

F. Orabona and D. Pál. From coin betting to parameter-free online learning, 2016a. Available from: http://arxiv.org/pdf/1602.04128.pdf.

F. Orabona and D. Pál. Scale-free online learning, 2016b. Submitted to Theoretical Computer Science, ALT 2015 Special Issue, arXiv:1601.01974, Available from: http://arxiv.org/pdf/1601.01974.pdf.

S. Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2011.

M. Streeter and B. McMahan. No-regret algorithms for unconstrained online convex optimization. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25 (NIPS 2012), Decmber 3-8, Lake Tahoe, Nevada, USA*, pages 2402–2410, 2012.