Online Learning of Noisy Data

Nicoló Cesa-Bianchi, Shai Shalev-Shwartz, and Ohad Shamir

***Abstract*—We study online learning of linear and kernel-based predictors, when individual examples are corrupted by random noise, and both examples and noise type can be chosen adversarially and change over time. We begin with the setting where some auxiliary information on the noise distribution is provided, and we wish to learn predictors with respect to the squared loss. Depending on the auxiliary information, we show how one can learn linear and kernel-based predictors, using just 1 or 2 noisy copies of each example. We then turn to discuss a general setting where virtually nothing is known about the noise distribution, and one wishes to learn with respect to general losses and using linear and kernel-based predictors. We show how this can be achieved using a random, essentially constant number of noisy copies of each example. Allowing multiple copies cannot be avoided: Indeed, we show that the setting becomes impossible when only one noisy copy of each instance can be accessed. To obtain our results we introduce several novel techniques, some of which might be of independent interest.**

I. Introduction

In many machine learning applications training data are typically collected by measuring certain physical quantities. Examples include bioinformatics, medical tests, robotics, and remote sensing. These measurements have errors that may be due to several reasons: low-cost sensors, communication and power constraints, or intrinsic physical limitations. In all such cases, the learner trains on a distorted version of the actual "target" data, which is where the learner's predictive ability is eventually evaluated. A concrete scenario matching this setting is an automated diagnosis system based on computed-tomography (CT) scans. In order to build a large dataset for training the system, we might use low-dose CT scans: although the images are noisier than those obtained through a standard-radiation CT scan, lower exposure to radiation will persuade more people to get a scan. On the other hand, at test time, a patient suspected of having a serious disease will agree to undergo a standard scan.

Manuscript received September 02, 2010; revised December 29, 2010; accepted July 08, 2011. Date of publication September 08, 2011; date of current version December 07, 2011. The material in this paper was presented at the COLT 2010 conference. This work was supported in part by the Israeli Science Foundation under Grant 590-10 and in part by the PASCAL2 Network of Excellence under EC Grant 216886.

N. Cesa-Bianchi is with the Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano, Milano 20135, Italy (e-mail: nicolo.cesa-bianchi@unimi.it).

S. Shalev-Shwartz is with the Computer Science and Engineering Department, The Hebrew University, Jerusalem 91904, Israel (e-mail: shais@cs.huji.ac.il).

O. Shamir is with Microsoft Research New England, Cambridge, MA 02142 USA (e-mail: ohadsh@microsoft.com).

Communicated by T. Weissman, Associate Editor for Shannon Theory.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2011.2164053

In this work, we investigate the extent to which a learning algorithm for training linear and kernel-based predictors can achieve a good performance when the features and/or target values of the training data are corrupted by noise. Note that, although in the noise-free case learning with kernels is generally not harder than linear learning, in the noisy case the situation is different due to the potentially complex interaction between the kernel and the noise distribution.

We prove upper and lower bounds on the learner's cumulative loss in the framework of online learning, where examples are generated by an arbitrary and possibly adversarial source. We model the measurement error via a random zero-mean perturbation which affects each example observed by the learner. The noise distribution may also be chosen adversarially, and change for each example.

In the first part of the paper, we discuss the consequences of being given some auxiliary information on the noise distribution. This is relevant in many applications, where the noise can be explicitly modeled, or even intentionally introduced. For example, in order to comply with privacy issues certain datasets can be published only after being "sanitized", which corresponds to perturbing each data item with enough Gaussian noise—see, e.g., [1]. In this work we show how to learn from such sanitized data.

Focusing on the squared loss, we discuss three different settings, reflecting different levels of knowledge about the noise distribution: known variance bound, known covariance structure, and Gaussian noise with known covariance matrix. Our results for these three settings can be summarized as follows:

Known Variance Bound: Linear predictors can be learnt with two independent noisy copies of each instance $\mathbf{x}_t$ (that is, two independent realizations of the example corrupted by random noise), and one noisy copy of each target value y_t.

Known covariance structure: Linear predictors can be learnt with only one noisy copy of $\mathbf{x}_t$ and y_t.

Gaussian distribution with known covariance matrix: Kernel-based (and therefore linear) predictors can be learnt using two independent noisy copies of each $\mathbf{x}_t$, and one noisy copy of y_t. (Although we focus on Gaussian kernels, we show how this result can be extended, in a certain sense, to general radial kernels.)

Thus, the positive learning results get stronger the more we can assume about the noise distribution. To obtain our results, we use online gradient descent techniques of increasing sophistication. The first two settings are based on constructing unbiased gradient estimates, while the third setting involves a novel technique based on constructing *surrogate* Hilbert spaces. The surrogate space is built such that gradient descent on the noisy examples in that space corresponds, in an appropriately defined

manner, to gradient descent on the noise-free examples in the original space.

In the second part of the paper we consider linear and kernel-based learning with respect to general loss functions (and not just the squared loss as before). Our positive results are quite general: by assuming just a variance bound on the noise we show how it is possible to learn functions in any dot-product (e.g., polynomial) or radial kernel Hilbert space, under any analytic convex loss function. Our techniques, which are readily extendable to other kernel types as well, require querying a random number of independently perturbed copies of each example. We show that this number is bounded by a constant with high probability. This is in sharp contrast with standard averaging techniques, which attempts to directly estimate the noisy instance, as these require a sample whose size depends on the scale of the problem. Moreover, the number of queries is controlled by the user, and can be reduced at the cost of receiving more examples overall.

Finally, we formally show in this setting that learning is impossible when only one perturbed copy of each example can be accessed. This holds even without kernels, and for any reasonable loss function.

A. Related Work

In the machine learning literature, the problem of learning from noisy examples, and, in particular, from noisy training instances, has traditionally received a lot of attention—see, for example, the recent survey [2]. On the other hand, there are comparably few theoretically-principled studies on this topic. Two of them focus on models quite different from the one studied here: random attribute noise in PAC boolean learning [3], [4], and malicious noise [5], [6]. In the first case learning is restricted to classes of boolean functions, and the noise must be independent across each boolean coordinate. In the second case an adversary is allowed to perturb a small fraction of the training examples in an arbitrary way, making learning impossible in a strong information-theoretic sense unless this perturbed fraction is very small (of the order of the desired accuracy for the predictor).

The previous work perhaps closest to the one presented here is [7], where binary classification mistake bounds are proven for the online Winnow algorithm in the presence of attribute errors. Similarly to our setting, the sequence of instances observed by the learner is chosen by an adversary. However, in [7] the noise process is deterministic and also controlled by the adversary, who may change the value of each attribute in an arbitrary way. The final mistake bound, which only applies when the noiseless data sequence is linearly separable without kernels, depends on the sum of all adversarial perturbations.

II. Framework and Notation

We consider a setting where the goal is to predict values $y \in \mathbb{R}$ based on instances $\mathbf{x} \in \mathbb{R}^d$. We focus on predictors which are either linear—i.e., of the form $\mathbf{x} \mapsto \langle \mathbf{w}, \mathbf{x} \rangle$ for some vector $\mathbf{w}$, or kernel-based—i.e., of the form $\mathbf{x} \mapsto \langle \mathbf{w}, \Psi(\mathbf{x}) \rangle$ where Ψ is a feature mapping into some reproducing kernel Hilbert space (RKHS)[1] $\mathcal{H}$. In the latter case, we assume there exists a kernel function $k : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$ that efficiently implements inner products in that space, i.e., $k(\mathbf{x}, \mathbf{x}') = \langle \Psi(\mathbf{x}), \Psi(\mathbf{x}') \rangle$ Note that in fact, linear predictors are just a special case of kernel-based predictors: we can take $\Psi(\cdot)$ to be the identity mapping and let $k(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle$. Other choices of the kernel allows us to learn nonlinear predictors over $\mathbb{R}^d$, while retaining much of the computational convenience and theoretical guarantees of learning linear predictors (see [8] for more details). In the remainder of this section, our discussion will use the notation of kernel-based predictors, but everything will apply to linear predictors as well.

The standard online learning protocol is defined as the following repeated game between the learner and an adversary: at each round $t = 1, 2, \ldots$, the learner picks a hypothesis $\mathbf{w}_t \in \mathcal{H}$. The adversary then picks an example $(\mathbf{x}_t, y_t)$, composed of a feature vector $\mathbf{x}_t$ and target value y_t, and reveals it to the learner. The loss suffered by the learner is $\ell(\langle \mathbf{w}_t, \Psi(\mathbf{x}_t) \rangle, y_t)$, where ℓ is a known and fixed loss function. The goal of the learner is to minimize *regret* with respect to a fixed convex set of hypotheses $\mathcal{W} \subseteq \mathcal{H}$, defined as

$$\sum_{t=1}^{T} \ell(\langle \mathbf{w}_t, \Psi(\mathbf{x}_t) \rangle, y_t) - \min_{\mathbf{w} \in \mathcal{W}} \sum_{t=1}^{T} \ell(\langle \mathbf{w}, \Psi(\mathbf{x}_t) \rangle, y_t).$$

Typically, we wish to find a strategy for the learner, such that no matter what is the adversary's strategy of choosing the sequence of examples, the expression above is sublinear in T. In this paper, we will focus for simplicity on a finite-horizon setting, where the number of online rounds T is fixed and known to the learner. All our results can easily be modified to deal with the infinite horizon setting, where the learner needs to achieve sublinear regret for all T simultaneously.

We now make the following modification, which limits the information available to the learner: In each round, the adversary also selects a vector-valued random variable $\mathbf{n}_t^x$ and a random variable n_t^y. Instead of receiving $(\mathbf{x}_t, y_t)$, the learner is given access to an *oracle* A_t, which can return independent realizations of $\tilde{\mathbf{x}}_t = \mathbf{x}_t + \mathbf{n}_t^x$ and $\tilde{y}_t = y_t + n_t^y$. In other words, the adversary forces the learner to see only a noisy version of the data, where the noise distribution can be changed by the adversary after each round. We will assume throughout the paper that $\mathbf{n}_t^x$ and n_t^y are zero-mean, independent, and there is some fixed known upper bound on $\mathbb{E}\left[\|\tilde{\mathbf{x}}_t\|^2\right]$ and $\mathbb{E}[\tilde{y}_t^2]$ for all t. Note that if $\mathbf{n}_t^x$ or n_t^y are not zero-mean, but the mean is known to the learner, we can always deduct those means from $\tilde{\mathbf{x}}_t$ and $\tilde{y}_t$, thus reducing to the zero-mean setting. The assumption that $\mathbf{n}_t^x$ is independent of n_t^y can be relaxed to uncorrelation or even disposed of entirely in some of the discussed settings, at the cost of some added technical complexity in the algorithms and proofs.

The learner may call the oracle A_t more than once. In fact, as we discuss later on, being able to call A_t more than once can be necessary for the learner to have any hope to succeed, when nothing more is known about the noise distribution. On the other hand, if the learner calls A_t an unlimited number of times, $\mathbf{x}_t$, y_t can be reconstructed arbitrarily well by averaging, and we

[1]Recall that a Hilbert space is a natural generalization of Euclidean space to possibly infinite dimensions. More formally, it is an inner product space which is complete with respect to the norm induced by the inner product.

are back to the standard learning setting. In this paper we focus on learning algorithms that call A_t only a small, essentially constant number of times, which depends only on our choice of loss function and kernel (rather than the horizon T, the norm of $\mathbf{x}_t$, or the variance of $\mathbf{n}_t^x$, n_t^y, which happens with naïve averaging techniques).

In this setting, we wish to minimize the regret in hindsight for any sequence of unperturbed data, and in expectation with respect to the noise introduced by the oracle, namely

$$\mathbb{E}\left[\sum_{t=1}^{T}\ell(\langle\mathbf{w}_t,\Psi(\mathbf{x}_t)\rangle,y_t)\right]-\min_{\mathbf{w}\in\mathcal{W}}\sum_{t=1}^{T}\ell(\langle\mathbf{w},\Psi(\mathbf{x}_t)\rangle,y_t). \tag{1}$$

Note that the stochastic quantities in the above expression are just $\mathbf{w}_1,\mathbf{w}_2,\ldots$, where each $\mathbf{w}_t$ is a measurable function of the previous perturbed examples $(\tilde{\mathbf{x}}_s,\tilde{y}_s)$ for $s=1,\ldots,t-1$. When the noise distribution is bounded or has sub-Gaussian tails, our techniques can also be used to bound the actual regret with high probability, by relying on Azuma's inequality or variants thereof (see for example [9]). However, for simplicity here we focus on the expected regret in (1).

The regret form in (1) is relevant where we actually wish to learn from data, without the noise causing a hindrance. In particular, consider the batch setting, where the examples $\{(\mathbf{x}_t,y_t)\}_{t=1}^{T}$ are actually sampled i.i.d. from some unknown distribution, and we wish to find a predictor which minimizes the expected loss with respect to new examples $(\mathbf{x},y)$. Using standard online-to-batch conversion techniques [9], if we can find an online algorithm with a sublinear bound on (1), then it is possible to construct learning algorithms for the batch setting which are robust to noise. That is, algorithms generating a predictor $\mathbf{w}$ with close to minimal expected loss $\mathbb{E}[\ell(\langle\mathbf{w},\mathbf{x}\rangle,y)]$ among all $\mathbf{w}\in\mathcal{W}$, despite getting only noisy access to the data. In Appendix A, we briefly discuss alternative regret measures.

In the first part of our paper, we assume that the loss function $\ell(\langle\mathbf{w},\Psi(\mathbf{x})\rangle,y)$ is the squared loss $(\langle\mathbf{w},\Psi(\mathbf{x})\rangle-y)^2$. In the second part of the paper, we deal with more general loss functions, which are convex in $\mathbf{w}$ and analytic, in the sense that $\ell(a,y)$ for a fixed y can be written as $\sum_{n=0}^{\infty}\gamma_n a^n$, for any a in its domain. This assumption holds for instance for the squared loss $\ell(a,y)=(a-y)^2$, the exponential loss $\ell(a,y)=\exp(-ya)$, and "smoothed" versions of loss functions such as the absolute loss $\ell(a,y)=|a-y|$ and the hinge loss $\ell(a,y)=\max\{1-ya,0\}$ (we discuss examples in more details in Section V-B. This assumption can be relaxed under certain conditions, and this is further discussed in Section III-C.

Turning to the issue of kernels, we note that the general presentation of our approach is somewhat hampered by the fact that it needs to be tailored to the kernel we use. In this paper, we focus on two important families of kernels:

Dot Product Kernels: the kernel $k(\mathbf{x},\mathbf{x}')$ can be written as a function of $\langle\mathbf{x},\mathbf{x}'\rangle$. Examples of such kernels $k(\mathbf{x},\mathbf{x}')$ are linear kernels $\langle\mathbf{x},\mathbf{x}'\rangle$; homogeneous polynomial kernels $(\langle\mathbf{x},\mathbf{x}'\rangle)^n$; inhomogeneous polynomial kernels $(1+\langle\mathbf{x},\mathbf{x}'\rangle)^n$; exponential kernels $e^{\langle\mathbf{x},\mathbf{x}'\rangle}$; binomial kernels $(1+\langle\mathbf{x},\mathbf{x}'\rangle)^{-\alpha}$, and more (see for instance [8], [10]).

Radial Kernels: $k(\mathbf{x},\mathbf{x}')$ can be written as a function of $\|\mathbf{x}-\mathbf{x}'\|$. A central and widely used member of this family is the Gaussian kernel, $\exp(-\|\mathbf{x}-\mathbf{x}'\|^2/s^2)$ for some $s^2>0$.

We emphasize that many of our techniques are extendable to other kernel types as well.

III. Techniques

We begin by presenting a high-level and mostly informal overview of the techniques we use to overcome the noise present in the data. The first technique we discuss ("stochastic" online gradient descent) is folklore, and forms a basis for our learning algorithms. The rest of the techniques are designed to overcome the noise in the data, and to the best of our knowledge, are novel to the machine learning community. Hence, they might be of independent interest and applicable to other learning problems with partial information on the examples.

A. "Stochastic" Online Gradient Descent

There exists a well-developed theory, as well as efficient algorithms, for dealing with the standard online learning setting, where the example $(\mathbf{x}_t,y_t)$ is revealed after each round, and for general convex loss functions. One of the simplest and most well known ones is the online gradient descent algorithm due to Zinkevich [11]. This algorithm, and its "stochastic" extension, form a basis for our results, and we briefly survey it below.

At the heart of the standard online gradient descent algorithm is the following observation: for any set of vectors $\nabla_1,\ldots,\nabla_T$ in some Hilbert space, suppose we define $\mathbf{w}_1=0$ and $\mathbf{w}_{t+1}=P(\mathbf{w}_t-\eta_t\nabla_t)$, where $P(\cdot)$ is a projection operator on a convex set $\mathcal{W}$, and η_t is a suitably chosen step size. Then for any $\mathbf{u}\in\mathcal{W}$, it holds that

$$\sum_{t=1}^{T}\langle\mathbf{w}_t-\mathbf{u},\nabla_t\rangle=O\left(\sqrt{T}\right) \tag{2}$$

where the $O(\cdot)$ notation hides dependencies on the norm of $\mathbf{u}$ and the norms of ∇_t. In particular, suppose that we let ∇_t be the gradient of $\ell(\langle\mathbf{w}_t,\mathbf{x}_t\rangle,y_t)$ with respect to $\mathbf{w}_t$ (we focus on linear predictors here for simplicity). Then by convexity, the left-hand side (LHS) of (2) is lower bounded by $\sum_{t=1}^{T}\ell(\langle\mathbf{w}_t,\mathbf{x}_t\rangle,y_t)-\sum_{t=1}^{T}\ell(\langle\mathbf{u},\mathbf{x}_t\rangle,y_t)$. Thus, if we are provided with $(\mathbf{x}_t,y_t)$ after each round, we can compute ∇_t, perform the update as above, and get an algorithm with sublinear regret with respect to any predictor $\mathbf{u}$ of bounded norm.

In our setting of noisy data, the algorithm described above is inapplicable, because $(\mathbf{x}_t,y_t)$ is unknown and we cannot compute ∇_t. However, suppose that instead of ∇_t, we pick *random* vectors $\tilde{\nabla}_t$ with bounded variance, such that $\mathbb{E}[\tilde{\nabla}_t|\mathbf{w}_t]=\nabla_t$, and use them to update $\mathbf{w}_t$. It turns out that based on (2), one can still show that

$$\mathbb{E}\left[\sum_{t=1}^{T}\left\langle\mathbf{w}_t-\mathbf{u},\tilde{\nabla}_t\right\rangle\right]=O\left(\sqrt{T}\right). \tag{3}$$

In our setting of noisy data, we cannot compute ∇_t, but suppose we can use the noisy data that we do have, in order to construct a random bounded-variance vector $\tilde{\nabla}_t$,

such that $\mathbb{E}[\tilde{\nabla}_t|\mathbf{w}_t] = \nabla_t$. In that case, the LHS of (3) can be shown to equal $\mathbb{E}\left[\sum_{t=1}^T \langle \mathbf{w}_t - \mathbf{u}, \nabla_t\rangle\right]$. The expectation here is again with respect to the noisy examples (recall that $\mathbf{w}_t$ is a random vector that depends on the noisy examples). Applying the same convexity argument as before, we get an $O(\sqrt{T})$ upper bound on the expected regret $\mathbb{E}\left[\sum_{t=1}^T \ell(\langle \mathbf{w}_t, \mathbf{x}_t\rangle, y_t) - \sum_{t=1}^T \ell(\langle \mathbf{u}, \mathbf{x}_t\rangle, y_t)\right]$. Thus, by doing updates using $\tilde{\nabla}_t$, we get an algorithm with a bound on the regret which scales sublinearly with T.

The idea that one can work with random unbiased estimates of ∇_t is not new, and has been used in previous work, such as online bandit learning (see for instance [12]–[14]). Here, we use this property in a new way, in order to devise algorithms which are robust to noise.

