Beyond Logarithmic Bounds in Online Learning

Francesco Orabona[*]
Toyota Technological Institute
Chicago, IL, USA
francesco@orabona.com

Nicolò Cesa-Bianchi
DSI, Università degli Studi di Milano
Milano, Italy
nicolo.cesa-bianchi@unimi.it

Claudio Gentile
DICOM, Università dell'Insubria
Varese, Italy
claudio.gentile@uninsubria.it

Abstract

We prove logarithmic regret bounds that depend on the loss L_T^* of the competitor rather than on the number T of time steps. In the general online convex optimization setting, our bounds hold for any smooth and exp-concave loss (such as the square loss or the logistic loss). This bridges the gap between the $\mathcal{O}(\ln T)$ regret exhibited by exp-concave losses and the $\mathcal{O}(\sqrt{L_T^*})$ regret exhibited by smooth losses. We also show that these bounds are tight for specific losses, thus they cannot be improved in general. For online regression with square loss, our analysis can be used to derive a sparse randomized variant of the online Newton step, whose expected number of updates scales with the algorithm's loss. For online classification, we prove the first logarithmic mistake bounds that do not rely on prior knowledge of a bound on the competitor's norm.

1 Introduction

Online convex optimization (e.g., [Zinkevich, 2003, Hazan et al., 2007]) is an abstract sequential prediction problem where, at each time step, the learner chooses a point from a fixed convex set $\mathcal{S}$ and then observes a convex loss function defined on the same set $\mathcal{S}$. The value of the function on the chosen point is viewed as the learner's instantaneous loss. The goal of the learner is to minimize regret, i.e., the difference between the learner's cumulative loss and the cumulative loss of the single best point in $\mathcal{S}$. Many problems such as prediction with expert advice, sequential investment, and online regression/classification can be viewed as special cases of this general framework.

In this work, we focus on the case where the convex set $\mathcal{S}$ is a bounded and closed subset of the d-dimensional Euclidean space. We also assume that the loss functions are differentiable and have uniformly bounded gradients over $\mathcal{S}$. In this setting it is well known that, in the absence of further conditions, the regret must grow at least as $\sqrt{T}$, where T is the total number of time steps. Moreover, a strategy as simple as Online Gradient Descent (OGD) is able to achieve this optimal rate. If each loss function observed by the learner is not only convex but strongly convex, then OGD (with proper tuning) achieves a regret that grows only logarithmically with time. This logarithmic rate cannot be improved for strongly convex losses [Abernethy et al., 2008]. Strong convexity is not necessary to achieve logarithmic regret. Indeed, more complex algorithms, such as Online Newton Step (ONS), have logarithmic regret over any sequence of exp-concave loss functions, a wider class than strongly convex functions.

A different way of constraining the loss functions is through the notion of smoothness [Srebro et al., 2010] (or self-boundedness [Shalev-Shwartz, 2007], see also the "subquadratic pairs" of Cesa-Bianchi and Lugosi [2006]). If the losses are smooth, then the regret has the form $D^2 + \sqrt{L_T^*}$, where L_T^* is the cumulative loss of the best point in $\mathcal{S}$ over the first T prediction steps and D is an upper bound on the diameter of $\mathcal{S}$. If L_T^* grows slowly enough this rate can be better than the logarithmic rate achieved by exp-concave losses. Put another way, the average per-step regret for smooth losses vanishes at a rate between $T^{-1/2}$ and T^{-1}, depending on the way L_T^* grows with time.

A natural question to ask is whether adding the assumption of smoothness may improve the regret for exp-concave losses as it does for regular convex losses. This question was left as an open problem by Vovk [2001] for the special case of online regression with square loss, which is smooth and exp-concave under reasonable assumptions on the domain. In that paper, Vovk introduced an algorithm for online linear regression in $\mathbb{R}^d$ with regret $D^2 + d \ln T$ (see also [Azoury and Warmuth, 2001] for a related result), and asked whether it is possible to bridge the gap between this rate and the rate $D^2 + \sqrt{L_T^*}$. In this paper, we indeed prove

[*] Work done while at Università degli Studi di Milano.

Appearing in Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS) 2012, La Palma, Canary Islands. Volume 22 of JMLR: W&CP 22.

that for any sequence of smooth and exp-concave losses, ONS has a regret of the order of $D^2 + \ln(1 + L_T^*)$. Thus, ONS has *constant* regret when $L_T^* = 0$ while the regret of Vovk's algorithm is bounded by $\mathcal{O}\big(\ln T\big)$. On the other hand, whereas ONS regret grows at most logarithmically (because $L_T^* = \mathcal{O}(T)$ anyway), OGD's regret can only be bounded by $\mathcal{O}\big(\sqrt{T}\big)$.

Our analysis cannot be improved in general: by extending an argument due to Vovk [2001], we also prove a matching non-asymptotic lower bound for the square loss that holds for any given $L \ge L_T^*$.

We then apply our techniques to derive a sparse randomized variant of ONS for regression with square loss. This variant performs, in expectation, a number of updates scaling linearly with the loss of ONS which, in turn, is essentially linear in the loss of the best competitor in $\mathcal{S}$.

In the last part of the paper we investigate logarithmic bounds for online classification. For this setting we obtain a bound that depends logarithmically of a new convex classification loss, smoothly interpolating between the hinge loss and the squared hinge loss. Unlike all previous bounds of similar form, our bound does not depend on previous knowledge on the norm of the competing hyperplane.

We finally mention that a paper similar in spirit to ours is [Hazan and Kale, 2009], where the authors show regret bounds depending logarithmically on the *variance* of the side information used to define the loss sequence. In the regression case, this corresponds to a bound that depends on the variance of the instance vectors $\boldsymbol{x}_t$, rather than on the loss of the competitor, as we do here.

2 Definitions

Fix a convex and closed subset $\mathcal{S} \subseteq \mathbb{R}^d$. At each step $t = 1, 2, \ldots$ of the online convex optimization protocol, the learner chooses $\boldsymbol{w}_t \in \mathcal{S}$ and then receives information about a convex and differentiable loss function $\ell_t : \mathcal{S} \to \mathbb{R}$. We assume that at each step t the loss $\ell_t(\boldsymbol{w}_t)$ and the loss gradient $\nabla\ell_t(\boldsymbol{w}_t)$ are both revealed to the learner. The goal is to control the regret

$$R_T(\boldsymbol{u}) = \sum_{t=1}^{T} \ell_t(\boldsymbol{w}_t) - \sum_{t=1}^{T} \ell_t(\boldsymbol{u})$$

uniformly over the horizon T and for all $\boldsymbol{u} \in \mathcal{S}$.

Our first set of results apply to loss functions that satisfy two conditions. The first one is exp-concavity: a loss function ℓ_t is c-exp-concave if $\exp(-c\ell_t)$ is a concave function, namely, the Hessian $\nabla^2 \exp\big(-c\ell_t(\boldsymbol{w})\big)$ is negative semidefinite for all $\boldsymbol{w} \in \mathcal{S}$. For example, functions that have the norm of the gradients upper bounded by G and the eigenvalues of the Hessian lower bounded by $H > 0$ are c-exp-concave for any $c \le \frac{H}{G^2}$ (see, e.g., [Hazan et al., 2007]). The second condition is smoothness: a loss function ℓ_t is H-smooth if there exists $H > 0$ such that $\left\|\nabla\ell_t(\boldsymbol{w})\right\|^2 \le 4H\ell_t(\boldsymbol{w})$ for all $\boldsymbol{w} \in \mathcal{S}$. It has been shown by Srebro et al. [2010] that a sufficient condition for smoothness of non-negative functions is the Lipschitzness of the gradient. That is, if $\|\nabla\ell_t(\boldsymbol{w}) - \nabla\ell_t(\boldsymbol{w}')\| \le H\,\|\boldsymbol{w} - \boldsymbol{w}'\|$ for all $\boldsymbol{w}, \boldsymbol{w}' \in \mathcal{S}$, then ℓ_t is H-smooth.

