Scale-free Unconstrained Online Learning for Curved Losses

Jack J. Mayo, J.J.MAYO@UVA.NL
Hédi Hadiji H.HADIJI@UVA.NL
and **Tim van Erven** TIM@TIMVANERVEN.NL
Korteweg-de Vries Institute for Mathematics, University of Amsterdam, Amsterdam, The Netherlands

Editors: Po-Ling Loh and Maxim Raginsky

Abstract

A sequence of works in unconstrained online convex optimisation have investigated the possibility of adapting simultaneously to the norm U of the comparator and the maximum norm G of the gradients. In full generality, matching upper and lower bounds are known which show that this comes at the unavoidable cost of an additive GU^3, which is not needed when either G or U is known in advance. Surprisingly, recent results by Kempka et al. (2019) show that no such price for adaptivity is needed in the specific case of 1-Lipschitz losses like the hinge loss. We follow up on this observation by showing that there is in fact never a price to pay for adaptivity if we specialise to any of the other common supervised online learning losses: our results cover log loss, (linear and non-parametric) logistic regression, square loss prediction, and (linear and non-parametric) least-squares regression. We also fill in several gaps in the literature by providing matching lower bounds with an explicit dependence on U. In all cases we obtain scale-free algorithms, which are suitably invariant under rescaling of the data. Our general goal is to establish achievable rates without concern for computational efficiency, but for linear logistic regression we also provide an adaptive method that is as efficient as the recent non-adaptive algorithm by Agarwal et al. (2021).

Keywords: Online convex optimisation, supervised online learning, comparator-adaptive, Lipschitz-adaptive, mixable loss

1. Introduction

The problem of hyperparameter tuning is ubiquitous across machine learning. We study it in the context of *online supervised learning* (see e.g. (Rakhlin et al., 2015)), in which a learner needs to issue sequential predictions over the course of T rounds. At the start of each round t, the learner first receives a feature vector x_t, and then issues a prediction a_t of the corresponding response y_t. Performance is measured by the *regret*, which is the difference between the sum of the learner's losses $\ell(a_t, y_t)$ and the sum of the losses $\ell(f_\theta(x_t), y_t)$ suffered by the best comparator function f_θ from a function class $\mathcal{F}$ indexed by parameters θ.

There are two main types of hyperparameters: on the one hand it is desirable to adapt automatically to the norm $\|\theta\|$ of the optimal comparator parameters; on the other hand we want algorithms that do not have to know the scale of the data x_t and y_t beforehand. These issues have frequently been studied in the context of *online convex optimisation* (OCO) (Hazan, 2016), where underlying details of the setup are abstracted away by assuming only that the functions $\ell_t(\theta) := \ell(f_\theta(x_t), y_t)$ are convex and requiring predictions to be of the form $a_t = f_{\theta_t}(x_t)$ for some θ_t. The scale of the data then comes in through the maximum length G of the gradients $g_t := \nabla \ell_t(\theta_t)$. Given G and an upper bound $U \geqslant \|\theta\|$ for the optimal parameters θ, the best regret that can be guaranteed is $O(UG\sqrt{T})$ (Zinkevich, 2003). This is even possible when only U is known, but not G (Duchi et al., 2011). Conversely, given G but not U, it has been found that the optimal rate is $O(\|\theta\|G\sqrt{T \log(1 + \|\theta\|)T})$

McMahan and Streeter (2012); McMahan and Abernethy (2013); Cutkosky and Orabona (2018), so then the price of adaptivity is a mere logarithmic factor in $\|\theta\|$. Simultaneous adaptivity to both G and $\|\theta\|$, however, has been shown to be impossible without a worse dependence on $\|\theta\|$ (Cutkosky and Boahen, 2017) and comes at the non-negligible cost of an additive $G\|\theta\|^3$, with matching upper and lower bounds establishing the rate to be $O(\|\theta\|G\sqrt{T\log(1+\|\theta\|)}+G\|\theta\|^3)$ (Cutkosky, 2019; Mhammedi and Koolen, 2020). Alternatively, $O((\|\theta\|^2+1)G\sqrt{T})$ is also possible (Orabona and Pál, 2018). This fully settles the issue of simultaneous adaptivity, but only for the OCO setting.

Since we have more information available in online supervised learning, the OCO lower bounds do not apply, and indeed Kempka et al. (2019); Mhammedi and Koolen (2020) obtain upper bounds of order $O(UX\sqrt{T\log(1+UXT)})$ for linear models $f_\theta(x_t) = \theta^\top x_t$ and losses of the form $\ell_t(\theta) = h_t(\theta^\top x_t)$, where h_t is 1-Lipschitz and $X = \max_{t\leqslant T}\|x_t\|_2$. 1-Lipschitzness is satisfied by important practical cases like the hinge loss $h_t(z) = \max\{1-y_t z, 0\}$, the two-class logistic loss $h_t(z) = \ln(1+e^{-y_t z})$ and the absolute loss $h_t(z) = |y_t - z|$. The key feature of this bound is that it depends on U and X only via their product UX, without having to know either hyperparameter in advance. It is therefore both adaptive to the norm of the comparator and scale-free: if all x_t get scaled by the same constant, then the optimal parameters θ undo this scaling, and the bound remains unchanged. In fact, even the algorithms are scale-free: scaling all x_t does not affect the predictions a_t at all. This tantalising possibility of circumventing lower bounds prompts us to ask the following general question:

> *Given a specific loss ℓ and function class $\mathcal{F}$ in online supervised learning, what is the price of adapting to $\|\theta\|$ while being at the same time scale-free?*

We focus on answering this question for two major classes of losses ℓ: the first is the logarithmic loss $\ell(p, y) = -\ln p(y)$ where predictions $a \equiv p$ are densities or probabilities, with (multiclass) logistic regression as its main special case; the second is the square loss $\ell(a, y) = \|y-a\|^2$, which pertains to least-squares regression. In the latter case, scale-freeness also requires the predictions a to scale linearly with the y_t, and the bounds also depend on $Y = \max_{t\leqslant T}\|y_t\|$. Prior work and our contributions are summarised in Table 1.

The main observation from Table 1 is that there is *never* a price to pay in the rates for adapting to $\|\theta\|$ with a scale-free algorithm, except possibly in the case that we do not study here: for the hinge loss there exists a gap between known upper and lower bounds in the regime where $\|\theta\|X > 1$, which we leave as an open issue.

Approach Our main technical tool in obtaining upper bounds is mixability of the logarithmic and square loss, which implies that we can aggregate over an exponentially spaced grid of hyperparameters $\alpha \geqslant \alpha_{\text{min}} > 0$ at the cost of a mere additive $O(\frac{1}{\eta}\log\log(\alpha/\alpha_{\text{min}}))$ term in the bound. This may be interpreted as the number of bits to encode α rounded up to the nearest grid point. A technical complication that requires considerable care is to specify a minimum value α_{min} without breaking either scale-freeness or paying a non-negligible price in the bound. This is related to the range-ratio problem of Mhammedi and Koolen (2020). As a consequence, we do end up with a dependence on the feature vector ratio X_T/X_{t^*} in some cases, where $X_t = \max_{s\leqslant t}\|x_s\|$ and t^* is the smallest t for which $\|x_t\| > 0$. A logarithmic dependence on this ratio has previously been considered acceptable by Kempka et al. (2019); Ross et al. (2013); Wintenberger (2017); Kotłowski (2017). In our case the term appears inside an even smaller double logarithm, which means that it can be neglected simply based on the range of numbers representable on a computer as double precision floating point numbers. A similar doubly logarithmic dependence was encountered by Gerchinovitz (2011). Mixability further depends on a parameter η, which is $\eta = 1$ for log loss and $\eta \propto 1/Y^2$ for square loss. In case of the square loss, the fact that Y^2 is unknown in advance introduces the need for online clipping and projecting of predictions to the range Y_t, where $Y_t = \max_{s \leqslant t} \|y_s\|$. Similar approaches have previously been used by Gerchinovitz (2011); Cutkosky (2019).

Loss	Function Class	Non-adaptive Rate	Adaptive Rate
Logarithmic $-\ln p(y)$	normal location	$d \ln \frac{UT}{\sigma}$ (Barron et al., 1998) (Stine and Foster, 2000), Thm. 2	$d \ln \frac{\|\theta\| T}{\sigma}$ (Grünwald, 2007), Thm. 3
Multiclass logistic regression (K classes)	linear	$\leqslant dK \ln \frac{UXT}{dK}$ $UX = \Omega(\sqrt{d} \ln T)$: $\geqslant d \ln \frac{UX}{\sqrt{d} \ln T}$ (Foster et al., 2018) $UX \leqslant 2\sqrt{d}:\ \geqslant d \ln \frac{UXT}{d}$ Thm. 6	$\leqslant X\|\theta\| \sqrt{T \ln(X\|\theta\| T)}$ (Mhammedi and Koolen, 2020) $\leqslant dK \ln \frac{\|\theta\| XT}{dK}$ Thm. 7, Thm. 8
	linear (efficient alg)	$\leqslant (UX + \ln K) dK \ln T$ (Agarwal et al., 2021)	$\leqslant (\|\theta\| X + \ln K)\, dK \ln T$ Thm. 10
	Besov	$\leqslant \tilde{O}(U^\beta T^\gamma)$ (Foster et al., 2018)	$\leqslant \tilde{O}(\|\theta\|^\beta T^\gamma)$ Thm. 20
Square $\frac{1}{2}(y-a)^2$	square loss prediction	$Y^2 \ln \frac{(U \wedge Y)T}{Y}$ (van der Hoeven et al., 2018), Thm. 18	$Y^2 \ln \frac{(\|\theta\| \wedge Y)T}{Y}$ Thm. 12
Least-squares regression	linear	$dY^2 \ln \frac{UXT}{dY}$ (Vovk, 1998; Azoury and Warmuth, 2001) Thm. 18	$dY^2 \ln \frac{\|\theta\| XT}{dY}$ Thm. 15
	Sobolev, $s \geqslant d/2$	$\tilde{O}(T^{d/(2s+d)})$ (Zadorozhnyi et al., 2021)	$\leqslant \tilde{O}(\|\theta\|^{s/(2s+d)} T^{d/(2s+d)})$ Thm. 17
Hinge $\max\{0, 1 - ya\}$	linear	$\leqslant UX\sqrt{T}$ $\geqslant (UX \wedge 1)\sqrt{T}$[1]	$\leqslant \|\theta\| X \sqrt{T \ln(\|\theta\| XT)}$ (Mhammedi and Koolen, 2020) $\geqslant (\|\theta\| X \wedge 1)\sqrt{T \ln((\|\theta\| X \wedge 1)T)}$[1] (McMahan and Streeter, 2012)

Table 1: Comparison of non-adaptive and adaptive rates for frequently used losses. All adaptive rates are achieved by scale-free algorithms, with no prior knowledge about the data.

Types of Scale-freeness Finally, we remark that the appropriate definition of 'scale-free' depends on the loss and setting. In OCO, the focus has been on algorithms whose predictions θ_t are invariant under scaling of the gradients g_t. Since this does not imply that $\theta_t^\top x_t$ is invariant under rescaling of x_t, Kempka et al. (2019); Mhammedi and Koolen (2020) add a post-processing step to scale θ_t to the range of $1/X_t$. While we consider only scale-freeness with respect to lengths of the whole vectors x_t, refined invariances with respect to the scale of individual features (Kempka et al., 2019; Orabona et al., 2015) or rotations (Mhammedi and Koolen, 2020) have also been studied. For the square loss, we also consider scale-freeness with respect to the data y_t. In non-parametric regression, the range of x_t is always assumed known, so we do not need to adapt to it.

Outline After preliminary definitions, we first study the logarithmic loss and logistic regression in Section 2. Then we consider the square loss and least-squares regression in Section 3.

Setting and Preliminaries We consider supervised online learning, in which the learner needs to issue a prediction $a_t \in \mathcal{A}$ for $y_t \in \mathcal{Y}$ at time t based on all the previous observations $\mathcal{S}_{t-1} =$

1. By reduction to linear loss, which works only when $UX \leqslant 1$; see Appendix D for details

$(x_1, y_1), \ldots, (x_{t-1}, y_{t-1}) \in (\mathcal{X} \times \mathcal{Y})^{t-1}$ as well as the features $x_t \in \mathcal{X}$. Given a loss function $\ell : \mathcal{A} \times \mathcal{Y} \to \mathbb{R} \cup \{\infty\}$, the performance of the learner after T time steps relative to a class of functions $\mathcal{F} = \{f_\theta : \mathcal{X} \to \mathcal{A} \mid \theta \in \Theta\}$ is evaluated by the regret

$$R_T(\theta) = \sum_{t=1}^{T} \ell(a_t, y_t) - \sum_{t=1}^{T} \ell(f_\theta(x_t), y_t) \qquad \text{for } \theta \in \Theta.$$

For logistic and square loss, define the running maximum of the feature norms $X_t = \max_{s \leqslant t} \|x_s\|$ and the responses $Y_t = \max_{s \leqslant t} \|y_s\|$, where the relevant norm will be clear from context.

2. Logarithmic Loss and Logistic Regression

For the log(arithmic) loss, the set of allowed predictions $\mathcal{A}$ corresponds to all probability density functions over $\mathcal{Y}$ with respect to some common σ-finite measure ν. Given a density $p \in \mathcal{A}$ and observation $y \in \mathcal{Y}$, the log loss is $\ell_{\text{log}}(p, y) = -\ln p(y)$.

To emphasize that predictions are densities (or probability mass functions if ν is the counting measure), we will write p_t instead of a_t and $p_{\theta,t}$ for $f_\theta(x_t)$. We will consider the log loss with respect to the normal location family and with respect to the multiclass logistic regression probability model, building our results on the Bayesian prediction strategy in both cases. Given a prior distribution π on Θ, the Bayesian prediction strategy predicts according to

$$p_t(y) = \int p_{\theta,t}(y) \,\mathrm{d}\pi(\theta \mid \mathcal{S}_{t-1}), \quad \text{where} \quad \mathrm{d}\pi(\theta \mid \mathcal{S}_{t-1}) = \frac{\prod_{s=1}^{t-1} p_{\theta,s}(y_s) \,\mathrm{d}\pi(\theta)}{\int \prod_{s=1}^{t-1} p_{\theta',s}(y_s) \,\mathrm{d}\pi(\theta')}, \qquad (1)$$

for which we assume throughout that the denominator is non-zero and finite. We also note that these definitions presume that the map $(\theta, y) \mapsto p_{\theta,t}(y)$ is measurable.

Adapting to a Hyperparameter The Bayesian prediction strategy can be applied directly to the normal location family or the logistic loss, but it may also be used to aggregate a finite or countable number of experts indexed by $\theta \in \Theta \subset \{0, 1, 2, \ldots\}$, whose predictions $p_{\theta,t}$ may vary arbitrarily over time. This fits into the general setting by letting $\mathcal{X} = \mathcal{A}^\Theta$ and $f_\theta(x) = x(\theta)$, with the interpretation that $p_{\theta,t} = f_\theta(x_t) = x_t(\theta)$ is the prediction of expert θ at time t. By making each expert correspond to a specific setting of a hyperparameter α, it then follows from Lemma 19 (Appendix A.3) that we can adapt to α with an overhead that is of order $O(\log \log \alpha)$:

Lemma 1 *Suppose that $A(\alpha)$ is an algorithm for the log loss that depends on hyperparameter $\alpha \geqslant 0$ and achieves a regret bound $B_T(\theta, \alpha) \geqslant R_T(\theta)$ for any $\theta \in \Theta_\alpha$, where $\Theta_\alpha \subseteq \Theta_\beta$ for $\alpha \leqslant \beta$. Then, for any $0 < \alpha_{\text{min}} < \alpha_{\text{max}} \leqslant \infty$, it is possible to adapt to $\alpha \leqslant \alpha_{\text{max}}$ with regret bounded by*

$$R_T(\theta) < \max_{\alpha' \in [\alpha, 2\alpha \vee \alpha_{\text{min}}]} B_T(\theta, \alpha') + 2 \ln \log_2 \left(\frac{8\alpha}{\alpha_{\text{min}}} \vee 4 \right) \quad \text{for all } \alpha \in [0, \alpha_{\text{max}}] \text{ and } \theta \in \Theta_\alpha \quad (2)$$

by aggregating experts $A(\alpha)$ for α in the exponential grid $\{\alpha_{\text{min}} 2^m \mid m = 0, 1, \ldots, M\}$ using the Bayesian prediction strategy with prior $\pi(m) = \frac{M+2}{(M+1)(m+1)(m+2)}$, where $M = \lceil \log_2(\alpha_{\text{max}}/\alpha_{\text{min}}) \rceil$.