For linear kernels and losses such as the squared loss, constructing such unbiased estimates based on 1 or 2 noisy copies of each example is not too hard. However, when we discuss nonlinear kernels, constructing an unbiased estimate becomes much more tricky: rather than a finite-dimensional vector, ∇_t might exist in a high or infinite dimensional Hilbert space. Even worse, due to the nonlinearity of virtually all feature mappings, the *unbiased* perturbation $\tilde{\mathbf{x}}_t$ of each instance $\mathbf{x}_t$ is mapped to a *biased* and complicated perturbation $\Psi(\tilde{\mathbf{x}}_t)$ of $\Psi(\mathbf{x}_t)$. This leads us to the next technique.

B. "Parallel Worlds" Online Gradient Descent

The technique described here is the central one we use to learn with kernel-based predictors and squared loss, in the case where the noise distribution is fixed and known to be a Gaussian. In the next subsections, we will describe our techniques for dealing with unknown noise distribution and more general loss functions, at the cost of more noisy copies per example.

Unlike the "stochastic" online gradient descent approach discussed in the previous subsection, the approach we discuss here does not rely directly on constructing unbiased estimates of ∇_t. In a nutshell, we construct a *surrogate* RKHS, with a *surrogate* feature mapping $\hat{\Psi}$, such that for any noisy copy $\tilde{\mathbf{x}}_t$ of $\mathbf{x}_t$, and any fixed instance $\mathbf{a}$, it holds that

$$\mathbb{E}\left[\langle \hat{\Psi}(\mathbf{a}), \hat{\Psi}(\tilde{\mathbf{x}}_t)\rangle\right] = \langle \Psi(\mathbf{a}), \Psi(\mathbf{x}_t)\rangle \tag{4}$$

where the expectation is with respect to the noise. Thus, "noisy" inner products in the surrogate RKHS correspond (in expectation) to "noise-free" inner products in the original RKHS. This allows us to use the noisy data in order to construct vectors $\hat{\nabla}_t$ in the surrogate RKHS with the following interesting property: if we apply online gradient descent on $\hat{\nabla}_1, \ldots, \hat{\nabla}_T$ (using kernels), to get predictors $\hat{\mathbf{w}}_1, \ldots, \hat{\mathbf{w}}_T$ in the RKHS of $\hat{\Psi}$, then for any $\hat{\mathbf{u}}$,

$$\mathbb{E}\left[\sum_{t=1}^T \left\langle \hat{\mathbf{w}}_t - \hat{\mathbf{u}}, \hat{\nabla}_t \right\rangle\right] = \mathbb{E}\left[\sum_{t=1}^T \langle \mathbf{w}_t - \mathbf{u}, \nabla_t\rangle\right]$$

where $\mathbf{w}_t$ and $\mathbf{u}$ are the images of $\hat{\mathbf{w}}_t$ and $\hat{\mathbf{u}}$ according to a certain mapping to the RKHS of Ψ, and ∇_t are the gradients with respect to the unperturbed examples $(\mathbf{x}_t, y_t)$. Since we applied online gradient descent in the surrogate RKHS, the LHS is $O(\sqrt{T})$ by (3). Thus, we get that $\mathbb{E}\left[\sum_{t=1}^T \langle \mathbf{w}_t - \mathbf{u}, \nabla_t\rangle\right]$ is $O(\sqrt{T})$, which implies a sublinear regret bound for $\mathbf{w}_1, \ldots, \mathbf{w}_T$. We emphasize that unlike the previous approaches, the expectation of $\hat{\nabla}_t$ is not equal to ∇_t. Indeed, they live in different mathematical spaces!

A technical issue which needs addressing is that the norm of $\hat{\mathbf{u}}$ has to be related to the norm of the actual predictor $\mathbf{u}$ we compare ourselves with. While this cannot be always done, such a relation does hold if $\mathbf{u}$ is reasonably "nice", in a sense which will be formalized later on.

Constructing a surrogate RKHS as in (4) can be done when the original RKHS corresponds to a Gaussian kernel. Nevertheless, we can extend our results, in a certain sense, to more general radial kernels. The basic tool we use is Schoenberg's theorem, which implies that any radial kernel can be written as an integral of Gaussian kernels of different widths. Using this result, we can show that one can still construct a surrogate RKHS, which has the property of (4) with respect to an approximate version of our original radial kernel.

C. Unbiased Estimators for Nonlinear Functions

We now turn to discuss our techniques for dealing with the most general setting: learning kernel-based predictors, with general loss functions, and with only a variance bound known on the noise distribution. At the heart of these techniques lies an apparently little-known method from sequential estimation theory to construct unbiased estimates of nonlinear and possibly complex functions.

Suppose that we are given access to independent copies of a real random variable X, with expectation $\mathbb{E}[X]$, and some real function f, and we wish to construct an unbiased estimate of $f(\mathbb{E}[X])$. If f is a linear function, then this is easy: just sample x from X, and return $f(x)$. By linearity, $\mathbb{E}[f(X)] = f(\mathbb{E}[X])$ and we are done. The problem becomes less trivial when f is a general, nonlinear function, since usually $\mathbb{E}[f(X)] \neq f(\mathbb{E}[X])$. In fact, when X takes finitely many values and f is not a polynomial function, one can prove that no unbiased estimator can exist (see [15], Proposition 8 and its proof). Nevertheless, we show how in many cases one can construct an unbiased estimator of $f(\mathbb{E}[X])$, including cases covered by the impossibility result. There is no contradiction, because we do not construct a "standard" estimator. Usually, an estimator is a function from a given sample to the range of the parameter we wish to estimate. An implicit assumption is that the size of the sample given to it is fixed, and this is also a crucial ingredient in the impossibility result. We circumvent this by constructing an estimator based on a random number of samples.

Here is the key idea: suppose $f : \mathbb{R} \to \mathbb{R}$ is any function continuous on a bounded interval. It is well known that one can construct a sequence of polynomials $(Q_n(\cdot))_{n=1}^\infty$, where $Q_n(\cdot)$ is a polynomial of degree n, which converges uniformly to f on the interval. If $Q_n(x) = \sum_{i=0}^n \gamma_{n,i} x^i$, let $Q'_n(x_1, \ldots, x_n) = \sum_{i=0}^n \gamma_{n,i} \prod_{j=1}^i x_j$. Now, consider the estimator which draws a positive integer N according to some distribution $\mathbb{P}(N = n) = p_n$, samples X for N times to get $x_1, x_2, \ldots, x_N$, and returns

$\frac{1}{p_N}\left(Q'_N(x_1,\ldots,x_N) - Q'_{N-1}(x_1,\ldots,x_{N-1})\right)$, where we assume $Q'_0 = 0$. The expected value of this estimator is equal to

$$\begin{aligned}
&\mathbb{E}_{N,x_1,\ldots,x_N}\left[\frac{1}{p_N}\left(Q'_N(x_1,\ldots,x_N) - Q'_{N-1}(x_1,\ldots,x_{N-1})\right)\right] \\
&= \sum_{n=1}^{\infty}\frac{p_n}{p_n}\mathbb{E}_{x_1,\ldots,x_n}\left[Q'_n(x_1,\ldots,x_n) - Q'_{n-1}(x_1,\ldots,x_{n-1})\right] \\
&= \sum_{n=1}^{\infty}\left(Q_n(\mathbb{E}[X]) - Q_{n-1}(\mathbb{E}[X])\right) = f(\mathbb{E}[X]).
\end{aligned}$$

Thus, we have an unbiased estimator of $f(\mathbb{E}[X])$.

This technique was introduced in a rather obscure early 1960's paper [16] from sequential estimation theory, and appears to be little known. However, we believe this technique is interesting, and expect it to have useful applications for other problems as well.

While this may seem at first like a very general result, the variance of this estimator must be bounded for it to be useful. Unfortunately, this is not true for general continuous functions. More precisely, let N be distributed according to p_n, and let θ be the value returned by the estimator of $f(\mathbb{E}[X])$. In [17], it is shown that if X is a Bernoulli random variable, and if $\mathbb{E}[\theta N^k] < \infty$ for some integer $k \geq 1$, then f must be k times continuously differentiable. Since $\mathbb{E}[\theta N^k] \leq (\mathbb{E}[\theta^2] + \mathbb{E}[N^{2k}])/2$, this means that functions f which yield an estimator with finite variance, while using a number of queries with bounded variance, must be continuously differentiable. Moreover, in case we desire the number of queries to be essentially constant (e.g., choose a distribution for N with exponentially decaying tails), we must have $\mathbb{E}[N^k] < \infty$ for all k, which implies that f should be infinitely differentiable (in fact, in [17] it is conjectured that f must be analytic in such cases).

Thus, we focus in this paper on functions f which are analytic, i.e., they can be written as $f(x) = \sum_{i=0}^{\infty}\gamma_i x^i$ for appropriate constants $\gamma_0, \gamma_1, \ldots$. In that case, Q_n can simply be the truncated Taylor expansion of f to order n, i.e., $Q_n = \sum_{i=0}^{n}\gamma_i x^i$. Moreover, we can pick $p_n \propto 1/p^n$ for any $p > 1$. So the estimator works as follows: we sample a nonnegative integer N according to $\mathbb{P}(N = n) = (p-1)/p^{n+1}$, sample X independently N times to get $x_1, x_2, \ldots, x_N$, and return $\theta = \gamma_N \frac{p^{N+1}}{p-1} x_1 x_2 \cdots x_N$ where we set $\theta = \frac{p}{p-1}\gamma_0$ if $N = 0$.[2] We have the following:

Lemma 1: For the above estimator, it holds that $\mathbb{E}[\theta] = f(\mathbb{E}[X])$. The expected number of samples used by the estimator is $1/(p-1)$, and the probability of it being at least z is p^{-z}. Moreover, if we assume that $f_+(x) = \sum_{n=0}^{\infty}|\gamma_n|x^n$ exists for any x in the domain of interest, then

$$\mathbb{E}[\theta^2] \leq \frac{p}{p-1} f_+^2\left(\sqrt{p\mathbb{E}[X^2]}\right).$$

Proof: The fact that $\mathbb{E}[\theta] = f(\mathbb{E}[X])$ follows from the discussion above. The results about the number of samples follow directly from properties of the geometric distribution. As for the second moment, $\mathbb{E}[\theta^2]$ equals

$$\begin{aligned}
&\mathbb{E}_{N,x_1,\ldots,x_N}\left[\gamma_N^2\frac{p^{2(N+1)}}{(p-1)^2}x_1^2x_2^2\cdots x_N^2\right] \\
&= \sum_{n=0}^{\infty}\frac{(p-1)p^{2(n+1)}}{(p-1)^2p^{n+1}}\gamma_n^2\mathbb{E}_{x_1,\ldots,x_n}\left[x_1^2x_2^2\cdots x_n^2\right] \\
&= \frac{p}{p-1}\sum_{n=0}^{\infty}\gamma_n^2p^n\left(\mathbb{E}[X^2]\right)^n \\
&= \frac{p}{p-1}\sum_{n=0}^{\infty}\left(|\gamma_n|\left(\sqrt{p\mathbb{E}[X^2]}\right)^n\right)^2 \\
&\leq \frac{p}{p-1}\left(\sum_{n=0}^{\infty}|\gamma_n|\left(\sqrt{p\mathbb{E}[X^2]}\right)^n\right)^2 \\
&= \frac{p}{p-1}f_+^2\left(\sqrt{p\mathbb{E}[X^2]}\right).
\end{aligned}$$

■

The parameter p provides a *tradeoff* between the variance of the estimator and the number of samples needed: the larger is p, the less samples we need, but the estimator has more variance. In any case, the sample size distribution decays exponentially fast.

It should be emphasized that the estimator associated with Lemma 1 is tailored for generality, and is suboptimal in some cases. For example, if f is a polynomial function, then $\gamma_n = 0$ for sufficiently large n, and there is no reason to sample N from a distribution supported on all nonnegative integers: it just increases the variance. Nevertheless, in order to keep the presentation uniform and general, we always use this type of estimator. If needed, the estimator can be optimized for specific cases.

We also note that this technique can be improved in various directions, if more is known about the distribution of X. For instance, if we have some estimate of the expectation and variance of X, then we can perform a Taylor expansion around the estimated $\mathbb{E}[X]$ rather than 0, and tune the probability distribution of N to be different than the one we used above. These modifications can allow us to make the variance of the estimator arbitrarily small, if the variance of X is small enough. Moreover, one can take polynomial approximations to f which are perhaps better than truncated Taylor expansions. In this paper, for simplicity, we ignore these potential improvements.

Finally, we note that a related result in [17] implies that it is impossible to estimate $f(\mathbb{E}[X])$ in an unbiased manner when f is discontinuous, even if we allow a number of queries and estimator values which are infinite in expectation. Since the derivatives of some well-known loss functions (such as the hinge loss) are discontinuous, estimating their gradient in an unbiased manner and arbitrary noise appears to be impossible. While our techniques allow us to work with "smoothed" approximate versions of such losses, the regret guarantees degrades with the quality of approximation, and this prevents us from saying anything nontrivial about learning with respect to the original losses. Thus, if online learning with noise and such loss functions is at all feasible, a rather different approach than ours needs to be taken.

[2]Admittedly, the event $N = 0$ should receive zero probability, as it amounts to "skipping" the sampling altogether. However, setting $\mathbb{P}(N = 0) = 0$ appears to improve the bound in this paper only in the smaller order terms, while making the analysis in the paper more complicated.

D. Unbiasing Noise in the RKHS

The second component in our approach to deal with unknown noise in the kernel setting involves the unbiased estimation of $\Psi(\mathbf{x}_t)$, when we only have unbiased noisy copies of $\mathbf{x}_t$. Here again, we have a nontrivial problem, because the feature mapping Ψ is usually highly nonlinear, so $\mathbb{E}[\Psi(\tilde{\mathbf{x}}_t)] \neq \Psi(\mathbb{E}[\tilde{\mathbf{x}}_t])$ in general. Moreover, Ψ is not a scalar function, so the technique of Section III-C will not work as-is.

To tackle this problem, we construct an explicit feature mapping, which needs to be tailored to the kernel we want to use. To give a very simple example, suppose we use the homogeneous second-degree polynomial kernel $k(\mathbf{x},\mathbf{a}) = (\langle \mathbf{x},\mathbf{a}\rangle)^2$. It is not hard to verify that the function $\Psi : \mathbb{R}^d \mapsto \mathbb{R}^{d^2}$, defined via $\Psi(\mathbf{x}) = (x_1x_1, x_1x_2, \ldots, x_dx_d)$, is an explicit feature mapping for this kernel. Now, if we query two independent noisy copies $\tilde{\mathbf{x}}$, $\tilde{\mathbf{x}}'$ of $\mathbf{x}$, we have that the expectation of the random vector $(\tilde{x}_1\tilde{x}'_1, \tilde{x}_1\tilde{x}'_2, \ldots, \tilde{x}_d\tilde{x}'_d)$ is nothing more than $\Psi(\mathbf{x})$. Thus, we can construct unbiased estimates of $\Psi(\mathbf{x})$ in the RKHS. Of course, this example pertains to a very simple RKHS with a finite dimensional representation. By a randomization technique somewhat similar to the one in Section III-C, we can adapt this approach to infinite dimensional RKHS as well. In a nutshell, we represent $\Psi(\mathbf{x})$ as an infinite-dimensional vector, and its noisy unbiased estimate is a vector which is nonzero on only finitely many entries, using finitely many noisy queries. Moreover, inner products between these estimates can be done efficiently, allowing us to implement the learning algorithms, and use the resulting predictors on test instances.

IV. Auxiliary Information on the Noise Distribution

In the first part of the paper, we focus on the squared loss, and discuss the implication of being provided different levels of auxiliary information on the noise distribution in each round.

The first setting assumes just a known upper bound on the variance of the noise. For the specific case of linear predictors, we show one can learn using two noisy copies of each $\mathbf{x}_t$ and one noisy copy of each y_t.

The second setting assumes that the covariance structure of the noise is known. In that case, we show that one can learn linear predictors with only one noisy copy of both $\mathbf{x}_t$ and y_t.

The third and most complex setting we consider is when the noise has a fixed Gaussian distribution with known covariance matrix. We show that one can even learn kernel-based predictors, using two independent noisy copies of each $\mathbf{x}_t$, and one noisy copy of y_t. We focus on Gaussian kernels, but also show how the result can be extended, in a certain sense, to general radial kernels.

Throughout the rest of the paper, we let $\mathbb{E}_t[\cdot]$ be a shorthand for expectation over $(\tilde{\mathbf{x}}_t, \tilde{\mathbf{x}}'_t, \tilde{y}_t)$ conditioned on $\{(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}'_i, \tilde{y}_i)\}_{i=1}^{t-1}$.

A. Setting 1: Upper Bound on the Variance

We begin with the simplest setting, which is when we only know that $\mathbb{E}_t\left[\|\tilde{\mathbf{x}}_t\|^2\right] \leq B^2_{\tilde{\mathbf{x}}}$ and $\mathbb{E}_t\left[\tilde{y}_t^2\right] \leq B^2_{\tilde{y}}$ for some known constants $B_{\tilde{\mathbf{x}}}, B_{\tilde{y}}$. Conditional expectation is used here because we are assuming the adversary can change the noise distribution after each round, depending on the realizations of the past noisy examples. We present an algorithm for learning linear predictors, using exactly two independent noisy copies of the instance $\mathbf{x}_t$ and one noisy copy of the target value y_t. As discussed in Section III, the algorithm is based on an adaptation of online gradient descent, and the main requirement is to construct an unbiased estimate of the gradient ∇_t. This follows from the following lemma.

Lemma 2: Let $\nabla_t = 2(\langle \mathbf{w}_t, \mathbf{x}_t\rangle - y_t)\mathbf{x}_t$ be the gradient of $(\langle \mathbf{w}_t, \mathbf{x}_t\rangle - y_t)^2$ at $\mathbf{w}_t$. Let $\tilde{\mathbf{x}}'_t$ be an additional independent copy of $\tilde{\mathbf{x}}_t$, and denote $\tilde{\nabla}_t = 2(\langle \mathbf{w}_t, \tilde{\mathbf{x}}_t\rangle - \tilde{y}_t)\tilde{\mathbf{x}}'_t$. Under the above assumptions, if $\|\mathbf{w}_t\| \leq B_{\mathbf{w}}$, then $\mathbb{E}_t[\tilde{\nabla}_t] = \nabla_t$ and $\mathbb{E}_t[\|\tilde{\nabla}_t\|^2] \leq G$, where $G = 4(B^2_{\mathbf{w}}B^2_{\tilde{\mathbf{x}}} + B^2_{\tilde{y}})B^2_{\tilde{\mathbf{x}}}$.

Proof: Because of the independence assumption, we have

$$\mathbb{E}_t[\tilde{\nabla}_t] = 2\,\mathbb{E}_t[\langle \mathbf{w}_t, \tilde{\mathbf{x}}_t\rangle - \tilde{y}_t]\,\mathbb{E}_t[\tilde{\mathbf{x}}'_t] = 2(\langle \mathbf{w}_t, \mathbf{x}_t\rangle - y_t)\mathbf{x}_t = \nabla_t.$$

For the second claim, we have by the independence assumption that

$$\begin{aligned}\mathbb{E}_t[\|\tilde{\nabla}_t\|^2] &= 4\mathbb{E}_t\left[(\langle \mathbf{w}_t, \tilde{\mathbf{x}}_t\rangle - \tilde{y}_t)^2\right]\mathbb{E}_t\left[\|\tilde{\mathbf{x}}'_t\|^2\right] \\ &= 4\left(\mathbb{E}_t\left[\langle \mathbf{w}_t, \tilde{\mathbf{x}}_t\rangle^2\right] + \mathbb{E}_t[\tilde{y}_t^2]\right)\mathbb{E}_t\left[\|\tilde{\mathbf{x}}'_t\|^2\right] \\ &\leq 4\left(B^2_{\mathbf{w}}B^2_{\tilde{\mathbf{x}}} + B^2_{\tilde{y}}\right)B^2_{\tilde{\mathbf{x}}}.\end{aligned}$$

■

The following theorem provides a bound on the regret for Algorithm 1. The proof is provided in Section VIII-A.

Algorithm 1 Learning with Upper Bound on Noise Variance

PARAMETERS: η, $B_{\mathbf{w}}$.

INITIALIZE: $\mathbf{w}_1 = \mathbf{0}$.