These conditions simplify when the losses ℓ_t can be written as $\ell_t(\boldsymbol{w}) = g_t\big(\boldsymbol{w}^\top\boldsymbol{x}_t\big)$ for some $g_t : \mathbb{R} \to \mathbb{R}$ and $\boldsymbol{x}_t \in \mathbb{R}^d$. In this case we have that $\nabla^2 \exp\big(-c\ell_t(\boldsymbol{w})\big) = f''(z)\,\boldsymbol{x}_t\boldsymbol{x}_t^\top$, where $z = \boldsymbol{w}^\top\boldsymbol{x}_t$ and $f(z) = \exp\big(-cg_t(z)\big)$. Thus the Hessian has rank one and the only eigenvalue is $f''(z)\|\boldsymbol{x}_t\|^2$. Therefore, verifying exp-concavity for such losses amounts to checking that $f''(z) \le 0$ for all $z = \boldsymbol{w}^\top\boldsymbol{x}_t$. Similarly, for H-smooth we have that $\nabla\ell_t(\boldsymbol{w}) = g_t'(z)\,\boldsymbol{x}_t$. Hence if $g_t'(z)$ is L-Lipschitz, then $\ell_t(\boldsymbol{w})$ is $(L\|x_t\|)$-smooth. An example of a function of this form that is both exp-concave and smooth is the square loss, $\ell_t(\boldsymbol{w}) = \big(\boldsymbol{w}^\top\boldsymbol{x}_t - y_t\big)^2$. Under the assumption $\left\|\boldsymbol{x}_t\right\|^2 \le H$, the square loss is naturally smooth. Moreover, under the assumption $y_t, \boldsymbol{w}^\top\boldsymbol{x}_t \in [-Y, Y]$ we have $\nabla^2 \exp\big(-c\ell_t(\boldsymbol{w})\big) \le 0$ for $c \le \frac{1}{8Y^2}$, see [Vovk, 2001, Remark 3]. Hence the square loss is also exp-concave. Another example is the logistic loss $\ell_t(\boldsymbol{w}) = \ln\big(1 + \exp(-\boldsymbol{w}^\top\boldsymbol{x}_t)\big)$. Indeed, it is easy to verify that this loss is exp-concave and smooth whenever $\boldsymbol{w}^\top\boldsymbol{x}_t \ge 0$ and $\max_t \|\boldsymbol{x}_t\|$ is bounded.

Throughout the rest of the paper, we use ∇_t as shorthand for $\nabla\ell_t(\boldsymbol{w}_t)$.

3 Regression and sparse regression

In this section we present three results. We show that the Online Newton Step (ONS) strategy of Hazan et al. [2007] (Algorithm 1) has regret bounds that are logarithmic in the loss of the competitor for any sequence of exp-concave and smooth losses. We then specialize the above result to the square loss. In particular, in the statistical setting with fixed design we derive a bound on the expected cumulative regret which is logarithmic in the cumulative variance of the noise. Finally, we show that the same machinery used to prove square loss results can be adapted to analyze a *sparse*[1] linear regression algorithm. This is a simple randomized algorithm whose logarithmic cumulative regret is achieved by updating the algorithm's internal state a number of times which scales with the total loss of the algorithm itself. This can be viewed as a natural regression counterpart to mistake driven algorithms for classification, where the total number of updates equals the total number of prediction mistakes of the algorithm —see also Section 5.

[1] Sparsity here refers to the dual variable representation of the learned regression function rather than the number of nonzero coefficients of the best offline comparator.

Algorithm 1 Online Newton Step (ONS) Algorithm

```
Input: $\alpha > 0, \beta > 0$.
Initialize: $\boldsymbol{w}'_1 = \boldsymbol{0}, A_1 = \alpha I$
for $t = 1, 2, \ldots, T$ do
    Receive $\mathcal{S}_t$
    $\boldsymbol{w}_t = \operatorname*{argmin}_{\boldsymbol{v} \in \mathcal{S}_t} (\boldsymbol{w}'_t - \boldsymbol{v})^\top A_t (\boldsymbol{w}'_t - \boldsymbol{v})$
    Suffer loss $\ell_t(\boldsymbol{w}_t)$
    $\boldsymbol{w}'_{t+1} = \boldsymbol{w}_t - \frac{1}{\beta} A_t^{-1} \nabla_t$
    $A_{t+1} = A_t + \nabla_t \nabla_t^\top$
end for
```

Theorem 1. *Let $\mathcal{S} \equiv \{\boldsymbol{u} \in \mathbb{R}^d : \|\boldsymbol{u}\| \le U\}$. If for all $t = 1, 2, \ldots$ each loss function $\ell_t : \mathbb{R}^d \to \mathbb{R}$ satisfies the following:*

1. *ℓ_t is H-smooth and c-exp-concave*
2. *$\min_{\boldsymbol{v} \in \mathcal{S}} \ell_t(\boldsymbol{v}) > -\infty$*
3. *$\max_{\boldsymbol{v} \in \mathcal{S}} \|\nabla \ell_t(\boldsymbol{v})\| \le G$*

then, for all $\boldsymbol{u} \in \mathcal{S}$, the regret of ONS Algorithm 1 run with $\beta \le \frac{1}{2} \min\{\frac{1}{8GU}, c\}$, $\alpha \ge 0$, and $\mathcal{S}_t \equiv \mathcal{S}$, satisfies

$$
\begin{aligned}
R_T \le{}& \frac{\alpha\,\beta}{2}\|u\|^2 \\
&+ \frac{d}{2\beta} \ln\left(\frac{8H}{\alpha d} \sum_{t=1}^T \left(\ell_t(\boldsymbol{u}) - \min_{\boldsymbol{v} \in \mathcal{S}} \ell_t(\boldsymbol{v})\right)\right. \\
&\left. + \frac{4H}{\alpha\,\beta} \ln \frac{4H}{e\alpha\,\beta} + \frac{4H\,\beta}{d}\|u\|^2 + 2\right) . \qquad (1)
\end{aligned}
$$

Proof. Set $d_t(\boldsymbol{u}, \boldsymbol{w}) = (\boldsymbol{w} - \boldsymbol{u})^\top A_t (\boldsymbol{w} - \boldsymbol{u})$, where A_t is as in Algorithm 1. From the proof of [Hazan et al., 2007, Theorem 2] one can extract the following inequality:

$$
\begin{aligned}
&\nabla_t^\top(\boldsymbol{w}_t - \boldsymbol{u}) - \frac{\beta}{2}\left(\nabla_t^\top(\boldsymbol{w}_t - \boldsymbol{u})\right)^2 \\
&\le \frac{1}{2\beta} \nabla_t^\top A_{t+1}^{-1} \nabla_t + \frac{\beta}{2}\left(d_t(\boldsymbol{u}, \boldsymbol{w}_t) - d_{t+1}(\boldsymbol{u}, \boldsymbol{w}_{t+1})\right) .
\end{aligned}
$$