In Appendix B, we recall how the Bayesian prediction strategy generalises to the Exponential Weights (EW) algorithm when replacing the log loss by *mixable* losses (Vovk, 2001). We apply this to the central case of the square loss in Section 3.

2.1. Warm-up: Normal Location Family

We start with the normal location family, which is simple because there are no features (i.e. $\mathcal{X}$ is a singleton), so scale-freeness is not an issue, and we can study comparator-adaptivity by itself. It also has the advantage that the non-adaptive minimax regret can be calculated in closed form. In this case ν is the Lebesgue measure, $\mathcal{Y} = \mathbb{R}^d$, and

$$p_{\theta,t}(y) = \frac{\exp\big(-\|y-\theta\|_2^2/(2\sigma^2)\big)}{(2\pi\sigma^2)^{d/2}},$$

with a fixed, known choice $\sigma > 0$ and $\Theta \subseteq \mathbb{R}^d$. We start with the exact minimax regret in the non-adaptive case, when θ is constrained to a ball of known radius U, and then apply Lemma 1 to adapt to U. As observed by Stine and Foster (2000); Barron et al. (1998), the minimax regret for $d = 1$ can be computed exactly when $\Theta = \{\theta : |\theta| \leqslant U\}$. The generalisation of their approach to higher dimensions gives the following:

Theorem 2 (Non-adaptive Minimax Rate) *For any $U > 0$, the minimax regret for the log loss with respect to the normal location family with $\Theta = \mathcal{B}(0, U)$ equals*

$$\min_{\mathbf{Algs}} \max_{\mathcal{S}} \max_{\theta \in \mathcal{B}(0,U)} R_T(\theta) = \frac{d}{2} \ln \frac{TU^2/\sigma^2}{2\Gamma(\frac{d}{2}+1)^{2/d}} + V(U,T), \tag{3}$$

where $V(U,T) = \ln\Big(1 + \frac{d}{\sqrt{T}U/\sigma}\int_0^\infty \Big(1 + \frac{r}{\sqrt{T}U/\sigma}\Big)^{d-1} e^{-r^2/2}\,\mathrm{d}r\Big) = O(1/\sqrt{T})$.

To interpret these expressions, note that $\Gamma(\frac{d}{2}+1)^{2/d} \approx \frac{d}{2e}$ by Stirling's approximation; in addition, $V(U,T)$ is a lower-order term, which simplifies to $V(U,T) = \ln(1 + \frac{\sqrt{\pi}}{\sqrt{2T}U/\sigma})$ for $d = 1$. We see here that the dependence on U is only logarithmic, rather than linear, which turns out to be common for curved losses when combined with parametric models. Adapting to U in Theorem 2 using Lemma 1 gives the following adaptive result, which may be viewed as the straightforward generalisation of Grünwald (2007, Examples 11.1 and 11.5) to $d > 1$:

Theorem 3 (Adaptive Rate) *There exists a learner whose regret for the log loss with respect to the normal location family with $\Theta = \mathbb{R}^d$ is at most*

$$R_T(\theta) \leqslant \frac{d}{2}\ln\frac{2T\|\theta\|^2/\sigma^2 + 1}{2\Gamma(\frac{d}{2}+1)^{2/d}} + 2\ln\log_2\Big(\frac{8T\|\theta\|^2}{\sigma^2} + 4\Big) + V(\|\theta\|, T) = O\bigg(\frac{d}{2}\ln\frac{T\|\theta\|^2}{\sigma^2}\bigg),$$

for all $\theta \in \mathbb{R}^d$, where $V(U,T) = O(1/\sqrt{T})$ is as in Theorem 2.

Comparing to Theorem 2, we see that the overhead for adaptivity is negligible compared to the non-adaptive rate. Adaptivity to T can be obtained by another application of Lemma 1 with $T = \alpha$.

Proof We apply Lemma 1 with $U = \sqrt{\alpha} = \|\theta\|$, $\alpha_{\text{min}} = \sigma^2/T$ and $\alpha_{\text{max}} = \infty$. The result then follows upon observing that $V(\sqrt{\alpha}, T)$ is decreasing in α. ■

2.2. Multiclass Logistic Regression

We proceed with multiclass logistic regression, which corresponds to the case where ν is the counting measure on K classes $\mathcal{Y} = \{1, \ldots, K\}$, and the corresponding probability mass functions are

$$p_{\theta,t}(y) = \frac{e^{h_\theta(x_t)_y}}{\sum_{y' \in \mathcal{Y}} e^{h_\theta(x_t)_{y'}}},$$

where $h_\theta : \mathcal{X} \to \mathbb{R}^K$ are predictors that map inputs $x \in \mathcal{X} \subseteq \mathbb{R}^d$ to vectors of class-scores. In particular, linear predictors $h_\theta(x) = \theta x$ are parameterised by weight matrices $\theta \in \Theta \subseteq \mathbb{R}^{K \times d}$ and lead to the multiclass logistic loss $\ell_{\log}(p_{\theta,t}, y) = \ln(1 + \sum_{y' \neq y} e^{(\theta x_t)_{y'} - (\theta x_t)_y})$ when combined with the log loss. The standard definition for binary logistic regression with a single vector $\theta' \in \mathbb{R}^d$ is recovered by setting $\Theta \subset \{\binom{\theta'}{0} : \theta' \in \mathbb{R}^d\}$. We call an algorithm for the logistic loss *scale-free* if scaling all x_t by the same positive constant does not change the predictions p_t.

We discuss linear predictors, both in terms of minimax rates and for the rates that are achievable by efficient algorithms. In Appendix A.3.1, we also consider adapting to the Besov norm of functions in non-parametric logistic regression.

2.2.1. LINEAR PREDICTORS

For linear predictors, there is a gap between the minimax rate and the best known upper bound for which there exists an efficient algorithm. We discuss the two cases in turn.

Let $\|\cdot\|$ be any norm, with corresponding dual norm $\|\cdot\|_*$, and define the following induced matrix norm: $\|\theta\| = \max_k \|\theta_k\|$, where θ_k is the k-th row of $\theta \in \mathbb{R}^{K \times d}$. Then Foster et al. (2018) provide the following upper and lower bounds for the minimax rate:

Theorem 4 (Non-adaptive Upper Bound, Foster et al. (2018)) *Suppose $\Theta \subseteq \{\theta \in \mathbb{R}^{K \times d} : \|\theta\| \leqslant U\}$ is a non-empty convex set, and $\mathcal{X} = \{x \in \mathbb{R}^d : \|x\|_* \leqslant X\}$. Then the Bayesian prediction strategy with uniform prior π on Θ satisfies*

$$R_T(\theta) \leqslant 5 d_\Theta \ln\left(\frac{UXT}{d_\Theta} + e\right) \qquad \text{for all } \theta \in \Theta,$$

where $d_\Theta \leqslant dK$ is the linear-algebraic dimension of Θ.

Indeed, the proof of this result by Foster et al. may be viewed as a specialisation of Lemma 19. (See Appendix A.3.) Many related results are shown by Shamir (2020), who also obtains tighter constants for binary logistic regression.

For UX larger than $\Omega(\sqrt{d} \ln T)$, Foster et al. (2018) further show a lower bound that matches their upper bound up to the dependence on T inside the logarithm, for the case of binary logistic regression with the L_2-norm[2]:

Theorem 5 (Lower Bound, Foster et al. (2018)) *Consider binary logistic regression with the L_2-norm, $\Theta = \{\binom{\theta'}{0} : \|\theta'\|_2 \leqslant U\}$ with $U = \Omega(\sqrt{d} \ln(T))$, and $\mathcal{X} = \{x \in \mathbb{R}^d : \|x\|_2 \leqslant 1\}$. Then*

$$\min_{\textbf{Algs}} \max_{\mathcal{S}} \max_{\theta \in \Theta} R_T(\theta) = \Omega\Big(d \ln\Big(\frac{U}{\sqrt{d} \ln T}\Big) \wedge T\Big).$$

2. In restating their result, we add a minimum with T, which appears to be missing from Foster et al. (2018, Lemma 4).

We complement this by the following lower bound, which matches the upper bound from Theorem 4 for the regime where UX is smaller than $O(\sqrt{d})$:

Theorem 6 (Lower Bound) *Consider binary logistic regression with the L_2-norm, $\Theta = \{\left(\begin{smallmatrix}\theta' \\ 0\end{smallmatrix}\right) : \|\theta'\|_2 \leqslant U\}$ and $\mathcal{X} = \{x \in \mathbb{R}^d : \|x\|_2 \leqslant X\}$ with $UX \leqslant 2\sqrt{d}$ and $T > d$. Then the minimax regret is at least*

$$\min_{\textbf{Algs}} \max_{\mathcal{S}} \max_{\theta \in \Theta} R_T(\theta) \geqslant d \ln\Big(\frac{UX\sqrt{T-d}}{4\sqrt{\pi}d} - \frac{2}{\sqrt{\pi(T/d-1)}}\Big) = \Omega\Big(d\ln\Big(\frac{UX\sqrt{T}}{d}\Big)\Big). \tag{4}$$

Shamir (2020) obtains related lower bounds, asymptotically when $T \to \infty$, but he does not spell out their dependence on U explicitly. Combining Theorem 4 with Lemma 1 to adapt to U, and instantiating $\Theta_U = \{\theta \in \mathbb{R}^{K \times d} : \|\theta\| \leqslant U\}$ for concreteness, gives the following scale-free adaptive result:

Theorem 7 (Scale-Free, Adaptive) *Let $\Theta = \mathbb{R}^{K\times d}$ and $\mathcal{X} = \mathbb{R}^d$. Then, for any $\varepsilon > 0$, there exists a scale-free strategy for the learner that guarantees*

$$R_T(\theta) \leqslant 5dK \ln\Big(\frac{2\|\theta\|X_T T}{dK} + \frac{\varepsilon X_T}{X_{t^*}} + e\Big) + 2\ln\Big(\log_2\Big(\frac{8\|\theta\|X_{t^*}T}{\varepsilon dK} \vee 4\Big)\Big) \qquad \text{for all } \theta \in \mathbb{R}^d,$$

where t^* is the first t such that $\|x_t\| > 0$.

We see that the overhead for adaptivity becomes negligible if we can take $\varepsilon = O(X_{t^*}/X_T)$. There is no automatic way available to achieve this completely for free, because X_T is unknown at the start of the algorithm, but there are two reasonable solutions: the first is to just take ε to be "very small", which is still fine in the bound, because the $O(\log\log(1/\varepsilon))$ term hardly grows with $1/\varepsilon$. In particular, even for the smallest possible positive value representable in a double precision floating point number, e.g. $\varepsilon \approx 2.2 \times 10^{-308}$, we still have that $\ln(\log_2(1/\varepsilon)) \leqslant 7$. The second solution is to aggregate multiple copies of the algorithm using Lemma 1 with $\varepsilon = e/\alpha = eX_{t^*}/X_T$, $\alpha_{\min} = 1$, $\alpha_{\max} = \infty$. We then obtain the following parameter-free result:

Theorem 8 (Scale-Free, Adaptive, Parameter-Free) *Let $\Theta = \mathbb{R}^{K\times d}$ and $\mathcal{X} = \mathbb{R}^d$. Then there exists a scale-free strategy for the learner that guarantees*

$$R_T(\theta) \leqslant 5dK \ln\Big(\frac{2\|\theta\|X_T T}{dK} + 2e\Big) + 2\ln\Big(\log_2\Big(\frac{16\|\theta\|X_T T}{edK} \vee 4\Big)\Big) + 2\ln\Big(\log_2\Big(\frac{8X_T}{X_{t^*}}\Big)\Big)$$

for all $\theta \in \mathbb{R}^d$, where t^* is the first t such that $\|x_t\| > 0$.

The term $2\ln(\log_2(\frac{8X_T}{X_{t^*}}))$ is again unbounded in theory, but it is at most 14 when X_T/X_{t^*} is restricted to the range of double precision floating point numbers, which goes up to 1.8×10^{308}. This seems acceptable for all practical purposes.

Efficient Algorithms Since the Bayesian algorithm from Theorem 4 is not computationally efficient, efficient algorithms based on quadratic approximations of the losses have been developed: by Jézéquel et al. (2020) for binary logistic regression and by Agarwal et al. (2021); Jézéquel et al. (2021) for the multiclass case. (In a different context, Mourtada and Gaïffas (2022) also obtain an efficient algorithm for misspecified *offline* logistic regression.) These efficient methods achieve worse regret rates, however, of order $O(dKU\ln(T))$, with a linear rather than logarithmic dependence on U. The state of the art for the multiclass case is the algorithm of Agarwal et al. (2021), which achieves the following run-time and regret bound with respect to the $2 \to \infty$-norm of θ, which is defined as $\|\theta\|_{2,\infty} = \sup_{x:\|x\|_2 \leqslant 1} \|\theta x\|_\infty$:

Theorem 9 (Non-adaptive, Efficient Algorithm, Agarwal et al. (2021)) *Suppose the set of parameters is $\Theta = \{\theta \in \mathbb{R}^{K\times d} : \|\theta\|_{2,\infty} \leqslant U\}$, and $\mathcal{X} = \{x \in \mathbb{R}^d : \|x\|_2 \leqslant X\}$. Then there exists a learning algorithm (depending on U and X, but not on T) that achieves*

$$R_T(\theta) = O\Big(\big(UX + \ln K\big)dK\ln T\Big) \qquad \text{for all } \theta \in \Theta,$$

and runs in time $O(d^2K^3 + UXK^2\ln(t(1+UX)))$ per round t.

The absolute constants in the theorem depend on a trade-off between optimisation accuracy and run-time, which Agarwal et al. (2021) leave open; we assumed here that the optimisation accuracy is $\text{poly}(1/t)$.

Since the dependence on U is now linear, adaptation to the norm of θ becomes a much more pressing issue. We pursue this with computational considerations in mind. Our starting point is the observation that the algorithm from Theorem 9 is only computationally efficient if both $UX \leqslant T^\beta$ and $d^2K \leqslant T^\beta\ln(T)$ for some small $\beta > 0$, in which case its run-time is $O(K^2T^\beta\ln(T))$ per round. We will design a scale-free adaptive algorithm with the same run-time by using the doubling trick to adapt to $X \leqslant T^\beta/U$ and then applying Lemma 1 with $U = \alpha$. In this case choosing α_{max} to be finite is desirable for computational reasons, because it reduces the number of copies of the base algorithm that we need to run to $\lceil\log_2(\alpha_{\text{max}}/\alpha_{\text{min}})\rceil$. A good choice for α_{max} exists, because $\alpha_{\text{max}} = T^\beta/X_{t^*} \geqslant T^\beta/X_T$ is sufficient to cover all U such that $UX_T \leqslant T^\beta$, where t^* is again the smallest t such that $\|x_t\|_2 > 0$. We then choose $\alpha_{\text{min}} = T^{-\gamma}/X_{t^*}$ for γ as large as possible given the computational budget, leading to the following result:

Theorem 10 (Scale-Free, Adaptive, Efficient Algorithm) *Let $\Theta = \mathbb{R}^{K\times d}$ and $\mathcal{X} = \mathbb{R}^d$. Then, for any $\beta > 0$ and $c > 0$, there exists a learner that achieves*

$$\begin{aligned} R_T(\theta) = O\Big(&\Big(\|\theta\|_{2,\infty}X_T + \ln K + T^{-cT^\beta/(d^2K)}\frac{X_T}{X_{t^*}}\Big)dK\ln T\Big) \\ &+ 2\ln\Big(\log_2\Big(8\|\theta\|_{2,\infty}X_T T^{\frac{cT^\beta}{d^2K}} \vee 4\Big)\Big) \qquad \text{for all } \theta \text{ such that } \|\theta\|_{2,\infty}X_T \leqslant T^\beta, \end{aligned} \tag{5}$$

where t^* is the first t such that $\|x_t\|_2 > 0$. Furthermore, this learner runs in time complexity $O(d^2K^3 + (1+c)K^2T^\beta\ln(T))$ per round.