For $t = 1, 2, \ldots, T$

Receive $(\tilde{\mathbf{x}}_t, \tilde{y}_t)$

Receive another independent copy $\tilde{\mathbf{x}}'_t$

$\tilde{\nabla}_t = 2(\langle \mathbf{w}_t, \tilde{\mathbf{x}}_t\rangle - \tilde{y}_t)\tilde{\mathbf{x}}'_t$

$\mathbf{w}' = \mathbf{w}_t - \eta\tilde{\nabla}_t$

$\mathbf{w}_{t+1} = \min\{1, B_{\mathbf{w}}/\|\mathbf{w}'\|\}\,\mathbf{w}'$

Theorem 1: Let $\ell(a,y) = (a-y)^2$ be the squared loss. For all t assume that $\mathbb{E}_t[\|\tilde{\mathbf{x}}_t\|^2] \leq B^2_{\tilde{\mathbf{x}}}$, $\mathbb{E}_t[\tilde{y}_t^2] \leq B^2_{\tilde{y}}$, and that $\tilde{\mathbf{x}}_t$, $\tilde{\mathbf{x}}'_t$, $\tilde{y}_t$ are mutually independent. If we run Algorithm 1 with parameters $B_{\mathbf{w}}$, $\eta = B_{\mathbf{w}}/\sqrt{GT}$ (where G is defined in Lemma 2), then

$$\mathbb{E}\left[\sum_{t=1}^{T}\ell(\langle \mathbf{w}_t, \mathbf{x}_t\rangle, y_t)\right] - \min_{\mathbf{w}\,:\,\|\mathbf{w}\|\leq B_{\mathbf{w}}}\sum_{t=1}^{T}\ell(\langle \mathbf{w}, \mathbf{x}_t\rangle, y_t) \leq B_{\mathbf{w}}\sqrt{GT}.$$

B. Setting 2: Known Covariance

We now turn to the case where rather than an upper bound on the variance, we actually know the covariance matrix of the noise at each round, which we denote as Σ_t. We assume that

$\|\Sigma_t\| \leq B_\Sigma$ for all t, where $\|\cdot\|$ denotes the spectral norm. As to $\tilde{y}_t$, we can still assume we only have an upper bound $B_{\tilde{y}}^2$ on $\mathbb{E}_t[\tilde{y}_t^2]$ (with our algorithmic approach, knowing $\mathbb{E}_t[\tilde{y}_t^2]$ does not help much).

In this setting, we show it is possible to learn linear predictors, using just a single noisy copy $(\tilde{\mathbf{x}}_t, \tilde{y}_t)$. This is opposed to the previous subsection, where we needed an additional independent copy of $\tilde{\mathbf{x}}_t$. The idea is that if we use just one noisy copy in our gradient estimate, we need to deal with bias terms. When the covariance structure is known, we can calculate and remove these bias terms, allowing an online gradient descent similar to Algorithm 1 to work. As in Algorithm 1, the basic building block is a construction of an unbiased estimate of the gradient ∇_t at each iteration. See Algorithm 2 for the pseudocode.

Algorithm 2 Learning with Known Noise Covariance

PARAMETERS: η, $B_{\mathbf{w}}$.

INITIALIZE: $\mathbf{w}_1 = \mathbf{0}$.

For $t = 1, 2, \ldots, T$

Receive $(\tilde{\mathbf{x}}_t, \tilde{y}_t)$

$\tilde{\nabla}_t = 2(\langle \mathbf{w}_t, \tilde{\mathbf{x}}_t \rangle - \tilde{y}_t)\tilde{\mathbf{x}}_t - \Sigma_t \mathbf{w}_t$

$\mathbf{w}' = \mathbf{w}_t - \eta \tilde{\nabla}_t$

$\mathbf{w}_{t+1} = \min\{1, B_{\mathbf{w}}/\|\mathbf{w}'\|\}\, \mathbf{w}'$

Lemma 3: Let $\nabla_t = 2(\langle \mathbf{w}_t, \mathbf{x}_t \rangle - y_t)\mathbf{x}_t$ be the gradient of $(\langle \mathbf{w}_t, \mathbf{x}_t \rangle - y_t)^2$ at $\mathbf{w}_t$. Denote $\tilde{\nabla}_t = 2(\langle \mathbf{w}_t, \tilde{\mathbf{x}}_t \rangle - \tilde{y}_t)\tilde{\mathbf{x}}_t - \Sigma_t \mathbf{w}_t$, where Σ_t is the covariance matrix of $\tilde{\mathbf{x}}_t$. Then under the assumptions above, if $\|\mathbf{w}_t\| \leq B_{\mathbf{w}}$, $\mathbb{E}_t[\|\tilde{\mathbf{x}}_t\|^2] \leq B_{\tilde{\mathbf{x}}}^2$, and $\mathbb{E}_t[\|\tilde{\mathbf{x}}_t\|^4] \leq B_{\tilde{\mathbf{x}}}'^4$, then $\mathbb{E}_t[\tilde{\nabla}_t] = \nabla_t$ and $\mathbb{E}_t[\|\tilde{\nabla}\|_t^2] \leq G$, where $G = 8B_{\mathbf{w}}^2 B_{\tilde{\mathbf{x}}}'^4 + 8B_{\tilde{y}}^2 B_{\tilde{\mathbf{x}}}^2 + 4B_{\mathbf{w}}^2 B_{\tilde{\mathbf{x}}}^2 B_\Sigma + B_\Sigma^2 B_{\mathbf{w}}^2$.

Proof: Using the zero-mean and independence assumptions on $\mathbf{n}_t^x$, n_t^y, we have

$$\begin{aligned}
&\mathbb{E}_t\left[2(\langle \mathbf{w}_t, \tilde{\mathbf{x}}_t \rangle - \tilde{y}_t)\tilde{\mathbf{x}}_t\right] \\
&\quad = 2\mathbb{E}_t\left[(\langle \mathbf{w}_t, \mathbf{x}_t + \mathbf{n}_t^x \rangle - y_t - n_t^y)(\mathbf{x}_t + \mathbf{n}_t^x)\right] \\
&\quad = 2(\langle \mathbf{w}_t, \mathbf{x}_t \rangle - y_t)\mathbf{x}_t + \mathbb{E}_t[\mathbf{n}_t^x \langle \mathbf{w}_t, \mathbf{n}_t^x \rangle] \\
&\quad = \nabla_t + \Sigma_t \mathbf{w}_t
\end{aligned}$$

which implies that $\mathbb{E}_t[\tilde{\nabla}_t] = \nabla_t$. As to the second claim, using the well-known inequality $\|\mathbf{a} + \mathbf{b}\|^2 \leq 2\|\mathbf{a}\|^2 + 2\|\mathbf{b}\|^2$, we have

$$\begin{aligned}
&\mathbb{E}_t[\|\tilde{\nabla}_t\|^2] = \mathbb{E}_t[\|2(\langle \mathbf{w}_t, \tilde{\mathbf{x}}_t - \tilde{y}_t \rangle)\tilde{\mathbf{x}}_t - \Sigma_t \mathbf{w}_t\|^2] \\
&= \mathbb{E}_t \|2(\langle \mathbf{w}_t, \tilde{\mathbf{x}}_t \rangle - \tilde{y}_t)\tilde{\mathbf{x}}_t\|^2 \\
&\qquad - 4\mathbf{w}_t^\top \Sigma_t \mathbb{E}_t[(\langle \mathbf{w}_t, \tilde{\mathbf{x}}_t \rangle - y_t)\tilde{\mathbf{x}}_t] + \|\Sigma_t \mathbf{w}_t\|^2 \\
&\leq 8\mathbb{E}_t[\|\langle \mathbf{w}_t, \tilde{\mathbf{x}}_t \rangle \tilde{\mathbf{x}}_t\|^2] + 8\mathbb{E}_t[\|\tilde{y}_t \tilde{\mathbf{x}}_t\|^2] \\
&\qquad - 4\mathbf{w}_t^\top \Sigma_t \mathbb{E}_t[\langle \mathbf{w}_t, \tilde{\mathbf{x}}_t \rangle \mathbf{x}_t] + \|\Sigma_t \mathbf{w}_t\|^2 \\
&\leq 8\|\mathbf{w}_t\|^2 \mathbb{E}_t[\|\tilde{\mathbf{x}}_t\|^4] + 8\mathbb{E}_t[\tilde{y}_t^2]\mathbb{E}_t[\|\tilde{\mathbf{x}}_t^2\|] \\
&\qquad + 4\|\mathbf{w}_t\|^2 \mathbb{E}_t[\|\tilde{\mathbf{x}}_t\|^2]\|\Sigma_t\| + \|\Sigma_t\|^2 \|\mathbf{w}_t\|^2 \\
&\leq 8B_{\mathbf{w}}^2 B_{\tilde{\mathbf{x}}}'^4 + 8B_{\tilde{y}}^2 B_{\tilde{\mathbf{x}}}^2 + 4B_{\mathbf{w}}^2 B_{\tilde{\mathbf{x}}}^2 B_\Sigma + B_\Sigma^2 B_{\mathbf{w}}^2.
\end{aligned}$$

■

Theorem 2: Let $\ell(a, y) = (a - y)^2$ be the squared loss. For all t assume that $\mathbf{x}_t$ and y_t are perturbed by independent noise such that the known covariance matrix Σ_t of the noise added to $\mathbf{x}_t$ satisfies $\|\Sigma_t\| \leq B_\Sigma$. Assume further that $\mathbb{E}_t[\|\tilde{\mathbf{x}}_t\|^2] \leq B_{\tilde{\mathbf{x}}}$, $\mathbb{E}_t[\|\tilde{\mathbf{x}}_t\|^4] \leq B_{\tilde{\mathbf{x}}}'^4$, and $\mathbb{E}_t[\tilde{y}_t^2] \leq B_{\tilde{y}}^2$. If we run Algorithm 2 with parameters $B_{\mathbf{w}}$ and $\eta = B_{\mathbf{w}}/\sqrt{T}$, where G is defined in Lemma 3, then

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^T \ell(\langle \mathbf{w}_t, \mathbf{x}_t \rangle, y_t)\right] - \min_{\mathbf{w}\,:\,\|\mathbf{w}\| \leq B_{\mathbf{w}}} \sum_{t=1}^T \ell(\langle \mathbf{w}, \mathbf{x}_t \rangle, y_t) \\
&\leq \frac{1}{2}(G+1)B_{\mathbf{w}}\sqrt{T}.
\end{aligned}$$

The proof is similar to the proof of Theorem 1, with Lemma 3 replacing Lemma 2. We note that if G is known (which requires knowing a bound on the fourth moment of $\tilde{\mathbf{x}}_t$), then by picking $\eta = B_{\mathbf{w}}/\sqrt{GT}$ one can improve the bound to $B_{\mathbf{w}}\sqrt{GT}$.

C. Setting 3: Gaussian Distribution

The third and most complex setting we consider in this section is when the noise is assumed to have a Gaussian distribution $\mathcal{N}(0, \Sigma)$. Clearly, if we know the distribution, then we can derive upper bounds on the moments of $\tilde{\mathbf{x}}_t$ (assuming bounds are known on the original instances $\mathbf{x}_t$). Thus, the results of Section IV-B carry through to our setting, and we can learn linear predictors. However, when we also know the noise has a specific Gaussian distribution, we can learn the much more powerful hypothesis class of kernel-based predictors.

Recall that the basic premise of kernel-based learning is that the data (originally in $\mathbb{R}^d$) is mapped to some reproducing kernel Hilbert space (RKHS), via a feature mapping $\Psi(\mathbf{x})$, and a linear predictor is learned in that space. In our original space, this corresponds to learning a nonlinear function. Using the well-known kernel trick, inner products $\langle \Psi(\mathbf{x}), \Psi(\mathbf{x}') \rangle$ in the RKHS (which might be infinite-dimensional) can be easily computed via a kernel function $k(\mathbf{x}, \mathbf{x}')$.

While there are many possible kernel functions, perhaps the most popular one is the Gaussian kernel, defined as $k(\mathbf{x}, \mathbf{x}') = \exp(-\|\mathbf{x} - \mathbf{x}'\|^2 / s^2)$ for some $s^2 > 0$ (the kernel width). This corresponds to the inner product $\langle \Psi(\mathbf{x}), \Psi(\mathbf{x}') \rangle$ in an appropriate RKHS. We we will show below how to learn from noisy data with Gaussian kernels. In Section IV-D, we show how this can be extended, in a certain sense, to general *radial* kernels, i.e., kernels of the form $k(\mathbf{x}, \mathbf{x}') = f(\|\mathbf{x} - \mathbf{x}'\|)$ for an appropriate real function f.

In this subsection, we assume that the noise distribution is fixed for all t. Hence, we may assume w.l.o.g. that Σ is a diagonal matrix, with element σ_i^2 at row/column i. To see why, notice that there always exists a rotation matrix R, such that $R\tilde{\mathbf{x}}_t$ has a Gaussian distribution with diagonal covariance matrix. Therefore, instead of learning with respect to $\{(\tilde{\mathbf{x}}_t, y_t)\}_{t=1}^T$, we can just learn with respect to $\{(R\tilde{\mathbf{x}}_t, y_t)\}_{t=1}^T$, and predict on any instance $\mathbf{x}$ by pre-rotating it using R. Since we focus here on rotationally invariant kernels, which depend just on the Euclidean distance between instances, we have that $k(\mathbf{x}, \mathbf{x}') = k(R\mathbf{x}, R\mathbf{x}')$ for any $\mathbf{x}, \mathbf{x}'$. Therefore, the data structure remains the same in the kernel space, and all our guarantees will still

hold. As to $\tilde{y}_t$, similar to the previous settings, we will only need to assume that $\mathbb{E}_t[\tilde{y}_t^2] \leq B_{\tilde{y}}^2$ for some known parameter $B_{\tilde{y}}$.

The algorithm that we present (Algorithm 3) is based on being able to receive two independent copies of each instance $\tilde{\mathbf{x}}_t$, as well as a single independent copy of $\tilde{y}_t$. As in the linear case, the learning algorithm that we use relies upon the online gradient descent technique due to [11], with the main difference being that instead of using a Gaussian kernel of width s^2, we use a surrogate kernel, as discussed in Section III.

Algorithm 3 Kernel Learning Algorithm with Gaussian Noise $\mathcal{N}(0, \Sigma)$

PARAMETERS: W, η

INITIALIZE: $\alpha_i := 0$ for all $i = 1, \ldots, T$

For $t = 1, \ldots, T$:

Define $\hat{\mathbf{w}}_t = \sum_{i=1}^{t-1} \alpha_i \hat{\Psi}(\tilde{\mathbf{x}}_i)$

Define $\mathbf{w}_t = \sum_{i=1}^{t-1} \alpha_i \Psi(\tilde{\mathbf{x}}_i)$

Receive $\tilde{y}_t$, $\tilde{\mathbf{x}}_t$, and independent copy $\tilde{\mathbf{x}}'_t$

Let $\tilde{g}_t := 2\left(\sum_{i=1}^{t-1} \alpha_i \hat{k}(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}'_t) - \tilde{y}_t\right)$

//$\tilde{g}_t$ is gradient length with respect to $\hat{\Psi}(\tilde{\mathbf{x}}'_t)$ at $\hat{\mathbf{w}}_t$

Let $\alpha_t := -\eta \tilde{g}_t$

Let $r_t := \sum_{i=1}^{t} \sum_{j=1}^{t} \alpha_i \alpha_j \hat{k}(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j)$

If $r_t > W^2$ // If $\|\hat{\mathbf{w}}_t\|^2 > W^2$, then project

Let $\alpha_i := \alpha_i \frac{W}{\sqrt{r_t}}$ for all $i = 1, \ldots, t$

In order to define the surrogate kernel that we use, consider the RKHS corresponding to the kernel

$$\hat{k}(\mathbf{x}, \mathbf{x}') = R_{\Sigma,s,d}^2 \exp\left(-\sum_{i=1}^{d} \frac{(x_i - x'_i)^2}{s^2 - 2\sigma_i^2}\right) \tag{5}$$

where we assume that $2\|\Sigma\| = 2\max_i \sigma_i^2$ is less than s^2 and

$$R_{\Sigma,s,d} = \left(\prod_{i=1}^{d} \frac{s^2}{s^2 - 2\sigma_i^2}\right)^{1/4}.$$

This can be shown to be a kernel by standard results (see for instance [8]). Note that $R_{\Sigma,s,d}$ can be bounded by a constant when $\sigma_i = O(1)$ for all i (constant noise) and $s^2 = \Theta(d)$—plausible when the feature values of observed instances $\mathbf{x}$ are of order $\Theta(1)$. Let $\hat{\Psi}$ be the feature mapping corresponding to this RKHS.

The pseudocode of our algorithm is presented below. Formally speaking, it is just applying online gradient descent, using kernels, in the surrogate RKHS that we constructed. However, it is crucial to note that the actual output are elements $\mathbf{w}_1, \mathbf{w}_2, \ldots$ in the RKHS corresponding to Ψ.

Before stating the bound for Algorithm 3 we need an auxiliary definition. Suppose that $\mathbf{w}$ is any element in the RKHS of Ψ, which can be written as $\sum_{t=1}^{T} a_t \Psi(\mathbf{x}_t)$ for some $a_1, \ldots, a_T \geq 0$. For example, this includes

$$\operatorname*{argmin}_{\mathbf{u}\,:\,\|\mathbf{u}\| \leq B_{\mathbf{w}}} \sum_{t=1}^{T} \ell(\langle \mathbf{u}, \Psi(\mathbf{x}_t) \rangle, y_t)$$

for any $B_{\mathbf{w}} > 0$ by the representer theorem. Define $\beta_{\mathbf{w}}$ to be the angle between $\sum_{t:a_t>0} a_t \Psi(\mathbf{x}_t)$ and $-\sum_{t:a_t<0} a_t \Psi(\mathbf{x}_t)$. In other words, this is the angle between the component due to positive support vectors, and the component due to the negative support vectors. If one of the components is zero, define $\beta_{\mathbf{w}}$ to be $\pi/2$. The main theorem of this section, whose proof is presented in Section VIII-B, is the following.

Theorem 3: Let $\ell(a, y) = (a - y)^2$ be the squared loss. For all t assume that $\mathbf{x}_t$ is perturbed by Gaussian noise with known distribution $\mathcal{N}(0, \Sigma)$, where Σ is diagonal, and y_t is perturbed by arbitrary independent noise with $\mathbb{E}_t[\tilde{y}_t^2] \leq B_{\tilde{y}}^2$. Let $B_{\mathbf{w}} > 0$ and $\beta \in (0, \pi/2]$ be fixed. If we run Algorithm 3 with the kernel (5) such that $s^2 \geq 2\|\Sigma\|$, and input parameters

$$W \geq \frac{\sqrt{5} B_{\mathbf{w}} R_{\Sigma,s,d}}{\sin(\beta)}$$

and

$$\eta = \frac{W}{2R_{\Sigma,s,d}\sqrt{\left(W^2 R_{\Sigma,s,d}^2 + B_{\tilde{y}}^2\right)T}}$$

then

$$\mathbb{E}\left[\sum_{t=1}^{T} \ell(\langle \mathbf{w}_t, \Psi(\mathbf{x}_t) \rangle, y_t)\right] - \min_{\mathbf{w} \in \mathcal{W}} \sum_{t=1}^{T} \ell(\langle \mathbf{w}, \Psi(\mathbf{x}_t) \rangle, y_t) \leq 2W R_{\Sigma,s,d}\sqrt{\left(W^2 R_{\Sigma,s,d}^2 + B_{\tilde{y}}^2\right)T}$$

where $\mathcal{W} = \left\{\mathbf{w} : \|\mathbf{w}\| \leq B_{\mathbf{w}},\ \beta_{\mathbf{w}} \geq \beta\right\}$ and Ψ is the feature mapping induced by the Gaussian kernel with width s^2. In particular, if $s^2 = \Omega(d)$, $\|\Sigma\| = O(1)$, and $B_{\tilde{y}} = O(1)$, then the above bound is $O\left(W^2\sqrt{T}\right)$.

