Efficient online learning with kernels for adversarial large scale problems

Rémi Jézéquel, Pierre Gaillard, Alessandro Rudi

EFFICIENT ONLINE LEARNING WITH KERNELS FOR ADVERSARIAL LARGE SCALE PROBLEMS

Rémi Jézéquel **Pierre Gaillard** **Alessandro Rudi**

INRIA - Département d'Informatique de l'École Normale Supérieure
PSL Research University
Paris, France

ABSTRACT

We are interested in a framework of online learning with kernels for low-dimensional but large-scale and potentially adversarial datasets. Considering the Gaussian kernel, we study the computational and theoretical performance of online variations of kernel Ridge regression.

The resulting algorithm is based on approximations of the Gaussian kernel through Taylor expansion. It achieves for d-dimensional inputs a (close to) optimal regret of order $O((\log n)^{d+1})$ with per-round time complexity and space complexity $O((\log n)^{2d})$. This makes the algorithm a suitable choice as soon as $n \gg e^d$ which is likely to happen in a scenario with small dimensional and large-scale dataset.

***K*eywords** Online learning · Kernel methods · Large scale

1 Introduction

Nowadays the volume and the velocity of data flows are deeply increasing. Consequently many applications need to switch from batch to online procedures that can treat and adapt to data on the fly. Furthermore to take advantage of very large datasets, non-parametric methods are gaining increasing momentum in practice. Yet the latter often suffer from slow rates of convergence and bad computational complexities. At the same time, data is getting more complicated and simple stochastic assumptions such as i.i.d. data are often not satisfied. In this paper, we try to combine these different aspects due to large scale and arbitrary data. We build a non-parametric online procedure based on kernels, which is efficient for large data sets and achieves close to optimal theoretical guarantees.

Online learning is a subfield of machine learning where some learner sequentially interacts with an environment and tries to learn and adapt on the fly to the observed data as one goes along. We consider the following sequential setting summarized in Figure 1. At each iteration $t \geq 1$, the learner receives some input $x_t \in \mathcal{X}$; makes a prediction $\widehat{y}_t \in \mathbb{R}$ and the environment reveals the output $y_t \in \mathbb{R}$. The inputs x_t and the outputs y_t are sequentially chosen by the environment and can be arbitrary. Learner's goal is to minimize his cumulative regret

$$R_n(f) := \sum_{t=1}^{n} (y_t - \widehat{y}_t)^2 - \sum_{t=1}^{n} \big(y_t - f(x_t)\big)^2 \tag{1}$$

uniformly over all functions f in a space of functions $\mathcal{H}$. We will consider Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$, [see next section or 1, for more details]. It is worth noting here that all the properties of a RKHS are controlled by the associated *kernel function* $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, usually known in closed form, and that many function spaces of interest are (or are contained in) RKHS, e.g. when $\mathcal{X} \subseteq \mathbb{R}^d$: polynomials of arbitrary degree, band-limited functions, analytic functions with given decay at infinity, Sobolev spaces and many others [2].

Previous work Kernel regression in a batch statistical setting has been widely studied by the statistics community. Our setting of online kernel regression with adversarial data is more recent.

Most of existing work focuses on the linear setting (i.e., linear kernel). First work on online linear regression dates back to [3]. [4] provided the minimax rates (together with an algorithm) and we refer to reader to references therein for

a recent overview of the literature in the linear case. We only recall relevant work for this paper. [5, 6] designed the nonlinear Ridge forecaster (referred to as AWV). In linear regression (linear kernel), it achieves the optimal regret of order $O(d \log n)$ uniformly over all ℓ_2-bounded vectors. As we will see in Inequality (5), in online kernel regression, the kernel version of nonlinear Ridge forecaster (referred to as Kernel-AWV) obtains a regret upper-bounded as

$$R_n(f) \lesssim \lambda \|f\|_{\mathcal{H}}^2 + B^2 d_{\text{eff}}(\lambda) \quad f \in \mathcal{H}\,, \tag{2}$$

where

$$d_{\text{eff}}(\lambda) := \operatorname{Tr}(K_{nn}\big(K_{nn} + \lambda I_n\big)^{-1}\big) \tag{3}$$

is the effective dimension, where $K_{tt} \in \mathbb{R}^{t\times t}$ denotes the *kernel matrix* at time t and is defined as $K_{tt} = \big(k(x_i, x_j)\big)_{1\le i,j\le t}$. The above upper-bound on the regret is essentially optimal (see next section). Yet the per round complexity and the space complexity of Kernel-AWV are $O(n^2)$. In this paper, we aims at reducing this complexity while keeping optimal regret guarantees.

Though the literature on online contextual learning is vast, little considalessandro.rudi@inria.frr a clipped version of kernel Ridge regression and by [7] for a clipped version of Kernel Online Newton Step (KONS) for general exp-concave loss functions.

The computational complexity ($O(n^2)$ per round) of these algorithms is however prohibitive for large datasets. [7] and [8] provide approximations of KONS to get manageable complexities. However these come with deteriorated regret guarantees. [7] improve the time and space complexities by a factor $\gamma > 0$ enlarging the regret upper-bound by $1/\gamma$. [8] propose an efficient approximation of KONS based on Nyström approximation [9, 10] and restarts with per-round complexities $O\big(d_{\text{eff}}(\lambda/n)^2\big)$. Yet their regret bound suffers an additional multiplicative factor $d_{\text{eff}}(\lambda/n)$ with respect to (2) because of the restarts. Another relevant approximation scheme of Online Kernel Learning was done by [11]. The authors consider online gradient descent algorithms which they approximate using Nyström approximation or a Fourier basis. However since they use general Lipschitz loss functions and consider ℓ_1-bounded functions f, their regret bounds of order $O(\sqrt{n})$ are hardly comparable to ours and seem suboptimal in n in our restrictive setting with square loss and Gaussian kernel.

Contributions and outline of the paper The main contribution of the paper is to propose for the Gaussian kernel an efficient algorithm for online kernel regression. Theorem 4.1 shows that the latter satisfies (up to log) the optimal regret bounds (2) while enjoying a total computational cost of $O\big(n\ \log^{2d}\left(\frac{n}{\lambda}\right)\big)$ in time and $O\big(\log^{2d}\left(\frac{n}{\lambda}\right)\big)$ in space. For the Gaussian kernel, these respectively correspond to $O\big(n d_{\text{eff}}(\lambda)^2\big)$ and $O\big(d_{\text{eff}}(\lambda)^2\big)$.

Our method denoted $\tilde{\phi}$-AWV is a based on Kernel-AWV. It consists in approximating the Gaussian kernel by a finite set of basis functions. The number of functions included in the basis is a parameter to be optimized and fixes an approximation-computational trade-off. Section 2 recalls the optimal regret bound of Kernel-AWV. Section 3 introduces our algorithm to approximate Kernel-AWV in general RKHS. Section 4 considers the important particular case of the Gaussian kernel and studies its approximation properties to state our main result (Theorem 4.1). Finally, we perform in Section 5 several experiments based on real and simulated data to compare the performance (in regret and in time) of our methods with competitors.

Notations We recall here basic notations that we will use throughout the paper. Given a vector $v \in \mathbb{R}^d$, we write $v = (v^{(1)}, \dots, v^{(d)})$. We denote by $\mathbb{N}_0 = \mathbb{N}\cup\{0\}$ the set of non-negative integers and for $p \in \mathbb{N}_0^d$, $|p| = p^{(1)} + \cdots + p^{(d)}$. We denote by $\|\cdot\|$ the Euclidean norm and by $\|\cdot\|_{\mathcal{H}}$ the norm for a Hilbert space $\mathcal{H}$. Write $v^\top w$, the dot product between $v, w \in \mathbb{R}^D$. The conjugate transpose for linear operator Z on $\mathcal{H}$ will be denoted Z^*. The notation $\lesssim$ will refer to rough inequalities up to logarithmic multiplicative factors. Finally we will denote $a \vee b = \max(a, b)$ and $a \wedge b = \min(a, b)$, for $a, b \in \mathbb{R}$.

For $t = 1, \dots, n$
- the learner receives input $x_t \in \mathcal{X} \subset \mathbb{R}^d$
- the learner predicts $\widehat{y}_t := \widehat{f}_t(x_t) \in \mathbb{R}$
- the environment reveals output $y_t \in \mathbb{R}$
- the learner suffers loss $(y_t - \widehat{y}_t)^2$

Figure 1: Setting of online non-parametric prediction.

2 Background

In this section we introduce some background on kernel methods and their known theoretical properties.