Summing over time we have

$$
\begin{aligned}
&\sum_{t=1}^T \left(\nabla_t^\top(\boldsymbol{w}_t - \boldsymbol{u}) - \frac{\beta}{2}\left(\nabla_t^\top(\boldsymbol{w}_t - \boldsymbol{u})\right)^2\right) \\
&\qquad \le \frac{1}{2\beta} \sum_{t=1}^T \nabla_t^\top A_{t+1}^{-1} \nabla_t + \frac{\beta}{2} \alpha \|u\|^2. \qquad (2)
\end{aligned}
$$

Now we could use [Hazan et al., 2007, Lemma 11] to get

$$
\sum_{t=1}^T \nabla_t^\top A_{t+1}^{-1} \nabla_t \le d \ln\left(1 + T \frac{\max_t \|\nabla_t\|^2}{\alpha}\right)
$$

but it is easy to show that we can bound the same sum with the tighter upper bound

$$
d \ln\left(1 + \sum_{t=1}^T \frac{\|\nabla_t\|^2}{d\alpha}\right) . \qquad (3)
$$

Define $\widetilde{\ell}_t := \ell_t - \min_{\boldsymbol{v} \in \mathcal{S}} \ell_t(\boldsymbol{v})$. Noting that $\nabla \widetilde{\ell}_t(\boldsymbol{w}_t) = \nabla_t$, we use [Srebro et al., 2010, Lemma 3.1] on the non-negative functions $\widetilde{\ell}_t$ and obtain $\|\nabla_t\|^2 \le 4H\widetilde{\ell}_t(\boldsymbol{w}_t)$. Hence, using [Hazan et al., 2007, Lemma 3], we get

$$
\begin{aligned}
&\sum_{t=1}^T \widetilde{\ell}_t(\boldsymbol{w}_t) - \sum_{t=1}^T \widetilde{\ell}_t(\boldsymbol{u}) \\
&\quad \le \sum_{t=1}^T \left(\nabla_t^\top(\boldsymbol{w} - \boldsymbol{u}) - \frac{\beta}{2}\left(\nabla_t^\top(\boldsymbol{w} - \boldsymbol{u})\right)^2\right) \\
&\quad \le \frac{d}{2\beta} \ln\left(1 + \frac{4H}{\alpha d} \sum_{t=1}^T \widetilde{\ell}_t(\boldsymbol{w}_t)\right) + \frac{\alpha\,\beta}{2}\|\boldsymbol{u}\|^2 .
\end{aligned}
$$

Using Corollary 5 in the Appendix with $n = 2$ yields the stated bound. □

Note that the algorithm and the bound are invariant to loss shifts. In fact, the terms $\ell_t(\boldsymbol{u}) - \min_{\boldsymbol{v} \in \mathcal{S}} \ell_t(\boldsymbol{v})$ in the logarithm can be viewed as shifting the loss function ℓ_t so as its minimal value is zero.

3.1 Square loss

Special consideration deserves the square loss $\ell_t(\boldsymbol{v}) = (\boldsymbol{v}^\top \boldsymbol{x}_t - y_t)^2$. Although Theorem 1 readily applies to this loss, we show how to obtain a tighter bound by applying a direct argument to ONS run with the choice of β shown in the next lemma. The same technique is also useful to derive the sparse regression bound contained in Section 3.2.

Lemma 1. *For any $\boldsymbol{u}, \boldsymbol{w} \in \mathbb{R}^d$, and $\beta > 0$ such that $\ell_t(\boldsymbol{w}) \le \frac{1}{2\beta}$ we have that*

$$
\ell_t(\boldsymbol{w}) - \ell_t(\boldsymbol{u}) \le \nabla_t^\top(\boldsymbol{w} - \boldsymbol{u}) - \frac{\beta}{2}\left(\nabla_t^\top(\boldsymbol{w} - \boldsymbol{u})\right)^2 .
$$

Proof. The statement results from the following chain of elementary inequalities:

$$
\begin{aligned}
&\ell_t(\boldsymbol{w}) - \ell_t(\boldsymbol{u}) \\
&= (\boldsymbol{w}^\top \boldsymbol{x}_t)^2 - 2y_t \boldsymbol{w}^\top \boldsymbol{x}_t - (\boldsymbol{u}^\top \boldsymbol{x}_t)^2 + 2y_t \boldsymbol{u}^\top \boldsymbol{x}_t \\
&= 2(\boldsymbol{w}^\top \boldsymbol{x}_t - y_t)\boldsymbol{x}_t^\top(\boldsymbol{w} - \boldsymbol{u}) - (\boldsymbol{w}^\top \boldsymbol{x}_t - \boldsymbol{u}^\top \boldsymbol{x}_t)^2 \qquad (4) \\
&\le 2(\boldsymbol{w}^\top \boldsymbol{x}_t - y_t)\boldsymbol{x}_t^\top(\boldsymbol{w} - \boldsymbol{u}) - 2\beta(\boldsymbol{w}^\top \boldsymbol{x}_t - \boldsymbol{u}^\top \boldsymbol{x}_t)^2 \ell_t(\boldsymbol{w}) \\
&= \nabla_t^\top(\boldsymbol{w} - \boldsymbol{u}) - \frac{\beta}{2}\left(\nabla_t^\top(\boldsymbol{w} - \boldsymbol{u})\right)^2 .
\end{aligned}
$$

□

This lemma implies the bound of Theorem 1, provided an upper bound on the maximum loss $\max_t \ell_t(\boldsymbol{w}_t)$ can be established. For this reason, we state the following corollary of Theorem 1 for square loss using two different projection strategies, both aimed at bounding $\ell_t(\boldsymbol{w}_t)$.

Corollary 1. *Assume for all t the loss function $\ell_t(\boldsymbol{v}) = (\boldsymbol{v}^\top \boldsymbol{x}_t - y_t)^2$ is such that $\|\boldsymbol{x}_t\| \le R$, and $|y_t| \le Y$. Let ONS Algorithm 1 be run with $\alpha > 0$,*

1. *if $(UR+Y)^2 \le \frac{1}{2\beta}$ and $\mathcal{S}_t \equiv \mathcal{S} \equiv \{\boldsymbol{v} : \|\boldsymbol{v}\| \le U\}$, then (1) holds for any $\boldsymbol{u} \in \mathcal{S}$;*
2. *if $(\widetilde{Y} + Y)^2 \le \frac{1}{2\beta}$ and $\mathcal{S}_t \equiv \left\{\boldsymbol{v} : |\boldsymbol{v}^\top \boldsymbol{x}_t| \le \widetilde{Y}\right\}$, then (1) holds for any $\boldsymbol{u} \in \mathcal{S} \equiv \bigcap_t \mathcal{S}_t$.*

Note that the both types of projections can be efficiently calculated, as shown in [Hazan et al., 2007] and [Dekel et al., 2010].