The intuition for $\beta_{\mathbf{w}}$ is that it measures how well separated are the training examples: if the "positive" and "negative" example groups are not too close together, then the angle between $\sum_{t:a_t>0} a_t \Psi(\mathbf{x}_t)$ and $-\sum_{i:a_t<0} a_t \Psi(\mathbf{x}_t)$ will be large, and the bound will be small. Note that in the RKHS corresponding to a Gaussian kernel, $\beta_{\mathbf{w}}$ is *always* between 0 and $\pi/2$, since the inner product between any two elements $\Psi(\mathbf{x})$ and $\Psi(\mathbf{x}')$ is positive. In addition, $\beta_{\mathbf{w}}$ can be shown to be exactly zero if and only if the positive and negative examples exactly coincide. Overall, on realistic datasets, assuming there exist some good predictor $\mathbf{w}$ with $\beta_{\mathbf{w}}$ not too small is a pretty mild assumption, if something interesting can be learned even on the unperturbed data.

D. Extension to General Radial Kernels

The Gaussian kernel we discussed previously is a member of the family of *radial kernels*, that is kernels on $\mathbf{x}$, $\mathbf{x}'$ which can be written as a function of $\|\mathbf{x} - \mathbf{x}'\|$. Although the Gaussian kernel is the most popular member of this family, there are many other radial kernels, such as $\exp(-\|\mathbf{x} - \mathbf{x}'\|/s)$ and $(1+$

$\|\mathbf{x}-\mathbf{x}'\|^2/m)^{-\alpha}$ for appropriate parameters s, m, α. Thus, a reasonable question is whether Algorithm 3 and its analysis can be extended to general radial kernels. The extension we are able to show is in the following sense: for any radial kernel $g(\mathbf{x},\mathbf{x}')$, there exists another radial kernel $k(\mathbf{x},\mathbf{x}')$, which approximates $g(\mathbf{x},\mathbf{x}')$ arbitrarily well, for which one can extend Algorithm 3 and its analysis. Although the approximation parameter is user-defined, the bound on the regret depends on this parameter and deteriorates as the approximation gets better.

Recall from Section III-B that the heart of our approach is constructing a surrogate RKHS, with surrogate kernel $\hat{k}$, such that $\mathbb{E}[\hat{k}(\mathbf{a},\tilde{\mathbf{x}})]=k(\mathbf{a},\mathbf{x})$. In the Gaussian kernel case, the required surrogate RKHS corresponds to the kernel defined in (5). To deal with other kernels, constructing an appropriate surrogate kernel becomes trickier. Luckily, we can still reduce the problem, in some sense, to the case of Gaussian kernels. The key technical result is the following theorem due to Schoenberg ([18], see also [19]), slightly paraphrased and adapted to our purposes[3]:

Theorem 4 (Schoenberg's Theorem): A function $g(\cdot,\cdot)$ is a radial kernel corresponding to a valid RKHS, if and only if there exists a finite nonnegative measure μ on $[0,\infty)$, such that for any $\mathbf{x},\mathbf{a}\in\mathbb{R}^d$

$$g(\mathbf{x},\mathbf{a})=\int_{u=0}^{\infty}\exp\left(-u\|\mathbf{x}-\mathbf{a}\|^2\right)\mu(u)du.$$

This result asserts that, up to normalization factors, radial kernels can be characterized as Laplace transforms of probability measures on the positive reals. Schoenberg's Theorem has been used by Micchelli *et al.* [20] to prove universality of radial kernels and by Scovel *et al.* [21] to establish approximation error bounds. A related result is Bochner's theorem (see, e.g., [22]), which characterizes the more general class of shift-invariant kernels as Fourier transforms of multivariate distributions on $\mathbb{R}^d$.

The above theorem implies that we can write inner products in our RKHS using the approximate kernel

$$k(\mathbf{x},\mathbf{a})=\int_{u=0}^{c}k_u(\mathbf{x},\mathbf{a})\mu(u)du \tag{6}$$

where $c>0$ is a parameter and k_u is the Gaussian kernel $k_u(\mathbf{x},\mathbf{a})=\exp(-u\|\mathbf{x}-\mathbf{a}\|^2)$ with kernel width $1/u$. Note that this is a valid kernel by the reverse direction of Theorem 4. If c is chosen not too small, then $k(\mathbf{x},\mathbf{a})$ is an excellent approximation to $g(\mathbf{x},\mathbf{a})$ for all $\mathbf{x},\mathbf{a}$. The reason why we must settle for approximations of the radial kernel, rather than the kernel itself, is the following: for each k_u in the above integral, we construct a surrogate kernel $\hat{k}_u$ such that $\mathbb{E}_{\tilde{\mathbf{x}}}[\hat{k}_u(\tilde{\mathbf{x}},\mathbf{a})]=k_u(\mathbf{x},\mathbf{a})$. The surrogate kernel $\hat{k}_u$ is based on subtracting certain constants from the kernel width $1/u$ along each dimension, and this cannot be done if u is larger than those constants.

By Fubini's theorem, we can write (6) as

$$k(\mathbf{x},\mathbf{a})=\int_{u=0}^{c}\mathbb{E}\left[\hat{k}_u(\tilde{\mathbf{x}},\mathbf{a})\right]\mu(u)du=\mathbb{E}\left[\int_{u=0}^{c}\hat{k}_u(\tilde{\mathbf{x}},\mathbf{a})du\right].$$

It turns out that the integral inside the expectation corresponds to an inner product, in a valid RKHS, between the noisy instance $\tilde{\mathbf{x}}$ and $\mathbf{a}$. This will be our surrogate kernel for k.

To provide a concrete case study, we will outline the results for the specific radial kernel[4]

$$g(\mathbf{x},\mathbf{x}')=\left(1+\frac{1}{d}\|\mathbf{x}-\mathbf{x}'\|^2\right)^{-1}$$

postponing the full technical details and proofs to Section VIII-C. Just to make our analysis simpler to present, we assume here that $\Sigma=\sigma^2 I$ for some parameter σ^2, where $\sigma^2=O(1)$ (this is a reasonable assumption to make when the feature values of the original data is $\Theta(1)$).

The approximate kernel we will consider is

$$\begin{aligned}k(\mathbf{x},\mathbf{x}')&=d\int_0^{c/d}\exp\left(-u\|\mathbf{x}-\mathbf{x}'\|^2-ud\right)du\\&=g(\mathbf{x},\mathbf{x}')\left(1-\exp\left(-\frac{c}{g(\mathbf{x},\mathbf{x}')}\right)\right)\end{aligned} \tag{7}$$

where $c\in(0,d/4\sigma)$ is a user-defined parameter, which trades off the quality of the bound on the regret and the similarity of $k(\mathbf{x},\mathbf{x}')$ to $g(\mathbf{x},\mathbf{x}')$. This is a valid kernel by the reverse direction of Theorem 4 since

$$d\int_0^{c/d}\exp(-ud)du=1-\exp(-c)>0.$$

Note that $g(\mathbf{x},\mathbf{x}')$ is always between 0 and 1, so

$$\frac{k(\mathbf{x},\mathbf{x}')}{g(\mathbf{x},\mathbf{x}')}\in[1-\exp(-c),1].$$

Therefore, $k(\mathbf{x},\mathbf{x}')$ is an excellent approximation of $g(\mathbf{x},\mathbf{x}')$ for values of c not too small (see Fig. 1 for a graphical illustration). As before, we let Ψ denote the feature mapping associated with the kernel k.

The surrogate kernel that we will pick is defined as follows:

$$\begin{aligned}&\hat{k}(\mathbf{x},\mathbf{x}')\\&\quad=d\int_{u=0}^{c/d}(1-2\sigma u)^{-d/2}\exp\left(-\frac{u\|\mathbf{x}-\mathbf{x}'\|^2}{1-2\sigma u}-ud\right)du.\end{aligned} \tag{8}$$

As before, we let $\hat{\Psi}$ denote the feature mapping associated with this kernel. This is a valid kernel by the reverse direction of Theorem 4.

Our algorithm looks exactly like Algorithm 3, only that now we use the new definitions of $\hat{\Psi}$, Ψ above. To state the bound, recall that for any $\mathbf{w}=\sum_{t=1}^{T}a_t\Psi(\mathbf{x}_t)$ for some $a_1,\ldots,a_T$, we define $\beta_{\mathbf{w}}$ to be the angle between $\sum_{t:a_t>0}a_t\Psi(\mathbf{x}_t)$ and $-\sum_{t:a_t<0}a_t\Psi(\mathbf{x}_t)$. The bound takes the following form.

Theorem 5: Let $\ell(a,y)=(a-y)^2$ be the squared loss. For all t assume that $\mathbf{x}_t$ is perturbed by Gaussian noise with known distribution $\mathcal{N}(0,\sigma^2 I)$, and y_t is perturbed by arbitrary independent noise with $\mathbb{E}_t[\tilde{y}_t^2]\le B_{\tilde{y}}^2$. Let $B_{\mathbf{w}}>0$ and $\beta\in(0,\pi/2]$

[3]To be precise, the theorem here is a corollary of Schoenberg's theorem, which discusses necessary and sufficient conditions for $k(\cdot,\cdot)$ to be positive definite, and Mercer's theorem (see [8]), which asserts that such a function is a kernel of a valid RKHS.

[4]Note that the scaling factor $1/d$ is the reasonable one to take, when we assume that the attribute values in the instances are on the order of $\Theta(1)$.

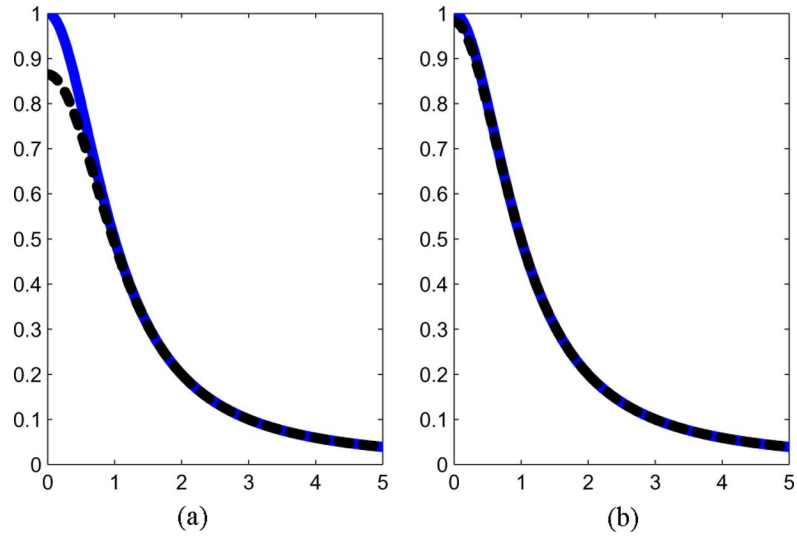

Fig. 1. Comparison of $g(\mathbf{x},\mathbf{x}')$ (solid line) and $k(\mathbf{x},\mathbf{x}')$ (dashed line) as a function of $\|\mathbf{x}-\mathbf{x}'\|$, for $c=2$ (left) and $c=4$ (right). Note that for $c=4$, the two graphs are visually indistinguishable.

be fixed. If we run Algorithm 3 with the kernel (7) where $c \in (0, d/4\sigma)$, and input parameters

$$W \geq \frac{\sqrt{5}B_{\mathbf{w}}\exp(\sigma c)}{\sin(\beta)}$$

and

$$\eta = \frac{W}{2\exp(\sigma c)\sqrt{(W^2\exp(2\sigma c)+B_{\tilde{y}}^2)T}}$$

then

$$\mathbb{E}\left[\sum_{t=1}^{T}\ell(\langle\mathbf{w}_t,\Psi(\mathbf{x}_t)\rangle,y_t)\right]-\min_{\mathbf{w}\in\mathcal{W}}\sum_{t=1}^{T}\ell(\langle\mathbf{w},\Psi(\mathbf{x}_t)\rangle,y_t) \leq 2W\exp(\sigma c)\sqrt{(W^2\exp(2\sigma c)+B_{\tilde{y}}^2)T}$$

where $\mathcal{W}=\{\mathbf{w} : \|\mathbf{w}\|\leq B_{\mathbf{w}},\ \beta_{\mathbf{w}}\geq\beta\}$ and Ψ is the feature mapping induced by the kernel (7).

The proof of the theorem is provided in Section VIII-C.

V. Unknown Noise Distribution

In this part of the paper, we turn to study the setting where we wish to learn kernel-based predictors, while having no information about the noise distribution other than an upper bound on its variance. This is relevant in cases where the noise is hard to model, or if it might change in an unexpected or even adversarial manner. Moreover, we provide results with respect to general analytic loss functions, which go beyond the squared loss on which we focused in Section IV. We emphasize that the techniques here are substantially different than those of Section IV, and do not rely on surrogate kernels. Instead, the techniques focus on construction of unbiased gradient estimates directly in the RKHS.

A. Algorithm

We present our algorithmic approach in a modular form. We start by introducing the main algorithm, which contains several subroutines. Then we prove our two main results, which bound the regret of the algorithm, the number of queries to the oracle, and the running time for two types of kernels: dot product and Gaussian (our results can be extended to other kernel types as well). In itself, the algorithm is nothing more than a standard online gradient descent algorithm with a standard $O(\sqrt{T})$ regret bound. Thus, most of the proofs are devoted to a detailed discussion of how the subroutines are implemented (including explicit pseudo-code). In this subsection, we describe just one subroutine, based on the techniques discussed in Section III. The other subroutines require a more detailed and technical discussion, and thus their implementation is described as part of the proofs in Section VIII. In any case, the intuition behind the implementations and the techniques used are described in Section III.

For the remainder of this subsection, we assume for simplicity that ℓ is a classification loss; namely, it can be written as a function of $\ell(y\langle\mathbf{w},\Psi(\mathbf{x})\rangle)$. It is not hard to adapt the results below to the case where ℓ is a regression loss (where ℓ is a function of $\langle\mathbf{w},\Psi(\mathbf{x})\rangle - y$). Another simplifying assumption we will make, purely in the interest of clarity, is that the noise will be restricted just to the instance $\mathbf{x}_t$, and not to the target value y_t. In other words, we assume that the learner is given access to y_t, and to an oracle A_t which provides noisy copies of $\mathbf{x}_t$. This does not make our lives easier, since the hard estimation problems relate to $\mathbf{x}_t$ and not y_t (e.g., estimating $\langle\mathbf{w}_t,\Psi(\mathbf{x}_t)\rangle$ in an unbiased manner, despite the nonlinearity of the feature mapping Ψ). On the other hand, it will help to make our results more transparent, and reduce tedious bookkeeping.

At each round, the algorithm below constructs an object which we denote as $\tilde{\Psi}(\mathbf{x}_t)$ (note that it has no relationship to $\hat{\Psi}(\mathbf{x}_t)$ used in the previous section). This object has two interpretations here: formally, it is an element of a reproducing kernel Hilbert space (RKHS) corresponding to the kernel we use, and is equal in expectation to $\Psi(\mathbf{x}_t)$. However, in terms

of implementation, it is simply a data structure consisting of a finite set of vectors from $\mathbb{R}^d$. Thus, it can be efficiently stored in memory and handled even for infinite-dimensional RKHS.

Like $\tilde{\Psi}(\mathbf{x}_t)$, $\mathbf{w}_{t+1}$ has also two interpretations: formally, it is an element in the RKHS, as defined in the pseudocode. In terms of implementation, it is defined via the data structures $\tilde{\Psi}(\mathbf{x}_1), \ldots, \tilde{\Psi}(\mathbf{x}_t)$ and the values of $\alpha_1, \ldots, \alpha_t$ at round t. To apply this hypothesis on a given instance $\mathbf{x}$, we compute $\sum_{i=1}^{t} \alpha_{t,i} \texttt{Mult}(\tilde{\Psi}(\mathbf{x}_i), \mathbf{x}')$, where $\texttt{Mult}(\tilde{\Psi}(\mathbf{x}_i), \mathbf{x}')$ is a subroutine which returns the inner product $\left\langle \tilde{\Psi}(\mathbf{x}_i), \Psi(\mathbf{x}') \right\rangle$ (a pseudocode is provided as part of the proofs in Section VIII).

We start by considering dot-product kernels; that is, kernels $k(\cdot,\cdot)$ that can be written as $k(\mathbf{x},\mathbf{x}') = Q(\langle \mathbf{x}, \mathbf{x}' \rangle)$, where $Q(\cdot)$ has a Taylor expansion $Q(a) = \sum_{n=0}^{\infty} \beta_n a^n$ such that $\beta_n \geq 0$ for all n—see theorem 4.19 in [8]. Our first result shows what regret bound is achievable by the algorithm for any dot-product kernel, as well as characterize the number of oracle queries per instance, and the overall running time of the algorithm. The proof is provided in Section VIII-E.

Theorem 6: Assume that the loss function ℓ has an analytic derivative $\ell'(a) = \sum_{n=0}^{\infty} \gamma_n a^n$ for all a in its domain, and let $\ell'_+(a) = \sum_{n=0}^{\infty} |\gamma_n| a^n$ (assuming it exists). Pick any dot-product kernel $k(\cdot,\cdot) = Q(\langle \cdot, \cdot \rangle)$. Finally, assume that $\mathbb{E}_t[\|\tilde{\mathbf{x}}_t\|^2] \leq B_{\tilde{\mathbf{x}}}$ for any $\tilde{\mathbf{x}}_t$ returned by the oracle at round t, for all $t = 1, \ldots, T$. Then, for all $B_{\mathbf{w}} > 0$ and $p > 1$, it is possible to implement the subroutines of Algorithm 4 such that:

1) The expected number of queries to each oracle A_t is

$$\frac{p}{(p-1)^2}.$$

2) The expected running time of the algorithm is

$$O\left(T^3\left(1 + \frac{dp}{(p-1)^2}\right)\right).$$

3) If we run Algorithm 4 with

$$\eta = \frac{B_{\mathbf{w}}}{\sqrt{u}\ell'_+\left(\sqrt{(p-1)u}\right)}$$

where

$$u = B_{\mathbf{w}}\left(\frac{p}{p-1}\right)^2 Q(pB_{\tilde{\mathbf{x}}})$$

then

$$\mathbb{E}\left[\sum_{t=1}^{T} \ell\left(\langle \mathbf{w}_t, \Psi(\mathbf{x}_t)\rangle, y_t\right)\right] - \min_{\mathbf{w}\,:\,\|\mathbf{w}\| \leq B_{\mathbf{w}}} \sum_{t=1}^{T} \ell\left(\langle \mathbf{w}, \Psi(\mathbf{x}_t)\rangle, y_t\right) \leq \ell'_+\left(\sqrt{(p-1)u}\right)\sqrt{uT}.$$

Algorithm 4 Kernel Learning Algorithm with Noisy Input

`Parameters`: Learning rate $\eta > 0$, number of rounds T, sample parameter $p > 1$

`Initialize`:

$\alpha_i = 0$ for all $i = 1, \ldots, T$.

$\tilde{\Psi}(\mathbf{x}_i)$ for all $i = 1, \ldots, T$

// $\tilde{\Psi}(\mathbf{x}_i)$ is a data structure which can store a

// variable number of vectors in $\mathbb{R}^d$

`For` $t = 1 \ldots T$

Define $\mathbf{w}_t = \sum_{i=1}^{t-1} \alpha_i \tilde{\Psi}(\mathbf{x}_i)$

Receive oracle A_t and y_t

Let $\tilde{\Psi}(\mathbf{x}_t) := \texttt{Map\_Estimate}(A_t, p)$

// Get unbiased estimates of $\Psi(\mathbf{x}_t)$ in the RKHS

Let $\tilde{g}_t := \texttt{Grad\_Length\_Estimate}(A_t, y_t, p)$

// Get unbiased estimate of $\ell'(y_t \langle \mathbf{w}_t, \Psi(\mathbf{x}_t)\rangle)$

Let $\alpha_t := -\tilde{g}_t \eta / \sqrt{T}$ // Perform gradient step

Let $\tilde{n}_t := \sum_{i=1}^{t} \sum_{j=1}^{t} \alpha_{t,i}\alpha_{t,j} \texttt{Prod}(\tilde{\Psi}(\mathbf{x}_i), \tilde{\Psi}(\mathbf{x}_j))$

// Compute squared norm, where

//$\texttt{Prod}(\tilde{\Psi}(\mathbf{x}_i), \tilde{\Psi}(\mathbf{x}_j))$ returns $\left\langle \tilde{\Psi}(\mathbf{x}_i), \tilde{\Psi}(\mathbf{x}_j) \right\rangle$

If $\tilde{n}_t > B_w$

Let $\alpha_i := \alpha_i \frac{\sqrt{B_w}}{\tilde{n}_t}$ for all $i = 1, \ldots, t$

//If squared norm is larger than B_w, then project

We note that the distribution of the number of oracle queries can be specified explicitly, and it decays very rapidly—see the proof for details.