Apart from the term involving X_T/X_{t^*}, this rate matches the non-adaptive rate for any θ such that $\|\theta\|_{2,\infty}X_T \leqslant T^\beta$, at no cost in the run-time compared to the efficient non-adaptive algorithm (see discussion above). Like in Theorem 8, the dependence on X_T/X_{t^*} is very minor: for $\beta = 1$, $c = 10$ and $\frac{X_T}{X_{t^*}} \leqslant 1.8 \times 10^{308}$ (the maximum value of a double precision float), we then have $T^{-cT^\beta/(d^2K)}\frac{X_T}{X_{t^*}} \leqslant 1$ as soon as $T \geqslant 23d^2K$.

3. Square Loss and Least-Squares Regression

Consider the square loss $\ell(a, y) = \|a - y\|^2/2$ over domains $\mathcal{A} = \mathcal{Y} = \mathbb{R}^d$. For both prediction with the square loss (Section 3.2) and for regression (Section 3.3), Exponential Weights (EW) with clipping adapts to the range of the data and to the norm of the comparator at essentially no cost.

The square loss is $1/(4Y^2)$-mixable over $\mathcal{A} = \mathcal{Y} = \mathcal{B}(0, Y)$, with the mean as a substitution function. In contrast with the log-loss, the mixability constant depends on the domain of the data points, preventing the application of the EW aggregation scheme of Lemma 21 (in Appendix B). We get around this issue thanks to a clipping trick of Cutkosky (2019).

For the square loss, there are two requirements to scale-freeness: the predictions a_t should *not* change when all x_t are scaled by the same constant, but, if all y_t are scaled by a constant, then the predictions *should* scale by the same constant.

3.1. An Aggregation Procedure Tailored to the Square Loss

In order to adapt to arbitrary hyperparameters without knowledge of the range of data Y, we propose a general aggregation scheme built upon the EW strategy with clipping. Throughout this section, we denote by Π_Y the projection to the ball of radius Y.

Cutkosky Clipping To get around the issue of tuning the learning rate in the EW strategy, we apply a trick from Cutkosky (2019), which is to feed an algorithm clipped data points

$$\tilde{y}_t := \frac{Y_{t-1}}{Y_t} y_t = \begin{cases} \frac{Y_{t-1}}{\|y_t\|} y_t & \text{if } \|y_t\| \geqslant Y_{t-1} \\ y_t & \text{otherwise,} \end{cases} \tag{6}$$

where $Y_t = \max_{s \leqslant t} \|y_s\|$. Then the algorithm knows in advance that its next data point will be bounded by Y_{t-1}. This is exactly the knowledge needed to tune the learning rate when using EW. *A priori*, one would need to ensure that the actions of the experts are bounded by Y_{t-1} to satisfy mixability. It turns out this is not necessary: it suffices to also feed clipped actions to EW.

Adapting Suppose that $A(\alpha)$ is an algorithm for the square loss that depends on a hyperparameter $\alpha \in [\alpha_{\min}, \alpha_{\max})$, with $0 \leqslant \alpha_{\min} \leqslant \alpha_{\max} \leqslant \infty$. Consider the grid of parameter values $\{2^m \alpha_{\min} | \, m = 0, \ldots, M\}$ where $M = \lceil \log_2(\alpha_{\max}/\alpha_{\min}) \rceil$. For a sequence of data points (y_t, x_t), denote by $a_{t,\alpha}$ the output of $A(\alpha)$ at time t. Apply the EW strategy based on the clipped data points $\tilde{y}_t$ and the clipped actions $\Pi_{Y_{t-1}}(a_{t,\alpha})$, with learning rate $\eta_t = 1/(4Y_{t-1}^2)$, and prior $\pi(m) = \frac{M+2}{(M+1)(m+1)(m+2)}$. The next result is an analogue to Lemma 1 for the square loss.

Lemma 11 *Let $A(\alpha)$ be an algorithm that achieves a regret bound $B_T(\theta, \alpha) \geqslant R_T(\theta)$ for any $\alpha \in [\alpha_{\min}, \alpha_{\max})$ and $\theta \in \Theta$. Then, it is possible to adapt to $\alpha \leqslant \alpha_{\max}$ with regret bounded by*

$$R_T(\theta) \leqslant \max_{\alpha' \in [\alpha, 2\alpha \vee \alpha_{\min}]} B_T(\theta, \alpha') + 8Y_T^2 \ln \log_2 \left(\frac{8\alpha}{\alpha_{\min}} \vee 4 \right) + 2Y_T^2 \quad \textit{for all } \theta \in \Theta \,. \tag{7}$$

Furthermore, if all algorithms $A(\alpha)$ are scale-free, then the aggregated procedure is scale-free as well.

3.2. Square Loss Prediction

Let us apply the results built above to the case of prediction with the square loss. In this case, $\Theta = \mathcal{A} = \mathbb{R}^d$, there are no observed features (i.e. $\mathcal{X}$ is a singleton), and the actions functions are $a_\theta \equiv \theta$. Slightly abusing notation, we shall denote $\ell(\theta, y) = \|\theta - y\|^2/2$.

Aggregated Gradient Descent We apply the aggregation procedure of Lemma 11 to multiple instances of Online Gradient Descent (cf. Appendix C.2.1) with step sizes $1/(\lambda + t)$, where we aggregate over $\lceil \log_2 T \rceil$ values of $\lambda = \alpha$ from $\alpha_{\min} = 1$ to $\alpha_{\max} = T$. Note that the clipping of the actions has no effect in this case, since the individual updates of every expert are already in $\mathcal{B}(0, Y_{t-1})$ at every t.

Theorem 12 *In square loss prediction, there exists a scale-free algorithm such that for any $\theta \in \mathbb{R}^d$,*

$$R_T(\theta) \leqslant 2Y_T^2 \log\left(2 + T\left(\frac{\|\theta\|^2}{Y_T^2} \wedge 1\right)\right) + 8Y_T^2 \ln \log_2\left(\frac{8Y_T^2}{\|\theta\|^2} \vee 8\right) + 3Y_T^2 .$$

The per-round computation time is $\lceil \log_2 T \rceil$ times the cost of a gradient descent update, with guarantees that match the non-adaptive lower bound of Theorem 18 for $d = 1$ and constant features.

3.3. Least-Squares Regression

Upon observing a feature point $x_t \in \mathcal{X}$, the learner outputs a predictions $a_t \in \mathcal{A} = \mathbb{R}$, then receives the answer $y_t \in \mathcal{Y} = \mathbb{R}$. The learnes competes against functions $\theta \in \mathcal{F}$ where $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$ is a known set of comparator functions. We assume $\mathcal{F}$ is a separately implementable Reproducing Kernel Hilbert Space (RKHS) with the kernel k over $\mathcal{X}$ (Gammerman et al., 2004, Definition 1). We refer to the kernel matrix $K_T = (k(x_s, x_t))_{(s,t) \in [T]^2}$ and to $X_t = \max_{s \in [t]} k(x_s, x_s)$.

Kernel methods are useful when the algorithm depends on the feature vectors $x_1, x_2, \dots$ exclusively via the quantities $k(x_s, x_t)$, thanks to the kernel trick. For such learners, X-scale-invariance generalises to invariance by scaling of k by a positive factor.

Analyses of (non-)parametric regression focus on the asymptotic dependence on the number of data points T. When $\mathcal{F}$ is a rich class of functions, even for the parametric case in large dimension, the norm of the comparator impacts the rates, hence the importance of adaptation. We show that adapting to $\|\theta\|$ comes at no cost on the regret, with a scale-free algorithm.

Aggregated-KAAR A key algorithm in online regression is the Azoury-Vovk-Warmuth forecaster (also called the forward algorithm) from Azoury and Warmuth (2001); Vovk (1998), and its kernelised version KAAR (Gammerman et al., 2004). This algorithm has been analysed and modified in a variety of settings; see, e.g., Orabona et al. (2015); Jézéquel et al. (2019); Gaillard et al. (2019); Jézéquel et al. (2019); Zadorozhnyi et al. (2021). Notably, for linear regression Gaillard et al. (2019) notice that the Vovk-Azoury-Warmuth forecaster with regularization parameter set to $\lambda = 0$ is scale-free and enjoys a regret bound that is optimal up to an additive term that is a constant for many reasonable sequences of feature vectors x_t, but can potentially blow up. Upon seeing x_t, KAAR with regularisation $\lambda > 0$ predicts $a_t = \theta_t(x_t)$ where $\theta_t \in \mathcal{F}$ is picked according to the rule

$$\theta_t = \operatorname*{argmin}_{\theta \in \mathcal{F}} \left\{ \frac{\lambda \|\theta\|^2}{2} + \frac{1}{2}\theta(x_t)^2 + \frac{1}{2}\sum_{s=1}^{t-1} (\theta(x_s) - y_s)^2 \right\} . \tag{8}$$

The update θ_t admits the closed-form expression

$$\theta_t(x) = (y_1, \dots, y_{t-1}, 0)^\top (\lambda I_t + K_t)^{-1} \big(k(x_1, x), \dots, k(x_t, x)\big) . \tag{9}$$

To ensure scale-invariance, we tune the regularisation proportionally to the first non-zero feature (and predict 0 until there is one); we call the ensuing algorithm KAAR-sf(α), which will be the building block in our aggregated algorithm.

For clarity, let us ignore computational issues and run an infinite number of instances. We apply Lemma 11 twice in order to aggregate both arbitrary small and arbitrary large values of α. Doing so, the dependence on the initial guess of the correct scale moves into a $\log\log$ factor.

Theorem 13 (Adaptive, Scale-Free) *In kernel least-squares regression, there exists a scale-free algorithm such that for any $\lambda > 0$,*

$$R_T(\theta) \leqslant \frac{\lambda \|\theta\|^2}{2} + \frac{Y_T^2}{2} \ln \det \left(I_T + \frac{1}{\lambda} K_T \right) + 8 Y_T^2 \ln \left(\frac{3e}{2} \left| \log_2 \left(\frac{\lambda}{X_{t^\star}^2} \right) \right| \right) . \tag{10}$$

where $Y = \max_{t \in [T]} |y_t|$ and $K_t = \big(k(x_u, x_v)\big)_{(u,v) \in [T]^2}$.

This new algorithm, which we call A-KAAR, is scale-free and enjoys the same guarantees as best tuned KAAR, up to a small $\log\log$ term. We state a first consequence that holds for any kernel.

Corollary 14 (Dimension-Free, Scale-Free) *A-KAAR enjoys the dimension-free regret bound*

$$R_T(\theta) \leqslant X_T Y_T \|\theta\| \sqrt{T} + 8 Y_T^2 \ln \left(\frac{3e}{2} \left| \log_2 \left(\frac{X_T Y_T \sqrt{T}}{\|\theta\| X_{t^\star}^2} \right) \right| \right) . \tag{11}$$

In the worst-case, the regret of A-KAAR grows at most at an $XY\|\theta\|\sqrt{T}$ rate. This matches the non-adaptive lower bound of Theorem 18 in the large-dimensional regime. Faster rates are achievable under additional assumptions on $\mathcal{F}$, as we shall see now.

3.3.1. PARAMETRIC REGRESSION

In this case, the set $\mathcal{F}$ is the set of linear functions over $\mathbb{R}^d$, identified with $\mathbb{R}^d$. Then $\mathcal{X} = \mathbb{R}^d$ and $k(x, x') = \langle x, x' \rangle$ and $X_t = \max_{s \in [T]} \|x_s\|$ and the RKHS norm is the Euclidean norm. KAAR specialises to the VAW forecaster, and its aggregated version enjoys the adaptive upper bound:

Theorem 15 *In d-dimensional linear regression, A-KAAR guarantees that for any $\theta \in \mathbb{R}^d$,*

$$R_T(\theta) \leqslant \frac{dY_T^2}{2} \ln \left(1 + \frac{T \|\theta\|^2 X_T^2}{d^2 Y_T^2} \right) + \frac{dY_T^2}{2} + 8 Y_T^2 \ln \left(\frac{3e}{2} \left| \log_2 \left(\frac{dY_T^2}{\|\theta\| X_{t^\star}^2} \right) \right| \right) .$$

The bound matches the non-adaptive lower bound of Theorems 18. Note that this implies a uniform regret bound over comparators in $\mathbb{R}^d$, by instantiating the comparator θ to be a minimiser $\theta^\star$ of the least-squares error on the data, i.e. a maximiser of the regret. See Corollary 3 in Gaillard et al. (2019) and its proof for upper bounds on the norm of $\theta^\star$.

3.3.2. COMPARATOR-ADAPTIVE BOUNDS UNDER THE CAPACITY CONDITION

In typical uses, $\mathcal{F}$ is vastly richer than a set of linear functions. The effective dimension $d_{\text{eff}}(\lambda)$ of the features K_T at scale λ (Zhang, 2003) provides a standard data-dependent complexity measure of $\mathcal{F}$ (cf. Appendix C.5). The space $\mathcal{F}$ is said to satisfy the γ-capacity condition if for any sequence of features of length T and for any $\lambda > 0$, the effective dimension grows at most at a rate of $d_{\text{eff}}(\lambda) \leqslant (C_k T/\lambda)^\gamma$ for some $C_k > 0$. Under this condition, the second term in (10) is polynomial in T/λ, and we obtain the following rates.

Theorem 16 *If $\mathcal{F}$ satisfies the γ-capacity condition, then A-KAAR guarantees thhat*

$$R_T(\theta) \leqslant \tilde{\mathcal{O}}\big(Y_T^{2/(1+\gamma)} \|\theta\|^{2\gamma/(1+\gamma)} T^{\gamma/(1+\gamma)}\big) \quad \textit{for any } \theta \in \mathcal{F}\,.$$

A finite-time version of the bound depending on X_T, $X_{t^\star}$ is available in (20), in Appendix C.5 .

The capacity condition is satisfied, e.g. when $\mathcal{F}$ is a space of smoothing splines (Zhang, 2003, Section 4), or for Sobolev spaces; we detail this application in the next section.

Comparator-Adaptive Regression over Sobolev Spaces By Theorem 3 of Zadorozhnyi et al. (2021), the results above imply adaptive rates when the class of functions is the Sobolev space $W_{s,p}([-1,1]^d)$ with $p \geqslant 2$; we refer the reader to Adams and Fournier (2003) for definitions and properties of Sobolev spaces, and to Wendland (2004, Chapter 10) for more details on Sobolev spaces as RKHS. For simplicity, let us state the results in the case when s is an integer and $s \geqslant d/2$. For fractional s, the same rates are valid, up to a T^ε factor with ε arbitrarily small, and the rates change when $s < d/2$.

Corollary 17 *For $\mathcal{F} = W_{s,p}([-1,1]^2)$ for $s \in \mathbb{N}$ with $s \geqslant d/2$, there exists an algorithm such that*

$$R_T(\theta) \leqslant \tilde{\mathcal{O}}\Big(Y_T^{4s/(2s+d)} \|\theta\|_{s,p}^{2d/(2s+d)} T^{d/(2s+d)}\Big) \quad \textit{for any } \theta \in W_{s,p}([-1,1]^2)\,.$$

The exponent on T is optimal (Rakhlin and Sridharan, 2014; Zadorozhnyi et al., 2021, Thm 9).

Efficient Methods The updates (9) can be computed in $\mathcal{O}(t^2)$ time and memory, and both these complexities can be improved for specific kernels. Running EW over experts KAAR-sf(α) with small α_{min} and large α_{max} would be an implementable strategy in cases when single instances are efficient, at the cost of limiting the adaptivity to a specific range of values of $\|\theta\|$. Jézéquel et al. (2019) build a faster version of KAAR enjoying essentially the same regret bound as (17), but with better computational complexity for large T. The aggregation we propose applies to this algorithm too, and would incorporate the improvements in computational complexity.

3.4. Lower Bounds

The main lower bound for the square loss, in Theorem 18, provides the optimal asymptotic rate together with the dependence on the comparator. While the dependence on T is a standard result in the literature (Takimoto and Warmuth, 2000; Abernethy et al., 2008; Hazan et al., 2007; Gaillard et al., 2019), we did not find a version of the lower bound that provided the dependence on U. We thus refined the proof of Vovk (2001, Theorem 2). In the large-dimensional regime where $d \geqslant T(UX/Y)^2$, the finite-time version of the first lower bound is vacuous. The effect of the curvature becomes negligible and rates behave like in the linear loss case, as shown in the upper bound (Corollary 14) and in the matching lower bound.