Using Part 1 of Corollary 1, $\alpha = \frac{4R^2}{\beta}$, and $\beta = \frac{1}{2(UR+Y)^2}$, the upper bound on the regret becomes

$$R_T(\boldsymbol{u}) \le 2R^2\|\boldsymbol{u}\|^2 + d(UR+Y)^2 \times \ln\left(\frac{2R^2\|\boldsymbol{u}\|^2 + \sum_{t=1}^T \ell_t(\boldsymbol{u})}{d(UR+Y)^2} + 1\right). \qquad (5)$$

Using a different algorithm, and a different analysis, Vovk [2001] obtains $R_T(\boldsymbol{u}) \le \|\boldsymbol{u}\|^2 + dY^2 \ln\big(1 + TR^2/d\big)$. We see that our bound has a logarithmic dependence on the loss of the competitor, while Vovk's bound depends on T. On the other hand, in (5) the multiplicative factor of the logarithm is $(UR+Y)^2$, which is strictly bigger than Y^2. Note also that our algorithm requires prior knowledge of U. Overall, our bound is better when T is large and the loss of the competitor grows sublinearly.

We now briefly consider statistical regression with fixed design. In this setting the labels y_t are random variables $Y_t = \boldsymbol{u}^\top \boldsymbol{x}_t + Q_t$, where $\boldsymbol{u} \in \mathbb{R}^d$ parameterizes the underlying linear regression function and the Q_t are zero-mean random variables with bounded variance $\mathrm{Var}[Q_t]$. We prove a bound on the expected regret that depends logarithmically on the cumulative variance of the noise Q_t. We are not aware of similar bounds for regression with fixed design. In the next section, we also prove a lower bound for this setting. Note that we do not require the Q_t to be independent for the bound to hold.

Corollary 2. *Let $\mathcal{S} \equiv \left\{\boldsymbol{u} \in \mathbb{R}^d : \|\boldsymbol{u}\| \le U\right\}$. Assume for all t the loss function $\ell_t(\boldsymbol{v}) = (\boldsymbol{v}^\top \boldsymbol{x}_t - Y_t)^2$ is such that $\|\boldsymbol{x}_t\| \le R$ and $Y_t = \boldsymbol{u}^\top \boldsymbol{x}_t + Q_t$ for some $\boldsymbol{u} \in \mathcal{S}$, where Q_t are zero-mean random variables such that*

$$\mathrm{Var}\big[Q_t \,\big|\, Q_1, \ldots, Q_{t-1}\big] \le V \qquad t = 1, 2, \ldots$$

Then the regret of ONS Algorithm 1 run with $\alpha = \frac{4R^2}{\beta}$, $\beta = \frac{1}{8U^2R^2+2V}$, and $\mathcal{S}_t \equiv \mathcal{S}$ satisfies

$$\mathbb{E}[R_T] \le 2R^2\|\boldsymbol{u}\|^2 + d(4U^2R^2+V)\ln\left(\frac{2R^2\|\boldsymbol{u}\|^2 + \sum_{t=1}^T \mathrm{Var}[Q_t]}{d(4U^2R^2+V)} + 1\right).$$

Algorithm 2 Sparse Online Newton Step (SONS) Algorithm

```
Input: $\alpha > 0, \beta > 0$.
Initialize: $\boldsymbol{w}'_1 = \boldsymbol{0}, A_1 = \alpha I$
for $t = 1, 2, \ldots, T$ do
    Receive $\boldsymbol{x}_t$
    $\boldsymbol{w}_t = \operatorname*{argmin}_{\boldsymbol{v} \in \mathcal{S}_t} (\boldsymbol{w}'_t - \boldsymbol{v})^\top A_t (\boldsymbol{w}'_t - \boldsymbol{v})$
    Suffer loss $\ell_t(w_t) = (\boldsymbol{w}_t^\top \boldsymbol{x}_t - y_t)^2$
    $B_t = \begin{cases} \frac{1}{\beta \ell_t(w_t)} & \text{with probability } 2\beta\ell_t(w_t) \\ 0 & \text{with probability } 1 - 2\beta\ell_t(w_t) \end{cases}$
    $\boldsymbol{w}'_{t+1} = \boldsymbol{w}_t - \frac{B_t}{\beta} A_t^{-1} \boldsymbol{x}_t$
    $A_{t+1} = A_t + B_t^2 \, \boldsymbol{x}_t \boldsymbol{x}_t^\top$
end for
```

Proof. Define $e_t = (\boldsymbol{w}_t^\top \boldsymbol{x}_t - \boldsymbol{u}^\top \boldsymbol{x}_t)^2$ and observe that $e_t \le 4R^2U^2$. Let $\mathbb{E}_t = \mathbb{E}[\,\cdot\,|Q_1, \ldots, Q_{t-1}]$ and $\mathrm{Var}_t = \mathrm{Var}[\,\cdot\,|Q_1, \ldots, Q_{t-1}]$. We have that $\mathbb{E}_t\left[\ell_t(\boldsymbol{w}_t)\right] = e_t + \mathrm{Var}_t[Q_t]$ and $\mathbb{E}_t\left[\ell_t(\boldsymbol{u})\right] = \mathrm{Var}_t[Q_t]$. Hence, using our choice of β,

$$\begin{aligned} \mathbb{E}_t\left[\ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{u})\right] = e_t &\le 2e_t\big(1 - \beta(4R^2U^2 + V)\big) \\ &\le 2e_t\big(1 - \beta(e_t + \mathrm{Var}_t[Q_t])\big) \\ &= \nabla_t^\top(\boldsymbol{w}_t - \boldsymbol{u}) - \frac{\beta}{2}\Big(\nabla_t^\top(\boldsymbol{w}_t - \boldsymbol{u})\Big)^2. \end{aligned}$$

Similarly to the proof of Corollary 1, we get

$$\begin{aligned} \mathbb{E}[R_T] &\le \mathbb{E}\left[\sum_{t=1}^T \mathbb{E}_t\big[\ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{u})\big]\right] \\ &\le \mathbb{E}\left[\sum_{t=1}^T \nabla_t^\top(\boldsymbol{w}_t - \boldsymbol{u}) - \frac{\beta}{2}\Big(\nabla_t^\top(\boldsymbol{w}_t - \boldsymbol{u})\Big)^2\right] \\ &\le 2R^2\|\boldsymbol{u}\|^2 + d(4U^2R^2 + V) \\ &\quad \times \mathbb{E}\left[\ln\left(\frac{2R^2\|\boldsymbol{u}\|^2 + \sum_{t=1}^T \mathrm{Var}[Q_t]}{d(4U^2R^2+V)} + 1\right)\right]. \end{aligned}$$