The parameter p is user-defined, and allows one to perform a *tradeoff* between the number of noisy copies required for each example, and the total number of examples. In other words, the regret bound will be similar whether many noisy measurements are provided on a few examples, or just a few noisy measurements are provided on many different examples.

The result pertaining to radial kernels is very similar, and uses essentially the same techniques. For the sake of clarity, we provide a more concrete result which pertains specifically to the most important and popular radial kernel, namely the Gaussian kernel. The proof is provided in Section VIII-F.

Theorem 7: Assume that the loss function ℓ has an analytic derivative $\ell'(a) = \sum_{n=0}^{\infty} \gamma_n a^n$ for all a in its domain, and let $\ell'_+(a) = \sum_{n=0}^{\infty} |\gamma_n| a^n$ (assuming it exists). Pick any Gaussian kernel $k(\mathbf{x},\mathbf{x}') = \exp(-\|\mathbf{x} - \mathbf{x}\|^2 / s^2)$ for some $s^2 > 0$. Finally, assume that $\mathbb{E}_t\left[\|\tilde{\mathbf{x}}_t\|^2\right] \leq B_{\tilde{\mathbf{x}}}$ for any $\tilde{\mathbf{x}}_t$ returned by the oracle at round t, for all $t = 1, \ldots, T$. Then for all $B_{\mathbf{w}} > 0$ and $p > 1$ it is possible to implement the subroutines of Algorithm 4 such that

1) The expected number of queries to each oracle A_t is

$$\frac{3p}{(p-1)^2}.$$

2) The expected running time of the algorithm is

$$O\left(T^3\left(1+\frac{dp}{(p-1)^2}\right)\right).$$

3) If we run Algorithm 4 with

$$\eta = \frac{B_{\mathbf{w}}}{\sqrt{u}\ell'_+\left(\sqrt{(p-1)u}\right)}$$

where

$$u = B_{\mathbf{w}}\left(\frac{p}{p-1}\right)^3 \exp\left(\frac{\sqrt{p}B_{\tilde{\mathbf{x}}}+2p\sqrt{B_{\tilde{\mathbf{x}}}}}{s^2}\right)$$

then

$$\mathbb{E}\left[\sum_{t=1}^{T}\ell(\langle\mathbf{w}_t,\Psi(\mathbf{x}_t)\rangle,y_t)\right] - \min_{\mathbf{w}:\|\mathbf{w}\|\le B_{\mathbf{w}}}\sum_{t=1}^{T}\ell(\langle\mathbf{w},\Psi(\mathbf{x}_t)\rangle,y_t) \le \ell'_+\left(\sqrt{(p-1)u}\right)\sqrt{uT}.$$

As in Theorem 6, note that the number of oracle queries has a fast decaying distribution. Also, note that with Gaussian kernels, s^2 is usually chosen to be on the order of the example's squared norms. Thus, if the noise added to the examples is proportional to their original norm, we can assume that $B_{\tilde{\mathbf{x}}}/s^2 = O(1)$, and thus u appearing in the bound is also bounded by a constant.

As previously mentioned, most of the subroutines are described in the proofs section, as part of the proof of Theorem 6. Here, we only show how to implement the `Grad_Length_Estimate` subroutine, which returns the gradient length estimate $\tilde{g}_t$. The idea is based on the technique described in Section III-C. We prove that $\tilde{g}_t$ is an unbiased estimate of $\ell'(y_t\langle\mathbf{w}_t,\Psi(\mathbf{x}_t)\rangle)$, and bound $\mathbb{E}_t[\tilde{g}_t^2]$. As discussed earlier, we assume that $\ell'(\cdot)$ is analytic and can be written as $\ell'(a)=\sum_{n=0}^{\infty}\gamma_n a^n$.

Subroutine 1 `Grad_Length_Estimate`(A_t, yt, p)

Sample nonnegative integer n according to $\mathbb{P}(n)=\frac{p-1}{p^{n+1}}$

For$j=1,\ldots,n$

Let $\tilde{\Psi}(\mathbf{x}_t)_j :=$ `Map_Estimate`(A_t)

// Get unbiased estimate of $\Psi(\mathbf{x}_t)$ in the RKHS

Return

$$y_t\gamma_n\frac{p^{n+1}}{p-1}\prod_{j=1}^{n}\left(\sum_{i=1}^{t-1}\alpha_{t-1,i}\texttt{Prod}(\tilde{\Psi}(\mathbf{x}_i),\tilde{\Psi}(\mathbf{x}_t)_j)\right).$$

Lemma 4: Assume that $\mathbb{E}_t[\tilde{\Psi}(\mathbf{x}_t)] = \Psi(\mathbf{x}_t)$, and that $\texttt{Prod}(\tilde{\Psi}(\mathbf{x}),\tilde{\Psi}(\mathbf{x}'))$ returns $\left\langle\tilde{\Psi}(\mathbf{x}),\tilde{\Psi}(\mathbf{x}')\right\rangle$ for all $\mathbf{x}$, $\mathbf{x}'$. Denote the output of the subroutine above as $\tilde{g}_t$, and define $\ell'_+(a)=\sum_{n=0}^{\infty}|\gamma_n|a^n$. Then for any given

$$\mathbf{w}_t = \alpha_{t-1,1}\tilde{\Psi}(\mathbf{x}_1)+\cdots+\alpha_{t-1,t-1}\tilde{\Psi}(\mathbf{x}_{t-1})$$

it holds that $\mathbb{E}_t[\tilde{g}_t] = y_t\ell'(y_t\langle\mathbf{w}_t,\Psi(\mathbf{x}_t)\rangle)$ and

$$\mathbb{E}_t[\tilde{g}_t^2] \le \frac{p}{p-1}\ell'_+\left(\sqrt{pB_{\mathbf{w}}B_{\tilde{\Psi}(\mathbf{x})}}\right)^2$$

where the expectation is with respect to the randomness of Subroutine 1.

Proof: The result follows from Lemma 1, where $\tilde{g}_t$ corresponds to the estimator θ, the function f corresponds to ℓ', and the random variable X corresponds to $\left\langle\mathbf{w}_t,\tilde{\Psi}(\mathbf{x}_t)\right\rangle$ (where $\tilde{\Psi}(\mathbf{x}_t)$ is random and $\mathbf{w}_t$ is held fixed). The term $\mathbb{E}[X^2]$ in Lemma 1 can be upper bounded as

$$\mathbb{E}_t\left[\left(\left\langle\mathbf{w}_t,\tilde{\Psi}(\mathbf{x}_t)\right\rangle\right)^2\right] \le \|\mathbf{w}_t\|^2\,\mathbb{E}_t\left[\left\|\tilde{\Psi}(\mathbf{x}_t)\right\|^2\right] \le B_{\mathbf{w}}B_{\tilde{\Psi}(\mathbf{x})}.$$

■

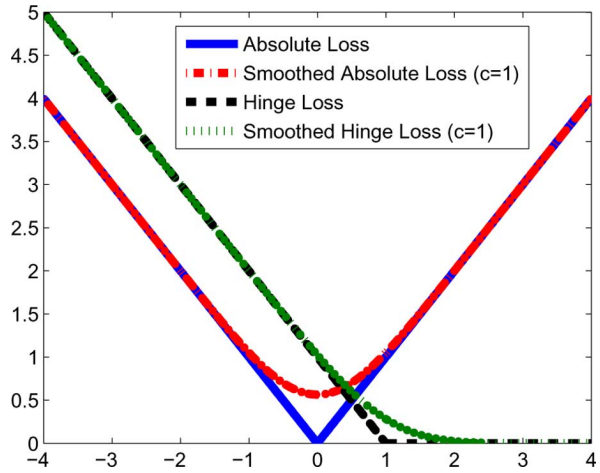

Fig. 2. Absolute loss, hinge loss, and analytic approximations. For the absolute loss, the line represents the loss as a function of $\langle\mathbf{w},\Psi(\mathbf{x})\rangle - y$. For the hinge loss, the lines represent the loss as a function of $y\langle\mathbf{w},\Psi(\mathbf{x})\rangle$.

B. Loss Function Examples

Theorems 6 and 7 both deal with generic loss functions ℓ whose derivative can be written as $\sum_{n=0}^{\infty}\gamma_n a^n$, and the regret bounds involve the functions $\ell'_+(a)=\sum_{n=0}^{\infty}|\gamma_n|a^n$. Below, we present a few examples of loss functions and their corresponding ℓ'_+. As mentioned earlier, while the theorems in the previous subsection are in terms of classification losses (i.e., ℓ is a function of $y\langle\mathbf{w},\Psi(\mathbf{x})\rangle$), virtually identical results can be proven for regression losses (i.e., ℓ is a function of $\langle\mathbf{w},\Psi(\mathbf{x})\rangle - y$), so we will give examples from both families. Working out the first two examples is straightforward. The proofs of the other two appear in Section VIII-G. The loss functions in the last two examples are illustrated graphically in Fig. 2.

Example 1: For the squared loss function, $\ell(\langle\mathbf{w},\Psi(\mathbf{x})\rangle, y) = (\langle\mathbf{w},\Psi(\mathbf{x})\rangle - y)^2$, we have $\ell'_+\left(\sqrt{(p-1)u}\right) = 2\sqrt{(p-1)u}$.

Example 2: For the exponential loss function,

$$\ell(\langle \mathbf{w}, \Psi(\mathbf{x})\rangle, y) = e^{y\langle \mathbf{w}, \Psi(\mathbf{x})\rangle},$$

we have $\ell'_+\left(\sqrt{(p-1)u}\right) = e^{\sqrt{(p-1)u}}$.

Example 3: Recall that the standard absolute loss is defined as $\ell(\langle \mathbf{w}, \Psi(\mathbf{x})\rangle, y) = |\langle \mathbf{w}, \Psi(\mathbf{x})\rangle - y|$. Consider a "smoothed" absolute loss function $\ell_c(\langle \mathbf{w}, \Psi(\mathbf{x})\rangle, y)$, defined as an antiderivative of $\mathrm{Erf}(sa)$ for some $c > 0$ (see proof for exact analytic form). Then we have that

$$\ell'_+\left(\sqrt{(p-1)u}\right) \le \frac{1}{2} + \frac{1}{c\sqrt{\pi(p-1)u}}\left(e^{c^2(p-1)u} - 1\right).$$

Example 4: Recall that the standard hinge loss is defined as $\ell(\langle \mathbf{w}, \Psi(\mathbf{x})\rangle, y) = \max\{1 - y\langle \mathbf{w}, \Psi(\mathbf{x})\rangle, 0\}$. Consider a "smoothed" hinge loss $\ell_c(y\langle \mathbf{w}, \Psi(\mathbf{x})\rangle)$, defined as an antiderivative of $(\mathrm{Erf}(c(a-1)) - 1)/2$ for some $c > 0$ (see proof for exact analytic form). Then we have that

$$\ell'_+\left(\sqrt{(p-1)u}\right) \le \frac{2}{c\sqrt{\pi(p-1)u}}\left(e^{c^2(p-1)u-1}\right).$$

For any c, the loss function in the last two examples is convex, and, respectively, approximate the absolute loss $|\langle \mathbf{w}, \Psi(\mathbf{x})\rangle - y|$ and the hinge loss $\max\{0, 1 - y\langle \mathbf{w}, \Psi(\mathbf{x})\rangle\}$ arbitrarily well for large enough c. Fig. 2 shows these loss functions graphically for $c = 1$. Note that c need not be large in order to get a good approximation. Also, we note that both the loss itself and its gradient are computationally easy to evaluate.

Finally, we remind the reader that as discussed in Section III-C, performing an unbiased estimate of the gradient for nondifferentiable losses directly (such as the hinge loss or absolute loss) appears to be impossible in general. On the flip side, if one is willing to use a random number of queries with polynomially decaying rather than exponentially-decaying tails, then one can achieve much better sample complexity results, by focusing on loss functions (or approximations thereof) which are only differentiable to a bounded order, rather than fully analytic. This again demonstrates the tradeoff between the number of examples, and the amount of information that needs to be gathered on each example.

VI. Are Multiple Noisy Copies Necessary?

The positive results discussed so far are mostly based on getting more than one noisy copy per example. However, one might wonder if this is really necessary. In some applications this is inconvenient, and one would prefer a method which works when just a single noisy copy of each example is made available. Moreover, in the setting of known noise covariance (Section IV-B), for linear predictors and squared loss, we needed just one noisy copy of each example $(\mathbf{x}_t, y_t)$ in order to learn. Perhaps a similar result can be obtained even when the noise distribution in unknown?

In this subsection we show that, unfortunately, such a method cannot be found. Specifically, we prove that if the noise distribution is unknown, then under very mild assumptions, no method can achieve sublinear regret, when it has access to just a single noisy copy of each instance $\mathbf{x}_t$ (even when y_t is known). On the other hand, for the case of squared loss and linear kernels, we know that we can learn based on two noisy copies of each instance (see Section IV-A). So without further assumptions, the lower bound that we prove here is indeed tight. It is an interesting open problem to show improved lower bounds when nonlinear kernels are used, or when the loss function is more complex.

Theorem 8: Let $\mathcal{W}$ be a compact convex subset of $\mathbb{R}^d$, and let $\ell(\cdot, 1) : \mathbb{R} \mapsto \mathbb{R}$ satisfies the following: (1) it is bounded from below; (2) it is differentiable at 0 with $\ell'(0, 1) < 0$. For any learning algorithm which selects hypotheses from $\mathcal{W}$ and is allowed access to a single noisy copy of the instance at each round t, there exists a strategy for the adversary such that the sequence $\mathbf{w}_1, \mathbf{w}_2, \ldots$ of predictors output by the algorithm satisfies

$$\limsup_{T\to\infty} \max_{\mathbf{w}\in\mathcal{W}} \frac{1}{T}\sum_{t=1}^{T}\left(\ell(\langle \mathbf{w}_t, \mathbf{x}_t\rangle, y_t) - \ell(\langle \mathbf{w}, \mathbf{x}_t\rangle, y_t)\right) > 0$$

with probability 1.

Note that condition (1) is satisfied by virtually any loss function other than the linear loss, while condition (2) is satisfied by most regression losses, and by all classification calibrated losses, which include all reasonable losses for classification (see [23]).

The intuition of the proof is very simple: the adversary chooses beforehand whether the examples are drawn i.i.d. from a distribution $\mathcal{D}$, and then perturbed by noise, or drawn i.i.d. from some other distribution $\mathcal{D}'$ without adding noise. The distributions $\mathcal{D}$, $\mathcal{D}'$ and the noise are designed so that the examples observed by the learner are distributed in the same way irrespective of which of the two sampling strategies the adversary chooses. Therefore, it is impossible for the learner accessing a single copy of each instance to be statistically consistent with respect to both distributions simultaneously. As a result, the adversary can always choose a distribution on which the algorithm will be inconsistent, leading to constant regret.

To prove the theorem, we use a more general result which leads to nonvanishing regret, and then show that under the assumptions of Theorem 8, the result holds. The proof of the result is given in Section VIII-I.

Theorem 9: Let $\mathcal{W}$ be a compact convex subset of $\mathbb{R}^d$ and pick any learning algorithm which selects hypotheses from $\mathcal{W}$ and is allowed access to a single noisy copy of the instance at each round t. If there exists a distribution over a compact subset of $\mathbb{R}^d$ such that

$$\operatorname*{argmin}_{\mathbf{w}\in\mathcal{W}} \mathbb{E}\left[\ell(\langle \mathbf{w}, \mathbf{x}\rangle, 1)\right] \text{ and } \operatorname*{argmin}_{\mathbf{w}\in\mathcal{W}} \ell\left(\langle \mathbf{w}, \mathbb{E}[\mathbf{x}]\rangle, 1\right) \quad (9)$$

are disjoint, then there exists a strategy for the adversary such that the sequence $\mathbf{w}_1, \mathbf{w}_2, \ldots \in \mathcal{W}$ of predictors output by the algorithm satisfies

$$\limsup_{T\to\infty} \max_{\mathbf{w}\in\mathcal{W}} \frac{1}{T}\sum_{t=1}^{T}\Big(\ell(\langle \mathbf{w}_t, \mathbf{x}_t\rangle, y_t) - \ell(\langle \mathbf{w}, \mathbf{x}_t\rangle, y_t)\Big) > 0$$

with probability 1.

Another way to phrase this theorem is that the regret cannot vanish, if given examples sampled i.i.d. from a distribution, the learning problem is more complicated than just finding the mean of the data. Indeed, the adversary's strategy we choose later on is simply drawing and presenting examples from such a distribution. Below, we sketch how we use Theorem 9 in order to prove Theorem 8. A full proof is provided in Section VIII-H.

We construct a very simple one-dimensional distribution, which satisfies the conditions of Theorem 9: it is simply the uniform distribution on $\{3\mathbf{x}, -\mathbf{x}\}$, where $\mathbf{x}$ is the vector $(1, 0, \ldots, 0)$. Thus, it is enough to show that

$$\operatorname*{argmin}_{w\,:\,|w|^2\le B_{\mathbf{w}}} \ell(3w,1)+\ell(-w,1) \quad \text{and} \quad \operatorname*{argmin}_{w\,:\,|w|^2\le B_{\mathbf{w}}} \ell(w,1) \tag{10}$$

are disjoint, for some appropriately chosen $B_{\mathbf{w}}$. Assuming the contrary, then under the assumptions on ℓ, we show that the first set in (10) is inside a bounded ball around the origin, in a way which is independent of $B_{\mathbf{w}}$, no matter how large it is. Thus, if we pick $B_{\mathbf{w}}$ to be large enough, and assume that the two sets in (10) are not disjoint, then there must be some w such that both $\ell(3w,1)+\ell(-w,1)$ and $\ell(w,1)$ have a subgradient of zero at w. However, this can be shown to contradict the assumptions on ℓ, leading to the desired result.

VII. Conclusions and Future Work

We have investigated the problem of learning, in an online fashion, linear and kernel-based predictors when the observed examples are corrupted by noise. We have shown bounds on the expected regret of learning algorithms under various assumptions on the noise distribution and the loss function (squared loss, analytic losses). A key ingredient of our results is the derivation of unbiased estimates for the loss gradients based on the possibility of obtaining a small but random number of independent copies of each noisy example. We also show that accessing more than one copy of each noisy example is a necessary condition for learning with sublinear regret.

There are several interesting research directions worth pursuing in the noisy learning framework introduced here. For instance, doing away with unbiasedness, which could lead to the design of estimators that are applicable to more types of loss functions, for which unbiased estimators may not even exist. Biased estimates may also help in designing improved estimates for kernel learning when the noise distribution is known, but not necessarily Gaussian. Another open question is whether our lower bound (Theorem 8) can be improved when nonlinear kernels are used.

VIII. Proofs

A. Proof of Theorem 1

First, we use the following lemma that can be easily adapted from [11].