Theorem 18 *Fix $X, Y, U > 0$. In linear least-squares regression over $\mathbb{R}^d$, if $d(Y/(UX))^2 \geqslant 1$, then for any algorithm,*

$$\sup_{\mathcal{S}} \sup_{\theta \in \mathcal{B}(0,U)} R_T(\theta) \geqslant \begin{cases} \frac{dY^2}{2} \log\left(\frac{TU^2X^2}{d^2Y^2}\right) + \mathcal{O}(\log\log T) & \textit{as} \quad T \to \infty\,, \\ (\sqrt{2}/8)UXY\sqrt{T} & \textit{for} \quad T \leqslant (d/8)(Y/(UX))^2\,. \end{cases}$$

Moreover, if $d = 1$, the first bound holds with $x_t = X$ for all t.

A finite-time version of the first bound can be found in (22) in Appendix C.6.

4. Discussion, Conclusions and Future Work

We have shown that scale-free algorithms can adapt to the norm of the comparator at almost no cost in common learning scenarios. While we have endeavored to complete the story, some points remain open. We note that the case of strongly convex losses with fixed strong-convexity parameter μ should be treatable by a proof directly analogous to that of Theorem 12. Time-varying μ or strong convexity with respect to other Bregman divergences (c.f. (Hazan et al., 2008)) would not be as easy, because the former would affect the mixability of the loss, and the latter might break (15). Additionally, avoiding $O(\log\log \frac{X_T}{X_{t^*}})$ terms in the logistic/least squares linear regression cases would be desirable, at least from a theoretical perspective; Gerchinovitz (2011) also observe this, while Gaillard et al. (2019) avoid it but in exchange find a different complicated dependence on the features. Lastly, it would be of interest to have comparator-adaptive lower bounds for regression in Sobolev spaces, with an explicit dependence on both Y and $\|\theta\|_{s,p}$.

Acknowledgments

All authors were supported by the Netherlands Organization for Scientific Research (NWO) under grant number VI.Vidi.192.095.

References

Jacob Abernethy, Peter L Bartlett, Alexander Rakhlin, and Ambuj Tewari. Optimal strategies and minimax lower bounds for online convex games, 2008.

Robert A. Adams and John J. F. Fournier. *Sobolev spaces*. Elsevier, 2003.

Dmitry Adamskiy, Wouter M. Koolen, Alexey Chernov, and Vladimir Vovk. A closer look at adaptive regret. *Journal of Machine Learning Research*, 17(23):1–21, 2016.

Naman Agarwal, Satyen Kale, and Julian Zimmert. Efficient methods for online multiclass logistic regression, 2021.

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

Andrew Barron, Jorma Rissanen, and Bin Yu. The minimum description length principle in coding and modeling. *IEEE Transactions on Iinformation Theory*, 44(6), 1998.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013.

Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.

Ashok Cutkosky. Artificial constraints and hints for unbounded online learning. In *Proceedings of the 32nd Annual Conference on Learning Theory*, volume 99, pages 874–894. PMLR, 25–28 Jun 2019.

Ashok Cutkosky and Kwabena A. Boahen. Online learning without prior information. In *Proceedings of The 30th Annual Conference on Learning Theory*, 2017.

Ashok Cutkosky and Francesco Orabona. Black-box reductions for parameter-free online learning in banach spaces. In *Proceedings of the 31st Conference On Learning Theory*, volume 75, pages 1493–1529. PMLR, 06–09 Jul 2018.

A Philip Dawid. Statistical theory: the prequential approach. *Journal of the Royal Statistical Society: Series A (General)*, 147(2):278–290, 1984.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(61):2121–2159, 2011.

Dylan J Foster, Satyen Kale, Haipeng Luo, Mehryar Mohri, and Karthik Sridharan. Logistic regression: The importance of being improper. In *Proceedings of the 31st Annual Conference On Learning Theory*, pages 167–208. PMLR, 2018.

Yoav Freund, Robert E. Schapire, Yoram Singer, and Manfred K. Warmuth. Using and combining predictors that specialize. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, STOC '97, pages 334–343, 1997.

Pierre Gaillard, Sébastien Gerchinovitz, Malo Huard, and Gilles Stoltz. Uniform regret bounds over R^d for the sequential linear regression problem with the square loss. *Proceedings of Machine Learning Research*, 98:404–432, 2019.

Alex Gammerman, Yuri Kalnishkan, and Vladimir Vovk. On-line prediction with kernels and the complexity approximation principle. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, UAI '04, pages 170–176, 2004.

Sébastien Gerchinovitz. Sparsity regret bounds for individual sequences in online linear regression. In *Proceedings of the 24th Annual Conference on Learning Theory*, volume 19, pages 377–396. PMLR, 09–11 Jun 2011.

Peter D. Grünwald. *The minimum description length principle*. MIT press, 2007.

Elad Hazan. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016. ISSN 2167-3888. doi: 10.1561/2400000013.

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2–3):169–192, dec 2007.

Elad Hazan, Alexander Rakhlin, and Peter Bartlett. Adaptive online gradient descent. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2008.

Rémi Jézéquel, Pierre Gaillard, and Alessandro Rudi. Efficient online learning with kernels for adversarial large scale problems. In *Advances in Neural Information Processing Systems*, volume 32, pages 9432–9441, 2019.

Rémi Jézéquel, Pierre Gaillard, and Alessandro Rudi. Efficient improper learning for online logistic regression. In *Proceedings of 33rd Annual Conference on Learning Theory*, volume 125, pages 2085–2108. PMLR, 09–12 Jul 2020.

Rémi Jézéquel, Pierre Gaillard, and Alessandro Rudi. Mixability made efficient: Fast online multiclass logistic regression. In *Advances in Neural Information Processing Systems (pre-proceedings)*, volume 34, 2021.

Sham M Kakade and Andrew Ng. Online bounds for bayesian algorithms. *Advances in neural information processing systems*, 17, 2004.

Michal Kempka, Wojciech Kotłowski, and Manfred K. Warmuth. Adaptive scale-invariant online algorithms for learning linear models. In *International Conference on Machine Learning*, pages 3321–3330. PMLR, 2019.

Wouter M. Koolen. Exploiting curvature using exponential weights. Blog post, September 2016.

Wojciech Kotłowski. Scale-invariant unconstrained online learning. In *International Conference on Algorithmic Learning Theory*, pages 412–433. PMLR, 2017.

Olivier Marchal and Julyan Arbel. On the sub-Gaussianity of the Beta and Dirichlet distributions. *Electronic Communications in Probability*, 22:1 – 14, 2017.

Brendan McMahan and Jacob Abernethy. Minimax optimal algorithms for unconstrained linear optimization. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

Brendan McMahan and Matthew Streeter. No-regret algorithms for unconstrained online convex optimization. In *Advances in Neural Information Processing Systems*, volume 25, 2012.

Zakaria Mhammedi and Wouter M. Koolen. Lipschitz and comparator-norm adaptivity in online learning. In *Proceedings of 33rd Conference on Learning Theory*, volume 125, pages 2858–2887. PMLR, 09–12 Jul 2020.

Jaouad Mourtada and Stéphane Gaïffas. An improper estimator with optimal excess risk in misspecified density estimation and logistic regression. *J. Mach. Learn. Res.*, 23:31–1, 2022.

Francesco Orabona and Dávid Pál. Scale-free online learning. *Theoretical Computer Science*, 716: 50–69, 2018.

Francesco Orabona, Koby Crammer, and Nicolò Cesa-Bianchi. A generalized online mirror descent with applications to classification and regression. *Machine Learning*, 99(3):411–435, 2015.

Erik Ordentlich and Thomas M. Cover. The cost of achieving the best portfolio in hindsight. *Mathematics of Operations Research*, 23(4), 1998.

Alexander Rakhlin and Karthik Sridharan. Online non-parametric regression. In *Proceedings of The 27th Annual Conference on Learning Theory*, pages 1232–1264. PMLR, 2014.

Alexander Rakhlin, Karthik Sridharan, and Ambuj Tewari. Online learning via sequential complexities. *Journal of Machine Learning Research*, 16(6):155–186, 2015.

Stéphane Ross, Paul Mineiro, and John Langford. Normalized online learning. In *Uncertainty in Artificial Intelligence*, page 537, 2013.

Gil I. Shamir. Logistic regression regret: What's the catch? In *Proceedings of the 33rd Annual Conference on Learning Theory*, pages 3296–3319. PMLR, 2020.

Robert A. Stine and Dean P. Foster. The competitive complexity ratio. In *Conference on Information Sciences and Systems*, 2000.

Eiji Takimoto and Manfred Warmuth. The minimax strategy for gaussian density estimation. In *Proceedings of the 13th Annual Conference on Learning Theory*, pages 100–106, 2000.

Dirk van der Hoeven, Tim van Erven, and Wojciech Kotłowski. The many faces of exponential weights in online learning. In *Proceedings of the 31st Annual Conference On Learning Theory*, pages 2067–2092. PMLR, 2018.

Volodya Vovk. Competitive on-line linear regression. In *Advances in Neural Information Processing Systems*, volume 10, 1998.

Volodya Vovk. Competitive on-line statistics. *International Statistical Review*, 69(2):213–248, 2001.

Holger Wendland. *Scattered data approximation*, volume 17. Cambridge university press, 2004.

Olivier Wintenberger. Optimal learning with bernstein online aggregation. *Machine Learning*, 106 (1):119–141, 2017.

Qun Xie and Andrew R. Barron. Asymptotic minimax regret for data compression, gambling, and prediction. *IEEE Transactions on Information Theory*, 46(2):431–445, 2000.

Oleksandr Zadorozhnyi, Pierre Gaillard, Sebastien Gerschinovitz, and Alessandro Rudi. Online nonparametric regression with sobolev kernels. arXiv preprint arXiv:2102.03594, 2021.

Tong Zhang. Effective dimension and generalization of kernel learning. In *Advances in Neural Information Processing Systems*, volume 15, 2003.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Machine Learning, Proceedings of the 20th International Conference (ICML)*, pages 928–936, August 21-24 2003.

Appendix A. Proofs from Sections 2

A.1. Proofs for the Aggregation

Guarantees for our aggregation scheme derive from a straightforward application of the following standard guarantee for the Bayesian prediction strategy (see, e.g., Section 10 in Dawid (1984) or Lemma 2.1 of Kakade and Ng (2004)).

Lemma 19 *The Bayesian prediction strategy with prior π achieves*

$$\sum_{t=1}^{T} \ell_{\log}(p_t, y_t) \leqslant \mathop{\mathbb{E}}_{\theta\sim\gamma}\left[\sum_{t=1}^{T} \ell_{\log}(p_{\theta,t}, y_t)\right] + \mathrm{KL}(\gamma\|\pi) \qquad \textit{for all distributions } \gamma, \tag{12}$$

with equality if $\gamma = \pi(\theta \mid \mathcal{S}_T)$.

Proof By telescoping, the cumulative loss of the Bayesian prediction strategy simplifies to

$$\begin{aligned}\sum_{t=1}^{T} \ell_{\log}(p_t, y_t) &= \sum_{t=1}^{T} -\ln \frac{\int \prod_{s=1}^{t} p_{\theta,s}(y_s)\,\mathrm{d}\pi(\theta)}{\int \prod_{s=1}^{t-1} p_{\theta,s}(y_s)\,\mathrm{d}\pi(\theta)} = -\ln \int \prod_{t=1}^{T} p_{\theta,t}(y_t)\,\mathrm{d}\pi(\theta)\\ &= -\ln \int e^{-\sum_{t=1}^{T} \ell_{\log}(p_{\theta,t}, y_t)}\,\mathrm{d}\pi(\theta).\end{aligned}$$

The result then follows by recognising the right-hand side as minus the convex conjugate of the Kullback-Leibler divergence (i.e., by applying the Donsker-Varadhan lemma (Boucheron et al., 2013, Corollary 4.14)). ■

Proof of Lemma 1 With minor abuse of notation, let $p_{\alpha,t}$ denote the prediction of $A(\alpha)$. Then, given any $\alpha \in [0, \alpha_{\max})$, let α^* be the smallest value in the grid exceeding α, such that $\alpha \leqslant \alpha^* \leqslant 2\alpha \vee \alpha_{\min}$, and let $m^* = \log_2(\alpha^*/\alpha_{\min})$. By Lemma 19, with γ a point-mass on α^*, we find that

$$\sum_{t=1}^{T} \ell_{\log}(p_t, y_t) \leqslant \sum_{t=1}^{T} \ell_{\log}(p_{\alpha^*,t}, y_t) - \ln \pi(m^*)\,.$$

Then, for any $\theta \in \Theta_\alpha \subseteq \Theta_{\alpha^*}$,

$$\begin{aligned}&R_T(\theta) \leqslant B_T(\theta, \alpha^*) - \ln \pi(m^*) < B_T(\theta, \alpha^*) + 2\ln(m^* + 2)\\ &\quad = B_T(\theta, \alpha^*) + 2\ln\Big(\log_2\big(\frac{\alpha^*}{\alpha_{\min}}\big) + 2\Big) \leqslant \max_{\alpha'\in[\alpha, 2\alpha\vee\alpha_{\min}]} B_T(\theta, \alpha') + 2\ln\Big(\log_2\big(\frac{8\alpha}{\alpha_{\min}} \vee 4\big)\Big),\end{aligned}$$

as required. ■

A.2. Proofs for the Normal Location Family

Proof of Theorem 2 Abbreviate $y^t = (y_1, \ldots, y_t)$, define $p_\theta(y^T) = \prod_{t=1}^{T} p_{\theta,t}(y_t)$, let $\hat{\mu}(y^T) = \operatorname{argmax}_{\theta\in\mathbb{R}^d} p_\theta(y^T) = \frac{1}{T}\sum_{t=1}^{T} y_t$ be the unconstrained maximum likelihood, and take $\hat{\theta}(y^T) =$

$\operatorname{argmax}_{\theta \in \mathcal{B}(0,U)} p_\theta(y^T)$ to be the maximum likelihood restricted to Θ, which is the projection onto $\mathcal{B}(0,U)$ of $\hat{\mu}$: $\hat{\theta}(y^T) = \operatorname{Proj}_{\mathcal{B}(0,U)}(\hat{\mu}) = \min\{1, \frac{U}{\|\hat{\mu}\|}\}\hat{\mu}$.

Since the horizon T and the predictions $p_{\theta,t}$ for $t = 1, \dots, T$ are known in advance, the exact minimax strategy (Grünwald, 2007) is to predict $p_t = p_{\text{nml}}(y_t \mid y^{t-1})$, where

$$p_{\text{nml}}(y^T) = \frac{p_{\hat{\theta}(y^T)}(y^T)}{Z}$$

is the normalised maximum-likelihood (NML) density, with normalising constant

$$Z = \int_{\mathbb{R}^{d \times T}} p_{\hat{\theta}(y^T)}(y^T) \, \mathrm{d}y^T.$$

The NML density is an equalizing strategy that ensures the regret is exactly

$$\max_{\theta \in \mathcal{B}(0,U)} R_T(\theta) = \ln Z$$

for all sequences y^T. The value $\ln Z$ is called the *stochastic complexity*. It therefore remains to evaluate the integral Z. To this end, we use that $\hat{\mu}(y^T)$ is a sufficient statistic for y^T, which means that the conditional density of p_θ given $\hat{\mu}$ does not depend on θ. We therefore define $\bar{p}(y^T \mid \hat{\mu}(y^T)) := p_\theta(y^T \mid \hat{\mu}(y^T))$, independently of θ. Consequently, $p_{\hat{\theta}(y^T)}(y^T) = \bar{p}(y^T \mid \hat{\mu}) p_{\hat{\theta}(\hat{\mu})}(\hat{\mu})$, where $\hat{\theta}(\hat{\mu}) := \operatorname{argmax}_{\theta \in \Theta} p_\theta(\hat{\mu})$ equals $\hat{\theta}(y^T)$ for any y^T for which $\hat{\mu} = \hat{\mu}(y^T)$. Hence

$$\begin{aligned} Z &= \int_{\mathbb{R}^d} \mathop{\mathbb{E}}_{\nu} \left[p_{\hat{\theta}(y^T)}(y^T) \;\middle|\; \hat{\mu}(y^T) = \hat{\mu} \right] \mathrm{d}\hat{\mu} = \int_{\mathbb{R}^d} p_{\hat{\theta}(\hat{\mu})}(\hat{\mu}) \mathop{\mathbb{E}}_{\nu} \left[\bar{p}(y^T \mid \hat{\mu}) \;\middle|\; \hat{\mu}(y^T) = \hat{\mu} \right] \mathrm{d}\hat{\mu} \\ &= \int_{\mathbb{R}^d} p_{\hat{\theta}(\hat{\mu})}(\hat{\mu}) \, \mathrm{d}\hat{\mu} = \frac{1}{(2\pi\sigma^2/T)^{d/2}} \int_{\mathbb{R}^d} e^{-\frac{\|\hat{\mu} - \operatorname{Proj}_{\mathcal{B}(0,U)}(\hat{\mu})\|_2^2}{2\sigma^2/T}} \, \mathrm{d}\hat{\mu}, \end{aligned}$$

where the first identity is the law of total probability for Lebesgue measure, the third identity uses that $\bar{p}(y^T \mid \hat{\mu})$ integrates to 1 over its domain, and the last identity comes from the fact that, for any θ, $\hat{\mu}$ is the average of T normal distributions $\mathcal{N}(\theta, \sigma^2 I)$, and is therefore distributed as $\mathcal{N}(\theta, \frac{\sigma^2}{T} I)$.