Applying Jensen's inequality we conclude the proof. □

3.2 Sparse regression

The sparse regression algorithm for square loss (Algorithm 2) simply replaces the gradient vector $\nabla_t = (\boldsymbol{w}_t^\top \boldsymbol{x}_t - y_t)\boldsymbol{x}_t$ by the stochastic vector $B_t \boldsymbol{x}_t$, where B_t is a random variable taking value $1/\big(\beta(\boldsymbol{w}_t^\top \boldsymbol{x}_t - y_t)\big)$ with probability $2\beta\big(\boldsymbol{w}_t^\top \boldsymbol{x}_t - y_t\big)^2 \le 1$, and zero otherwise. Using $\mathbb{E}_t$ as a shorthand for the conditional expectation $\mathbb{E}[\,\cdot\,|\,B_1, \ldots, B_t]$, it is immediate to see that $\mathbb{E}_{t-1}[B_t] = 2\big(\boldsymbol{w}_t^\top \boldsymbol{x}_t - y_t\big)$ and $\mathbb{E}_{t-1}[B_t^2] = 2/\beta$. We are interested in proving upper bounds on both the expected regret $\mathbb{E}[R_T]$ and the expected number of times the algorithm makes a weight update, i.e., $\sum_{t=1}^T \mathbb{P}(B_t \ne 0)$, where probabilities and expectations are w.r.t. the random draws of $B_1, \ldots, B_T$. A sparse regression algorithm is useful, for instance, when we would like to force constraints on the overall running time of the learning process. But also when we are running our algorithm in a RKHS, to reduce the number of "support vectors" in the dual representation of the learned regression function, as in budget algorithms [Orabona et al., 2009, and references therein]. In the following theorem, we show how we can achieve a regret logarithmic (in the time horizon T, rather than the total loss of the best offline comparator as in Theorem 1), by a sparse regression function. Once again, we are unaware of similar results in the online linear regression literature. In fact, it is not clear to us whether similar results could be obtained by a direct adaptation of existing ridge regression algorithms, such as Vovk's or Azoury and Warmuth's.

Theorem 2. *Assume for all t the loss function $\ell_t(\boldsymbol{v}) = (\boldsymbol{v}^\top \boldsymbol{x}_t - y_t)^2$ is such that $\|\boldsymbol{x}_t\| \le R$, and $|y_t| \le Y$. Assuming the SONS Algorithm 2 is run with $\alpha > 0$, then*

$$\mathbb{E}[R_T] \le \frac{d}{2\beta}\,\ln\left(1 + \frac{2\,T\,R^2}{\beta\,d\,\alpha}\right) + \frac{\alpha\,\beta}{2}\|\boldsymbol{u}\|^2$$

holds

1. *for any $\boldsymbol{u} \in \mathcal{S}$ whenever $(UR + Y)^2 \le \frac{1}{2\beta}$ and $\mathcal{S}_t \equiv \mathcal{S} \equiv \{\boldsymbol{v} \,:\, \|\boldsymbol{v}\| \le U\}$;*
2. *for any $\boldsymbol{u} \in \mathcal{S} \equiv \bigcap_t \mathcal{S}_t$ whenever $(\widetilde{Y} + Y)^2 \le \frac{1}{2\beta}$ and $\mathcal{S}_t \equiv \left\{\boldsymbol{v} \,:\, |\boldsymbol{v}^\top \boldsymbol{x}_t| \le \widetilde{Y}\right\}$.*

Moreover, the expected number of updates satisfies

$$\sum_{t=1}^T \mathbb{P}(B_t \neq 0) \le 2\beta \sum_{t=1}^T \ell_t(\boldsymbol{u}) + d\,\ln\left(1 + \frac{2\,T\,R^2}{\beta\,d\,\alpha}\right) + \beta^2\,\alpha\|\boldsymbol{u}\|^2\,.$$

Proof. We start from (2) where we replace every occurrence of ∇_t by $B_t\boldsymbol{x}_t$. This yields the deterministic inequality

$$\sum_{t=1}^T \left(B_t\,(\boldsymbol{w}_t^\top \boldsymbol{x}_t - \boldsymbol{u}^\top \boldsymbol{x}_t) - \frac{\beta}{2}\,B_t^2\,(\boldsymbol{w}_t^\top \boldsymbol{x}_t - \boldsymbol{u}^\top \boldsymbol{x}_t)^2\right) \le \frac{1}{2\beta}\sum_{t=1}^T B_t\boldsymbol{x}_t^\top\,A_{t+1}^{-1}\,B_t\boldsymbol{x}_t + \frac{\beta}{2}\,\alpha\|u\|^2. \tag{6}$$

Now note that

$$\mathbb{E}\big[B_t\,(\boldsymbol{w}_t^\top \boldsymbol{x}_t - \boldsymbol{u}^\top \boldsymbol{x}_t)\big] = \mathbb{E}\left[\mathbb{E}_{t-1}[B_t]\,(\boldsymbol{w}_t^\top \boldsymbol{x}_t - \boldsymbol{u}^\top \boldsymbol{x}_t)\right] = 2\,\mathbb{E}\left[(\boldsymbol{w}_t^\top \boldsymbol{x}_t - y_t)\,\boldsymbol{x}_t^\top(\boldsymbol{w}_t - \boldsymbol{u})\right]$$

and, for similar reasons,

$$\mathbb{E}[B_t^2\,(\boldsymbol{w}_t^\top \boldsymbol{x}_t - \boldsymbol{u}^\top \boldsymbol{x}_t)^2] = \frac{2}{\beta}\,\mathbb{E}\left[(\boldsymbol{w}_t^\top \boldsymbol{x}_t - \boldsymbol{u}^\top \boldsymbol{x}_t)^2\right]\,.$$

Taking expectation of (6) and using identity (4), we get

$$\begin{aligned}\mathbb{E}[R_T] \le& \mathbb{E}\left[\sum_{t=1}^T 2(\boldsymbol{w}^\top \boldsymbol{x}_t - y_t)\,\boldsymbol{x}_t^\top(\boldsymbol{w} - \boldsymbol{u}) - (\boldsymbol{w}^\top \boldsymbol{x}_t - \boldsymbol{u}^\top \boldsymbol{x}_t)^2\right]\\ \le& \frac{1}{2\beta}\mathbb{E}\left[\sum_{t=1}^T B_t\boldsymbol{x}_t^\top\,A_{t+1}^{-1}\,B_t\boldsymbol{x}_t\right] + \frac{\beta}{2}\,\alpha\|u\|^2\\ \le& \frac{d}{2\beta}\mathbb{E}\left[\ln\left(1 + \sum_{t=1}^T B_t^2\,\|\boldsymbol{x}_t\|^2\right)\right] + \frac{\beta}{2}\,\alpha\|u\|^2\\ \le& \frac{d}{2\beta}\,\ln\left(1 + \frac{2\,T\,R^2}{\beta\,d\,\alpha}\right) + \frac{\beta}{2}\,\alpha\|u\|^2\end{aligned}$$

where in the penultimate step we used the upper bound (3) and in the last step we used the concavity of the logarithm. Further overapproximations result in the claimed upper bound on the expected regret $\mathbb{E}[R_T]$. The bound on the expected number of updates follows from

$$\sum_{t=1}^T \mathbb{P}(B_t \neq 0) = \mathbb{E}\left[\sum_{t=1}^T \mathbb{P}_{t-1}[B_t \neq 0]\right] = 2\beta\,\mathbb{E}\left[\sum_{t=1}^T \ell_t(\boldsymbol{w}_t)\right]\,.$$

Using the previous upper bound on $\mathbb{E}[R_T]$ gives the desired result. □

A few remarks are in order. First, observe the role played by parameter β which enables a significant trade-off of regret against number of updates. For instance, if $\sum_{t=1}^T \ell_t(\boldsymbol{u})$ is more than logarithmic in T, then decreasing β tends to sparsify the final hypothesis $\boldsymbol{w}_{T+1}$ at the cost of increasing the contribution to the regret R_T due to the logarithmic term $d\,\ln\left(1 + \frac{2\,T\,R^2}{\beta\,d\,\alpha}\right)$. When T is large, setting $\beta = O(1/\sqrt{T})$ yields an algorithm achieving expected regret $O(\sqrt{T})$ with an expected number of weight updates which is again $O(\sqrt{T})$.