Lemma 5: Let $\mathbf{v}_1, \ldots, \mathbf{v}_T$ be a sequence of vectors. Let $\mathbf{w}_1 = \mathbf{0}$ and for $t \ge 1$ let $\mathbf{w}_t = P(\mathbf{w}_t - \eta \mathbf{v}_t)$, where $P(\cdot)$ is the projection operator on an origin-centered ball of radius $B_{\mathbf{w}}$. Then, for all $\mathbf{u}$ such that $\|\mathbf{u}\| \le B_{\mathbf{w}}$ we have

$$\sum_{t=1}^{m} \langle \mathbf{v}_t, \mathbf{w}_t - \mathbf{u}\rangle \le \frac{B_{\mathbf{w}}^2}{2\eta} + \frac{\eta \sum_{t=1}^{T} \|\mathbf{v}_t\|^2}{2}.$$

Applying Lemma 5 with $\mathbf{v}_t = \tilde{\nabla}_t$ as defined in Lemma 2, we obtain

$$\sum_{t=1}^{T} \left\langle \tilde{\nabla}_t, \mathbf{w}_t - \mathbf{u}\right\rangle \le \frac{B_{\mathbf{w}}^2}{2\eta} + \frac{\eta \sum_{t=1}^{T} \|\tilde{\nabla}_t\|^2}{2}.$$

Taking expectation of both sides and using again Lemma 2, we obtain that

$$\mathbb{E}\left[\sum_{t=1}^{T} \langle \nabla_t, \mathbf{w}_t - \mathbf{u}\rangle\right] \le \frac{B_{\mathbf{w}}^2}{2\eta} + \frac{\eta T G}{2}.$$

Now, using convexity we get that

$$\langle \nabla_t, \mathbf{w}_t - \mathbf{u}\rangle \ge (\langle \mathbf{w}_t, \mathbf{x}_t\rangle - y_t)^2 - (\langle \mathbf{w}_t, \mathbf{u}\rangle - y_t)^2$$

which gives

$$\mathbb{E}\left[\sum_{t=1}^{T} (\langle \mathbf{w}_t, \mathbf{x}_t\rangle - y_t)^2\right] \le \sum_{t=1}^{T} (\langle \mathbf{u}, \mathbf{x}_t\rangle - y_t)^2 + \frac{B_{\mathbf{w}}^2}{2\eta} + \frac{\eta T G}{2}.$$

Picking η as in the theorem statement concludes our proof.

B. Proof of Theorem 3

To prove the theorem, we will need a few auxiliary lemmas. In particular, Lemma 6 is a key technical lemma, which will prove crucial in connecting the RKHS with respect to $\Psi(\cdot)$, $k(\cdot,\cdot)$, and the RKHS with respect to $\hat{\Psi}(\cdot)$, $\hat{k}(\cdot,\cdot)$. Lemma 8 connects between the norms of elements in the two RKHS's.

To state the lemmas and proofs conveniently, recall the shorthand

$$R_{\Sigma,s,d} = \left(\prod_{i=1}^{d} \frac{s^2}{s^2 - 2\sigma_i^2}\right)^{1/4}.$$

Lemma 6: For any $\mathbf{a}, \mathbf{x} \in \mathbb{R}^d$, if we let $\tilde{\mathbf{x}} = \mathbf{x} + \mathbf{n}$ where $\mathbf{n} \sim \mathcal{N}(0, \Sigma)$ is a Gaussian random vector with covariance matrix Σ, then it holds that

$$\mathbb{E}_{\mathbf{n}} \left\langle \hat{\Psi}(\mathbf{a}), \hat{\Psi}(\tilde{\mathbf{x}})\right\rangle = \langle \Psi(\mathbf{a}), \Psi(\mathbf{x})\rangle.$$

Proof: The expectation in the lemma can be written as

$$\mathbb{E}_{\mathbf{n}}\left[R_{\Sigma,s,d}^2 \exp\left(-\sum_{i=1}^{d} \frac{(a_i - x_i - n_i)^2}{s^2 - 2\sigma_i^2}\right)\right] = \prod_{i=1}^{d} R_{\Sigma,s,d}^{2/d} \int_{z_i} (2\pi s^2)^{-1/2} \exp\left(-\frac{z_i^2}{2\sigma_i^2} - \frac{(a_i - x_i - z_i)^2}{s^2 - 2\sigma_i^2}\right) dz_i. \tag{11}$$

A purely technical integration exercise reveals that each element i in this product equals $\exp(-(a_i - x_i)^2/s^2)$. Therefore, (11) equals

$$\prod_{i=1}^{d} \exp\left(-\frac{(a_i - x_i)^2}{s^2}\right) = \exp\left(-\frac{\|\mathbf{a} - \mathbf{x}\|^2}{s^2}\right)$$

which is exactly $\langle \Psi(\mathbf{x}), \Psi(\mathbf{a}) \rangle$. ∎

Lemma 7: Let $\Psi(\cdot)$ denote a feature mapping to an arbitrary RKHS. Let $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_T$ be vectors in $\mathbb{R}^d$, and $a_1, a_2, \ldots, a_T$ scalars, such that $\|\sum_i a_i \Psi(\mathbf{z}_i)\|^2 \le B^2$ for some $B > 0$. Then it holds that

$$\left\| \sum_{i:a_i>0} a_i \Psi(\mathbf{z}_i) \right\|^2 + \left\| \sum_{i:a_i<0} a_i \Psi(\mathbf{z}_i) \right\|^2 \le \frac{5B^2}{\sin^2(\beta)}$$

where β is the angle between $\sum_{i:a_i>0} a_i \Psi(\mathbf{z}_i)$ and $\sum_{i:a_i<0} a_i \Psi(\mathbf{z}_i)$ in the RKHS (or $\beta = \pi/2$ if one of these elements is zero).

We remark that this bound is designed for readability—it is not the tightest upper bound possible.

Proof: The bound trivially holds if $\sum_{i:a_i>0} a_i \Psi(\mathbf{z}_i)$ or $\sum_{i:a_i<0} a_i \Psi(\mathbf{z}_i)$ are zero, so we will assume w.l.o.g. that they are both nonzero.

To simplify notation, let

$$\mathbf{w} = \sum_i a_i \Psi(\mathbf{z}_i)$$
$$\mathbf{w}_+ = \sum_{i:a_i>0} a_i \Psi(\mathbf{z}_i)$$
$$\mathbf{w}_- = \sum_{i:a_i<0} -a_i \Psi(\mathbf{z}_i)$$

and notice that $\mathbf{w} = \mathbf{w}_+ - \mathbf{w}_-$. By the cosine theorem and the fact that $\mathbf{w} = \mathbf{w}_+ - \mathbf{w}_-$, we have that

$$\|\mathbf{w}\|^2 = \|\mathbf{w}_+\|^2 + \|\mathbf{w}_-\|^2 - 2\|\mathbf{w}_+\|\,\|\mathbf{w}_-\|\cos(\beta).$$

Solving for $\|\mathbf{w}_-\|$ and taking the larger root in the resulting quadratic equation, we have that

$$\|\mathbf{w}_-\| \le \|\mathbf{w}_+\|\cos(\beta) + \sqrt{\|\mathbf{w}\|^2 - \|\mathbf{w}_+\|^2 \sin^2(\beta)} \quad (12)$$

(it is easy to verify that the term in the squared root is always nonnegative). Therefore

$$\begin{aligned}
&\|\mathbf{w}_+\|^2 + \|\mathbf{w}_-\|^2 \\
&\le \|\mathbf{w}_+\|^2 + \left(\|\mathbf{w}_+\|\cos(\beta) + \sqrt{\|\mathbf{w}\|^2 - \|\mathbf{w}_+\|^2 \sin^2(\beta)}\right)^2 \\
&\le \|\mathbf{w}_+\|^2 + \left(\|\mathbf{w}_+\|\,|\cos(\beta)| + \|\mathbf{w}\|\right)^2.
\end{aligned}$$

From straightforward geometric arguments, we must have $\|\mathbf{w}_+\|^2 \le \|\mathbf{w}\|^2 / \sin^2(\beta)$ (this is the same reason the term in the squared root in (12) is nonnegative). Plugging this into the right-hand side (RHS) of the inequality above, we get an upper bound of the form

$$\begin{aligned}
&\frac{\|\mathbf{w}\|^2}{\sin^2(\beta)} + \left(\|\mathbf{w}\|\left|\frac{\cos(\beta)}{\sin(\beta)}\right| + \|\mathbf{w}\|\right)^2 \\
&\qquad \le \|\mathbf{w}\|^2 \left(1 + \frac{2}{|\sin(\beta)|} + \frac{2}{\sin^2(\beta)}\right)
\end{aligned}$$

where we used the fact that $|\cos(\beta)| \le 1$. A straightforward upper bounding leads to the lemma statement. ∎

The following lemma is basically a corollary of Lemma 7.

Lemma 8: Let $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_T$ be vectors in $\mathbb{R}^d$, and $a_1, a_2, \ldots, a_T$ scalars, such that $\|\sum_i a_i \Psi(\mathbf{z}_i)\|^2 \le B^2$. Then $\sum_i a_i \hat{\Psi}(z_i)$ is an element in the RKHS with respect to $\hat{\Psi}(\cdot)$, whose norm squared is at most

$$\frac{5B^2}{\sin^2(\beta)} R^2_{\Sigma,s,d}.$$

Here, β is the angle between $\sum_{i:a_i>0} a_i \Psi(\mathbf{z}_i)$ and $-\sum_{i:a_i<0} a_i \Psi(\mathbf{z}_i)$ in the RKHS (or $\beta = \pi/2$ if one of the elements is zero).

Proof: Picking some $\mathbf{z}_1, \mathbf{z}_2, \ldots$ and $a_1, a_2, \ldots$ as in the lemma statement, we have

$$\left\| \sum_i a_i \hat{\Psi}(z_i) \right\|^2 = \sum_{i,j} a_i a_j \hat{k}(\mathbf{z}_i, \mathbf{z}_j) \le \sum_{i,j:a_i a_j>0} a_i a_j \hat{k}(\mathbf{z}_i, \mathbf{z}_j) \quad (13)$$

where the last transition is by the fact that $\hat{k}$ is always positive. Now, by definition of $k(\cdot,\cdot)$, $\hat{k}(\cdot,\cdot)$, it holds for any $\mathbf{z}_i, \mathbf{z}_j$ that

$$\frac{\hat{k}(\mathbf{z}_i, \mathbf{z}_j)}{k(\mathbf{z}_i, \mathbf{z}_j)} = R^2_{\Sigma,s,d} \exp\left(\sum_{l=1}^{d} \frac{(z_{i,l} - z_{j,l})^2}{s^2} - \frac{(z_{i,l} - z_{j,l})^2}{s^2 - 2\sigma_l^2}\right)$$

which is at most $R^2_{\Sigma,s,d}$. Therefore, we can upper bound (13) by

$$\sum_{i,j\,:\,a_i a_j>0} a_i a_j \hat{k}(\mathbf{z}_i, \mathbf{z}_j) \le R^2_{\Sigma,s,d} \sum_{i,j\,:\,a_i a_j>0} a_i a_j k(\mathbf{z}_i, \mathbf{z}_j).$$

The lemma follows by noting that according to Lemma 7

$$\begin{aligned}
&\sum_{i,j\,:\,a_i a_j>0} a_i a_j k(\mathbf{z}_i, \mathbf{z}_j) \\
&\qquad = \left\| \sum_{i\,:\,a_i>0} a_i \Psi(\mathbf{z}_i) \right\|^2 + \left\| \sum_{i\,:\,a_i<0} a_i \Psi(\mathbf{z}_i) \right\|^2 \\
&\qquad \le \frac{5B^2}{\sin^2(\beta)}.
\end{aligned}$$

∎

With these lemmas in hand, we are now ready to prove the main theorem.

To make the proof clearer, let $\alpha_{t,i}$ denote the value of α_i in algorithm 3 at the beginning of round t.

The first step of the proof consists of applying Lemma 5, since our algorithm follows the protocol outlined in that lemma, using kernels. We therefore have that for any $\hat{\mathbf{u}}$ in the RKHS corresponding to $\hat{\Psi}(\cdot)$, such that $\|\hat{\mathbf{u}}\|^2 \leq W^2$, it holds that

$$\sum_{t=1}^{T}\left\langle \hat{\mathbf{w}}_t, \tilde{g}_t\hat{\Psi}(\tilde{\mathbf{x}}_t)\right\rangle - \sum_{t=1}^{T}\left\langle \hat{\mathbf{u}}, \tilde{g}_t\hat{\Psi}(\tilde{\mathbf{x}}_t)\right\rangle \leq \frac{W^2}{2\eta} + \frac{\eta\sum_{t=1}^{T}\left\|\tilde{g}_t\hat{\Psi}(\tilde{\mathbf{x}}_t)\right\|^2}{2}. \quad (14)$$

In particular, consider $\mathbf{u} = \sum_{i=1}^{T} a_t\Psi(\mathbf{x}_t)$ from the theorem's statement, and define

$$\hat{\mathbf{u}}^* = \sum_{i=1}^{T} a_t\hat{\Psi}(\mathbf{x}_t).$$

This is an element in the RKHS corresponding to $\hat{\Psi}(\cdot)$, but it shares the same set of weights as $\mathbf{u}$, which is an element in the RKHS corresponding to $\Psi(\cdot)$. Since $\|\mathbf{u}\|^2 \leq B_{\mathbf{w}}^2$, it follows from Lemma 8 and the definition of W that $\|\hat{\mathbf{u}}^*\|^2 \leq W^2$. Therefore, (14) applies, and we get

$$\sum_{t=1}^{T}\left\langle \hat{\mathbf{w}}_t, \tilde{g}_t\hat{\Psi}(\tilde{\mathbf{x}}_t)\right\rangle - \sum_{t=1}^{T}\left\langle \hat{\mathbf{u}}^*, \tilde{g}_t\hat{\Psi}(\tilde{\mathbf{x}}_t)\right\rangle \leq \frac{W^2}{2\eta} + \frac{\eta\sum_{t=1}^{T}\left\|\tilde{g}_t\hat{\Psi}(\tilde{\mathbf{x}}_t)\right\|^2}{2}.$$

This inequality holds for any $\{\tilde{\mathbf{x}}_t, \tilde{\mathbf{x}}'_t, \tilde{y}_t\}_{t=1}^{T}$. In particular, it will remain valid if we take expectations of both sides with respect to the Gaussian noise injected into the unperturbed data

$$\mathbb{E}\left[\sum_{t=1}^{T}\left\langle \hat{\mathbf{w}}_t, \tilde{g}_t\hat{\Psi}(\tilde{\mathbf{x}}_t)\right\rangle - \sum_{t=1}^{T}\left\langle \hat{\mathbf{u}}^*, \tilde{g}_t\hat{\Psi}(\tilde{\mathbf{x}}_t)\right\rangle\right] \leq \frac{W^2}{2\eta} + \frac{1}{2}E\left[\eta\sum_{t=1}^{T}\mathbb{E}_t\left[\left\|\tilde{g}_t\hat{\Psi}(\tilde{\mathbf{x}}_t)\right\|^2\right]\right]. \quad (15)$$

Starting with the RHS, we note that by definition of $\tilde{g}_t$ from the algorithm's pseudocode, and the fact that $\left\|\hat{\Psi}(\tilde{\mathbf{x}}_t)\right\|^2 = \hat{k}(\tilde{\mathbf{x}},\tilde{\mathbf{x}}) \leq R_{\Sigma,s,d}^2$ by definition of the kernel $\hat{k}$ in (5)

$$\begin{aligned}
\mathbb{E}_t\left[\left\|\tilde{g}_t\hat{\Psi}(\tilde{\mathbf{x}}'_t)\right\|^2\right] &= 4\mathbb{E}_t\left[\left\|\left(\left\langle \hat{\mathbf{w}}_t, \hat{\Psi}(\tilde{\mathbf{x}}_t)\right\rangle - \tilde{y}_t\right)\hat{\Psi}(\tilde{\mathbf{x}}'_t)\right\|^2\right] \\
&= 4\mathbb{E}_t\left[\left\|\hat{\Psi}(\tilde{\mathbf{x}}'_t)\right\|^2\right]\mathbb{E}_t\left[\left(\left\langle \hat{\mathbf{w}}_t, \hat{\Psi}(\tilde{\mathbf{x}}_t)\right\rangle - \tilde{y}_t\right)^2\right] \\
&= 4\mathbb{E}_t\left[\left\|\hat{\Psi}(\tilde{\mathbf{x}}'_t)\right\|^2\right]\mathbb{E}_t\left[\left(\left\langle \hat{\mathbf{w}}_t, \hat{\Psi}(\tilde{\mathbf{x}}_t)\right\rangle\right)^2 + \tilde{y}_t^2\right] \\
&\leq 4\mathbb{E}_t\left[\left\|\hat{\Psi}(\tilde{\mathbf{x}}'_t)\right\|^2\right]\left(\|\hat{\mathbf{w}}_t\|^2\,\mathbb{E}_t\left[\left\|\hat{\Psi}(\tilde{\mathbf{x}}_t)\right\|^2\right] + \mathbb{E}_t[\tilde{y}_t^2]\right) \\
&= 4R_{\Sigma,s,d}^2(W^2R_{\Sigma,s,d}^2 + B_{\tilde{y}}^2).
\end{aligned}$$

Plugging this back into (15), and choosing η as in the theorem's statement, we finally get

$$\mathbb{E}\left[\sum_{t=1}^{T}\left\langle \hat{\mathbf{w}}_t, \tilde{g}_t\hat{\Psi}(\tilde{\mathbf{x}}_t)\right\rangle - \sum_{t=1}^{T}\left\langle \hat{\mathbf{u}}^*, \tilde{g}_t\hat{\Psi}(\tilde{\mathbf{x}}_t)\right\rangle\right] \leq 2WR_{\Sigma,s,d}\sqrt{(W^2R_{\Sigma,s,d}^2 + B_{\tilde{y}}^2)T}. \quad (16)$$

We now turn to analyze the more interesting LHS of (16). The LHS of (16) can be written as

$$\mathbb{E}\left[\sum_{t=1}^{T}\mathbb{E}_t\left[\left\langle \hat{\mathbf{w}}_t, \tilde{g}_t\hat{\Psi}(\tilde{\mathbf{x}}_t)\right\rangle\right] - \sum_{t=1}^{T}\mathbb{E}_t\left[\left\langle \hat{\mathbf{u}}^*, \tilde{g}_t\hat{\Psi}(\tilde{\mathbf{x}}_t)\right\rangle\right]\right]. \quad (17)$$

In order to analyze the first sum inside the expectation, recall that $\hat{\mathbf{w}}_t$ can be written as $\sum_{i=1}^{t-1}\alpha_{t,i}\hat{\Psi}(\tilde{\mathbf{x}}_i)$. Therefore, we have that

$$\begin{aligned}
\mathbb{E}_t\left[\left\langle \hat{\mathbf{w}}_t, \tilde{g}_t\hat{\Psi}(\tilde{\mathbf{x}}_t)\right\rangle\right] &= \sum_{i=1}^{t-1}\mathbb{E}_t\left[\alpha_{t,i}\left\langle \hat{\Psi}(\tilde{\mathbf{x}}_i), \tilde{g}_t\hat{\Psi}(\tilde{\mathbf{x}}_t)\right\rangle\right] \\
&= \sum_{i=1}^{t-1}\mathbb{E}_t[\tilde{g}_t]\mathbb{E}_t\left[\alpha_{t,i}\left\langle \hat{\Psi}(\tilde{\mathbf{x}}_i), \hat{\Psi}(\tilde{\mathbf{x}}_t)\right\rangle\right]
\end{aligned}$$

where the last transition is by the fact that $\tilde{\mathbf{x}}_t$, $\tilde{\mathbf{x}}'_t$, $\tilde{y}_t$ are mutually independent, and therefore $\tilde{g}_t$ is independent of $\hat{\Psi}(\tilde{\mathbf{x}}_t)$ conditioned on $\{(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}'_i, \tilde{y}_i)\}_{i=1}^{t-1}$.

We now make two crucial observations, which are really the heart of our proof: First, by Lemma 6, we have that

$$\begin{aligned}
\sum_{i=1}^{t-1}\mathbb{E}_t\left[\alpha_{t,i}\left\langle \hat{\Psi}(\tilde{\mathbf{x}}_i), \hat{\Psi}(\tilde{\mathbf{x}}_t)\right\rangle\right] &= \sum_{i=1}^{t-1}\alpha_{t,i}\left\langle \Psi(\tilde{\mathbf{x}}_i), \Psi(\mathbf{x}_t)\right\rangle \\
&= \left\langle \mathbf{w}_t, \Psi(\mathbf{x}_t)\right\rangle.
\end{aligned}$$

Second, using Lemma 6 in a similar manner, we also have

$$\begin{aligned}
\mathbb{E}_t[\tilde{g}_t] &= 2\left(\sum_{i=1}^{t-1}\mathbb{E}_t\left[\alpha_{t,i}\left\langle \hat{\Psi}(\tilde{\mathbf{x}}_i), \hat{\Psi}(\mathbf{x}_t)\right\rangle - \tilde{y}_t\right]\right) \\
&= 2(\left\langle \mathbf{w}_t, \Psi(\mathbf{x}_t)\right\rangle - y_t).
\end{aligned}$$

Define this expression as g_t. Notice that it is exactly the gradient of $\mathbf{w}_t$ with respect to $\ell(\left\langle \mathbf{w}_t, \Psi(\mathbf{x}_t)\right\rangle, y_t)$.