Kernels. Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a positive definite kernel [1]. The function k is characterized by the existence of a *feature map* $\phi : \mathcal{X} \to \mathbb{R}^D$, with $D \in \mathbb{N} \cup \{\infty\}$[1], such that

$$k(x, x') = \phi(x)^\top \phi(x'),$$

moreover the *reproducing kernel Hilbert space* (RKHS) associated to k is characterized by

$$\mathcal{H} = \{f \mid f(x) = w^\top \phi(x),\ w \in \mathbb{R}^D, x \in \mathcal{X}\},$$

with inner product $\langle f, g\rangle_{\mathcal{H}} := v^\top w$, for $f, g \in \mathcal{H}$ defined by $f(x) = v^\top \phi(x)$, $g(x) = w^\top \phi(x)$ and $v, w \in \mathbb{R}^D$. For more details and different characterizations of $k, \mathcal{H}$, see [1, 2]. It's worth noting that the knowledge of ϕ is not necessary when working with functions of the form $f = \sum_{i=1}^p \alpha_i \phi(x_i)$, with $\alpha_i \in \mathbb{R}$, $x_i \in \mathcal{X}$ and finite $p \in \mathbb{N}$, indeed

$$f(x) = \sum_{i=1}^p \alpha_i \phi(x_i)^\top \phi(x) = \sum_{i=1}^p \alpha_i k(x_i, x),$$

and moreover $\|f\|_{\mathcal{H}}^2 = \alpha^\top K_{pp} \alpha$, with K_{pp} the kernel matrix associated to the set of points $x_1, \dots, x_p$.

Kernel-AWV. The nonlinear Ridge forecaster (denoted *AWV*) on the space of linear functions on $\mathcal{X} = \mathbb{R}^d$ has been introduced and analyzed in [5, 6]. It predicts $\widehat{y}_t = \widehat{w}_t^\top x_t$, where

$$\widehat{w}_t = \operatorname*{argmin}_{w \in \mathbb{R}^d} \sum_{s=1}^{t-1} \big(y_s - w^\top x_s)\big)^2 + \lambda \|w\|^2 + (w^\top x_t)^2.$$

We consider here a straightforward generalization to kernels (we denote it *Kernel-AWV*) in which the forecaster produces $\widehat{y}_t = \widehat{f}_t(x_t)$, where

$$\widehat{f}_t = \operatorname*{argmin}_{f \in \mathcal{H}} \sum_{s=1}^{t-1} \big(y_s - f(x_s)\big)^2 + \lambda \big\|f\big\|_{\mathcal{H}}^2 + f(x_t)^2. \tag{4}$$

A variant of this algorithm, more used in the context of data independently sampled from distribution, is known as *Ridge regression* (*kernel Ridge regression*) corresponding to solving the problem above, without the last penalization term $(w^\top x_t)^2$, $(f(x_t)^2$ for the kernel case).

In the next proposition we state a preliminary result which proves that Kernel-AWV achieves a regret depending on the eigenvalues of the kernel matrix. In the next paragraph we show that the regret is optimal (up to log terms).

Proposition 2.1. *Let λ, B. For any RKHS $\mathcal{H}$, for all $n \geq 1$, for all inputs $x_1, \dots, x_n \in \mathcal{X}$ and all $y_1, \dots, y_n \in [-B, B]$, the regret of Kernel-AWV is upper-bounded for all $f \in \mathcal{H}$ as*

$$R_n(f) \leq \lambda \big\|f\big\|_{\mathcal{H}}^2 + B^2 \sum_{k=1}^n \log\left(1 + \frac{\lambda_k(K_{nn})}{\lambda}\right),$$

where $\lambda_k(K_{nn})$ denotes the k-th largest eigenvalue of K_{nn}.

The proof is a direct consequence of the known regret bound of *AWV* in the finite dimensional linear regression setting—see Theorem 11.8 of [12] or Theorem 2 of [13]. For completeness, we reproduce the analysis for infinite dimensional space (RKHS) in Appendix A. In online linear regression in dimension d, the above result implies the optimal rate of convergence $dB^2 \log(n) + O(1)$ (see [13] and [6]).

Optimal regret for Kernel-AWV. We show here that Proposition 2.1 yields optimal regret (up to log) of the form (2) for online kernel regression. The following proposition upper-bounds for any bounded kernel (i.e. $\sup_{x \in \mathcal{X}} k(x, x) \leq \kappa^2$, with $\kappa > 0$) the right-hand side of Inequality (5) with the effective dimension.

Proposition 2.2. *For all $n \geq 1$ $\lambda > 0$ and all input sequences $x_1, \dots, x_n \in \mathcal{X}$,*

$$\sum_{k=1}^n \log\left(1 + \frac{\lambda_k(K_n)}{\lambda}\right) \leq \log\Big(e + \frac{en\kappa^2}{\lambda}\Big) d_{\mathit{eff}}(\lambda)\,.$$

[1] when $D = \infty$ we consider $\mathbb{R}^D$ as the space of squared summable sequences.

Algorithm 1 $\tilde{\phi}$-AWV

Input: $\lambda > 0$, $\tilde{\phi} : \mathcal{X} \to \mathbb{R}^r$ for $r \geq 1$
Initialization: $A_0^{-1} = \lambda^{-1} I_r$, $b_0 = 0$
For $t = 1, \ldots, n$

- receive $x_t \in \mathcal{X}$
- compute $v_t = \tilde{\phi}(x_t) \in \mathbb{R}^r$
- update $A_t^{-1} = A_{t-1}^{-1} - \frac{(A_t^{-1} v_t)(A_t^{-1} v_t)^\top}{1 + v_t^\top A_t^{-1} v_t}$
- predict $\widehat{y}_t = \tilde{\phi}(x_t)^\top A_t^{-1} b_{t-1}$
- receive $y_t \in \mathbb{R}$
- update $b_t = b_{t-1} + v_t y_t$

Combined with Proposition 2.1, this entails that Kernel-AWV satisfies the close to be optimal regret bound

$$R_n(f) \leq \lambda \|f\|_{\mathcal{H}}^2 + B^2 \log\Big(e + \frac{en\kappa^2}{\lambda}\Big) d_{\text{eff}}(\lambda)\,. \tag{5}$$

As discussed in the introduction, such an upper-bound on the regret is not new and was already proved by [14] or by [7] for other algorithms. An advantage of Kernel-AWV is that it does not require any clipping and thus the beforehand knowledge of $Y > 0$ to obtained Proposition 2.1. Furthermore, we slightly improve the constants in the above proposition. We note that the regret bound for Kernel-AWV is optimal up to log terms when λ is chosen minimizing r.h.s. of Eq. (5), since it meets known minimax lower bounds for the setting where $(x^{(i)}, y_i)_{i=1}^n$ are sampled independently from a distribution. For more details on minimax lower bounds see [15], in particular Eq. (9), related discussion and references therein, noting that their λ correspond to our λ/n and our $d_{\text{eff}}(\lambda)$ corresponds to their $\gamma(\lambda/n)$.

It is worth pointing out that in the worst case $d_{\text{eff}}(\lambda) \leq \kappa^2 n/\lambda$ for any bounded kernel. In particular, optimizing the bound yields $\lambda = O(\sqrt{n})$ and a regret bound of order $O(\sqrt{n})$. In the special case of the Gaussian kernel (which we consider in Section 4), the latter can be improved to $d_{\text{eff}}(\lambda) \lesssim \big(\log(n/\lambda)\big)^d$ (see [16]) which entails $R_n(f) \leq O\big((\log n)^{d+1}\big)$ for well tuned value of λ.

3 Regression with non-linear embedding

In previous section we have seen that, while Kernel-AWV achieves optimal regret, it has computational requirements that are $O(n^3 + n^2 d)$ in time and $O(n^2)$ in space, for n steps of the algorithm, making it unfeasible in the context of large scale datasets, i.e. $n \gg 10^5$. In this paper we consider and analyze a variation of Kernel-AWV denoted $\tilde{\phi}$-AWV.

Let $G = \{g_1, \ldots, g_r\} \subset \mathcal{H}$ be a set of functions and denote by $\tilde{\phi} : \mathcal{X} \to \mathbb{R}^r$ the map

$$\tilde{\phi}(x) = Q^{-1/2} v(x), \tag{6}$$

with $v(x) = (g_1(x), \ldots, g_r(x))$, and $Q \in \mathbb{R}^{r \times r}$ defined as $Q_{ij} = \langle g_i, g_j \rangle_{\mathcal{H}}$. The algorithms consists in embedding x in $\mathbb{R}^r$ via $\tilde{\phi}$ and then performing linear AWV (see Algorithm 1). This reduces the total computational complexity to $O(nr^2 + nrd + r^3)$ in time and $O(r^2)$ in space. While the algorithm is very simple, it achieves optimal regret in some interesting non-parametric cases, with dramatically reduced computational complexity w.r.t. Kernel-AWV, as analyzed in the next section.

Now we are ready to consider the effect on the regret of learning using the embedding of X in $\mathbb{R}^r$ via the map $\tilde{\phi} : \mathcal{X} \to \mathbb{R}^r$. In the next theorem we assume that k is a positive definite kernel, such that $\sup_{x \in \mathcal{X}} k(x, x) \leq \kappa^2$, for $\kappa > 0$ and denote by $\mathcal{H}$ the associated RKHS.

Theorem 3.1. *When $\tilde{\phi}$-AWV is run with $\lambda > 0$ and $\tilde{\phi}$ built using linearly independent set of functions $G \subset \mathcal{H}$ (see Eq.* (6)*), then for all $f \in \mathcal{H}$*

$$R_n(f) \leq \lambda \|f\|_{\mathcal{H}}^2 + B^2 \sum_{j=1}^n \log\left(1 + \frac{\lambda_j(K_{nn})}{\lambda}\right) + 2n\|f\|_{\mathcal{H}}(\|f\|_{\mathcal{H}}\kappa + B) \sup_{x \in \mathcal{X}} \sqrt{k(x,x) - \|\tilde{\phi}(x)\|^2}\,, \tag{7}$$

for any sequence $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times [-B, B]$.