We evaluate the remaining integral, starting from the observation that $\|\hat{\mu} - \operatorname{Proj}_{\mathcal{B}(0,U)}(\hat{\mu})\|_2 = \max\{\|\hat{\mu}\| - U, 0\}$ depends only on the length of $\hat{\mu}$. For $d = 1$, computing Z is straightforward, so assume for the remainder that $d \geqslant 2$. Switching to hyperspherical coordinates with radial parameter $r \in [0, \infty)$ and angular parameters $\phi \in \Phi := [0, \pi]^{d-2} \times [0, 2\pi]$, then implies that

$$\begin{aligned} \int_{\mathbb{R}^d} e^{-\frac{\|\hat{\mu} - \operatorname{Proj}_{\mathcal{B}(0,U)}(\hat{\mu})\|_2^2}{2\sigma^2/T}} \, \mathrm{d}\hat{\mu} &= \int_0^\infty \int_\Phi e^{-\frac{\max\{r-U,0\}^2}{2\sigma^2/T}} r^{d-1} \prod_{i=1}^{d-2} \sin^{d-1-i}(\phi_i) \, \mathrm{d}\phi \, \mathrm{d}r \\ &= \int_0^\infty e^{-\frac{\max\{r-U,0\}^2}{2\sigma^2/T}} r^{d-1} \, \mathrm{d}r \times \int_\Phi \prod_{i=1}^{d-2} \sin^{d-1-i}(\phi_i) \, \mathrm{d}\phi. \end{aligned}$$

The second factor evaluates to

$$\int_\Phi \prod_{i=1}^{d-2} \sin^{d-1-i}(\phi_i) \, \mathrm{d}\phi = d \int_0^1 \int_\Phi r^{d-1} \prod_{i=1}^{d-2} \sin^{d-1-i}(\phi_i) \, \mathrm{d}\phi \, \mathrm{d}r = d \operatorname{Vol}(\mathcal{B}(0,1)) = \frac{d\pi^{d/2}}{\Gamma(\frac{d}{2}+1)},$$

where $\mathrm{Vol}(\mathcal{B}(0,r)) = \frac{\pi^{d/2} r^d}{\Gamma(\frac{d}{2}+1)}$ is the volume of a ball of radius r; the first factor can be re-expressed as

$$\begin{aligned}\int_0^\infty e^{-\frac{\max\{r-U,0\}^2}{2\sigma^2/T}} r^{d-1}\,\mathrm{d}r &= \int_0^U r^{d-1}\,\mathrm{d}r + \int_U^\infty r^{d-1} e^{-T(r-U)^2/(2\sigma^2)}\,\mathrm{d}r \\ &= \frac{U^d}{d} + \int_0^\infty (r+U)^{d-1} e^{-Tr^2/(2\sigma^2)}\,\mathrm{d}r \\ &= \frac{U^d}{d} + \frac{\sigma}{\sqrt{T}} \int_0^\infty \left(\frac{r\sigma}{\sqrt{T}} + U\right)^{d-1} e^{-r^2/2}\,\mathrm{d}r.\end{aligned}$$

Putting all equalities together establishes (3). The proof is completed upon observing that $V(U,T) = O(1/\sqrt{T})$. ■

A.3. Proofs for the Logistic Loss

Proof of Theorem 4 Consider the statement of Lemma 19

$$\sum_{t=1}^T \ell_{\log}(p_t, y_t) \leqslant \mathop{\mathbb{E}}_{\theta\sim\gamma}\left[\sum_{t=1}^T \ell_{\log}(p_{\theta,t}, y_t)\right] + \mathrm{KL}(\gamma\|\pi) \qquad \text{for all distributions } \gamma.$$

Set $\gamma = \pi(\cdot|A)$, where $A := \{a\theta^* + (1-a)\theta | \theta \in \Theta\} \subset \Theta$ where $a \in [0,1)$ as in Foster et al. (2018) and π is chosen to be uniform. Then the Kullback-Leibler divergence reads

$$\mathrm{KL}(\pi(\cdot|A)\|\pi) = \int_{\theta\in\Theta} \ln\left(\frac{\mathrm{d}\pi(\theta|A)}{\mathrm{d}\pi(\theta)}\right) \mathrm{d}\pi(\theta|A) = -\ln \pi(A) = \ln \frac{\mathcal{V}(\Theta)}{\mathcal{V}(A)} = d_\Theta \ln \frac{1}{1-a},$$

where the last equality follows from $\mathcal{V}(A) = (1-a)^{d_\Theta}\mathcal{V}(\Theta)$, and so we can bound

$$\begin{aligned}\mathop{\mathbb{E}}_{\theta\sim\pi(\cdot|A)}\left[\sum_{t=1}^T \ell_{\log}(p_{\theta,t}, y_t)\right] &\leqslant \max_{\theta\in\Theta}\left\{\sum_{t=1}^T \ell_{\log}(p_{a\theta^*+(1-a)\theta,t}, y_t)\right\} \\ &\leqslant \sum_{t=1}^T \left(\ell_{\log}(p_{\theta^*,t}, y_t) + 4(1-a)B\|x_t\|_*\right),\end{aligned}$$

where in the second inequality we have used the 2-Lipschitzness of the logistic loss with respect to the L_∞-norm (Foster et al. (2018), Lemma 1) and their observation that $\|(a\theta^* + (1-a)\theta - \theta^*)\,x_t\|_\infty = (1-a)\max_{k\in[K]} |\langle \theta_k - \theta_k^*, x_t\rangle| \leqslant 2(1-a)U\|x_t\|_*$ for all $\theta \in \Theta$. It follows that

$$\sum_{t=1}^T \ell_{\log}(p_t, y_t) \leqslant \sum_{t=1}^T \left(\ell_{\log}(p_{\theta^*,t}, y_t) + 4(1-a)U\|x_t\|_*\right) + d_\Theta \ln \frac{1}{1-a}.$$

Setting $1-a := 1 \wedge \frac{d_\Theta}{U\sum_{t=1}^T \|x_t\|_*}$ and bounding appropriately completes the proof. ■

Proof of Theorem 6 We first lower bound the minimax regret by restricting the maximum over θ to $\Theta' = \{\binom{\theta'}{0} : \|\theta'\|_\infty \leqslant U/\sqrt{d}\} \subset \Theta$. To construct a hard data sequence $\mathcal{S}$, we then set

$x_t = Xe_{(t \bmod d)+1}$, which reduces the learning task to d independent one-dimensional learning tasks with $n \geqslant \frac{T}{d} - 1$ learning rounds each. Consequently, our lower bound will be d times a lower bound that holds for each of the one-dimensional tasks.

So consider a one-dimensional binary logistic regression task with $|\theta| \leqslant U/\sqrt{d}$ and $x_t = X$ for all $t = 1, \ldots, n$. This is equivalent to log loss prediction of $z_t = (y_t + 1)/2 \in \{0, 1\}$ with respect to the Bernoulli distributions $\mathcal{B}_\mu$ with means restricted to $\mu \in [a, b]$ where

$$a = \frac{1}{1 + e^{XU/\sqrt{d}}}, \qquad b = \frac{1}{1 + e^{-XU/\sqrt{d}}}.$$

The minimax regret for the case $a = 0, b = 1$ is well known (see (Xie and Barron, 2000) and references therein). We can handle general a, b by adapting the proof of Ordentlich and Cover (1998). Like in the proof of Theorem 2, the minimax regret equals the stochastic complexity (Grünwald, 2007):

$$\begin{aligned}\ln \sum_{z^n \in \{0,1\}^n} \max_{\mu \in [a,b]} \mathcal{B}_\mu(z^n) &= \ln \sum_{k=0}^{n} \binom{n}{k} \max_{\mu \in [a,b]} \mu^k (1-\mu)^{n-k} \\ &\geqslant \ln \sum_{k=\lceil an \rceil}^{\lfloor bn \rfloor} \binom{n}{k} \left(\frac{k}{n}\right)^k \left(\frac{n-k}{n}\right)^{n-k}.\end{aligned}$$

As shown by Ordentlich and Cover (1998, Proof of Lemma 2), the terms in this sum are at least

$$\binom{n}{k} \left(\frac{k}{n}\right)^k \left(\frac{n-k}{n}\right)^{n-k} \geqslant \frac{1}{\sqrt{\pi(n+1)/2}} \qquad \text{for all } k.$$

Hence

$$\begin{aligned}\sum_{k=\lceil an \rceil}^{\lfloor bn \rfloor} \binom{n}{k} \left(\frac{k}{n}\right)^k \left(\frac{n-k}{n}\right)^{n-k} &\geqslant \frac{\lfloor bn \rfloor - \lceil an \rceil}{\sqrt{\pi(n+1)/2}} \geqslant \frac{(b-a)n - 2}{\sqrt{\pi(n+1)/2}} \geqslant \frac{(b-a)n-2}{\sqrt{\pi n}} \\ &= \frac{(b-a)\sqrt{n}}{\sqrt{\pi}} - \frac{2}{\sqrt{\pi n}}.\end{aligned}$$

It remains to bound

$$b - a = \tanh\left(\frac{UX}{2\sqrt{d}}\right) \geqslant \left(\frac{UX}{4\sqrt{d}}\right),$$

where the inequality follows from $\tanh(x) \geqslant x/2$ for $x \in [0, 1]$, which applies because $UX \leqslant 2\sqrt{d}$ by assumption. The proof is completed by combining all previous steps. ∎

Proof of Theorem 7 For $t < t^*$ the learner can play $p_t(y) = 1/K$, which incurs 0 instantaneous regret, because $\|x_t\| = 0$ implies that $p_{\theta,t}(y) = 1/K$ for all θ. Then, from $t \geqslant t^*$, the learner aggregates multiple copies of the algorithm from Theorem 4 for $U = \alpha$, specialized to the case that $\Theta_U = \{\theta \in \mathbb{R}^{K \times d} : \|\theta\| \leqslant U\}$ and with $X = X_T$ (which is possible because the Bayesian algorithm described there does not depend on X). Lemma 1 is applied with $\alpha = \|\theta\|$, $\alpha_{\text{min}} = \varepsilon dK/(X_{t^*}T)$ and $\alpha_{\text{max}} = \infty$. All together this gives the bound

$$\begin{aligned} R_T(\theta) &\leqslant 5dK \ln\Big(\frac{(2\|\theta\| \wedge \frac{\varepsilon dK}{X_{t^*}T})X_T(T - t^* + 1)}{dK} + e\Big) + 2\ln\Big(\log_2\Big(\frac{8\|\theta\|X_{t^*}T}{\varepsilon dK} \vee 4\Big)\Big) \\ &\leqslant 5dK \ln\Big(\frac{2\|\theta\|X_T T}{dK} + \frac{\varepsilon X_T}{X_{t^*}} + e\Big) + 2\ln\Big(\log_2\Big(\frac{8\|\theta\|X_{t^*}T}{\varepsilon dK} \vee 4\Big)\Big), \end{aligned}$$

as required. The algorithm is scale-free, because $x_t \mapsto \lambda x_t$ for all t implies that $\alpha_{\text{min}} \mapsto \alpha_{\text{min}}/\lambda$, which is equivalent to calling Lemma 1 with α_{min} unchanged but $U = \alpha/\lambda$, leading the algorithm from Theorem 4 to produce the same predictions, and as a consequence p_t is also unchanged. ■

Proof of Theorem 10 As in the proof of Theorem 9 we can predict with the uniform distribution p_t for all $t < t^*$ without incurring instantaneous regret, so assume without loss of generality that $t^* = 1$.

We then start by using a doubling trick to make the algorithm from Theorem 9 adapt to X: starting from $X = X_{\text{min}} := \ln(K)/U$ we restart the algorithm with new value $2X$ any time that $\|x_t\|_2 > X$. (NB. If $\|x_t\|_2 > 2X$ at the time of a restart, we interpret it as immediately triggering more restarts until $X \geqslant \|x_t\|_2$.) Since

$$\sum_{i=0}^{\lceil \log_2(X_T/X_{\text{min}}) \rceil \vee 0} X_{\text{min}} 2^i \leqslant X_{\text{min}} 2^{(\lceil \log_2(X_T/X_{\text{min}}) \rceil + 1) \vee 0} \leqslant 4X_T \vee X_{\text{min}} = 4X_T \vee \frac{2\ln(K)}{U},$$

this leads to a regret bound of the same order as in Theorem 9:

$$\begin{aligned} R_T(\theta) &= O\Big(\sum_{i=0}^{\lceil \log_2(X_T/X_{\text{min}} \rceil \vee 0} \big(UX_{\text{min}}2^i + \ln K\big) dK \ln T\Big) \\ &= O\Big(\big(UX_T \vee \ln K\big) dK \ln T\Big) \qquad \text{for all } \theta \text{ such that } \|\theta\|_{2,\infty} \leqslant U, \end{aligned}$$

and runs in time $O(d^2K^3 + (UX_t \vee \ln K)K^2 \ln(t(1 + UX_t \vee \ln K)))$ in round t.

Let $A(U)$ denote this algorithm. We will aggregate multiple copies of $A(U)$ using Lemma 1 with $U = \alpha = \|\theta\|_{2,\infty}$, $\alpha_{\text{max}} = \frac{T^\beta}{X_{t^*}} \geqslant \frac{T^\beta}{X_T}$, $\Theta_\alpha = \{\theta \in \mathbb{R}^{K \times d} : \|\theta\|_{2,\infty} \leqslant \alpha\}$, and $\alpha_{\text{min}} = \frac{T^{-\gamma}}{X_{t^*}} \geqslant \frac{T^{-\gamma}}{X_T}$ for $\gamma \geqslant 0$ to be chosen below. We further observe that algorithms with $U > 2T^\beta/X_t \geqslant T^\beta/X_T$ will never be useful because of the restriction to $\|\theta\|_{2,\infty}X_T \leqslant T^\beta$ in (5), so as soon as X_t becomes large enough for this to happen, we stop expending computation on $A(U)$. This can be implemented either by treating $A(U)$ as a sleeping expert in the sense of Freund et al. (1997); Adamskiy et al. (2016) or by simply setting the algorithm's predictions to the uniform distribution for all remaining rounds. All together, this aggregation procedure guarantees regret at most

$$R_T(\theta) = O\Big(\Big(\|\theta\|_{2,\infty}X_T + \frac{T^{-\gamma}X_T}{X_{t^*}} + \ln K\Big) dK \ln T\Big) + 2\ln\Big(\log_2\Big(8\|\theta\|_{2,\infty}X_T T^\gamma \vee 4\Big)\Big)$$

for all θ such that $\|\theta\|_{2,\infty}X_T \leqslant T^\beta$, which establishes (5) for $\gamma = \frac{cT^\beta}{d^2K}$.