As a result of these two observations, we get overall that

$$\begin{aligned}
\mathbb{E}_t\left[\left\langle \hat{\mathbf{w}}_t, \tilde{g}_t\hat{\Psi}(\tilde{\mathbf{x}}_t)\right\rangle\right] &= \sum_{i=1}^{t-1}\alpha_{t,i}\left\langle \Psi(\tilde{\mathbf{x}}_i), g_t\Psi(\mathbf{x}_t)\right\rangle \\
&= \left\langle \mathbf{w}_t, g_t\Psi(\mathbf{x}_t)\right\rangle. \quad (18)
\end{aligned}$$

Moving to the second sum in the LHS of (17), recall that there exist some $a_1, \ldots, a_T$ such that $\hat{\mathbf{u}}^* = \sum_{t=1}^{T} a_t\hat{\Psi}(\mathbf{x}_t)$.

Therefore

$$\mathbb{E}_t\left[\left\langle \hat{\mathbf{u}}^*, \tilde{g}_t\hat{\Psi}(\tilde{\mathbf{x}}_t)\right\rangle\right] = \sum_{i=1}^{T}\mathbb{E}_t\left[a_i\left\langle \hat{\Psi}(\mathbf{x}_i), \tilde{g}_t\hat{\Psi}(\tilde{\mathbf{x}}_t)\right\rangle\right].$$

As before, we have by Lemma 6 that $\mathbb{E}_t\left[\left\langle \hat{\Psi}(\mathbf{x}_i), \hat{\Psi}(\tilde{\mathbf{x}}_t)\right\rangle\right] = \langle\Psi(\mathbf{x}_i), \Psi(\mathbf{x}_t)\rangle$, and that $\tilde{g}_t$ is conditionally independent with

expected value $g_t = 2(\sum_{i=1}^{t-1} \alpha_{t,i} k(\tilde{\mathbf{x}}_i, \mathbf{x}_t) - y_t)$. Substituting this into the expression above, we get that it is equal to

$$\sum_{i=1}^{T} a_i \left\langle \Psi(\mathbf{x}_i), g_t \Psi(\mathbf{x}_t) \right\rangle = \left\langle \mathbf{u}, g_t \Psi(\mathbf{x}_t) \right\rangle .$$

Combining this and (18), and summing over t, we get that

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{E}_t\left[\left\langle \hat{\mathbf{w}}_t, \tilde{g}_t \hat{\Psi}(\tilde{\mathbf{x}}_t)\right\rangle\right] - \sum_{t=1}^{T} \mathbb{E}_t\left[\left\langle \hat{\mathbf{u}}^*, \tilde{g}_t \hat{\Psi}(\tilde{\mathbf{x}}_t)\right\rangle\right]\right] = \mathbb{E}\left[\sum_{t=1}^{T} \left\langle \mathbf{w}_t, g_t \Psi(\mathbf{x}_t)\right\rangle - \sum_{t=1}^{T} \left\langle \mathbf{u}, g_t \Psi(\mathbf{x}_t)\right\rangle\right]. \quad (19)$$

Remarkably, this equation links between classifiers $\hat{\mathbf{w}}_t$ in the RKHS corresponding to $\Psi(\cdot)$, and the classifiers $\mathbf{w}_t$ in another RKHS, corresponding to $\Psi(\cdot)$.

Substituting (19) into (16), we get that

$$\mathbb{E}\left[\sum_{t=1}^{T} \left\langle \mathbf{w}_t, g_t \Psi(\mathbf{x}_t)\right\rangle - \sum_{t=1}^{T} \left\langle \mathbf{u}, g_t \Psi(\mathbf{x}_t)\right\rangle\right] \le 2W R_{\Sigma,s,d} \sqrt{(W^2 R^2_{\Sigma,s,d} + B^2_{\tilde{y}})T}.$$

Now, since $\ell(\langle \mathbf{w}_t, \Psi(\mathbf{x}_t)\rangle, y_t) = (\langle \mathbf{w}_t, \Psi(\mathbf{x}_t)\rangle - y_t)^2$ is a convex function of $\mathbf{w}_t$, and since $g_t \Psi(\mathbf{x}_t)$ is the gradient at $\mathbf{w}_t$, we can lower bound the LHS as

$$\mathbb{E}\left[\sum_{t=1}^{T} \ell\big(\langle \mathbf{w}_t, \Psi(\mathbf{x}_t)\rangle, y_t\big)\right] - \sum_{t=1}^{T} \ell\big(\langle \mathbf{u}, \Psi(\mathbf{x}_t)\rangle, y_t\big)$$

from which the theorem follows.

C. Proof of Theorem 5

The proof follows the same lines as the proof of Theorem 3 in the previous subsection. The changes mostly have to do with the auxiliary lemmas, which we present below. The proof of the theorem itself is virtually identical to the one of Theorem 3, and is thus skipped.

The auxiliary lemmas below modify the parallel lemmas in Section VIII-B, based on the new definitions of the feature mapping Ψ and the surrogate feature mapping $\hat{\Psi}$. But before that, we begin with a lemma which explicitly upper bounds $\|\hat{\Psi}(\mathbf{x})\|^2$ for any $\mathbf{x}$. With Gaussian kernels, this was trivial, but now we need to work a bit harder.

Lemma 9: For any vector $\mathbf{x} \in \mathbb{R}^d$, we have

$$\left\|\hat{\Psi}(\mathbf{x})\right\|^2 \le \exp(2\sigma c).$$

Proof: By (8)

$$\begin{aligned}\left\|\hat{\Psi}(\mathbf{x})\right\|^2 &= \hat{k}(\mathbf{x},\mathbf{x}) = d\int_{u=0}^{c/d} (1-2\sigma u)^{-d/2} \exp(-ud)du \\ &\le (1-2\sigma c/d)^{-d/2} \int_{u=0}^{c/d} d\exp(-ud)du \\ &= (1-2\sigma c/d)^{-d/2}(1-\exp(-c)). \end{aligned} \quad (20)$$

Also, by a Taylor expansion of the log function, and using the fact that $2\sigma c/d < 1/2$ by the assumption that $c < d/4\sigma$, we get

$$\begin{aligned}\left(1-\frac{2\sigma c}{d}\right)^d &= \exp\left(d\log\left(1-\frac{2\sigma c}{d}\right)\right) \\ &\ge \exp\left(d\left(-\frac{4\log(2)\sigma c}{d}\right)\right) = \exp(-4\log(2)\sigma c). \end{aligned} \quad (21)$$

Plugging this into (20), we get the upper bound

$$\exp\big(2\log(2)\sigma c\big)(1-\exp(-c)) \le \exp(2\sigma c).$$

■

Lemma 10: For any $\mathbf{a}, \mathbf{x} \in \mathbb{R}^d$, if we let $\tilde{\mathbf{x}} = \mathbf{x} + \mathbf{n}$ where $\mathbf{n} \sim \mathcal{N}(0, \Sigma)$ is a Gaussian random vector with covariance matrix Σ, then it holds that

$$\mathbb{E}_{\mathbf{n}}\left\langle \hat{\Psi}(\tilde{\mathbf{x}}), \hat{\Psi}(\mathbf{a})\right\rangle = \left\langle \Psi(\mathbf{x}), \Psi(\mathbf{a})\right\rangle .$$

Proof: On one hand, based on the definition of k in (7), it can be verified that $\langle \Psi(\mathbf{x}), \Psi(\mathbf{a})\rangle$ equals

$$k(\mathbf{x},\mathbf{a}) = \int_{u=0}^{c/d} d\exp\left(-u\left\|\mathbf{x}-\mathbf{a}\right\|^2 - ud\right)du. \quad (22)$$

On the other hand, using the proof of Lemma 6 and Fubini's theorem, the expectation in the lemma can be written as

$$\begin{aligned}&\mathbb{E}_{\mathbf{n}}\left\langle \hat{\Psi}(\tilde{\mathbf{x}}), \hat{\Psi}(\mathbf{a})\right\rangle \\ &= \mathbb{E}_{\mathbf{n}}\left[d\int_{u=0}^{c/d} (1-2\sigma u)^{-d/2} \exp\left(-\frac{u\left\|\tilde{\mathbf{x}}-\mathbf{a}\right\|^2}{1-2\sigma u} - ud\right)du\right] \\ &= \int_{u=0}^{c/d} de^{-ud}\, \mathbb{E}_{\mathbf{n}}\left[(1-2\sigma u)^{-d/2} \exp\left(-\frac{\left\|\tilde{\mathbf{x}}-\mathbf{a}\right\|^2}{1/u-2\sigma}\right)\right]du \\ &= \int_{u=0}^{c/d} de^{-ud} \exp\left(-\frac{\left\|\mathbf{x}-\mathbf{a}\right\|^2}{1/u}\right)du \\ &= \int_{u=0}^{c/d} d\exp\left(-u\left\|\mathbf{x}-\mathbf{a}\right\|^2 - ud\right)du. \end{aligned}$$

■

Lemma 11: Let $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_T$ be vectors in $\mathbb{R}^d$, and $a_1, a_2, \ldots, a_T$ scalars, such that $\left\|\sum_i a_i \Psi(\mathbf{z}_i)\right\|^2 \le B^2$. Then $\sum_i a_i \hat{\Psi}(z_i)$ is an element in the RKHS with respect to $\hat{\Psi}(\cdot)$, whose norm squared is at most

$$\frac{5B^2}{\sin^2(\beta)} \exp(2\sigma c).$$

Here, β is the angle between $\sum_{i:a_i>0} a_i \Psi(\mathbf{z}_i)$ and $-\sum_{i:a_i<0} a_i \Psi(\mathbf{z}_i)$ in the RKHS (or $\beta = \pi/2$ is one of the elements is zero).

Proof: Picking some $\mathbf{z}_1, \mathbf{z}_2, \ldots$ and $a_1, a_2, \ldots$ as in the lemma statement, we have

$$\left\|\sum_i a_i \hat{\Psi}(z_i)\right\|^2 = \sum_{i,j} a_i a_j \hat{k}(\mathbf{z}_i, \mathbf{z}_j) \le \sum_{i,j:a_i a_j>0} a_i a_j \hat{k}(\mathbf{z}_i, \mathbf{z}_j). \quad (23)$$

Now, by definition of $\hat{k}(\cdot,\cdot)$ in (8), and the representation of $k(\cdot,\cdot)$ as in (22), it holds for any $\mathbf{z}_i, \mathbf{z}_j$ that $\hat{k}(\mathbf{z}_i,\mathbf{z}_j)/k(\mathbf{z}_i,\mathbf{z}_j)$ equals

$$\begin{aligned}&\frac{d\int_{u=0}^{c/d}(1-2\sigma u)^{-d/2}\exp\left(-\frac{u\|\mathbf{z}_i-\mathbf{z}_j\|^2}{1-2\sigma u}-ud\right)du}{d\int_{u=0}^{c/d}\exp\left(-u\|\mathbf{z}_i-\mathbf{z}_j\|^2-ud\right)du}\\&\le(1-2\sigma c/d)^{-d/2}\frac{\int_{u=0}^{c/d}\exp\left(-\frac{u\|\mathbf{z}_i-\mathbf{z}_j\|^2}{1-2\sigma u}-ud\right)du}{\int_{u=0}^{c/d}\exp\left(-u\|\mathbf{z}_i-\mathbf{z}_j\|^2-ud\right)du}\\&\le(1-2\sigma c/d)^{-d/2}\le e^{2\sigma c}\end{aligned}$$

where the last transition can be verified as in (21). Therefore, we can upper bound (23) by

$$\sum_{i,j\,:\,a_ia_j>0}a_ia_j\hat{k}(\mathbf{z}_i,\mathbf{z}_j)\le e^{2\sigma c}\sum_{i,j\,:\,a_ia_j>0}a_ia_jk(\mathbf{z}_i,\mathbf{z}_j).$$

The lemma follows by noting that

$$\begin{aligned}&\sum_{i,j\,:\,a_ia_j>0}a_ia_jk(\mathbf{z}_i,\mathbf{z}_j)\\&\qquad=\left\|\sum_{i\,:\,a_i>0}a_i\Psi(\mathbf{z}_i)\right\|^2+\left\|\sum_{i\,:\,a_i<0}a_i\Psi(\mathbf{z}_i)\right\|^2\end{aligned}$$

which according to Lemma 7 is at most $5B^2/\sin^2(\beta)$. ∎

D. Preliminary Result for Proving Theorem 6 and Theorem 7

To prove Theorem 6 and Theorem 7, we need a theorem which basically states that if all subroutines in algorithm 4 behave as they should, then one can achieve an $O(\sqrt{T})$ regret bound. This is provided in the following theorem, which is an adaptation of a standard result of online convex optimization (see, e.g., [11]).

Theorem 10: Assume the following conditions hold with respect to Algorithm 4:

1) For all t, $\tilde{\Psi}(\mathbf{x}_t)$ and $\tilde{g}_t$ are independent of each other (as random variables induced by the randomness of Algorithm 4) as well as independent of any $\tilde{\Psi}(\mathbf{x}_i)$ and $\tilde{g}_i$ for $i<t$.
2) For all t, $\mathbb{E}_t[\tilde{\Psi}(\mathbf{x}_t)]=\Psi(\mathbf{x}_t)$, and there exists a constant $B_{\tilde{\Psi}}>0$ such that
$$\mathbb{E}_t\left[\left\|\tilde{\Psi}(\mathbf{x}_t)\right\|^2\right]\le B_{\tilde{\Psi}}.$$
3) For all t, $\mathbb{E}_t[\tilde{g}_t]=y_t\ell'(y_t\langle\mathbf{w}_t,\Psi(\mathbf{x}_t)\rangle)$, and there exists a constant $B_{\tilde{g}}>0$ such that $\mathbb{E}_t[\tilde{g}_t^2]\le B_{\tilde{g}}$.
4) For any pair of instances $\mathbf{x}$, $\mathbf{x}'$
$$\texttt{Prod}(\tilde{\Psi}(\mathbf{x}),\tilde{\Psi}(\mathbf{x}'))=\left\langle\tilde{\Psi}(\mathbf{x}),\tilde{\Psi}(\mathbf{x}')\right\rangle.$$

If Algorithm 4 is run with $\eta=B_{\mathbf{w}}/\sqrt{B_{\tilde{g}}B_{\tilde{\Psi}}}$, then

$$\begin{aligned}&\mathbb{E}\left[\sum_{t=1}^{T}\ell(\langle\mathbf{w}_t,\Psi(\mathbf{x}_t)\rangle,y_t)\right]-\min_{\mathbf{w}\,:\,\|\mathbf{w}\|\le B_{\mathbf{w}}}\sum_{t=1}^{T}\ell(\langle\mathbf{w},\Psi(\mathbf{x}_t)\rangle,y_t)\\&\le B_{\mathbf{w}}\sqrt{B_{\tilde{g}}B_{\tilde{\Psi}}T}.\end{aligned}$$

Here the expectation is with respect to both the randomness of the oracles and of the algorithm throughout its run.

Proof: Our algorithm corresponds to Zinkevich's online gradient descent algorithm [11] in a finite horizon setting, where we assume the sequence of examples is $\tilde{g}_1\tilde{\Psi}(\mathbf{x}_1),\ldots,\tilde{g}_T\tilde{\Psi}(\mathbf{x}_T)$, the cost function is linear, and the learning rate at round t is $\eta/\sqrt{T}$. By a straightforward adaptation of the standard regret bound for that algorithm (see [11]), we have that for any $\mathbf{w}$ such that $\|\mathbf{w}\|\le B_{\mathbf{w}}$

$$\begin{aligned}&\sum_{t=1}^{T}\left\langle\mathbf{w}_t,\tilde{g}_t\tilde{\Psi}(\mathbf{x}_t)\right\rangle-\sum_{t=1}^{T}\left\langle\mathbf{w},\tilde{g}_t\tilde{\Psi}(\mathbf{x}_t)\right\rangle\\&\qquad\le\frac{1}{2}\left(\frac{B_{\mathbf{w}}^2}{\eta}+\frac{\eta}{T}\sum_{t=1}^{T}\left\|\tilde{g}_t\tilde{\Psi}(\mathbf{x}_t)\right\|^2\right)\sqrt{T}.\end{aligned}$$

We now take expectation of both sides in the inequality above. The expectation of the RHS is simply

$$\begin{aligned}&\mathbb{E}\left[\frac{1}{2}\left(\frac{B_{\mathbf{w}}^2}{\eta}+\frac{\eta}{T}\sum_{t=1}^{T}\mathbb{E}_t[\tilde{g}_t^2]\,\mathbb{E}_t\left[\left\|\tilde{\Psi}(\mathbf{x}_t)\right\|^2\right]\right)\sqrt{T}\right]\\&\le\frac{1}{2}\left(\frac{B_{\mathbf{w}}^2}{\eta}+\eta B_{\tilde{g}}B_{\tilde{\Psi}}\right)\sqrt{T}.\end{aligned}$$

As to the LHS, note that

$$\begin{aligned}&\mathbb{E}\left[\sum_{t=1}^{T}\left\langle\mathbf{w}_t,\tilde{g}_t\tilde{\Psi}(\mathbf{x}_t)\right\rangle\right]=\mathbb{E}\left[\sum_{t=1}^{T}\mathbb{E}_t\left[\left\langle\mathbf{w}_t,\tilde{g}_t\tilde{\Psi}(\mathbf{x}_t)\right\rangle\right]\right]\\&\quad=\mathbb{E}\left[\sum_{t=1}^{T}\left\langle\mathbf{w}_t,y_t\ell'\left(y_t\langle\mathbf{w}_t,\Psi(\mathbf{x}_t)\rangle\right)\Psi(\mathbf{x}_t)\right\rangle\right].\end{aligned}$$

Also

$$\mathbb{E}\left[\sum_{t=1}^{T}\left\langle\mathbf{w},\tilde{g}_t\tilde{\Psi}(\mathbf{x}_t)\right\rangle\right]=\sum_{t=1}^{T}\left\langle\mathbf{w},\ell'\left(y_t\langle\mathbf{w},\Psi(\mathbf{x}_t)\rangle\right)\Psi(\mathbf{x}_t)\right\rangle.$$

Plugging in these expectations and choosing η as in the statement of the theorem, we get that for any $\mathbf{w}$ such that $\|\mathbf{w}\|\le B_{\mathbf{w}}$

$$\begin{aligned}&\mathbb{E}\left[\sum_{t=1}^{T}\left\langle\mathbf{w}_t,y_t\ell'\left(y_t\langle\mathbf{w}_t,\Psi(\mathbf{x}_t)\rangle\right)\Psi(\mathbf{x}_t)\right\rangle\right.\\&\qquad\left.-\sum_{t=1}^{T}\left\langle\mathbf{w},\ell'\left(y_t\langle\mathbf{w}_t,\Psi(\mathbf{x}_t)\rangle\right)\Psi(\mathbf{x}_t)\right\rangle\right]\\&\qquad\le B_{\mathbf{w}}\sqrt{B_{\tilde{g}}B_{\tilde{\Psi}}T}.\end{aligned}$$

To get the theorem, we note that by convexity of ℓ, the LHS above can be lower bounded by

$$\mathbb{E}\left[\sum_{t=1}^{T}\ell(y_t\langle\mathbf{w}_t,\Psi(\mathbf{x}_t)\rangle)-\sum_{t=1}^{T}\ell(y_t\langle\mathbf{w},\Psi(\mathbf{x}_t)\rangle)\right].$$

∎

E. Proof of Theorem 6

Based on the preliminary result of Section VIII-D, we present in this subsection the proof of Theorem 6. We first show how to implement the subroutines of Algorithm 4, and prove the relevant results on their behavior. Then, we prove the theorem itself.