The result above bounds the regret of the $\tilde{\phi}$-AWV for any function $f \in \mathcal{H}$ and holds for any bounded kernel and any explicit feature map built as in Eq. (6), with G satisfying the basic properties required by the theorem. This result will be used in Thm. 4.1 for Gaussian kernel and a specific set of functions derived from the Taylor expansion of the kernel.

3.1 Proof

First note that by defining the linear operator $Z : \mathcal{H} \to \mathbb{R}^r$ as $Zf = (\langle g_1, f\rangle_{\mathcal{H}}, \dots, \langle g_r, f\rangle_{\mathcal{H}})$, for any $f \in \mathcal{H}$, we have that $Q = ZZ^*$ and $v(x) = Z\phi(x)$, so

$$\tilde{\phi}(x) = V\phi(x),$$

with $V := (ZZ^*)^{-1/2}Z$. Now note that V is a partial isometry, indeed $VV^* = I_r$ and $P := V^*V$ is an orthogonal projection operator.

Let us denote by $\widehat{y}_t$ the predictions done by $\tilde{\phi}$-AWV. Let $f \in \mathcal{H}$ and denote by $\tilde{f}$ the function $\tilde{f} = Pf \in \mathcal{H}$. We have that

$$\begin{aligned} R_n(f) &:= \sum_{t=1}^{n}(y_t - \widehat{y}_t)^2 - \sum_{t=1}^{n}(y_t - f(x_t))^2 \\ &= \widetilde{R}_n(w) \;+\; c_n(f), \end{aligned}$$

where $\widetilde{R}_n(\tilde{f}) = \sum_{t=1}^{n}(y_t - \widehat{y}_t)^2 - \sum_{t=1}^{n}(y_t - \tilde{f}(x_t))^2$ and $c_n(f) = \sum_{t=1}^{n}(y_t - \tilde{f}(x_t))^2 - \sum_{t=1}^{n}(y_t - f(x_t))^2$.

Bounding $\tilde{R}_n(\tilde{f})$. Denote by $w \in \mathbb{R}^r$ the vector $w = Vf$. Note that

$$\tilde{\phi}(x)^\top w = (V\phi(x))^\top (Vf) = \langle \phi(x), Pf\rangle_{\mathcal{H}} = \tilde{f}(x),$$

then $\widetilde{R}_n(\widetilde{f})$ is just the regret of linear AWV computed with respect to the vector w and data $(\tilde{\phi}(x_t), y_t)_{t=1}^n$. By applying standard result for linear AWV, we have then

$$\widetilde{R}_n(\widetilde{f}) = R_n(w) \le \lambda\|w\|^2_{\mathbb{R}^r} + B^2\sum_{i=1}^{r}\log(1 + \lambda_i(\widetilde{C}_n)/\lambda),$$

with $\widetilde{C}_n = \sum_{i=1}^{n}\tilde{\phi}(x^{(i)})\tilde{\phi}(x^{(i)})^\top$ and $\lambda_i(\widetilde{C}_n)$ the i-th eigenvalue, considering the multiplicity, of $\widetilde{C}_n$.

Bounding the bias. By definition of w, we have that

$$\|w\|^2_{\mathbb{R}^r} = \|Vf\|^2_{\mathbb{R}^r} \le \|V\|^2\|f\|^2_{\mathcal{H}} \le \|f\|^2_{\mathcal{H}},$$

where the last step is due to the fact that V is a partial isometry.

Bounding the variance. Then $\widetilde{C}_n = VC_nV^*$, with $C_n = \sum_{t=1}^{n}\phi(x)\otimes\phi(x)$. By defining the linear operator $S_n : \mathcal{H} \to \mathbb{R}^n$ as $S_nf = (\langle\phi(x_1), f\rangle_{\mathcal{H}}, \dots, \langle\phi(x_n), f\rangle_{\mathcal{H}})$, we have that $C_n = S_n^*S_n$ and $K_{nn} = S_nS_n^*$. Moreover, denoting by $A \preceq B$, the fact that $B - A$ is a positive semidefinite Hermitian operator (when A, B Hermitian operators), we have that $P \preceq I$ and so $S_nPS_n^* \preceq S_nS_n^* = K_{nn}$. By recalling that $A \preceq B$, when A, B both positive Hermitian operators, implies that $\lambda_i(A) \le \lambda_i(B)$ [17], and the fact that $\lambda_i(A^*A) = \lambda_i(AA^*)$ for any compact linear operator A, we have

$$\begin{aligned} \lambda_i(\widetilde{C}_n) &= \lambda_i(VC_nV^*) = \lambda_i(VS_nS_n^*V^*) \\ &= \lambda_i(S_n^*V^*VS_n) = \lambda_i(S_n^*PS_n) \\ &\le \lambda_i(K_{nn}). \end{aligned}$$

Finally, by bounding $\lambda_i(K_{nn})$ with $\kappa^2 n$ and using that $\log(1+x)$ with $x/(1+x)(1+\log(1+x_{\max}))$ for $0 \le x \le x_{\max}$ as in proposition 2.1, we have

$$\begin{aligned} \log(1 + \lambda_i(\widetilde{C}_n)/\lambda) &\le \log(1 + \lambda_i(K_{nn})/\lambda) \\ &\le \frac{\lambda_i(K_{nn})}{\lambda_i(K_{nn}) + \lambda}(1 + \log(1 + \kappa^2 n/\lambda)). \end{aligned}$$

So $\sum_{i=1}^{n}\log(1 + \lambda_i(\widetilde{C}_n)/\lambda) \le d_{\text{eff}}(\lambda)(1 + \log(1 + \kappa n/\lambda))$.

Bounding approximation error $c_n(f)$. Note that

$$\begin{aligned} &(y_t - \tilde{f}(x_t))^2 - (y_t - f(x_t))^2 \\ &\quad = (f(x_t) - \tilde{f}(x_t))(2y_t - f(x_t) - \tilde{f}(x_t)). \end{aligned} \tag{8}$$

Now note that $|y_t| \leq B$, that

$$f(x) = \langle f, \phi(x) \rangle_{\mathcal{H}} \leq \|f\|_{\mathcal{H}} \|\phi(x)\|_{\mathcal{H}} \leq \kappa f,$$

and analogously $\tilde{f}(x) \leq \kappa \|Pf\|_{\mathcal{H}} \leq \kappa \|f\|_{\mathcal{H}}$. Therefore,

$$\left|2y_t - f(x_t) - \tilde{f}(x_t)\right| \leq 2(B + \kappa)\,. \tag{9}$$

Moreover

$$\begin{aligned} f(x) - \tilde{f}(x) &= \langle \phi(x), f - Pf \rangle = \langle \phi(x), (I - P)f \rangle \\ &= \langle (I - P)\phi(x), f \rangle \\ &\leq \|(I - P)\phi(x)\|_{\mathcal{H}} \|f\|_{\mathcal{H}}. \end{aligned}$$

Finally, since $I - P$ is a projection operator, $(I - P)^2 = I - P$, then

$$\begin{aligned} \|(I - P)\phi(x)\|_{\mathcal{H}}^2 &= \langle \phi(x), (I - P)\phi(x) \rangle_{\mathcal{H}} \\ &= \langle \phi(x), \phi(x) \rangle_{\mathcal{H}} - \langle \phi(x), P\phi(x) \rangle_{\mathcal{H}}\,. \end{aligned}$$

To conclude, note that $\langle \phi(x), \phi(x) \rangle_{\mathcal{H}} = k(x, x)$ and that

$$\begin{aligned} \langle \phi(x), P\phi(x) \rangle_{\mathcal{H}} &= \langle \phi(x), V^* V \phi(x) \rangle_{\mathcal{H}} \\ &= (V\phi(x))^\top (V\phi(x)) = \|\tilde{\phi}(x)\|^2. \end{aligned}$$

So, by defining $c_{\mathcal{X},G} = \sup_{x \in \mathcal{X}} |k(x, x) - \|\tilde{\phi}(x)\|^2|$, we get

$$\left|f(x) - \tilde{f}(x)\right| \leq c_{\mathcal{X},G}^{1/2} \|f\|_{\mathcal{H}}.$$

Combining with (8) and (9) and summing over n yields

$$\begin{aligned} &\textstyle\sum_{t=1}^n (y_t - \tilde{f}(x_t))^2 - (y_t - f(x_t))^2 \\ &\qquad \leq 2n\|f\|_{\mathcal{H}}(\|f\|_{\mathcal{H}}\kappa + B) c_{\mathcal{X},G}^{1/2}\,, \end{aligned}$$

which concludes the proof.