Let us proceed to analyse the run-time. To this end, define $U_i = \alpha_{\text{min}} 2^i$ and recall that algorithm U_i is still running in round t only if $U_i \leqslant T^\beta / X_t$. The run-time in round t therefore comes to

$$\begin{aligned}
&O\Big(\sum_{i=0}^{\lceil \log_2(\alpha_{\text{max}}/\alpha_{\text{min}}) \rceil} 1_{[U_i \leqslant T^\beta / X_t]} \Big\{ d^2 K^3 + (U_i X_t \vee \ln K) K^2 \ln(t(1 + U_i X_t \vee \ln K)) \Big\} \Big) \\
&= O\Big(\sum_{i=0}^{\lceil \log_2(\alpha_{\text{max}}/\alpha_{\text{min}}) \rceil} \Big\{ 1_{[U_i \leqslant T^\beta / X_t]} U_i X_t K^2 \ln(T(1 + U_i X_t)) \Big\} \\
&\qquad + \log_2 \Big(\frac{\alpha_{\text{max}}}{\alpha_{\text{min}}} \Big) \Big\{ d^2 K^3 + \ln(K) K^2 \ln(T(1 + \ln K)) \Big\} \Big) \\
&= O\Big(\sum_{i=0}^{\operatorname{argmin}_j \{U_j \leqslant T^\beta / X_t\}} \Big\{ U_i X_t K^2 \ln(T(1 + T^\beta)) \Big\} + (\beta + \gamma) \Big\{ d^2 K^3 + \ln(K) K^2 \ln(T(1 + \ln K)) \Big\} \Big) \\
&= O\Big(T^\beta K^2 \ln(T) + (\beta + \gamma) \Big\{ d^2 K^3 + \ln(K) K^2 \ln(T(1 + \ln K)) \Big\} \Big).
\end{aligned}$$

We aim to choose γ (nearly) as large as possible to ensure that

$$(\beta + \gamma) \Big\{ d^2 K^3 + \ln(K) K^2 \ln(T(1 + \ln K)) \Big\} = O\Big(d^2 K^3 + c T^\beta K^2 \ln(T) \Big),$$

so that the run-time per round is $O(d^2 K^3 + (1 + c) T^\beta K^2 \ln(T))$. The choice $\gamma = \frac{c T^\beta}{d^2 K}$ indicated above satisfies this requirement.

Finally, it remains to establish that the algorithm is scale-free. To see this, note that if we multiply all x_t by some $\lambda > 0$, then $A(U/\lambda)$ makes the same predictions as $A(U)$ does without multiplication. This type of compensation is built into the aggregation procedure, because the definitions of α_{min} and α_{max} scale inversely with λ. ■

A.3.1. BESOV CLASSES

Let $\mathcal{X} \subset \mathbb{R}^d$ is compact, let Θ be the Besov space $B^s_{p,q}(\mathcal{X})$ and identify $h_\theta \equiv \theta$ for $\theta \in \Theta$. Suppose $\Theta_U = \{\theta \in \Theta : \|\theta\|_{B^s_{p,q}} \leqslant U$ is a ball of radius U, where $\|\theta\|_{B^s_{p,q}}$ is the corresponding Besov-norm. Foster et al. (2018, Example 2) show, non-constructively, that the minimax regret is bounded by

$$\min_{\textbf{Algs}} \max_{\mathcal{S}} \max_{\theta \in \Theta_U} R_T(\theta) = \tilde{O}(U^\beta T^\gamma),$$

where

1. If $s \geqslant d/2$, then $\beta = 2d/(d + 2s), \gamma = d/(d + 2s)$;
2. If $s < d/2$, then $\beta = 1$ and γ depends on p: if $p > 1 + d/(2s)$, then $\gamma = 1 - s/d$; otherwise $\gamma = 1 - 1/p$.

We see that in all cases the rate depends heavily on U. Adaptation to U using Lemma 1 with $\alpha = U^\beta$, $\alpha_{\text{min}} = T^{-\gamma}$ and $\alpha_{\text{max}} = \infty$ gives

Theorem 20 *Consider the Besov space setup described above for any fixed p, q and s. Then there exists a learning algorithm with respect to the entire Besov space $B^s_{p,q}(\mathcal{X})$ that guarantees*

$$R_T(\theta) = \tilde{O}\Big(\|\theta\|^\beta_{B^s_{p,q}} T^\gamma + \ln(\log_2(\|\theta\|_{B^s_{p,q}} T^\gamma \vee 1) \Big) \qquad \textit{for all } \theta \in B^s_{p,q}(\mathcal{X}).$$

This adaptive upper bound matches the non-adaptive bound.

Appendix B. From Log Loss to General Mixable Losses

For general loss functions, the Bayesian prediction strategy from the previous section generalises to the Exponential Weights (EW) algorithm, which produces distributions π_t that generalize the posterior distribution from (1) to

$$\mathrm{d}\pi_t(\theta) = \frac{e^{-\eta_t \sum_{s=1}^{t-1} \ell(f_\theta(x_t), y_t)} \mathrm{d}\pi(\theta)}{\int e^{-\eta_t \sum_{s=1}^{t-1} \ell(f_{\theta'}(x_t), y_t)} \mathrm{d}\pi(\theta')}.$$

These depend not just on a prior π, but also on (possibly time-varying) learning rates $\eta_t > 0$. The log loss case is recovered for $\eta_t = 1$. Generalizing (1) for the log loss, we need a way to map the distributions π_t over Θ to actual predictions a_t. To this end, let $P_t = f_\theta(x_t)_\# \pi_t$ be the distribution over predictions in $\mathcal{A}$ induced by $f_\theta(x_t)$ when the parameters θ are distributed according to π_t (i.e. the pushforward of π_t for the map $\theta \mapsto f_\theta(x_t)$). Then the predictions are determined by a *substitution function* ζ_t which maps distributions on $\mathcal{A}$ to a single action: $a_t = \zeta_t(P_t)$. In case of the log loss, actions are densities and $\zeta_t(P) = \mathbb{E}_P[a]$ is simply the mean.

For so-called mixable loss functions ℓ, there exists a direct generalization of Lemma 19. For $\eta > 0$, a loss function ℓ is said to be *η-mixable* with respect to $(\mathcal{A}, \mathcal{Y})$ (Vovk, 2001) if there exists a substitution function ζ that maps any probability distribution P over $\mathcal{A}$ to a single prediction $a_P = \zeta(P) \in \mathcal{A}$ satisfying

$$\ell(a_P, y) \leqslant -\tfrac{1}{\eta} \ln \mathop{\mathbb{E}}_{a \sim P} \left[e^{-\eta \ell(a,y)} \right] \qquad \text{for all } y \in \mathcal{Y}.$$

For the log loss, 1-mixability (trivially) holds with equality when ζ is the mean. In general, we will also cover the case that we have prior knowledge that $y_t \in \mathcal{Y}_t \subseteq \mathcal{Y}$. Lemma 19 then generalizes to:

Lemma 21 *For $t = 1, \ldots, T$, suppose the loss ℓ is η_t-mixable with respect to $(\mathcal{A}, \mathcal{Y}_t)$ with $\mathcal{Y}_t \subseteq \mathcal{Y}$ for substitution function ζ_t. Then the exponential weights algorithm with non-increasing learning rates $\eta_1 \geqslant \cdots \geqslant \eta_T > 0$ and substitution functions $\zeta_1, \ldots, \zeta_T$ achieves*

$$\sum_{t=1}^T \ell(a_t, y_t) \leqslant \mathop{\mathbb{E}}_{\theta \sim \gamma} \left[\sum_{t=1}^T \ell(f_\theta(x_t), y_t) \right] + \frac{\mathrm{KL}(\gamma \| \pi)}{\eta_T} \qquad \text{for all } \gamma \text{ such that } \mathrm{KL}(\gamma, \pi_t) < \infty, \tag{13}$$

provided that the prior knowledge that $y_t \in \mathcal{Y}_t$ is correct..

Note that Lemma 21 specialises to any countable set of experts or continuously parameterised set of static experts.

Proof The proof is a straightforward specialisation of Lemma 1 from van der Hoeven et al. (2018):

Lemma *The FTRL version of EW with prior $\pi(\theta)$ generates a sequence of distributions P_t over θ that satisfies*

$$\sum_{t=1}^T g_t(\theta_t) - \mathop{\mathbb{E}}_{\theta \sim \gamma} \left[\sum_{t=1}^T g_t(\theta) \right] \leqslant \sum_{t=1}^T \left(g_t(\theta_t) + \frac{\log \mathbb{E}_{\theta \sim P_t} \left[\exp\left(-\eta_t g_t(\theta) \right) \right]}{\eta_t} \right) + \frac{\mathrm{KL}(\gamma \| \pi)}{\eta_T}. \tag{14}$$

We specify the result to $g_t : \theta \mapsto \ell(f_\theta(x_t), y)$. Then, by definition of a_t, and by the mixability property

$$\ell(a_t, y_t) + \frac{\log \mathbb{E}_{\theta \sim P_t} [\exp(-\eta_t \ell(f_\theta((x_t)), y_t))]}{\eta_t} \leqslant 0.$$

Summing over t and substituting in (14) (replacing the sum over $g_t(\theta_t)$ by a sum over $\ell(a_t, y_t)$) yields the claimed result. ■

Appendix C. Proofs and Additions to Sections 3

C.1. Proof of the Aggregation Lemma

Proof of Lemma 11 We apply the procedure described above. Let $a_{\alpha,t}$ denote the prediction of $A(\alpha)$. Then, given any $\alpha \in (0, \alpha_{\text{max}})$, let α^* be the smallest value in the grid exceeding α, such that $\alpha \leqslant \alpha^* \leqslant 2\alpha \vee \alpha_{\text{min}}$, and let $m^* = \log_2 \alpha^*$.

$$\sum_{t=1}^{T} \ell(a_t, y_t) - \ell(a_\theta(x_t), y_t) = \sum_{t=1}^{T} \ell(a_t, y_t) - \ell(a_{\alpha^\star,t}, y_t) + \underbrace{\sum_{t=1}^{T} \ell(a_{\alpha^\star,t}, y_t) - \ell(a_\theta(x_t), y_t)}_{\leqslant B_T(\theta,\alpha^\star)} .$$

Since $y_t \in \mathcal{B}(0, Y_t)$, by the Pythagorean inequality, the loss $\ell(a_{\alpha^\star,t}, y_t)$ can only reduced by a projection of $a_{\alpha^\star,t}$ on $\mathcal{B}(0, Y_t)$, so

$$\sum_{t=1}^{T} \ell(a_t, y_t) - \ell(a_{\alpha^\star,t}, y_t) \leqslant \sum_{t=1}^{T} \ell(a_t, y_t) - \ell(\Pi_{Y_t}(a_{\alpha^\star,t}), y_t) .$$

By clipping the losses (cf. details at the end of the proof), then applying the Pythagorean inequality again, thanks to the fact that

$$\Pi_{Y_{t-1}}(\Pi_{Y_t}(a_{\alpha^\star,t})) = \Pi_{Y_{t-1}}(a_{\alpha^\star,t}) \tag{15}$$

and that $\tilde{y}_t \in \mathcal{B}(0, Y_{t-1})$:

$$\begin{aligned}\sum_{t=1}^{T} \ell(a_t, y_t) - \ell(\Pi_{Y_t}(a_{\alpha^\star,t}), y_t) &\leqslant \sum_{t=1}^{T} \ell(a_t, \tilde{y}_t) - \ell(\Pi_{Y_t}(a_{\alpha^\star,t}), \tilde{y}_t) + Y_T \max_{t\in[T]} \|a_t - \Pi_{Y_t}(a_{\alpha^\star,t})\| \\ &\leqslant \sum_{t=1}^{T} \ell(a_t, \tilde{y}_t) - \ell(\Pi_{Y_{t-1}}(a_{\alpha^\star,t}), \tilde{y}_t) + Y_T \max_{t\in[T]} \|a_t - \Pi_{Y_t}(a_{\alpha^\star,t})\| .\end{aligned}$$

We now apply Lemma 21 with γ a point mass at $\alpha^\star$, each expert being $\Pi_{Y_{t-1}}(a_{\alpha_j,t})$, and the losses being $\tilde{y}_t$, which are both in $\mathcal{B}(0, Y_{t-1})$ to see that

$$\sum_{t=1}^{T} \ell(a_t, \tilde{y}_t) - \ell(\Pi_{Y_{t-1}}(a_{\alpha^\star,t}), \tilde{y}_t) \leqslant \frac{-\log \pi(m^\star)}{\eta_T} = 4Y_{T-1}^2\big(-\log \pi(m^\star)\big) .$$

Then, since the substitution function we use for the aggregation is the mean, a_t is a convex combination of the $\Pi_{Y_{t-1}}(a_{\alpha_j,t})$'s, so $\|a_t\| \leqslant Y_{t-1}$, and $\|a_t - \Pi_{Y_t}(a_{\alpha^\star,t})\| \leqslant Y_{t-1} + Y_t \leqslant 2Y_t$. Therefore for any $\theta \in \Theta$

$$R_T(\theta) \leqslant B(T, \theta, \alpha^\star) - 4Y_{T-1}^2 \log \pi(m^\star) + 2Y^2 \leqslant \max_{\alpha' \in [\alpha, 2\alpha\vee\alpha_{\text{min}}]} B_T(\theta, \alpha') - 4Y_{T-1}^2 \log \pi(m^\star) + 2Y_T^2$$

Then observe that

$$-\ln \pi(m^\star) < 2\ln(m^\star + 2) \leqslant 2\ln(\log_2(\alpha^\star) + 2) \leqslant 2\ln(\log_2(8(\alpha/\alpha_{\text{min}}) \vee 4))$$

to conclude.

Details on the Clipping Clipping works similarly to the linear case, because regret depends affinely on the data y_t. Indeed, by expanding the squares,

$$\|a_t - y\|^2 - \|\tilde{a}_{t,\alpha^\star} - y\|^2 = \|a_t - \tilde{y}_t\|^2 - \|\tilde{a}_{t,\alpha^\star} - \tilde{y}_t\|^2 - 2\langle y_t - \tilde{y}_t, a_t - \tilde{a}_{t,\alpha^\star}\rangle\,,$$

where we denoted $\tilde{a}_{t,\alpha^\star} = \Pi_{Y_t}(a_{t,\alpha^\star})$ to reduce clutter. The linear overhead can be bounded by Cauchy-Schwarz, $|\langle y_t - \tilde{y}_t,\, a_t - \tilde{a}_{t,\alpha^\star}\rangle| \leqslant \|y_t - \tilde{y}_t\|\|a_t - \tilde{a}_{t,\alpha^\star}\|$, and

$$\|y_t - \tilde{y}_t\| = \|y_t\|\,|1 - Y_{t-1}/Y_t| \leqslant (Y_t - Y_{t-1})\,.$$

Therefore, by summing over t, upper bounding $\|a_t - \tilde{a}_{t,\alpha^\star}\|$ by its maximum over t, simplifying the telescoping sum and dividing by 2 to recover the square loss,

$$\sum_{t=1}^T \ell(a_t, y_t) - \ell(\tilde{a}_{t,\alpha^\star}, y_t) \leqslant \sum_{t=1}^T \ell(a_t, y_t) - \ell(\tilde{a}_{t,\alpha^\star}, y_t) + Y_T \max_{t\in[T]} \|a_t - \tilde{a}_{t,\alpha^\star}\|.$$

Scale-invariance Scale-freeness with respect to the features is straightforward, as the aggregation procedure does not look at the features.

Let us prove the scale-invariance with respect to the data points. If all y_t's are multiplied by a factor a, then the actions returned by the experts are multiplied by a. The clipping threshold is multiplied by a, and thus both the clipped actions and the clipped data points are multiplied by a. Therefore the losses fed to the aggregation procedure is multiplied by a^2. The learning rate in the aggregation procedure is multiplied by a^2. Therefore the mass put on every expert is kept the same. Since the output of every expert was multiplied by a, the final action is also multiplied by a. ■

C.2. Details for the Square Loss

C.2.1. GRADIENT DESCENT

For any $\lambda > 0$, Gradient Descent (GD) tuned with step size $\eta_t = 1/(\lambda + t)$ on square losses is equivalent to Exponential Weights with learning rate $1/\lambda$ and Gaussian prior $\Sigma = I_d$ with the mean as a substitution function; this was observed by Koolen (2016) and van der Hoeven et al. (2018). We recall a slightly modified version of Corollary 6 in the latter reference.

Theorem 22 *For prediction with the square loss, for any $\alpha > 0$, gradient descent with step size $\eta_t = 1/(\lambda + t)$ is scale-free and enjoys the regret bound*

$$R_T(\theta) \leqslant \frac{\lambda\|\theta\|^2}{2} + 2Y_T^2 \log\left(1 + \frac{T}{\lambda}\right), \quad \textit{for any } \theta \in \mathbb{R}^d. \tag{16}$$

Furthermore, the updates are such that $\|\theta_t\| \leqslant Y_{t-1}$ for all t.

The analysis can be made tighter so that the bound does not diverge when $\alpha \to \infty$, but we chose the bound simplest to read.

C.2.2. KAAR

Let us recall the guarantees for scale-free KAAR, from Gammerman et al. (2004, Theorem 2). We slightly adapt the statement to include the scaling of the regularisation by $X_{t^\star}$. Note that if $k(x_t, x_t) = 0$, then $k(x_t, \cdot) = 0$ and $\theta(x_t) = 0$ for any x_t. Thefore predicting $a_t = 0$ on all the rounds for which $k(x_t, x_t) = 0$ has no impact on the regret.