Second, compared to standard ways of sparsifying a linear regression function (e.g., an ϵ-insensitive square loss, as often used in the SVM literature) the advantage of our approach lies in the ability to provide a detailed quantification of the outcome of the sparsification procedure, with the additional advantage of measuring the bounds i.t.o. the desired loss instead of its ϵ-insensitive version.

Third, it is worth stressing that SONS is not a selective sampling (or active learning) algorithm à la Cesa-Bianchi et al. [2006], Orabona and Cesa-Bianchi [2011]. There, sparsification is obtained as a by-product of actively selecting labels in a stream of training examples, and the decision to update does not depend on the label of the current instance vector. Here, the decision to update or not at time t depends on the value of variable B_t, whose bias is set after seeing the current label y_t.

4 Lower Bounds for the Square Loss

In this section we prove lower bounds for the square loss in both the adversarial and fixed design settings. In what follows, let

$$L_T^* = \inf_{\boldsymbol{u}\in\mathbb{R}^d} \sum_{t=1}^{T} \left(\boldsymbol{u}^\top \boldsymbol{x}_t - y_t\right)^2$$

where $\boldsymbol{u}$ is understood from the context.

We start with a sligthly tighter version of the lower bound due to Vovk [2001].

Theorem 3. *Fix the dimension of the space d, the upper bound $2Y$ on the range of outcomes, and $T \in \mathbb{N}$ such that T is a multiple of d. For any $a > 0$, there exists a sequence $\boldsymbol{x}_1, \dots, \boldsymbol{x}_T \in \mathbb{R}^d$ with $\|\boldsymbol{x}_t\|_\infty = 1$ for all t, and a joint distribution of outcomes $Y_1, \dots, Y_T$ with $Y_t \in \{0, 2Y\}$ for all t, such that for any online algorithm we have that*

$$\mathbb{E}\big[L_T - L_T^*\big] \ge dY^2 \frac{2a}{2a+1}\left(\ln\left(\frac{T/d}{2a}+1\right)+1\right) .$$

Proof. First consider the one-dimensional case, using T/d instances $x_1 = \cdots = x_{T/d} = 1$, then we generalize it to the d-dimensional case. We sample p from a Beta distribution with parameters (a, a), $a > 0$. For each t we set $Y_t = 2Y$ with probability p and $Y_t = 0$ with probability $1 - p$. We proceed as in Vovk [2001], obtaining that

$$\begin{aligned}\mathbb{E}[L_{T/d}] = 4Y^2\mathbb{E}[p(1-p)] &\sum_{t=0}^{T/d-1}\left(\frac{t}{(t+2a)^2}+1\right)\\ &+ 4Y^2\mathbb{E}[(2ap-a)^2]\sum_{t=0}^{T/d-1}\frac{1}{(t+2a)^2}\\ &= \frac{4Y^2a}{2(2a+1)}\left(\sum_{t=0}^{T/d-1}\frac{1}{t+2a}+T/d\right)\end{aligned}$$

where we have used the fact that $\mathbb{E}[p(1-p)] = \frac{a}{2(2a+1)}$, and $\mathbb{E}[(2p-1)^2] = \frac{1}{2a+1}$. We now lower bound this quantity with

$$\begin{aligned}&\frac{4Y^2a}{2(2a+1)}\left(\sum_{t=0}^{T/d-1}\frac{1}{t+2a}+T/d\right)\\ &\ge \frac{4Y^2a}{2(2a+1)}\left(\int_0^{T/d}\frac{1}{t+2a}dt+T/d\right)\\ &= \frac{4Y^2a}{2(2a+1)}\left(\ln\left(\frac{T/d}{2a}+1\right)+T/d\right) .\end{aligned}$$

On the other hand, using the hypothesis that the Y_t are i.i.d., it is easy to show that

$$\mathbb{E}[L_{T/d}^*] = \mathbb{E}\left[\inf_{u\in\mathbb{R}}\sum_{t=1}^{T/d}(u-Y_t)^2\right] = \frac{4Y^2a(T/d-1)}{2(2a+1)} .$$

We have

$$\mathbb{E}[L_{T/d} - L_{T/d}^*] \ge \frac{4Y^2a}{2(2a+1)}\left(\ln\left(\frac{T/d}{2a}+1\right)+1\right) .$$

The d-dimensional bound can be easily obtained proceeding as in Vovk [2001]. □

We now use this lower bound to obtain a lower bound that depends on the loss of the competitor.

Corollary 3. *Fix the dimension of the space d and the upper bound $2Y$ on the range of outcomes. For any $L < Y^2T$ there exists a sequence $(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_T, \boldsymbol{y}_T)$ of examples with $\|\boldsymbol{x}_t\|_\infty = 1$, $|y_t| = Y$, and $L_T^* \le L$, such that for any online algorithm we have that*

$$L_T - L_T^* \ge dY^2(1-\epsilon)\left(\ln\left(\frac{\epsilon(L/Y^2-1)}{(1-\epsilon)d}+1\right)+1\right) .$$

Proof. Fix $\epsilon = 1 - \frac{4a}{2(2a+1)}$, so that $\frac{1}{2a} = \frac{\epsilon}{1-\epsilon}$. We apply the bound of Theorem 3 on the first $T' = \left\lfloor \frac{L}{Y^2} \right\rfloor$ steps. We then sets the labels of the remaining $T - T'$ steps so to have $L_T^* - L_{T'}^* = 0$, hence we have that $L_T^* \le L$ deterministically. We have

$$\begin{aligned}&E[L_T - L_T^*] \ge E[L_{T'} - L_{T'}^*]\\ &\ge dY^2(1-\epsilon)\left(\ln\left(\frac{\epsilon T'}{(1-\epsilon)d}+1\right)+1\right)\\ &\ge dY^2(1-\epsilon)\left(\ln\left(\frac{\epsilon(L/Y^2-1)}{(1-\epsilon)d}+1\right)+1\right) .\end{aligned}$$

Hence there exists a sequence of labels such that the claimed lower bound holds. □

The lower bound is of order $dY^2 \ln \frac{L}{dY^2}$. Using Part 1 of Corollary 1, the dominant term in the upper bound is $d(U+Y)^2 \ln \frac{L}{d(U+Y)^2}$, matching the lower bound whenever U and Y are of the same order.

We can also specialize Theorem 3 to the case of statistical regression with fixed design, although the resulting bound is suboptimal.