4 Learning with Taylor expansions and Gaussian kernel for very large data set

In this section we focus on non-parametric regression with a widely used basis function, the *Gaussian kernel*

$$k(x, x') = \exp(-\|x - x'\|^2/(2\sigma^2)),$$

for $\sigma > 0$ and the associated RKHS $\mathcal{H}$. Using the results from the previous section, we prove that $\tilde{\phi}$-AWV with the considered basis, achieves optimal regret with r only in the order of $O(\text{polylog}(n/\lambda))$. Leading to a computational complexity that is only $O(n \text{ polylog}(n/\lambda))$ for optimal regret.

To really exploit the result in the previous section, we need a basis that (1) approximates very well the Gaussian kernel and at the same time (2) whose projection is easy to compute. We consider the following basis of functions, for $k \in \mathbb{N}_0^d$,

$$g_k(x) = \prod_{i=1}^d \psi_{k_i}(x^{(i)}), \quad \psi_t(x) = \frac{x^t}{\sigma^t \sqrt{t!}} e^{-\frac{x^2}{2\sigma^2}}\,. \tag{10}$$

Let us mention that for one dimensional data this corresponds to Taylor expansion of the Gaussian kernel. In particular we will use $\tilde{\phi}$-AWV with the following subset of functions

$$G_M = \{g_k \mid |k| \leq M, k \in \mathbb{N}_0^d\},$$

where $|k| := k_1 + \cdots + k_d$, for $k \in \mathbb{N}_0^d$.

Our main theorem below states that $\tilde{\phi}$-AWV using the basis G_M gets optimal regret while enjoying low complexity. The size of the basis M controls the trade-off between approximating well the Gaussian kernel (to incur low regret) and large computational cost. Theorem 4.1 optimizes M so that the approximation term of Theorem 3.1 (due to kernel approximation) is of the same order than the optimal regret.

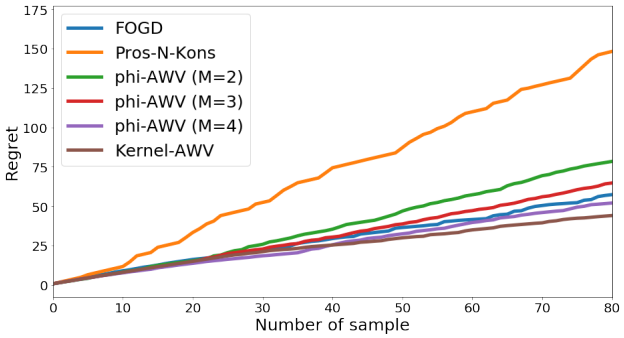

Figure 2: Regret in adversarial setting, see Sec. 5.

Theorem 4.1. *Let $\lambda > 0, n \in \mathbb{N}$ and let $R, B > 0$. Assume that $\|x_t\| \leq R$ and $|y_t| \leq B$. When M is chosen as*

$$M = \left\lceil \frac{8R^2}{\sigma^2} \vee 2\log\frac{2n(1+\kappa)}{\lambda \wedge \lambda^{-1}} \right\rceil, \quad \textit{then}$$

1. *running $\tilde{\phi}$-AWV using G_M as set of functions leads to a map $\tilde{\phi} : \mathcal{X} \to \mathbb{R}^r$, with*

$$r \leq \left(3 + \frac{21R^2}{\sigma^2 d} \vee \frac{6}{d}\log\frac{2n(1+\kappa)}{\lambda \wedge \lambda^{-1}}\right)^d$$

and a computational cost per iteration of

$$O\left(\left(3 + \frac{1}{d}\log\frac{n}{\lambda \wedge \lambda^{-1}}\right)^{2d}\right) \quad \textit{in space and time},$$

2. *moreover, for any $f \in \mathcal{H}$, the algorithm achieves a regret bounded by*

$$R_n(f) \leq 2\lambda\|f\|_{\mathcal{H}}^2 + 2B^2 \sum_{j=1}^{n} \log\left(1 + \frac{\lambda_j(K_{nn})}{\lambda}\right).$$

Therefore $\tilde{\phi}$-AWV achieves a regret-bound only deteriorated by a multiplicative factor of 2 with respect to the bound obtained by Kernel-AWV (see Proposition 2.1). From Proposition 2.2 this also yields (up to log) the optimal bound (2).

In particular, it is known [16] for the Gaussian kernel that

$$d_{\text{eff}}(\lambda) \leq 3\Big(6 + \frac{41}{d}\frac{R^2}{2\sigma^2} + \frac{3}{d}\log\frac{n}{\lambda}\Big)^d = O\Big(\Big(\log\frac{n}{\lambda}\Big)^d\Big).$$

The upper-bound is matching even in the i.i.d. setting for non trivial distributions. In this case, we have $r \lesssim d_{\text{eff}}(\lambda)$. The per-round space and time complexities are thus $O\big(d_{\text{eff}}(\lambda)^2\big)$. Though our method is quite simple (since it uses fixed explicit embedding) it is able to recover results -in terms of computational time and bounds in the adversarial setting- that are similar to results obtained in the more restrictive i.i.d. setting obtained via much more sophisticated methods, like learning with (1) Nyström with importance sampling via leverage scores [18], (2) reweighted random features [19, 20], (3) volume sampling [21]. By choosing λ as follows,

$$\lambda = \frac{B^2}{\|f\|^2},$$

to minimize the r.h.s. of the regret bound of the theorem, we get

$$R_n(f) \lesssim \left(\log\frac{n\|f\|_{\mathcal{H}}^2}{B^2}\right)^{d+1} B^2.$$

Note that the optimal λ does not depend on n and can be optimized in practice through standard online calibration methods such as doubling trick or with an expert advice algorithm [12]. Similarly, though we use a fixed number of features M in the experiments, the latter could be increased slowly over time thanks to online calibration techniques.

It is worth mentioning that there is a multiplicative gap of order $O(\log(n))$ with the known minimax lower bound (see [15]) in the i.i.d. offline setting. Yet, it is known that such a gap between the offline i.i.d. setting and the online adversarial framework is unavoidable in some cases such as finite linear regression (see [6]). We leave as an open question whether this gap can be closed in this case.

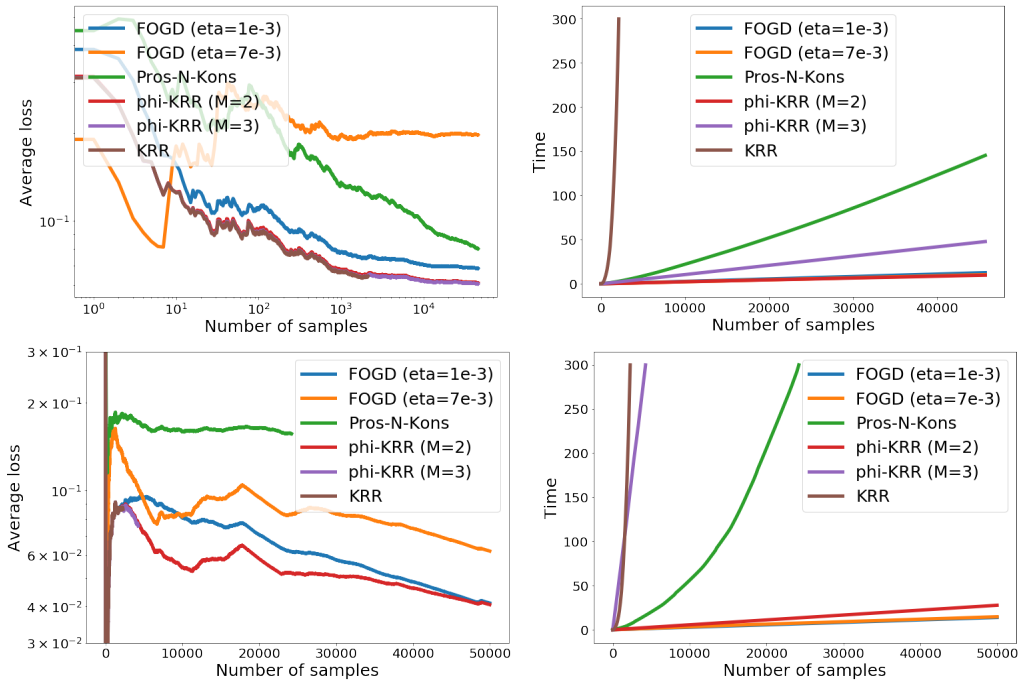

Figure 3: Average loss and time on (top): regression casp ($n = 4.5 \times 10^4$, $d = 9$); (bottom) classification ijcnn1 ($n = 1.5 \times 10^5$, $d = 22$).

4.1 Proof of Theorem 4.1

To apply our Thm. 3.1, we need first (1) to recall that the functions g_k, $k \in \mathbb{N}_0^d$ are in $\mathcal{H}$, (2) to show that they can approximate perfectly the kernel and (3) to quantify the approximation error of G_M for the kernel function. First we recall some important existing results about the considered set of functions. For completeness, we provide self-contained (and often shorter and simplified) proofs of the following lemmas in the supplementary material.

The next lemma states that g_k with $k \in \mathbb{N}$ is an orthonormal basis for $\mathcal{H}$ induced by the Gaussian kernel.