The formula for the updates of KAAR gives the updates of KAAR-sf for all $t \geqslant t^\star$ and are still given by (9). The scale-free property derives directly from the updates formula.

Theorem 23 *KAAR-sf(α) over the RKHS $\mathcal{F}$ is scale-free and guarantees that for any $\theta \in \mathcal{F}$.*

$$R_T(\theta) \leqslant \frac{\alpha X_{t^\star} \|\theta\|^2}{2} + \frac{Y_T^2}{2} \ln \det \left(I_T + \frac{1}{\alpha X_{t^\star}} K_T \right), \tag{17}$$

where $K_T = \big(k(x_u, x_v)\big)_{(u,v)\in[T]^2}$ *and* $t^\star = \min\{t \,|\, k(x_t, x_t) > 0\}$.

We use the convention that $1/(X_{t^\star})K_T = 0$ if $T \leqslant t^\star$. (In this case, all features up to time t are 0, the algorithm predicts only 0 and the regret at time T is exactly 0.)

C.3. Proofs of the Regret Bounds

Proof of Theorem 12 We prove the result for arbitrary values of α_{min} and α_{max}, then specialize to $\alpha_{\text{min}} = 1$ and $\alpha_{\text{max}} = T$. Define $Y = Y_T$ to simplify notation. First, assume that $\|\theta\| \leqslant Y$. Plug in the value $\alpha = (Y^2/\|\theta\|^2) \wedge \alpha_{\text{max}}$ in the upper bound

$$\begin{aligned}
R_T(\theta) &\leqslant \max_{\alpha' \in [\alpha,\, 2\alpha \vee \alpha_{\text{min}}]} \left\{ \frac{\alpha' \|\theta\|^2}{2} + 2Y^2 \log\big(1 + \frac{1}{\alpha'} T\big) \right\} + \frac{1}{\eta_{T-1}} \ln\big(\log_2((\frac{8\alpha}{\alpha_{\text{min}}} \vee 4\alpha_{\text{min}})\big) + 2Y^2 \\
&\leqslant \frac{(2\alpha \vee \alpha_{\text{min}})\|\theta\|^2}{2} + 2Y^2 \log\left(1 + \frac{1}{\alpha} T\right) + 8Y^2 \ln\big(\log_2((8\alpha/\alpha_{\text{min}}) \vee 4)\big) + 2Y^2 \\
&\leqslant Y^2 \vee \frac{\alpha_{\text{min}} \|\theta\|^2}{2} + 2Y^2 \log\left(1 + \big(\frac{\|\theta\|^2}{Y^2} \vee \frac{1}{\alpha_{\text{max}}}\big) T\right) + 8Y^2 \ln\big(\log_2(\frac{8Y^2}{\alpha_{\text{min}} \|\theta\|^2} \vee 4)\big) + 2Y^2 .
\end{aligned}$$

Note that as $\alpha_{\text{min}} \leqslant 1$ and $(\|\theta\|/Y)^2 \leqslant 1$ we have $Y^2 \vee (\alpha_{\text{min}} \|\theta\|^2/2) = Y^2$. The claimed bound follows after replacing $\alpha_{\text{min}} = 1$ and $\alpha_{\text{max}} = T$, applying the bound $a \vee b \leqslant a + b$ for $a, b > 0$.

If $\|\theta\| > Y$ consider $\tilde{\theta} = \Pi_Y(\theta)$. Then by the Pythagorean inequality $R_T(\theta) \leqslant R_T(\tilde{\theta})$. ■

Proof of Theorem 13 We build an algorithm via a double infinite aggregation procedure. Define the algorithm $A(\alpha)$ to be KAAR-sf(α), and define its regret bound:

$$B_T(\alpha, \theta) := \frac{\alpha X_{t^\star}^2 \|\theta\|^2}{2} + \frac{Y^2}{2} \ln \det \left(I_T + \frac{1}{\alpha X_{t^\star}} K_T \right);$$

with the convention that $B_T(\alpha, \theta) = 0$ if $t \leqslant t^\star$. For any $c > 0$, define $\tilde{A}(c)$ to be result of the aggregation procedure applied to $A(\alpha)$ tuned with $\alpha_{\text{min}} = 1/c$, and $\alpha_{\text{max}} = \infty$. Since each instance is scale-free, the aggregated version is also scale-free, by Lemma 11. The algorithm $\tilde{A}(c)$ enjoys the regret bound

$$R_T(\theta) \leqslant \max_{\alpha' \in [\alpha, 2\alpha \vee (1/c)]} B_T(\theta, \alpha') + 8Y^2 \ln(\log_2((8\alpha c) \vee 4)) + 2Y^2$$
$$\leqslant \underbrace{\left((2\alpha) \vee \frac{1}{c}\right) \frac{X_{t^\star}^2 \|\theta\|^2}{2} + \frac{Y^2}{2} \ln\det\left(I_T + \frac{1}{\alpha X_{t^\star}^2} K_T\right) + 8Y^2 \ln(\log_2((8\alpha c) \vee 4)) + 2Y^2}_{:= \tilde{B}_T(\theta, c)} .$$

Now run the aggregation procedure again, with each expert being $\tilde{A}(c)$, this time with the parameters $c_{\min} = 1$ and $c_{\max} = \infty$. Again, Lemma 11 guarantees that the total algorithm is also scale-free. Then for any $c \geqslant 0$

$$\begin{aligned} R_T(\theta) &\leqslant \max_{c' \in [c, 2c \vee 1]} \tilde{B}_T(c, \theta) + 2Y^2 + 8Y^2 \ln\log_2(8c \vee 4) \\ &\leqslant \left((2\alpha) \vee \frac{1}{c}\right) \frac{X_{t^\star}^2 \|\theta\|^2}{2} + 8Y^2 \ln\log_2((8\alpha(2c \vee 1)) \vee 4) \\ &+ \frac{Y^2}{2} \ln\det\left(I_T + \frac{1}{\alpha X_{t^\star}^2} K_T\right) + 2Y^2 + 2Y^2 + 8Y^2 \ln\log_2(8c \vee 4) . \end{aligned}$$

In particular, for $c = 1/(2\alpha)$, noting that $8\alpha(2c \vee 1) = 8 \vee 8\alpha$

$$R_T(\theta) \leqslant \frac{\alpha X_{t^\star}^2 \|\theta\|^2}{2} + \frac{Y^2}{2} \ln\det\left(I_T + \frac{1}{\alpha X_{t^\star}^2} K_T\right) + 4Y^2 + 8Y^2 \ln\log(8\alpha \vee 8) + 8Y^2 \ln\log\left(\frac{4}{\alpha} \vee 4\right) .$$

Finally, upper bounding 4 by 8 inside the logarithms and using a case disjunction on whether $\alpha \geqslant 1$,

$$\begin{aligned} \ln\log_2(8\alpha \vee 8) + \ln\log_2(4\alpha^{-1} \vee 4) &\leqslant \ln\log_2(8\alpha \vee 8) + \ln\log_2(8\alpha^{-1} \vee 8) \\ &= \ln\log_2 8 + \ln\log_2(\alpha \vee \alpha^{-1}) = \ln\left(3|\log_2 \alpha|\right) . \end{aligned}$$

Reparameterize by $\lambda = \alpha / X_{t^\star}^2$ to obtain the regret bound. ■

C.4. Consequences of the General Regret Bound

C.4.1. Dimension-independent Bound

Note that regardless of the kernel, denoting by $\lambda_n(K_T)$ the n-the largest eigenvalue of K_T,

$$\ln\det\left(I_T + \frac{1}{\lambda} K_T\right) = \sum_{n=1}^{T} \ln\left(1 + \frac{\lambda_n(K_T)}{\lambda}\right) \leqslant \frac{\mathrm{Tr}(K_T)}{\lambda} \leqslant \frac{T X_T^2}{\lambda} . \tag{18}$$

In particular, after applying this upper bound, optimizing (10) over λ to get $\lambda = X_T Y_T \sqrt{T} / \|\theta\|$, we see that A-KAAR enjoys the dimension-independent bound Corollary 14.

C.4.2. PARAMETRIC CASE

Using concavity of the logarithm,

$$\ln\det\left(I_T + \frac{1}{\lambda}K_T\right) = \sum_{n=1}^{d} \ln\left(1 + \frac{\lambda_n(K_T)}{\lambda}\right) \leqslant d\ln\left(1 + \frac{\mathrm{Tr}(K_T)}{\lambda d}\right) \leqslant d\ln\left(1 + \frac{TX_T^2}{\lambda d}\right).$$

Apply the inequality above in (10) and plug in the value $\lambda = (dY^2)/\|\theta\|^2$ to obtain Theorem 15.

C.5. RKHS with the Capacity Condition

The effective dimension of the kernel matrix K_T at scale λ is defined as

$$d_{\mathrm{eff}}(\lambda) = \mathrm{Tr}\left(K_T(K_T + \lambda I_T)^{-1}\right).$$

It is a quantity that appears naturally in the analysis of kernel ridge regression, a widely studied variant of KAAR in the batch version of the problem. To analyse KAAR, Jézéquel et al. (2019, Proposition 2) prove that

$$\ln\det\left(I_T + \frac{1}{\lambda}K_T\right) \leqslant d_{\mathrm{eff}}(\lambda)\left(1 + \log\left(1 + \frac{TX_T^2}{\lambda}\right)\right). \tag{19}$$

The capacity condition then provides an upper bound on the term above, which yields and explicit regret bound when used in (10). We plug in the value of λ that optimises this regret bound.

Proof of Theorem 16 Plug in the order optimal value in the A-KAAR upper bound (10)

$$\lambda = \left(\frac{Y^2(TC_k)^\gamma}{\|\theta\|^2}\right)^{1/(1+\gamma)},$$

which roughly balances the two terms (that is, up to the logarithms). Then the first term in (10) becomes

$$\frac{\lambda\|\theta\|^2}{2} = \frac{1}{2}Y^{2/(1+\gamma)}\|\theta\|^{2\gamma/(1+\gamma)}(C_kT)^{2\gamma/(1+\gamma)},$$

and, after applying (19), using the capacity condition and replacing λ by its value

$$\begin{aligned} Y^2\ln\det\left(I_T + \frac{1}{\lambda}K_T\right) &\leqslant \left(\frac{TC_k}{\lambda}\right)^\gamma\left(1 + \log\left(1 + \frac{TX_T^2}{\lambda}\right)\right) \\ &\leqslant 2^\gamma Y^{2/(1+\gamma)}\|\theta\|^{2\gamma/(1+\gamma)}(C_kT)^{2\gamma/(1+\gamma)}\left(1 + \log\left(1 + X^2\left(\frac{\|\theta\|^2T^\gamma}{Y^2C_k^\gamma}\right)^{1/(1+\gamma)}\right)\right) \end{aligned}$$

to obtain the final regret bound

$$\begin{aligned} R_T(\theta) \leqslant Y^{2/(1+\gamma)}\|\theta\|^{2\gamma/(1+\gamma)}(C_kT)^{2\gamma/(1+\gamma)}&\left(\frac{1}{2} + 2^{\gamma-1} + \log\left(1 + X^2\left(\frac{\|\theta\|^2T^\gamma}{Y^2C_k^\gamma}\right)^{1/(1+\gamma)}\right)\right) \\ &+ 8Y^2\ln\left(\frac{3e}{2(1+\gamma)}\left|\log_2\left(\frac{Y^2(TC_k)^\gamma}{\|\theta\|^2X_{t^\star}^{2+2\gamma}}\right)\right|\right), \end{aligned} \tag{20}$$

which is the finite-time version of the claimed result. ■

Proof of Corollary 17 By Zadorozhnyi et al. (2021, Theorem 3), $\mathcal{F} = W_{s,2}([-1,1]^d)$ is an RKHS that satisfies the capacity condition with $\gamma = d/(2s)$ for some $C_k > 0$, which depends on d and s. By playing according to A-KAAR on $\mathcal{F}$, Theorem 16 gives the claimed regret bound against any comparator $\theta \in \mathcal{F} = W_{s,2}([-1,1]^2)$, yielding the result for the $p = 2$ case.

For $p \geqslant 2$, by standard L_p-inclusions, $W_{s,p}([-1,1]^2) \subset W_{s,2}([-1,1]^2)$, and for any element $f \in W_{s,p}([-1,1]^2)$, denoting by $\|\cdot\|_p$ the p-norm with respect to the Lebesgue measure, and by D^α the (weak) partial differential of order α,

$$\|f\|_{s,p} = \left(\sum_{|\alpha|\leqslant s} \|D^\alpha f\|_p^p\right)^{1/p} \leqslant \left(\sum_{|\alpha|\leqslant s} 1\right)^{1/2-1/p} \left(\sum_{|\alpha|\leqslant s} \|D^\alpha f\|_p^2\right)^{1/2}$$
$$\leqslant \left(2^s \operatorname{Vol}(\mathcal{X})\right)^{1/2-1/p} \left(\sum_{|\alpha|\leqslant s} \|D^\alpha f\|_2^2\right)^{1/2} = \left(2^s \operatorname{Vol}(\mathcal{X})\right)^{1/2-1/p} \|f\|_{s,2}\,.$$

Therefore the result for $p \geqslant 2$ follows from the $p = 2$ case. ■

C.6. Lower Bound for the Square Loss

We separate the proof into three statements, considering different parameter regimes; in particular, even though the bounds of Propositions 25 and 26 are of the same order, we separate them since the proofs are different.

Proposition 24 *In linear least-squares regression in $\mathbb{R}^d$, for any values $Y, U, X > 0$ that satisfy $(UX/Y)^2 \leqslant d$, for any algorithm, there exists a sequence of examples $(x_t, y_t) \in \mathcal{B}(0,X) \times \mathcal{B}(0,Y)$ such that*

$$\sup_{\theta \in \mathcal{B}(0,U)} R_T(\theta) \geqslant 0.36\, dY^2 \log\left(\left\lfloor \frac{T}{d} \right\rfloor \frac{(UX/2Y)^2}{2d \log(2\lfloor T/d \rfloor)} + 1 \right) - 4dY^2\,.$$

The constant 0.36 in the bound can be replaced by a T-dependent quantity that converges to $1/2$ as $T \to \infty$ and other parameters are kept constant, cf. (22). Moreover, if $d = 1$, the same bound holds with $x_t = X$ for all t.

Proposition 25 *In linear least-squares regression in $\mathbb{R}^d$, for any values $Y, U, X > 0$ that satisfy $d \leqslant T \leqslant (d/8)(Y/(XU))^2$, for any algorithm, there exists a sequence of examples (x_t, y_t) in $\mathcal{B}(0,X) \times [-Y, Y]$ such that*

$$\sup_{\theta \in \mathcal{B}(0,U)} R_T(\theta) \geqslant \frac{\sqrt{2}}{8} \min(UX,\, Y) Y \sqrt{T}\,.$$

Proposition 26 *In linear least-squares regression in $\mathbb{R}^d$, for any values $Y, U, X > 0$ such that $T \leqslant d$, for any algorithm, there exists a sequence of examples (x_t, y_t) in $\mathcal{B}(0,X) \times [-Y, Y]$, such that*

$$\sup_{\theta \in \mathcal{B}(0,U)} R_T(\theta) \geqslant \frac{1}{2} \min(UX,\, Y) Y \sqrt{T}\,.$$

Proof of Theorem 24 The proof follows from an alteration of the proof of Theorem 2 from Vovk (2001). We reproduce it in detail for completeness. We start with the case $d = 1$ and $x_t = 1$ for all times t, and leave the general case for later.

A standard method in lower bounds for online learning is to build a distribution over y_t, and to lower bound the regret on average according to that distribution. Let $y_t \in \{0, 1\}$ be i.i.d. Bernoulli random variables with parameter $p \in [0, 1]$, itself drawn from a (symmetric) beta distribution with parameters (A, A).