Corollary 4. *Fix the dimension of the space d and the upper bound $2Y$ on the range of outcomes. For any $0 < \sigma \le Y$ there exists a sequence $\boldsymbol{x}_1, \dots, \boldsymbol{x}_T \in \mathbb{R}^d$ with $\|\boldsymbol{x}_t\|_\infty = 1$ for all t, and a joint distribution of outcomes $Y_1, \dots, Y_T$ with $Y_t \in \{0, 2Y\}$ and $\mathrm{Var}[Y_t] = \sigma^2$ for all t, such that for any online algorithm we have that*

$$\mathbb{E}[L_T - L_T^*] \ge d\sigma^2\left(\ln\left(\left(\frac{Y^2}{\sigma^2}-1\right)\frac{T}{d}+1\right)+1\right) .$$

5 Classification

Unlike regression, in online classification we are usually interested in mistake bounds rather than regret bounds. At first glance, this makes the problem easier, since applying the ONS algorithm to any (not necessarily smooth) exp-concave upper bound on the zero-one loss, one obtains a bound on the number M of mistakes of the form $M \leq L^* + \mathcal{O}(\ln M)$. This can be easily solved for M via Corollary 5 —see the appendix. However, standard classification algorithms (such as Perceptron) need not know a bound on the norm of the competing hyperplane $\boldsymbol{u}$, a relevant piece of information required by the approach of Section 3. In this section, we propose a different strategy based on an entire family of loss functions, rather than a single one. The mistake bound is shown to depend on the smallest of the functions in this family and, as a result, we get rid of the dependence on (an upper bound on) $\|\boldsymbol{u}\|$ in the algorithm[2]. We use the Second-Order Perceptron algorithm of Cesa-Bianchi et al. [2005] (see also AROW [Crammer et al., 2009]) whose pseudocode is given in Algorithm 3. However, the analysis here is original, and it also differs from the one we carried out in the regression case.

Algorithm 3 Second-Order Perceptron

```
Input: $\alpha > 0$.
Initialize: $\boldsymbol{w}_1 = \boldsymbol{0}, A_1 = \alpha I$
for $t = 1, 2, \ldots, T$ do
    Receive $\boldsymbol{x}_t$
    Predict with $\boldsymbol{w}_t^\top \boldsymbol{x}_t$ and receive $y_t \in \{-1, +1\}$
    if $\text{sign}(\boldsymbol{w}_t^\top \boldsymbol{x}_t) \neq y_t$ then
        $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - (1 - y_t \boldsymbol{w}_t^\top \boldsymbol{x}_t) \frac{A_t^{-1} y_t \boldsymbol{x}_t}{1 + \boldsymbol{x}_t^\top A_t^{-1} \boldsymbol{x}_t}$
        $A_{t+1} = A_t + \boldsymbol{x}_t \boldsymbol{x}_t^\top$
    else
        $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t, \quad A_{t+1} = A_t$
    end if
end for
```

Theorem 4. *Let $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_T, y_T) \in \mathbb{R}^d \times \{-1, +1\}$ be a sequence of examples such that $\|\boldsymbol{x}_t\| \leq R$, and assume that Algorithm 3 is run with $\alpha = R^2$. Then, for any $\boldsymbol{u} \in \mathbb{R}^d$ and $0 \leq \eta \leq \min\{\frac{2}{\|\boldsymbol{u}\| R + 1}, 1\}$, the number M of prediction mistakes is upper bounded by*[3]

$$L_\eta + \frac{\eta R^2 \|\boldsymbol{u}\|^2}{2 - \eta} + \frac{d}{\eta(2 - \eta)} \ln\left(\frac{2}{\eta(2 - \eta)} \ln \frac{2}{e\eta(2 - \eta)} + \frac{2}{d}\left(L_\eta + \eta R^2 \|\boldsymbol{u}\|^2\right) + 2\right)$$

where $L_\eta = \sum_t \ell_{\eta,t}(\boldsymbol{u})$ and

$$\ell_{\eta,t}(\boldsymbol{u}) = \left[1 - \frac{2}{2 - \eta} y_t \boldsymbol{u}^\top \boldsymbol{x}_t + \frac{\eta}{2 - \eta} (\boldsymbol{u}^\top \boldsymbol{x}_t)^2\right]_+ .$$

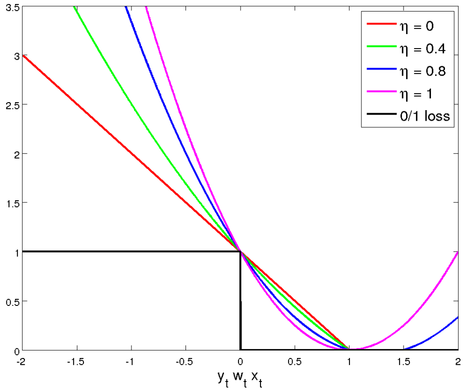

Figure 1: $\ell_{\eta,t}$ for various settings of η.

Before proving the theorem, we discuss the properties of the loss function $\ell_{\eta,t}(\boldsymbol{u})$. For any valid setting of η, $\ell_{\eta,t}(\boldsymbol{u})$ upper bounds the zero-one loss —see Figure 1. Specifically, for $\eta = 0$ we have that $\ell_{\eta,t}$ becomes the hinge loss, and for $\eta = 1$ it becomes the square loss. Hence, the loss is positive when $y_t \boldsymbol{u}^\top \boldsymbol{x}_t < 1$, is equal to zero when $1 \leq y_t \boldsymbol{u}^\top \boldsymbol{x}_t \leq \frac{2-\eta}{\eta}$, and is greater than zero when $y_t \boldsymbol{u}^\top \boldsymbol{x}_t > \frac{2-\eta}{\eta}$. The fact that the loss becomes positive when the margin is big enough may appear strange. However, thanks to the constraint on η, that range of values is never reached. This means that when the problem is linearly separable, there exists a valid η such that $L_\eta = 0$ (hence the algorithm makes a finite number of mistakes, just like Perceptron). On the other hand, if the problem is not linearly separable, the algorithm has a mistake bound that grows logarithmically with the loss of the competitor.

Proof of Theorem 4. Let $\mathcal{M}$ be the set of time steps when the algorithm makes a mistake. Define $p_t = (\boldsymbol{w}_t - \eta \boldsymbol{u})^\top A_t (\boldsymbol{w}_t - \eta \boldsymbol{u})$. Using the standard difference-of-norms proof technique together with the lemmas in Crammer et al. [2009] we have, for any $\eta > 0$,

$$\alpha \eta \|\boldsymbol{u}\|^2 \geq \frac{1}{\eta}(p_1 - p_{T+1}) = \frac{1}{\eta} \sum_{t=1}^{T} (p_t - p_{t+1})$$
$$= \sum_{t \in \mathcal{M}} \left(2 y_t \boldsymbol{u}^\top \boldsymbol{x}_t - \frac{\boldsymbol{x}_t^\top A_t^{-1} \boldsymbol{x}_t + 1 - (1 - y_t \boldsymbol{w}_t^\top \boldsymbol{x}_t)^2}{\eta(1 + \boldsymbol{x}_t^\top A_t^{-1} \boldsymbol{x}_t)} - \eta (\boldsymbol{u}^\top \boldsymbol{x}_t)^2\right) .$$

From this inequality we can derive the second-order Perceptron bound of Cesa-Bianchi et al. [2005]. If updates are performed also when the margin is smaller than 1, then we recover the mistake bound of AROW [Crammer et al., 2009]. We now show how to obtain our new mistake bound, that depends logarithmically on the loss of the competitor. Adding $b\,M$ to both sides of the last inequality, we have

[2] Knowing $\|\boldsymbol{u}\|$ in classification roughly corresponds to knowing ahead of time the margin level of the best hyperplane for the data at hand.