Lemma 4.2 ([22]). *For any $k, k' \in \mathbb{N}_0^d$,*

$$g_k \in \mathcal{H}, \quad \|g_k\|_{\mathcal{H}} = 1, \quad \langle g_k, g_{k'} \rangle_{\mathcal{H}} = \mathbb{1}_{k=k'} .$$

Note that byproduct of the lemma, we have that $G_M \subset \mathcal{H}$ and moreover that the matrix Q is the identity, indeed $Q_{ij} = \langle g_{k_i}, g_{k_j} \rangle_{\mathcal{H}} = \mathbb{1}_{k_i=k_j}$). This means that the functions in G_M are linearly independent. Moreover the fact that $Q = I_r$ further simplifies the computation of $\tilde{\phi}$.

The next lemma recalls the expansion of $k(x, x')$ in terms of the given basis.

Lemma 4.3 ([23]). *For any $x \in \mathcal{X}$,*

$$k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}} = \sum_{k \in \mathbb{N}_0^d} g_k(x) g_k(x'). \tag{11}$$

Finally, next lemma provides approximation error of $k(x, x')$ in terms of the set of functions in G_M, when the data is contained in a ball or radius R.

Lemma 4.4 ([23]). *Let $R > 0$. For any $x, x' \in \mathbb{R}^d$ such that $\|x\|, \|x'\| \leq R$*

$$\left| k(x, x') - \sum_{g \in G_M} g(x) g(x') \right| \leq \frac{(R/\sigma)^{2M+2}}{(M+1)!}. \tag{12}$$

Now we are ready to prove Thm. 4.1.

Proof of point 1. First, note that $r := |G_M|$, the cardinality of G_M, corresponds to number of monomials of the polynomial $(1 + x_1 + \cdots + x_d)^d$, i.e. $r := |G_M| = \binom{M+d}{M}$. By recalling that $\binom{n}{k} \leq (en/k)^k$ for any $n, k \in \mathbb{N}$, we have

$$r = \binom{M+d}{M} = \binom{M+d}{d} \leq e^d(1 + M/d)^d.$$

We conclude the proof of the first point of the theorem, by considering that $\tilde{\phi}$-AWV used with the set of functions G_M incurs in a computational cost of $O(nr^2 + nrd)$ in time (no r^3 since we don't need to invert Q which we have proven to be the identity matrix as consequence of Lemma 4.2) and $O(r^2)$ in memory.

Proof of point 2. By Lemma 4.2 we have that $G_M \subset \mathcal{H}$ and $Q = I_r$, so the functions in G_M are linearly independent. Then we can apply Thm. 3.1 obtaining the regret bound in Eq. (7). Denoting by c the constant $2\|f\|_{\mathcal{H}}(\|f\|_{\mathcal{H}}\kappa + B)$, by definition of $\tilde{\phi}$ (see (6)) and the fact that $Q = I_r$, we have $\|\tilde{\phi}(x)\|^2 = \sum_{g \in G_M} g(x)^2$ and so the approximation term in Eq. (7) is bounded by

$$\begin{aligned} c \sup_{x \in \mathcal{X}} \sqrt{k(x,x) - \|\tilde{\phi}(x)\|^2} &\overset{\text{Lem. 4.4}}{\leq} c\frac{(R/\sigma)^{M+1}}{\sqrt{(M+1)!}} \\ &\leq c(2\pi(M+1))^{-1/4} e^{-(M+1)/2 \log \frac{M+1}{eR^2/\sigma^2}} \\ &\leq c(2\pi)^{-1/4} e^{-(M+1)/2}, \end{aligned}$$

where we used the fact that $n!$ is lower bounded by the Stirling approximation as $n! \geq \sqrt{2\pi n} e^{n \log \frac{n}{e}}$, for $n \in \mathbb{N}_0$ and that $M + 1 \geq e^2R^2/\sigma^2$, so $\log \frac{M+1}{eR^2/\sigma^2} \geq 1$. Now, since $M \geq 2\log(2n(1+\kappa)/(\lambda \wedge \lambda^{-1}))$, we have

$$c(2\pi)^{-1/4} e^{-(M+1)/2} \leq \frac{c}{3}\, e^{-M/2} \leq \frac{c\lambda \wedge \lambda^{-1}}{6n(1+\kappa)}.$$

Now note that when $\|f\|_{\mathcal{H}} \geq B$, we have that $c \leq 2n(1+\kappa)\|f\|_{\mathcal{H}}^2$, while, when $\|f\|_{\mathcal{H}} < B$, we have $c \leq 2n(1+\kappa)B^2$. So, in the first case

$$\frac{c(\lambda \wedge \lambda^{-1})}{6n(1+\kappa)} \leq \frac{c\lambda}{6n(1+\kappa)} \leq \frac{1}{3}\lambda\|f\|_{\mathcal{H}}^2.$$

For the second case, first note that $(\lambda \wedge \lambda^{-1})/2 \leq \log(1 + 1/\lambda)$ for any $\lambda > 0$, so

$$\frac{c(\lambda \wedge \lambda^{-1})/2}{3n(1+\kappa)} \leq B^2 \log\left(1 + \frac{1}{\lambda}\right).$$

Now, since $\log(1 + x)$ is concave on $[0, \infty)$, by subadditivity

$$\textstyle\sum_{j=1}^n \log\left(1 + \frac{\lambda_j(K_{nn})}{\lambda}\right) \geq \log\left(1 + \sum_{j=1}^n \frac{\lambda_j(K_{nn})}{\lambda}\right).$$

By definition of trace in terms of eigenvalues and of the diagonal of K_{nn}, we have

$$\sum_{j=1}^n \lambda_j(K_{nn}) = \mathrm{Tr}(K_{nn}) = \sum_{j=1}^n k(x_j, x_j) = \kappa^2 n,$$

where the last step is due to the fact that for the Gaussian kernel we have $k(x,x) = 1 =: \kappa^2$, for any $x \in \mathcal{X}$. Then

$$\begin{aligned} B^2 \log\left(1 + \frac{1}{\lambda}\right) &\leq B^2 \log\left(1 + \frac{\kappa^2 n}{\lambda}\right) \\ &\leq B^2 \sum_{j=1}^n \log\left(1 + \frac{\lambda_j(K_{nn})}{\lambda}\right). \end{aligned}$$

5 Experiments

We empirically test $\tilde{\phi}$-AWV against several state-of-the-art algorithms for online kernel regression. In particular we test our algorithms in (1) an adversarial setting, (2) on large scale datasets. The following algorithms have been tested:

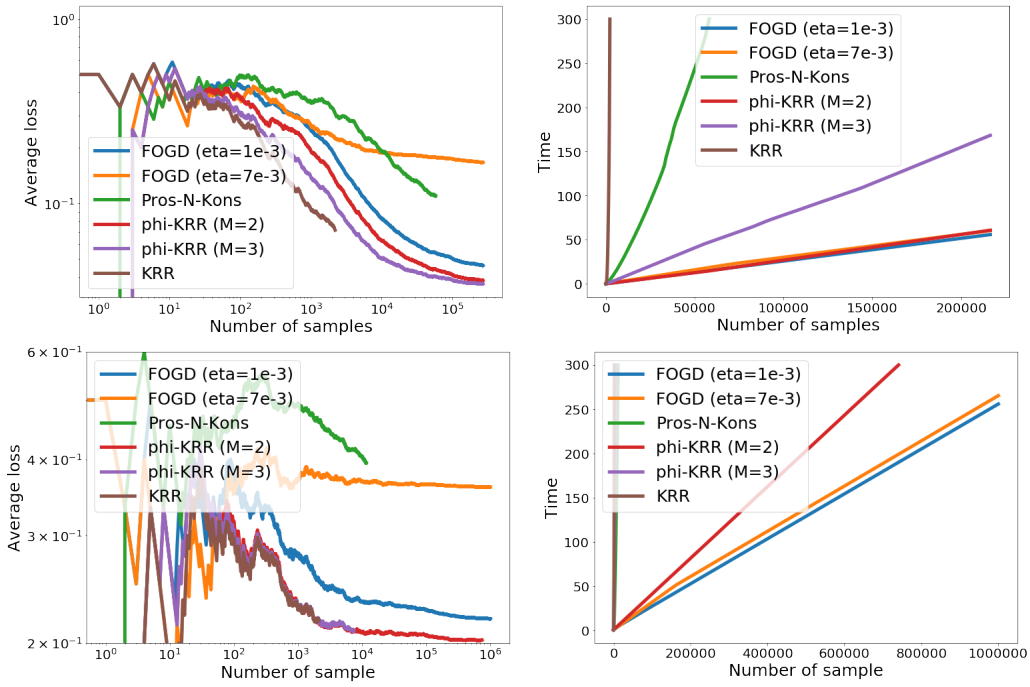

Figure 4: Average classification error and time on: (top) code-rna ($n = 2.7 \times 10^5$, $d = 8$); (bottom) SUSY ($n = 6 \times 10^6$, $d = 22$).

- Kernel-AWV for adversarial setting or Kernel Ridge Regression for i.i.d. and real data settings;
- Pros-N-Kons [7];
- Fourier Online Gradient Descent (FOGD, [11]);
- $\tilde{\phi}$-AWV with $M \in \{2, 3, 4\}$.