Denote by $\mathbb{E}$ the expectation with respect to the whole randomness, that is, both the prior π and the distribution of the y_t's. The natural comparator is p in this construction, and the quantity $\mathbb{E}[R_T(p)]$ can be explicitly lower bounded, as we shall see in (21). Since the distribution $\beta(A, A)$ puts mass on values of p outside of the set of comparators $\mathcal{B}(1/2, U)$, we lower bound the worst-case regret as

$$\begin{aligned}
\sup_{(y_t)_{t\in[T]}} \sup_{\theta\in\mathcal{B}(1/2,U)} R_T(\theta) &\geqslant \mathbb{E}\left[\sup_{\theta\in\mathcal{B}(1/2,U)} R_T(\theta)\right] \\
&\geqslant \mathbb{E}\left[\sup_{\theta\in\mathcal{B}(1/2,U)} R_T(\theta)\,\mathbf{1}\{p \in \mathcal{B}(1/2, U)\}\right] \\
&\geqslant \mathbb{E}\Big[R_T(p)\mathbf{1}\{p \in \mathcal{B}(1/2, U)\}\Big] \\
&= \mathbb{E}\big[R_T(p)\big] - \mathbb{E}\Big[R_T(p)\mathbf{1}\{p \notin \mathcal{B}(1/2, U)\}\Big] \\
&\geqslant \mathbb{E}\big[R_T(p)\big] - T\pi\big([0, 1] \setminus \mathcal{B}(1/2, U)\big)\,.
\end{aligned}$$

Let us now bound these two terms separately.

Probability of p Being Outside $B(1/2, U)$ Since the $\beta(A, A)$ distribution is subgaussian with subgaussianity constant $1/(4(2A + 1))$ (see, e.g., Marchal and Arbel (2017)), the Chernoff bound holds:

$$\mathbb{P}_{p\sim\beta(A,A)}\left[\left|\frac{1}{2} - p\right| \geqslant U\right] \leqslant 2e^{-2U^2(2A+1)}.$$

Picking $A \geqslant \frac{\log(2T)}{4U^2} - \frac{1}{2}$ guarantees that

$$\pi\big([0, 1] \setminus \mathcal{B}(1/2, U)\big) \leqslant \frac{1}{T}.$$

Bayesian Regret Consider the expected value of the regret against the comparator p.

$$\mathbb{E}[R_T(p)] = \mathbb{E}\left[\sum_{t=1}^{T}(y_t - a_t)^2 - (y_t - p)^2\right] = \mathbb{E}\left[\sum_{t=1}^{T}(y_t - a_t)(p - a_t) + (y_t - p)(p - x_t)\right].$$

Now the law of y_t given p is $\mathrm{Ber}(p)$, and y_t given p is independent from x_t. So upon conditioning over p and applying the tower rule, we get

$$\mathbb{E}[R_T(p)] = \mathbb{E}\left[\sum_{t=1}^{T}(a_t - p)^2\right].$$

Moreover, note that since a_t is $\sigma(y_1, \dots, y_{t-1})$-measurable,

$$\mathbb{E}\left[(a_t - p)^2 \mid y_1, \dots, y_{t-1}\right] \geqslant \mathbb{E}\left[\left(\mathbb{E}\left[p \mid y_1, \dots, y_{t-1}\right]\right) - p\right)^2 \mid y_1, \dots, y_{t-1}\Big] .$$

(One can interpret this as saying that if the player knows in advance that the adversary will pick p according to π and then generate y_t's iid, then the best the player can do is play the expected value of p given the observations.) Denote by $R_T^\star(p)$ the regret against the value of p of the optimal strategy playing $a_t^\star := \mathbb{E}\left[p \mid y_1, \dots, y_{t-1}\right]$ at every time step; then for any strategy of the learner, $\mathbb{E}[R_T(p)] \geqslant \mathbb{E}[R_T^\star(p)]$.

For the specific choice of prior $\pi = \beta(A, A)$, Vovk (2001) computes the nice closed-form expression for $a_t^\star$, namely,

$$a_t^\star = \frac{\sum_{s=1}^{t-1} y_s + A}{t - 1 + 2A} .$$

Now the whole expected regret is amenable to computation. Indeed, conditionally on p

$$\begin{aligned}
\mathbb{E}[(a_{t+1}^\star - p)^2 | p] &= \frac{1}{(t+2A)^2} \mathbb{E}\left[\left(\sum_{s=1}^{t} y_t + A - pt - 2Ap\right)^2 \middle| p\right] \\
&= \frac{1}{(t+2A)^2} \mathbb{E}\left[\left(\sum_{s=1}^{t} (y_t - p) + A(1-2p)\right)^2 \middle| p\right] \\
&= \frac{1}{(t+2A)^2} \left(\sum_{s=1}^{t} \mathbb{E}\left[(y_t - p)^2 | p\right] + A^2(1-2p)^2\right) \\
&= \frac{1}{(t+2A)^2} \left(tp(1-p) + A^2(1-2p)^2\right) \geqslant p(1-p)\frac{t}{(t+2A)^2} ,
\end{aligned}$$

where we used the fact that the variables $(y_t - p)$ are independent and centered conditionally on p, and that their variance given p is $p(1-p)$. Therefore,

$$\mathbb{E}\left[R_T^\star(p) \middle| p\right] \geqslant p(1-p) \sum_{t=1}^{T} \frac{t-1}{(t-1+2A)^2}.$$

Let us lower bound the sum by comparing it to an integral

$$\sum_{t=1}^{T} \frac{t-1}{(t-1+2A)^2} \geqslant \int_0^T \frac{u}{(u+2A)^2} \mathrm{d}u = \frac{1}{2} \log\left(\frac{T}{2A} + 1\right) ,$$

Finally, averaging over the prior distribution gives that $\mathbb{E}_{p\sim\beta(A,A)}\left[p(1-p)\right] = \frac{A}{4A+2}$, and consequently, for any strategy of the learner

$$\mathbb{E}[R_T(p)] \geqslant \mathbb{E}[R_T(p)^\star] \geqslant \frac{A}{2(4A+2)} \log\left(\frac{T}{2A} + 1\right) . \tag{21}$$

Concluding the 1-Dimensional Case We have shown that for $A \geqslant \log(2T)/(4U^2) - 1/2$, and for any strategy, the worst-case regret against the comparator set $\mathcal{B}(1/2, U)$ is lower bounded by

$$\frac{A}{2(4A+2)} \log\left(\frac{T}{2A} + 1\right) - 1 \,.$$

So replacing $A = \log(2T)/U^2$

$$\begin{aligned}\sup_{\theta \in \mathcal{B}(1/2,U)} R_T(\theta) &\geqslant \frac{\log(2T)/U^2}{2(4\log(2T)/U^2 + 2)} \log\left(\frac{TU^2}{2\log(2T)} + 1\right) - 1 \\ &\geqslant \frac{\log(2)/2}{2(4\log(2)/2+2)} \log\left(\frac{TU^2}{2\log(2T)} + 1\right) - 1 \geqslant 0.09 \log\left(\frac{TU^2}{2\log(2T)} + 1\right) - 1 \,. \quad (22)\end{aligned}$$

(We used the fact that $a \mapsto a/(4a + a)$ is decreasing, and we bounded $U \leqslant 1/2$ and $T \geqslant 1$.)

Scaling To generalize to an adversary playing in $y_t \in [-Y, Y]$ and comparators in $[-U, U]$, note that $y_t/(2Y) + 1/2 \in [0, 1]$, so

$$\begin{aligned}\sup_{(y_t)\in[-Y,Y]^T} \sup_{\theta\in\mathcal{B}(0,U)} R_T(\theta) &= (2Y)^2 \sup_{(y_t)\in[0,1]^T} \sup_{\theta\in\mathcal{B}(1/2,U/(2Y))} R_T(\theta) \\ &\geqslant 0.36\, Y^2 \log\left(\frac{T(U/2Y)^2}{2\log(2T)} + 1\right) - 4Y^2 \,.\end{aligned}$$

Generalizing to d-Dimensional Regression As in Vovk (2001), consider the sequence of features $x_1 = (X, 0, \dots, 0)$, $x_2 = (0, X, 0, \dots, 0)$, etc. Then partition the time steps according to the feature values. The regret over the d partitions of $\lfloor T/d \rfloor$ time steps is then lower bounded as

$$\sup_{(y_t)\in[-Y,Y]^T} \sup_{\theta\in[-U,U]^d} R_T(\theta) \geqslant 0.36\, dY^2 \log\left(\left\lfloor\frac{T}{d}\right\rfloor \frac{(UX/2Y)^2}{2\log(2T/d)} + 1\right) - 4dY^2 \,.$$

Note that the best comparator could *a priori* be anywhere in $[-U, U]^d \subset \mathcal{B}\big(0, \sqrt{d}U\big)$. We rescale the value of U by $\sqrt{d}$ to obtain the claimed result. ∎

Proof of Theorem 25 Again, we start in the 1-dimensional case and consider a constant sequence of features $x_t = 1$. Let (y_t) be a sequence of i.i.d. random variables that take values $-Y$ or Y with probability $1/2$, and set $x_t = 1$ for all t. The general case follows from rescaling $\theta \leftarrow X\theta$. We only consider the comparators $-U$ and U, then

$$\begin{aligned}\mathbb{E}\left[\max_{\theta\in[-U,U]} R_T(\theta)\right] &\geqslant \mathbb{E}\left[\max_{\theta\in\{-U,U\}} \left\{\sum_{t=1}^T (\theta_t - y_t)^2 - (\theta - y_t)^2\right\}\right] \\ &= \mathbb{E}\left[\max_{\theta\in\{-U,U\}} \left\{\sum_{t=1}^T \theta_t^2 - \theta^2 + 2y_t\theta - 2y_t\theta_t\right\}\right] \\ &= \mathbb{E}\left[\sum_{t=1}^T \left(\theta_t^2 - U^2 - 2y_t\theta_t\right) + U\left|\sum_{t=1}^T y_t\right|\right] \\ &= \mathbb{E}\left[\sum_{t=1}^T \theta_t^2\right] - TU^2 + U\,\mathbb{E}\left[\left|\sum_{t=1}^T y_t\right|\right] \geqslant U\,\mathbb{E}\left[\left|\sum_{t=1}^T y_t\right|\right] - TU^2 \,.\end{aligned}$$

We used the fact that $\mathbb{E}[y_t\theta_t] = \mathbb{E}[y_t]\ \mathbb{E}[\theta_t] = 0$, since y_t is independent from θ_t. Now since each y_t is either $-Y$ or Y with probability $1/2$, by Lemma A.9. in Cesa-Bianchi and Lugosi (2006)

$$\mathbb{E}\left[\Big|\sum_{t=1}^{T} y_t\Big|\right] \geqslant Y\sqrt{\frac{T}{2}}\,.$$

Then

$$\max_{y_{1:T}\in[-Y,Y]^T}\ \max_{\theta\in[-U,U]} R_T(\theta) \geqslant \mathbb{E}\left[\max_{\theta\in[-U,U]} R_T(\theta)\right] \geqslant UY\sqrt{\frac{T}{2}}\left(1-\sqrt{2T}\frac{U}{Y}\right).$$

The claimed bound follows by plugging in the condition that $T \leqslant (1/8)(Y/(XU))^2$.

Extending to Dimension d Using the sequence of feature $x_t = Xe_{(t \mod d)+1}$ and partitioning the time steps depending on the feature value, for any $T \leqslant (1/8)(Y/(XU))^2$,

$$\sup_{(y_t)\in[-Y,Y]^T}\ \sup_{\theta\in[-U,U]^d} R_T(\theta) \geqslant \frac{\sqrt{2}}{4}UXY\sum_{i=1}^{d}\sqrt{T_i}$$

where $T_i = \lfloor T/d\rfloor + \mathbb{1}\{i \leqslant (T \mod d)\}$ is the number of time steps for which $x_i = Xe_i$. Now note that as $T \geqslant d$,

$$\sum_{i=1}^{d}\sqrt{T_i} \geqslant \frac{1}{2}\sqrt{dT}\,.$$

Indeed, let us check this by case disjunction. If $T \mod d \geqslant T/2$, then the sum is at least $d/2\sqrt{T/d+1} \geqslant \sqrt{dT}/T$. Otherwise, $T \mod d < T/2$, then $\lfloor T\rfloor/d\rfloor \geqslant T/d - 1/2 \geqslant T/(2d)$, using the assumption that $T \geqslant d$. In this case, the sum is at least $d\sqrt{\lfloor T\rfloor/d\rfloor} \geqslant \sqrt{dT/2}$.

Then, after rescaling U by $1/\sqrt{d}$, and noting that $[-U,U]^d \subset \mathcal{B}(0,\sqrt{d}U)$, for any $T \leqslant (d/8)(Y/(XU))^2$,

$$\sup_{(y_t)\in[-Y,Y]^T}\ \sup_{\theta\in\mathcal{B}(0,U)} R_T(\theta) \geqslant \frac{\sqrt{2}}{8}UXY\sqrt{T}\,.$$

■

Proof of Theorem 26 Once again, we assume that $UX \leqslant Y$, as the general result follows from applying it with $U = Y/X$ when $UX \geqslant Y$.

The result in the $T \leqslant d$ regime follows from a somewhat trivial construction. At time t, let $x_t = Xe_t$. Given the action from the learner, set $y_t = -Y \operatorname{sign} a_t$, and define $u_t = (U/\sqrt{T}) \operatorname{sign} a_t$. Consier the comparator with coordinates u_t for $t \leqslant T$ and 0 otherwise. Then u_t has norm less than U, and the the total regret of the learner against θ is at least $TUXY/\sqrt{T}$.

Indeed, at all times t, we have $(a_t - y_t)^2 \geqslant Y^2$ since, e.g., $a_t \leqslant 0$ when $y_t = Y$. Similarly $((\langle\theta, x_t\rangle - y_t)^2) = (Y - UX/\sqrt{T})^2$ and

$$\begin{aligned}(a_t - y_t)^2 - (\langle\theta, x_t\rangle - y_t)^2 &= (a_t - y_t)^2 - \big((UX/\sqrt{T}) \operatorname{sign} a_t - y_t\big)^2 \\ &\geqslant Y^2 - \big(Y - UX/\sqrt{T}\big)^2 = 2\frac{UXY}{\sqrt{T}} - \frac{(UX)^2}{T} \geqslant \frac{UXY}{\sqrt{T}}\,.\end{aligned}$$

The final inequality holds as long as $UX \leqslant Y$ and $T \geqslant 1$. Summing over $t \leqslant T$ gives the result. ■

Appendix D. Lower Bound for the Hinge Loss

The next result shows a lower bound for the hinge loss that matches the upper bound by Mhammedi and Koolen (2020) in the regime where $UX \leqslant 1$. We prove it by relating the hinge loss to linear losses and then applying a result of McMahan and Streeter (2012).

Theorem 27 *In* 1*-dimensional online classification with the hinge loss, consider an algorithm that guarantees* $R_T(0) \leqslant \varepsilon$ *for any sequence of data. Then for any* $U, X > 0$ *and for any* $T_0 \geqslant 0$*, there exists a sequence* (x_t, y_t) *in* $[-X, X] \times [-Y, Y]$ *such that for some* $T \geqslant T_0$*,*

$$\sup_{\theta \in [-U,U]} R_T(\theta) \geqslant 0.336\,(UX \wedge 1)\sqrt{T \ln\left(\frac{(UX \wedge 1)\sqrt{T}}{\delta}\right)}\,.$$

Proof Let us assume $UX \leqslant 1$, as the general case can be derived by applying the result with $U = 1/X$. Consider the feature sequence $x_t = X$, and denote by $a_t \in \mathbb{R}$ the sequence of actions produced by the algorithm. Then for any $|u| \leqslant U \leqslant 1/X$, we have $|ux_t y_t| \leqslant 1$ for all t and therefore

$$\ell(a_t, y_t) - \ell(ux_t, y_t) \geqslant 1 - a_t y_t - (1 - uXy_t) = -y_t(a_t - uX)\,.$$

This implies that the regret of any algorithm for the hinge loss is lower bounded by its regret for a sequence of linear losses $\theta \mapsto -y_t\theta$ against the comparator uX; denote this regret by $\tilde{R}_T(uX)$. Then for any sequence of y_t, we have $R_T(u) \geqslant R_T(uX)$.

This implies in particular that $\tilde{R}_T(0) \leqslant R_T(0)$, which is assumed to be less than ε for any data sequence: the assumptions of McMahan and Streeter (2012, Theorem 7) are satisfied by the algorithm. Therefore, for any $\tilde{U} > 0$ and $T_0 \geqslant 0$, there exists a sequence of y_t's, and a comparator u with norm $\tilde{U}$ such that for some $T \geqslant T_0$,

$$R_T(u/X) \geqslant \tilde{R}_T(u) \geqslant 0.336\,\tilde{U}\sqrt{T \log\left(\frac{\tilde{U}\sqrt{T}}{\varepsilon}\right)}\,.$$

The claimed bound follows by reparameterising u by u/X. ■