[3] Note that $\|\boldsymbol{u}\|$ and η are free parameters in this statement.

that, for any $\eta, b > 0$, the number of mistakes M is less than

$$\frac{r\eta\|\boldsymbol{u}\|^2}{b} + \frac{1}{b\eta}\ln|A_{T+1}| \\ + \sum_{t\in\mathcal{M}}\left(1 + \frac{\eta(\boldsymbol{u}^\top\boldsymbol{x}_t)^2 - 2y_t\boldsymbol{u}^\top\boldsymbol{x}_t}{b}\right)$$

where we used

$$t \in \mathcal{M} \Rightarrow y_t\boldsymbol{w}_t^\top\boldsymbol{x}_t \le 0 \Rightarrow 1 - (1 - y_t\boldsymbol{w}_t^\top\boldsymbol{x}_t)^2 \le 0 \ .$$

Setting $b = 2 - \eta$, we have that

$$1 - \tfrac{2}{b}y_t\boldsymbol{u}^\top\boldsymbol{x}_t + \tfrac{\eta}{b}(\boldsymbol{u}^\top\boldsymbol{x}_t)^2 \le \ell_{\eta,t}(\boldsymbol{u}) \ .$$

Hence, for any

$$0 \le \eta \le \min\left\{\frac{2}{\|\boldsymbol{u}\|R+1}, 1\right\}$$

the following bound holds

$$M \le \frac{\alpha\eta\|\boldsymbol{u}\|^2}{2-\eta} + \frac{d}{\eta(2-\eta)}\ln\left(1 + \frac{MR^2}{\alpha d}\right) + L_\eta$$

where we upper bounded $\ln|A_{T+1}|$ with $d\ln\big(1 + \frac{MR^2}{\alpha d}\big)$ and used $R \ge \|\boldsymbol{x}_t\|$. Using Corollary 5 in the appendix with $n = 2$ yields the stated bound on M. □

6 Conclusions and ongoing research

We have shown that for smooth and exp-concave losses, variants of the Online Newton Step (ONS) algorithm exist whose regrets are logarithmic in the loss of the best comparison vector. When adapted to the square loss in the statistical setting with fixed design, these lead to regret bounds depending logarithmically on the cumulative variance of the label noise. Matching lower bounds are provided for the individual sequence setting. The same tools we used for the analysis of square loss regret can be adapted to design a sparse variant of ONS that trades off accuracy vs. sparsity. Finally, in the classification setting, we have given a new analysis of the Second-Order Perceptron, where a regret bound logarithmic in the loss of the best offline linear classifier $\boldsymbol{u}$ is achieved without prior knowledge of the norm of $\boldsymbol{u}$. The loss of $\boldsymbol{u}$ is measured according to the convex proxy to the zero-one loss which best interpolates between linear and quadratic hinge losses.

We close with a few directions of current research. First, it would be nice to extend to exp-concave losses our results that currently hold for the square loss only. Second, we would like to close the gap between upper and lower bounds in Corollary 2 and Corollary 4, at least in the case of square loss. Third, we plan to test the empirical behavior of the sparse regression algorithm of Section 3.2.

Acknowledgements

This work was partially supported by the PASCAL2 Network of Excellence under EC grant 216886. The first author was also supported by "Dote Ricerca": FSE, Regione Lombardia.

Appendix

This appendix contains technical lemmas that are needed to obtain explicit bounds for logarithmic inequalities. These lemmas are improvements of the statements in [Lihong et al., 2011, Lemma 4].

Lemma 2. *Let $a, x > 0$ satisfy $x \le a\ln x$, then $\forall n \ge 1$*

$$x \le \frac{n}{n-1}a\ln\frac{na}{e} \ .$$

Proof. For the purpose of contradiction, suppose that

$$x > \frac{n}{n-1}a\ln\frac{na}{e} \ .$$

In the following we use the inequality $\ln x \le \frac{n}{e}x^{\frac{1}{n}}$ for all $n, x > 0$. Note that $x \le a\ln x \le \frac{a}{e}x$, hence $a \ge e$. We have that

$$\frac{n}{n-1}a\ln\frac{na}{e} < x \le a\ln x \ .$$

This implies that $\left(\frac{na}{e}\right)^{\frac{n}{n-1}} < x$. On the other hand,

$$x \le a\ln x \le \frac{an}{e}x^{\frac{1}{n}} \ .$$

Hence $x \le \left(\frac{na}{e}\right)^{\frac{n}{n-1}}$. Comparing the lower and upper bound on x we reach a contradiction. This shows that $x \le \frac{n}{n-1}a\ln\frac{na}{e}$. □

We now use Lemma 2 to prove a more powerful inequality. This inequality allows us to prove regret bounds that have constant 1 in front of the loss of the competitor.

Lemma 3. *Let $a, x > 0$ satisfy $x \le a\ln x$, then $\forall n \ge 1$*

$$x \le a\ln\left(\frac{n}{n-1}a\ln\frac{na}{e}\right) \ .$$

Proof. Lemma 2 in the inequality $x \le a\ln x$ gives us

$$x \le a\ln x \le a\ln\Big(\frac{n}{n-1}a\ln\frac{na}{e}\Big) \ . \qquad \square$$

Corollary 5. *Let $a, b, c, d, x > 0$ satisfy*

$$x \le a\ln(bx + c) + d \ .$$

Then for all $n \ge 1$

$$x \le a\ln\left(\frac{n}{n-1}\left(ab\ln\frac{abn}{e} + db + c\right)\right) + d \ .$$

References

J. Abernethy, P. L. Bartlett, A. Rakhlin, and A. Tewari. Optimal strategies and minimax lower bounds for online convex games. In *Proceedings of the 21st Annual Conference on Learning Theory*, pages 414–424. Omnipress, 2008.

K. S. Azoury and M. K. Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning*, 43(3):211–246, 2001.

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

N. Cesa-Bianchi, A. Conconi, and C. Gentile. A second-order Perceptron algorithm. *SIAM Journal on Computing*, 34(3):640–668, 2005.

N. Cesa-Bianchi, C. Gentile, and L. Zaniboni. Worst-case analysis of selective sampling for linear-threshold algorithms. *Journal of Machine Learning Research*, 7:1205–1230, 2006.

K. Crammer, A. Kulesza, and M. Dredze. Adaptive regularization of weight vectors. *Advances in Neural Information Processing Systems*, 23, 2009.

O. Dekel, C. Gentile, and K. Sridharan. Robust selective sampling from single and multiple teachers. In *Proc. of the 23rd International Conference on Learning Theory*. MIT Press, 2010.

E. Hazan and S. Kale. On stochastic and worst-case models for investing. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 709–717. 2009.

E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

L. Lihong, M. Littman, T. Walsh, and A. Strehl. Knows what it knows: a framework for self-aware learning. *Machine Learning*, 82:399–443, 2011.

F. Orabona and N. Cesa-Bianchi. Better algorithms for selective sampling. In *Proceedings of the 28th International Conference (ICML)*, pages 433–440, 2011.

F. Orabona, J. Keshet, and B. Caputo. Bounded kernel-based online learning. *Journal of Machine Learning Research*, 10:2571–2594, 2009.

S. Shalev-Shwartz. Online learning: Theory, algorithms, and applications. Technical report, The Hebrew University, 2007. PhD thesis.

N. Srebro, K. Sridharan, and A. Tewari. Smoothness, low noise and fast rates. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 2199–2207. 2010.

V. Vovk. Competitive on-line statistics. *International Statistical Review*, 69:213–248, 2001.

M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference (ICML)*, pages 928–936, 2003.