The algorithms above have been implemented in python with numpy (the code for our algorithm is in Appendix D). For most algorithms we used hyperparameters from the respective papers. One exception is the learning rate of FOGD because the algorithm is very sensitive to the learning rate which has been globally optimized by grid search. For all algorithms and all experiments, we set $\sigma = 1$ [except for SUSY where $\sigma = 4$, to match accuracy results from 24] and $\lambda = 1$. For Pros-N-Kons, we set $\beta = 1$, $\varepsilon = 0.5$ as in the original paper. The number of random-features in FOGD is fixed to 100 and the learning rate η is 10^{-3} unless otherwise stated. All experiments have been done on a single desktop computer (Intel Xeon CPU E5-2620) with a timeout of 10-min per algorithm. The results of the algorithms are only recorded until this time.

Adversarial simulated data. In this experiment we produced the sequence $(x_t, y_t)_{t \in \mathbb{N}}$ adversarially on the regret function. In particular, given the learning algorithm, we use `scipy` as a greedy adversary i.e. at each step an optimization is done on the regret function to find the worst next x and y. In Figure 2, we plot thus simulations until $n = 80$, with $(x_t, y_t) \in [-1, 1]^d \times [-1, 1]$ where $d = 5$. We see that Kernel-AWV, which does not use any approximation, leads to the best regret. Furthermore, $\tilde{\phi}$-AWV approximations converges very fast to the regret of Kernel-AWV when M increases. The poor performance of Pros-N-Kons is likely because of its frequent restarts which is harmful when n is small. On the contrary, FOGD has surprisingly good performance. We runned the simulation up to $n = 80$ for the high computational cost required by the adversary (especially for algorithms like Kernel-AWV or Pros-N-Kons).

Large scale datasets. The algorithms are evaluated on four datasets from UCI machine learning repository. In particular `casp` (regression) and `ijcnn1`, `cod-rna`, `SUSY` (classification) ranging from 4×10^4 to 6×10^6 datapoints. For all datasets, we scaled x in $[-1, 1]^d$ and y in $[-1, 1]$. In Figs. 3 and 4 we show the average loss (square loss for regression and classification error for classification) and the computational costs of the considered algorithm.

In all the experiments $\tilde{\phi}$-AWV with $M = 2$ approximates reasonably well the performance of kernel forecaster and is usually very fast. We remark that using $\tilde{\phi}$-AWV $M = 2$ on the first million examples of `SUSY`, we achieve in 10 minutes on a single desktop, the same average classification error obtained with specific large scale methods for i.i.d. data [24], although Kernel-AWV is using a number of features reduced by a factor 100 with respect to the one used in for

FALKON in the same paper. Indeed they used $r = 10^4$ Nyström centers, while with $M = 2$ we used $r = 190$ features, validating empirically the effectiveness of the chosen features for the Gaussian kernel. This shows the effectiveness of the proposed approach for large scale machine learning problems with moderate dimension d.

References

[1] Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404, 1950.

[2] Alain Berlinet and Christine Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011.

[3] Dean P. Foster. Prediction in the worst case. *The Annals of Statistics*, 19(2):1084–1090, 1991.

[4] Peter L. Bartlett, Wouter M. Koolen, Alan Malek, Eiji Takimoto, and Manfred K. Warmuth. Minimax Fixed-Design Linear Regression. *JMLR: Workshop and Conference Proceedings*, 40:1–14, 2015. Proceedings of COLT'2015.

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

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

[7] Daniele Calandriello, Alessandro Lazaric, and Michal Valko. Second-order kernel online convex optimization with adaptive sketching. In *International Conference on Machine Learning*, 2017.

[8] Daniele Calandriello, Alessandro Lazaric, and Michal Valko. Efficient second-order online kernel learning with adaptive embedding. In *Neural Information Processing Systems*, 2017.

[9] AJ Smola, B Schölkopf, and P Langley. Sparse greedy matrix approximation for machine learning. In *17th International Conference on Machine Learning, Stanford, 2000*, pages 911–911, 2000.

[10] Christopher KI Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In *Advances in neural information processing systems*, pages 682–688, 2001.

[11] Jing Lu, Steven CH Hoi, Jialei Wang, Peilin Zhao, and Zhi-Yong Liu. Large scale online kernel learning. *The Journal of Machine Learning Research*, 17(1):1613–1655, 2016.

[12] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[13] Pierre Gaillard, Sébastien Gerchinovitz, Malo Huard, and Gilles Stoltz. Uniform regret bounds over $\mathbb{R}^d$ for the sequential linear regression problem with the square loss. *arXiv preprint arXiv:1805.11386*, 2018.

[14] Fedor Zhdanov and Yuri Kalnishkan. An identity for kernel ridge regression. In *International Conference on Algorithmic Learning Theory*, pages 405–419. Springer, 2010.

[15] Yuchen Zhang, John Duchi, and Martin Wainwright. Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *The Journal of Machine Learning Research*, 16(1):3299–3340, 2015.

[16] Jason Altschuler, Francis Bach, Alessandro Rudi, and Jonathan Weed. Massively scalable sinkhorn distances via the nystr\" om method. *arXiv preprint arXiv:1812.05189*, 2018.

[17] Rajendra Bhatia. *Matrix analysis*, volume 169. Springer Science & Business Media, 2013.

[18] Alessandro Rudi, Raffaello Camoriano, and Lorenzo Rosasco. Less is more: Nyström computational regularization. In *Advances in Neural Information Processing Systems*, pages 1657–1665, 2015.

[19] Francis Bach. On the equivalence between kernel quadrature rules and random feature expansions. *Journal of Machine Learning Research*, 18(21):1–38, 2017.

[20] Alessandro Rudi and Lorenzo Rosasco. Generalization properties of learning with random features. In *Advances in Neural Information Processing Systems*, pages 3215–3225, 2017.

[21] Michał Dereziński, Manfred K Warmuth, and Daniel Hsu. Correcting the bias in least squares regression with volume-rescaled sampling. *arXiv preprint arXiv:1810.02453*, 2018.

[22] Ingo Steinwart, Don Hush, and Clint Scovel. An explicit description of the reproducing kernel hilbert spaces of gaussian rbf kernels. *IEEE Transactions on Information Theory*, 52(10):4635–4643, 2006.

[23] Andrew Cotter, Joseph Keshet, and Nathan Srebro. Explicit approximations of the gaussian kernel. *arXiv preprint arXiv:1109.4603*, 2011.

[24] Alessandro Rudi, Luigi Carratino, and Lorenzo Rosasco. Falkon: An optimal large scale kernel method. In *Advances in Neural Information Processing Systems*, pages 3888–3898, 2017.

[25] Frank WJ Olver, Daniel W Lozier, Ronald F Boisvert, and Charles W Clark. *NIST handbook of mathematical functions*. Cambridge University Press, 2010.

6 Supplementary material

A Proof of Proposition 2.1

The proof follows the classical analysis of online Ridge regression (see for instance the proof of Theorem 2 of [13]). Let us first define some notation that will ease the proof. For all $t \geq 1$, we denote $Y_t = (y_1, \dots, y_t)^\top \in \mathbb{R}^t$ and $\widehat{Y}_t = (\widehat{y}_1, \dots, \widehat{y}_t)^\top \in \mathbb{R}^t$. We also define the operator $S_t : \mathcal{H} \to \mathbb{R}^t$ such that $[S_t f]_j = f(x^{(j)}) = \left\langle f, \phi(x^{(j)}) \right\rangle_{\mathcal{H}}$ for any $f \in \mathcal{H}$ and we define $C_t = \sum_{s=1}^t \phi(x_s) \otimes \phi(x_s)$, $A_t = C_t + \lambda I$. Finally, we denote $L_t(f) := \left\|Y_t - S_t f\right\|^2 + \lambda \|f\|_{\mathcal{H}}^2$ for $f \in \mathcal{H}$.

Under this notation, we successively prove the following two inequalities

$$\begin{aligned} R_n(f) &\leq \lambda \|f\|_{\mathcal{H}}^2 + \sum_{t=1}^n y_t^2 \left\langle A_t^{-1} \phi(x_t), \phi(x_t) \right\rangle_{\mathcal{H}} \\ &\leq \lambda \left\|f\right\|_{\mathcal{H}}^2 + B^2 \sum_{k=1}^{\infty} \log\left(1 + \frac{\lambda_k(C_n)}{\lambda}\right). \end{aligned}$$

Let us start with the first inequality. For $t \geq 1$ we define

$$\widehat{g}_t := \operatorname*{argmin}_{f \in \mathcal{H}} \left\{ L_{t-1}(f) \right\} = A_{t-1}^{-1} S_{t-1}^* Y_{t-1}$$

and we remark that

$$\widehat{f}_t = A_t^{-1} S_{t-1}^* Y_{t-1}$$

where S_t^* is the conjugate transpose of S_t. Let $f \in \mathcal{H}$. By definition of $\widehat{g}_{n+1}$ which minimizes $L_n(f)$ over $f \in \mathcal{H}$,

$$L_n(\widehat{g}_{n+1}) \leq L_n(f)$$

so that

$$\left\|Y_n - S_n \widehat{g}_{n+1}\right\|^2 - \left\|Y_n - S_n f\right\|^2 \leq \lambda \|f\|_{\mathcal{H}}^2 .$$

and

$$\begin{aligned} R_n(f) &= \left\|Y_n - \widehat{Y}_n\right\|^2 - \left\|Y_n - S_n f\right\|^2 \\ &\leq \left\|Y_n - \widehat{Y}_n\right\|^2 - \left\|Y_n - S_n \widehat{g}_{n+1}\right\|^2 + \lambda \|f\|_{\mathcal{H}}^2 \\ &\leq \lambda \|f\|_{\mathcal{H}}^2 + \sum_{t=1}^n \left((y_t - \widehat{y}_t)^2 + L_{t-1}(\widehat{g}_t) - L_t(\widehat{g}_{t+1}) \right). \end{aligned} \tag{13}$$

Now, we need to analyze the terms inside the sum. First we remark that by expanding the square norm

$$\begin{aligned} L_t(f) &= \|Y_t\|^2 - 2 Y_t^\top S_t f + \left\|S_t f\right\|^2 + \lambda \|f\|_{\mathcal{H}} \\ &= \|Y_t\|^2 - 2 Y_t^\top S_t f + \langle f, A_t f \rangle_{\mathcal{H}} \end{aligned} \tag{14}$$

where we used

$$\begin{aligned} \left\|S_t f\right\|^2 &= \sum_{t=1}^n f(x_t)^2 \\ &= \sum_{t=1}^n \langle f, \phi(x_t) \rangle \\ &= \sum_{t=1}^n \langle f, \phi(x_t) \otimes \phi(x_t) f \rangle \\ &= \langle f, C_t f \rangle . \end{aligned}$$

Therefore, substituting $\widehat{g}_{t+1}$ into (14) we get

$$\begin{aligned} L_t(\widehat{g}_{t+1}) &= \|Y_t\|^2 - 2 \underbrace{Y_t^\top S_t A_t^{-1}}_{\widehat{g}_{t+1}^\top} A_t \widehat{g}_{t+1} + \langle \widehat{g}_{t+1}, A_t \widehat{g}_{t+1} \rangle_{\mathcal{H}} \\ &= \|Y_t\|^2 - \langle \widehat{g}_{t+1}, A_t \widehat{g}_{t+1} \rangle_{\mathcal{H}} \end{aligned}$$

Thus,

$$L_{t-1}(\widehat{g}_t) - L_t(\widehat{g}_{t+1}) \\ = -y_t^2 + \langle \widehat{g}_{t+1}, A_t \widehat{g}_{t+1} \rangle_{\mathcal{H}} - \langle \widehat{g}_t, A_{t-1} \widehat{g}_t \rangle_{\mathcal{H}} \,. \quad (15)$$

Furthermore,

$$\begin{aligned} A_t(\widehat{g}_{t+1} - \widehat{f}_t) &= A_t\big(A_t^{-1} S_t^* Y_t - A_t^{-1} S_{t-1}^* Y_{t-1}\big) \\ &= S_t^* Y_t - S_{t-1}^* Y_{t-1} = y_t \phi(x_t)\,, \end{aligned} \quad (16)$$

where the last inequality is because

$$\begin{aligned} \langle S_t^* Y_t, f \rangle_{\mathcal{H}} &= (S_t f)^\top Y_t \\ &= \sum_{s=1}^{t} f(x_s) y_s \\ &= \left\langle \sum_{s=1}^{t} y_s \phi(x_s), f \right\rangle_{\mathcal{H}} . \end{aligned}$$

Then, using, $\widehat{y}_t = \widehat{f}_t(x_t) = \left\langle \widehat{f}_t, \phi(x_t) \right\rangle$, we get

$$\begin{aligned} &(y_t - \widehat{y}_t)^2 = y_t^2 - 2 y_t \widehat{y}_t + \widehat{y}_t^2 \\ &= y_t^2 - 2 \left\langle \widehat{f}_t, y_t \phi(x_t) \right\rangle_{\mathcal{H}} + \left\langle \widehat{f}_t, \phi(x_t) \otimes \phi(x_t) \widehat{f}_t \right\rangle_{\mathcal{H}} \\ &= y_t^2 - 2 \left\langle \widehat{f}_t, A_t(\widehat{g}_{t+1} - \widehat{f}_t) \right\rangle_{\mathcal{H}} + \left\langle \widehat{f}_t, (A_t - A_{t-1}) \widehat{f}_t \right\rangle_{\mathcal{H}} \end{aligned} \quad (17)$$

where in the last inequality we used $A_t - A_{t-1} = \phi(x_t) \otimes \phi(x_t)$. Putting Equations (15) and (17) together, we get

$$\begin{aligned} &(y_t - \widehat{y}_t)^2 + L_{t-1}(\widehat{g}_t) - L_t(\widehat{g}_{t+1}) \\ &= \langle \widehat{g}_{t+1}, A_t \widehat{g}_{t+1} \rangle_{\mathcal{H}} - \langle \widehat{g}_t, A_{t-1} \widehat{g}_t \rangle_{\mathcal{H}} \\ &\qquad - 2 \left\langle \widehat{f}_t, A_t(\widehat{g}_{t+1} - \widehat{f}_t) \right\rangle + \left\langle \widehat{f}_t, (A_t - A_{t-1}) \widehat{f}_t \right\rangle_{\mathcal{H}} \\ &= \Big(\langle \widehat{g}_{t+1}, A_t \widehat{g}_{t+1} \rangle_{\mathcal{H}} - 2 \left\langle \widehat{f}_t, A_t \widehat{g}_{t+1} \right\rangle_{\mathcal{H}} + \left\langle \widehat{f}_t, A_t \widehat{f}_t \right\rangle_{\mathcal{H}} \Big) \\ &\quad - \Big(\langle \widehat{g}_t, A_{t-1} \widehat{g}_t \rangle_{\mathcal{H}} - 2 \Big\langle \widehat{f}_t, \underbrace{A_t \widehat{f}_t}_{A_{t-1}\widehat{g}_t} \Big\rangle_{\mathcal{H}} + \left\langle \widehat{f}_t, A_{t-1} \widehat{f}_t \right\rangle_{\mathcal{H}} \Big) . \end{aligned}$$

Then, using the equality $A_t \widehat{f}_t = S_{t-1}^* Y_{t-1} = A_{t-1} \widehat{g}_t$, we recognize the difference between two quadratic forms

$$\begin{aligned} &(y_t - \widehat{y}_t)^2 + L_{t-1}(\widehat{g}_t) - L_t(\widehat{g}_{t+1}) \quad (18) \\ &= \left\langle \widehat{g}_{t+1} - \widehat{f}_t, A_t(\widehat{g}_{t+1} - \widehat{f}_t) \right\rangle_{\mathcal{H}} - \left\langle \widehat{f}_t - \widehat{g}_t, A_{t-1}(\widehat{f}_t - \widehat{g}_t) \right\rangle_{\mathcal{H}} \\ &\leq \left\langle \widehat{g}_{t+1} - \widehat{f}_t, A_t(\widehat{g}_{t+1} - \widehat{f}_t) \right\rangle_{\mathcal{H}} \\ &\overset{(16)}{=} y_t^2 \left\langle A_t^{-1} \phi(x_t), \phi(x_t) \right\rangle_{\mathcal{H}} . \quad (19) \end{aligned}$$

Substituting into (13) concludes the proof: for any $f \in \mathcal{H}$

$$R_n(f) \leq \lambda \|f\|_{\mathcal{H}}^2 + \sum_{t=1}^{n} y_t^2 \left\langle A_t^{-1} \phi(x_t), \phi(x_t) \right\rangle_{\mathcal{H}}$$

where we recall that $A_t = \sum_{s=1}^{t} \phi(x_s) \otimes \phi(x_s) + \lambda I$. This proves the first inequality of the theorem.

Let us now prove the second inequality. Upper-bounding $|y_t| \leq B$, it suffices to show

$$\sum_{t=1}^{n} \left\langle A_t^{-1} \phi(x_t), \phi(x_t) \right\rangle_{\mathcal{H}} \leq \sum_{k=1}^{\infty} \log \left(1 + \frac{\lambda_k(C_n)}{\lambda} \right) .$$

Remarking that $A_t = A_{t-1} + \phi(x_t) \otimes \phi(x_t)$ and applying Lemma A.1 stated below we have

$$\left\langle A_t^{-1}\phi(x_t), \phi(x_t)\right\rangle_{\mathcal{H}} = 1 - \frac{\det(A_{t-1}/\lambda)}{\det(A_t/\lambda)}\,.$$

It is worth pointing out that $\det(A_t/\lambda)$ is well defined since $A_t = I + C_t$ with $C_t = \sum_{s=1}^{t} \phi(x_s) \otimes \phi(x_s)$ at most of rank $t \geq 0$. Then we use $1 - u \leq \log(1/u)$ for $u > 0$ which yields

$$\left\langle A_t^{-1}\phi(x_t), \phi(x_t)\right\rangle_{\mathcal{H}} \leq \log \frac{\det(A_t/\lambda)}{\det(A_{t-1}/\lambda)}\,.$$

Summing over $t = 1, \ldots, n$, using $A_0 = \lambda I$ and $A_n = \lambda I + C_n$ we get

$$\begin{aligned}\sum_{t=1}^{n} \left\langle A_t^{-1}\phi(x_t), \phi(x_t)\right\rangle_{\mathcal{H}} &\leq \log\left(\det\left(I + \frac{C_n}{\lambda}\right)\right) \\ &= \sum_{k=1}^{\infty} \log\left(1 + \frac{\lambda_k(C_n)}{\lambda}\right)\,,\end{aligned}$$

which concludes the proof.

The following Lemma is a standard result of online matrix theory (see Lemma 11.11 of [12]).

Lemma A.1. *Let $V : \mathcal{H} \to \mathcal{H}$ be a linear operator. Let $u \in \mathcal{H}$ and let $U = V - u \otimes u$. Then,*

$$\left\langle V^{-1}u, u\right\rangle_{\mathcal{H}} = 1 - \frac{\det(U)}{\det(V)}\,.$$

B Proof of proposition 2.2

Using that for $x > 0$

$$\log(1 + x) \leq \frac{x}{x+1}(1 + \log(1 + x))\,,$$

and denoting by $a(\lambda)$ the quantity $a(s, \lambda) := 1 + \log(1 + s/\lambda)$, we get for any $n \geq 1$

$$\log\Big(1 + \frac{\lambda_k(K_{nn})}{\lambda}\Big) \leq \frac{\lambda_k(K_{nn})}{\lambda + \lambda_k(K_{nn})} a(\lambda_k(K_{nn}), \lambda).$$

Therefore, summing over $k \geq 1$ and denoting by λ_1 the largest eigenvalue of K_{nn}

$$\begin{aligned}\sum_{k=1}^{n} \log\left(1 + \frac{\lambda_k(K_{nn})}{\lambda}\right) &\leq a(\lambda_1, \lambda) \sum_{k=1}^{n} \frac{\lambda_k(K_{nn})}{\lambda + \lambda_k(K_{nn})} \\ &= a(\lambda_1, \lambda)\,\mathrm{Tr}\left(K_{nn}(K_{nn} + \lambda I)^{-1}\right) \\ &= a(\lambda_1, \lambda) d_{\text{eff}}(\lambda)\,,\end{aligned}$$

where the last equality is from the definition of $d_{\text{eff}}(\lambda)$. Combining with Proposition 2.1, substituting a and upper-bounding

$$\lambda_1(K_{nn}) \leq \mathrm{Tr}(K_{nn}) = \sum_{t=1}^{n} \|\phi(x_t)\|_{\mathcal{H}}^2 \leq n\kappa^2$$

concludes the proof.

C Additional Proofs

C.1 Proof of Lemma 4.2

Recall the following characterization of scalar product for translation invariant kernels (i.e. $k(x, x') = v(x - x')$ for a $v : \mathbb{R}^d \to \mathbb{R}$) [see 2]

$$\langle f, g\rangle_{\mathcal{H}} = \int \frac{\mathcal{F}[f](\omega)\mathcal{F}[g](\omega)}{\mathcal{F}[v](\omega)},$$

where $\mathcal{F}[f]$ is the unitary Fourier transform of f. Let start from the one dimensional case and denote by $\mathcal{H}_0$ the Gaussian RKHS on $\mathbb{R}$. First note that when $d = 1$, we have $g_k = \psi_k$. Now, the Fourier transform of ψ_k is $\mathcal{F}[\psi_k](\omega) = \frac{1}{\sqrt{k!}} H_k(x/\sigma^2) e^{-\omega^2/(2\sigma^2)}$, for any $k \in \mathbb{N}_0^d$, where $H_k(x)$ is the k-th Hermite polynomial [see 25, Eq. 18.17.35 pag. 457], and $\mathcal{F}[v] = e^{-\omega^2/2}$, then, by the fact that Hermite are orthogonal polynomial with respect to $e^{-\omega^2/2}$ forming a complete basis, we have

$$\langle \psi_k, \psi_{k'} \rangle_{\mathcal{H}_0} = \frac{1}{k!} \int H_k(\omega) H_{k'}(\omega) e^{-\omega^2/2} d\omega = \mathbb{1}_{k=k'}.$$

The multidimensional case is straightforward since Gaussian is a product kernel, i.e. $k(x, x') = \prod_{i=1}^d k(x^{(i)}, x^{(i)})$ and $\mathcal{H} = \otimes_{i=1}^d \mathcal{H}_0$, so $\left\langle \otimes_{i=1}^d f_i, \otimes_{i=1}^d g_i \right\rangle_{\mathcal{H}} = \prod_{i=1}^d \langle f_i, g_i \rangle_{\mathcal{H}_0}$ [see 1]. Now, since $g_k = \otimes_{i=1}^d \psi_{k_i}$, we have $\langle g_k, g_{k'} \rangle_{\mathcal{H}} = \prod_{i=1}^d \left\langle \psi_{k_i}, \psi_{k_i'} \right\rangle_{\mathcal{H}_0} = \mathbb{1}_{k=k'}$.

C.2 Proof of Lemma 4.3

First, for $j \in \mathbb{N}_0$ define

$$Q_j(x, x') := e^{-\frac{\|x\|^2}{2\sigma^2} - \frac{\|x'\|^2}{2\sigma^2}} \frac{(x^\top x'/\sigma^2)^j}{j!}.$$

First note that, by multinomial expansion of $(x^\top x')^j$,

$$\begin{aligned} Q_j(x, x') &= \frac{e^{-\frac{\|x\|^2 + \|x'\|^2}{2\sigma^2}}}{\sigma^{2j} j!} \sum_{|t|=j} \binom{j}{t_1 \dots t_d} \prod_{i=1}^d (x^{(i)})^{t_i} (x'^{(i)})^{t_i} \\ &= \sum_{|t|=j} g_t(x) g_t(x'). \end{aligned}$$

Now note that, by Taylor expansion of $e^{x^\top x'/\sigma^2}$ we have

$$\begin{aligned} k(x, x') = \sum_{j=0}^\infty Q_j(x, x') &= \sum_{j=0}^\infty \sum_{|t|=j} g_t(x) g_t(x') \\ &= \sum_{k \in \mathbb{N}_0^d} g_k(x) g_k(x'). \end{aligned}$$

Finally, with ϕ defined as above, and the fact that g_k forms an orthonormal basis for $\mathcal{H}$, leads to

$$\langle \phi(x), \phi'(x) \rangle = \sum_{k \in \mathbb{N}_0^d} g_k(x) g_k(x') = k(x, x').$$

C.3 Proof of Lemma 4.4

Here we use the same notation of the proof of Lemma 4.3. Since by Taylor expansion, we have that $k(x, x') = \sum_{j=0}^\infty Q_j(x, x')$, by mean value theorem for the function $f(s) = e^{s/\sigma^2}$, we have that there exists $c \in [0, x^\top x']$ such that

$$\begin{aligned} |k(x, x') - \sum_{j=0}^M Q_j(x, x')| &= e^{-\frac{\|x\|^2 + \|x'\|^2}{2\sigma^2}} \frac{c^{M+1}}{(M+1)!} \frac{d^{M+1} e^{\frac{s}{\sigma^2}}}{ds^{M+1}}|_{s=c} \\ &\leq \frac{(|x^\top x'|/\sigma^2)^{M+1}}{(M+1)!} \\ &\leq \frac{(R/\sigma)^{2M+2}}{(M+1)!} \end{aligned}$$

where the last step is obtained assuming $\|x\|, \|x'\| \leq R$. Finally note that, by definition of G_M,

$$\sum_{g \in G_M} g(x) g(x') = \sum_{|k| \leq M} g_k(x) g_k(x') = \sum_{j=0}^M Q_j(x, x').$$

D Python code for $\tilde{\phi}$-AWV with Taylor features

```
import numpy as np
from math import factorial,sqrt

class PhiAWV:
    def __init__(self, d, sigma=1.0, lbd=1.0, M=2):
        self.b = None
        self.A_inv = None
        self.M = M
        self.lbd = lbd
        self.sigma = sigma

    def taylor_phi(self, x):
        res = np.array([1.])
        mm = np.array([1.])
        for k in range(1,self.M+1):
            mm = (np.outer(mm, x)).flatten()
            q = self.sigma**k*sqrt(factorial(k))
            res = np.concatenate((res,mm/q))
        c  = 2*self.sigma**2
        res *= np.exp(-np.dot(x,x)/c)
        return np.array(res)

    def predict(self, x):
        z = self.taylor_phi(x)
        if self.b is None:
            r = len(z)
            self.b = np.zeros(r)
            self.A_inv = (1/self.lbd)*np.eye(r)

        v = np.dot(self.A_inv, z)
        v /= sqrt(1 + np.dot(z, v))
        self.A_inv -= np.outer(v, v)
        w_hat = np.dot(self.A_inv, self.b)
        self.z = z
        return np.dot(w_hat, z)

    def update(self, y):
        self.b += y*self.z
```