We start by constructing an explicit feature mapping $\Psi(\cdot)$ corresponding to the RKHS induced by our kernel. For any $\mathbf{x}$, $\mathbf{x}'$, we have that

$$\begin{aligned} k(\mathbf{x},\mathbf{x}') &= \sum_{n=0}^{\infty}\beta_n(\langle\mathbf{x},\mathbf{x}'\rangle)^n = \sum_{n=0}^{\infty}\beta_n\left(\sum_{i=1}^{d}x_i x_i'\right)^n \\ &= \sum_{n=0}^{\infty}\beta_n\sum_{k_1=1}^{d}\cdots\sum_{k_n=1}^{d}x_{k_1}x_{k_2}\cdots x_{k_n}x'_{k_1}x'_{k_2}\cdots x'_{k_n} \\ &= \sum_{n=0}^{\infty}\sum_{k_1=1}^{d}\cdots\sum_{k_n=1}^{d}\left(\sqrt{\beta_n}x_{k_1}\cdots x_{k_n}\right)\left(\sqrt{\beta_n}x'_{k_1}\cdots x'_{k_n}\right). \end{aligned}$$

This suggests the following feature representation: for any $\mathbf{x}$, $\Psi(\mathbf{x})$ returns an infinite-dimensional vector, indexed by n and $k_1,\ldots,k_n \in \{1,\ldots,d\}$, with the entry corresponding to $n,k_1,\ldots,k_n$ being $\sqrt{\beta_n}x_{k_1}\cdots x_{k_n}$. The inner product between $\Psi(\mathbf{x})$ and $\Psi(\mathbf{x}')$ is similar to a standard dot product between two vectors, and by the derivation above equals $k(\mathbf{x},\mathbf{x}')$ as required.

We now use a slightly more elaborate variant of our unbiased estimate technique, to derive an unbiased estimate of $\Psi(\mathbf{x})$. First, we sample N according to $\mathbb{P}(N=n)=(p-1)/p^{n+1}$. Then, we query the oracle for $\mathbf{x}$ for N times to get $\tilde{\mathbf{x}}^{(1)},\ldots,\tilde{\mathbf{x}}^{(N)}$, and formally define $\tilde{\Psi}(\mathbf{x})$ as

$$\tilde{\Psi}(\mathbf{x}) = \sqrt{\beta_n}\frac{p^{n+1}}{p-1}\sum_{k_1=1}^{d}\cdots\sum_{k_n=1}^{d}\tilde{x}_{k_1}^{(1)}\cdots\tilde{x}_{k_n}^{(n)}\mathbf{e}_{n,k_1,\ldots,k_n} \tag{24}$$

where $\mathbf{e}_{n,k_1,\ldots,k_n}$ represents the unit vector in the direction indexed by $n,k_1,\ldots,k_n$ as explained above. Since the oracle queries are i.i.d., the expectation of this expression is

$$\begin{aligned} &\sum_{n=0}^{\infty}\frac{p-1}{p^{n+1}}\sqrt{\beta_n}\frac{p^{n+1}}{p-1}\sum_{k_1=1}^{d}\cdots\sum_{k_n=1}^{d}\mathbb{E}\left[\tilde{x}_{k_1}^{(1)}\cdots\tilde{x}_{k_n}^{(n)}\right]\mathbf{e}_{n,k_1,\ldots,k_n} \\ &\quad= \sum_{n=0}^{\infty}\sum_{k_1=1}^{d}\cdots\sum_{k_n=1}^{d}\sqrt{\beta_n}x_{k_1}^{(1)}\cdots x_{k_n}^{(n)}\mathbf{e}_{n,k_1,\ldots,k_n} \end{aligned}$$

which is exactly $\Psi(\mathbf{x})$. We formalize the needed properties of $\tilde{\Psi}(\mathbf{x})$ in the following lemma.

Lemma 12: Assuming $\tilde{\Psi}(\mathbf{x})$ is constructed as in the discussion above, it holds that $\mathbb{E}[\tilde{\Psi}(\mathbf{x})]=\Psi(\mathbf{x})$ for any $\mathbf{x}$. Moreover, if the noisy samples $\tilde{\mathbf{x}}_t$ returned by the oracle A_t satisfy $\mathbb{E}_t[\|\tilde{\mathbf{x}}_t\|^2]\le B_{\tilde{\mathbf{x}}}$, then

$$\mathbb{E}_t\left[\left\|\tilde{\Psi}(\mathbf{x}_t)\right\|^2\right] \le \frac{p}{p-1}Q(pB_{\tilde{\mathbf{x}}})$$

where we recall that Q defines the kernel by $k(\mathbf{x},\mathbf{x}')=Q(\langle\mathbf{x},\mathbf{x}'\rangle)$.

Proof: The first part of the lemma follows from the discussion above. As to the second part, note that by (24)

$$\begin{aligned} \mathbb{E}_t\left[\left\|\tilde{\Psi}(\mathbf{x}_t)\right\|^2\right] &= \mathbb{E}_t\left[\beta_n\frac{p^{2n+2}}{(p-1)^2}\sum_{k_1\ldots,k_n=1}^{d}\left(\tilde{x}_{t,k_1}^{(1)}\cdots\tilde{x}_{t,k_n}^{(N)}\right)^2\right] \\ &= \mathbb{E}_t\left[\beta_n\frac{p^{2n+2}}{(p-1)^2}\prod_{j=1}^{n}\left\|\tilde{\mathbf{x}}_t^{(j)}\right\|^2\right] \\ &= \sum_{n=0}^{\infty}\frac{p-1}{p^{n+1}}\beta_n\frac{p^{2n+2}}{(p-1)^2}\left(\mathbb{E}_t\left[\tilde{\mathbf{x}}_t^2\right]\right)^n \\ &= \frac{p}{p-1}\sum_{n=0}^{\infty}\beta_n\left(p\mathbb{E}_t\left[\tilde{\mathbf{x}}_t^2\right]\right)^n \\ &\le \frac{p}{p-1}\sum_{n=0}^{\infty}\beta_n\left(pB_{\tilde{\mathbf{x}}}\right)^n = \frac{p}{p-1}Q(pB_{\tilde{\mathbf{x}}}) \end{aligned}$$

where the second-to-last step used the fact that $\beta_n\ge 0$ for all n. ∎

Of course, explicitly storing $\tilde{\Psi}(\mathbf{x})$ as defined above is infeasible, since the number of entries is huge. Fortunately, this is not needed: we just need to store $\tilde{\mathbf{x}}_t^{(1)},\ldots,\tilde{\mathbf{x}}_t^{(N)}$. The representation above is used implicitly when we calculate inner products between $\tilde{\Psi}(\mathbf{x})$ and other elements in the RKHS. We note that while N is a random quantity which might be unbounded, its distribution decays exponentially fast, so the number of vectors to store is essentially bounded.

After the discussion above, the pseudocode for `Map_Estimate` below should be self-explanatory.

Subroutine 2 `Map_Estimate`(A_t,p)

Sample nonnegative integer N according to

$$\mathbb{P}(N=n)=(p-1)/p^{n+1}$$

Query A_t for N times to get $\tilde{\mathbf{x}}_t^{(1)},\ldots,\tilde{\mathbf{x}}_t^{(N)}$

Return $\tilde{\mathbf{x}}_t^{(1)},\ldots,\tilde{\mathbf{x}}_t^{(N)}$ as $\tilde{\Psi}(\mathbf{x}_t)$.

We now turn to the subroutine `Prod`, which given two elements $\tilde{\Psi}(\mathbf{x})$, $\tilde{\Psi}(\mathbf{x}'))$ in the RKHS, returns their inner product.

Subroutine 3 `Prod`$(\tilde{\Psi}(\mathbf{x}),\tilde{\Psi}(\mathbf{x}'))$

Let $\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(n)}$ be the vectors comprising $\Psi(\mathbf{x})$

Let $\mathbf{x}'^{(1)},\ldots,\mathbf{x}'^{(n')}$ be the vectors comprising $\Psi(\mathbf{x}')$

If $n\ne n'$ return 0, else return $\beta_n\frac{p^{2n+2}}{(p-1)^2}\prod_{j=1}^{n}\left\langle\tilde{\mathbf{x}}^{(j)},\tilde{\mathbf{x}}'^{(j)}\right\rangle$

Lemma 13: `Prod`$(\tilde{\Psi}(\mathbf{x}),\tilde{\Psi}(\mathbf{x}'))$ returns $\left\langle\tilde{\Psi}(\mathbf{x})\tilde{\Psi}(\mathbf{x}')\right\rangle$.

Proof: Using the formal representation of $\tilde{\Psi}(\mathbf{x})$, $\tilde{\Psi}(\mathbf{x}')$ in (24), we have that $\left\langle \tilde{\Psi}(\mathbf{x}), \tilde{\Psi}(\mathbf{x}') \right\rangle$ is 0 whenever $n \neq n'$ (because then these two elements are composed of different unit vectors with respect to an orthogonal basis). Otherwise, we have that

$$\begin{aligned}
&\left\langle \tilde{\Psi}(\mathbf{x})\tilde{\Psi}(\mathbf{x}') \right\rangle \\
&= \beta_n \frac{p^{2n+2}}{(p-1)^2} \sum_{k_1,\ldots,k_n=1}^{d} \tilde{x}_{k_1}^{(1)} \cdots \tilde{x}_{k_n}^{(n)} \tilde{x}_{k_1}^{\prime(1)} \cdots \tilde{x}_{k_n}^{\prime(n)} \\
&= \beta_n \frac{p^{2n+2}}{(p-1)^2} \left(\sum_{k_1=1}^{d} \tilde{x}_{k_1}^{(1)} \tilde{x}_{k_1}^{\prime(1)} \right) \cdots \left(\sum_{k_N=1}^{d} \tilde{x}_{k_N}^{(n)} \tilde{x}_{k_N}^{\prime(n)} \right) \\
&= \beta_n \frac{p^{2n+2}}{(p-1)^2} \prod_{j=1}^{N} \left(\left\langle \tilde{\mathbf{x}}^{(j)}, \tilde{\mathbf{x}}^{\prime(j)} \right\rangle \right)
\end{aligned}$$

which is exactly what the algorithm returns, hence the lemma follows. ∎

As discussed in the main text, in order to apply the learned predictor on a new given instance $\mathbf{x}'$, we present another subroutine $\texttt{Mult}(\tilde{\Psi}(\mathbf{x}), \mathbf{x}')$, which calculates the inner product $\left\langle \tilde{\Psi}(\mathbf{x}), \Psi(\mathbf{x}') \right\rangle$. The pseudocode is very similar to the `Prod` subroutine, and the proof of correctness is essentially the same.

Subroutine 4 $\texttt{Mult}(\tilde{\Psi}(\mathbf{x}), \mathbf{x}')$

Let $n, \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}$ be the vectors comprising $\Psi(\mathbf{x})$

Return $\beta_n \frac{p^{n+1}}{p-1} \prod_{j=1}^{n} \left\langle \tilde{\mathbf{x}}^{(j)}, \mathbf{x}' \right\rangle$

We are now ready to prove Theorem 6. First, regarding the expected number of queries, notice that to run Algorithm 4, we invoke `Map_Estimate` and `Grad_Length_Estimate` once at round t. `Map_Estimate` uses a random number B of queries distributed as $\mathbb{P}(B = n) = (p-1)/p^{n+1}$, and `Grad_Length_Estimate` invokes `Map_Estimate` a random number C of times, distributed as $\mathbb{P}(C = n) = (p-1)/p^{n+1}$. The total number of queries is therefore $\sum_{j=1}^{C+1} B_j$, where B_j for all j are i.i.d. copies of B. The expected value of this expression, using a standard result on the expected value of a sum of a random number of independent random variables, is equal to $(1 + \mathbb{E}[C])\mathbb{E}[B_j]$, or $\left(1 + \frac{1}{p-1}\right)\frac{1}{p-1} = \frac{p}{(p-1)^2}$.

In terms of running time, we note that the expected running time of `Prod` is $O\left(1 + \frac{d}{p-1}\right)$, this because it performs N multiplications of inner products, each one with running time $O(d)$, and $\mathbb{E}[N] = \frac{1}{p-1}$. The expected running time of `Map_Estimate` is $O\left(1 + \frac{1}{p-1}\right)$. The expected running time of `Grad_Length_Estimate` is

$$\begin{aligned}
&O\left(1 + \frac{1}{p-1}\left(1 + \frac{1}{p-1}\right) + T\left(1 + \frac{d}{p-1}\right)\right) \\
&\quad = O\left(\frac{p}{(p-1)^2} + T\left(1 + \frac{d}{p-1}\right)\right).
\end{aligned}$$

Since Algorithm 4 at each of T rounds calls `Map_Estimate` once, `Grad_Length_Estimate` once, `Prod` for $O(T^2)$ times, and performs $O(1)$ other operations, we get that the overall runtime is

$$\begin{aligned}
O\Bigg(T\Bigg(1 + \frac{1}{p-1} + \frac{p}{(p-1)^2} \\
+T\left(1 + \frac{d}{p-1}\right) + T^2\left(1 + \frac{d}{p-1}\right)\Bigg)\Bigg).
\end{aligned}$$

Since $\frac{1}{p-1} \leq \frac{p}{(p-1)^2}$, we can upper bound this by

$$\begin{aligned}
&O\left(T\left(1 + \frac{p}{(p-1)^2} + T^2\left(1 + \frac{dp}{(p-1)^2}\right)\right)\right) \\
&\quad = O\left(T^3\left(1 + \frac{dp}{(p-1)^2}\right)\right).
\end{aligned}$$

The regret bound in the theorem follows from Theorem 10, with the expressions for constants following from Lemma 4, Lemma 12, and Lemma 13.

F. Proof of Theorem 7

The proof here is based on the preliminary result of Section VIII-D. The analysis in the Gaussian kernel case is rather similar to the one for inner product kernel case (in Section VIII-E), with some technical changes. Thus, we provide the proof here mostly for completeness.

We start by constructing an explicit feature mapping $\Psi(\cdot)$ corresponding to the RKHS induced by our kernel. For any $\mathbf{x}$, $\mathbf{x}'$, we have that

$$\begin{aligned}
k(\mathbf{x}, \mathbf{x}') &= \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{s^2}\right) \\
&= \exp\left(-\frac{\|\mathbf{x}\|^2}{s^2}\right) \exp\left(-\frac{\|\mathbf{x}'\|^2}{s^2}\right) \exp\left(\frac{2\langle \mathbf{x}, \mathbf{x}' \rangle}{s^2}\right) \\
&= \exp\left(-\frac{\|\mathbf{x}\|^2}{s^2}\right) \exp\left(-\frac{\|\mathbf{x}'\|^2}{s^2}\right) \left(\sum_{n=0}^{\infty} \frac{(2\langle \mathbf{x}, \mathbf{x}' \rangle)^n}{s^{2n} n!}\right) \\
&= \exp\left(-\frac{\|\mathbf{x}\|^2}{s^2}\right) \exp\left(-\frac{\|\mathbf{x}'\|^2}{s^2}\right) \\
&\quad \times \left(\sum_{n=0}^{\infty} \sum_{k_1=1}^{d} \cdots \sum_{k_n=1}^{d} \frac{(2/s^2)^n}{n!} x_{k_1} \cdots x_{k_n} x'_{k_1} \cdots x'_{k_n}\right).
\end{aligned}$$

This suggests the following feature representation: for any $\mathbf{x}$, $\Psi(\mathbf{x})$ returns an infinite-dimensional vector, indexed by n and $k_1, \ldots, k_n \in \{1, \ldots, d\}$, with the entry corresponding to $n, k_1, \ldots, k_n$ being $e^{-\|\mathbf{x}\|^2/s^2} \frac{(2/s^2)^n}{n!} x_{k_1} \ldots x_{k_n}$. The inner product between $\Psi(\mathbf{x})$ and $\Psi(\mathbf{x}')$ is similar to a standard inner product between two vectors, and by the derivation above equals $k(\mathbf{x}, \mathbf{x}')$ as required.

The idea of deriving an unbiased estimate of $\Psi(\mathbf{x})$ is the following: first, we sample N_1, N_2 independently according to $\mathbb{P}(N_1 = n_1) = \mathbb{P}(N_2 = n_2) = (p-1)/p^{n+1}$. Then, we query

the oracle for $\mathbf{x}$ for $2N_1 + N_2$ times to get $\tilde{\mathbf{x}}^1, \ldots, \tilde{\mathbf{x}}^{(2N_1+N_2)}$, and formally define $\tilde{\Psi}(\mathbf{x})$ as

$$\tilde{\Psi}(\mathbf{x}) = \frac{(-1)^{N_1} p^{N_1+N_2+2} 2^{N_2}}{N_1! N_2! s^{2N_1+2N_2}(p-1)^2} \times \left(\prod_{j=1}^{N_1} \left\langle \tilde{\mathbf{x}}^{(2j-1)}, \tilde{\mathbf{x}}^{(2j)} \right\rangle \right) \times \left(\sum_{k_1,\ldots,k_{N_2}=1}^{d} \tilde{x}_{k_1}^{(2N_1+1)} \cdots \tilde{x}_{k_{N_2}}^{(2N_1+N_2)} \mathbf{e}_{N_2,k_1,\ldots,k_{N_2}} \right) \tag{25}$$

where $\mathbf{e}_{N_2,k_1,\ldots,k_{N_2}}$ represents the unit vector in the direction indexed by $N_2, k_1, \ldots, k_{N_2}$ as explained above. Since the oracle calls are i.i.d., it is not hard to verify that the expectation of the expression above is

$$\begin{aligned}
&\left(\sum_{n_1=0}^{\infty} \frac{p-1}{p^{n_1+1}} \frac{(-1)^{n_1} p^{n_1+1}}{n_1! s^{2n_1}(p-1)} (\langle \mathbf{x}, \mathbf{x} \rangle)^{n_1} \right) \\
&\quad \times \left(\sum_{n_2=0}^{\infty} \frac{p-1}{p^{n_2+1}} \frac{p^{n_2+1} 2^{n_2}}{n_2! s^{2n_2}(p-1)} \right. \\
&\qquad \left. \sum_{k_1,\ldots,k_{n_2}=1}^{d} x_{k_1} \cdots x_{k_{n_2}} \mathbf{e}_{n_2,k_1,\ldots,k_{n_2}} \right) \\
&= \left(\sum_{n_1=0}^{\infty} \frac{(-\|\mathbf{x}\|^2/s^2)^{n_1}}{n_1!} \right) \\
&\quad \times \left(\sum_{n_2=0}^{\infty} \frac{(2/s^2)^{n_2}}{n_2!} \sum_{k_1,\ldots,k_{n_2}=1}^{d} x_{k_1} \cdots x_{k_{n_2}} \mathbf{e}_{n_2,k_1,\ldots,k_{n_2}} \right) \\
&= \exp\left(-\frac{\|\mathbf{x}\|^2}{s^2} \right) \\
&\quad \times \left(\sum_{n_2=0}^{\infty} \sum_{k_1,\ldots,k_{n_2}=1}^{d} \frac{(2/s^2)^{n_2}}{n_2!} x_{k_1} \cdots x_{k_{n_2}} \mathbf{e}_{n_2,k_1,\ldots,k_{n_2}} \right)
\end{aligned}$$

which is exactly $\Psi(\mathbf{x})$ as defined above.

To actually store $\tilde{\Psi}(\mathbf{x})$ in memory, we simply keep and $\tilde{\mathbf{x}}^{(1)}, \ldots, \tilde{\mathbf{x}}^{(2N_1+N_2)}$. The representation above is used implicitly when we calculate inner products between $\tilde{\Psi}(\mathbf{x})$ and other elements in the RKHS, via the subroutine Prod. We formalize the needed properties of $\tilde{\Psi}(\mathbf{x})$ in the following lemma.

Lemma 14: Assuming the construction of $\tilde{\Psi}(\mathbf{x})$ as in the discussion above, it holds that $\mathbb{E}_t[\tilde{\Psi}(\mathbf{x})] = \Psi(\mathbf{x})$ for all $\mathbf{x}$. Moreover, if the noisy sample $\tilde{\mathbf{x}}_t$ returned by the oracle A_t satisfies $\mathbb{E}_t[\|\tilde{\mathbf{x}}_t\|^2] \le B_{\tilde{\mathbf{x}}}$, then

$$\mathbb{E}_t \left[\left\| \tilde{\Psi}(\mathbf{x}_t) \right\|^2 \right] \le \left(\frac{p}{p-1} \right)^2 \exp\left(\frac{\sqrt{p} B_{\tilde{\mathbf{x}}} + 2p\sqrt{B_{\tilde{\mathbf{x}}}}}{s^2} \right).$$

Proof: The first part of the lemma follows from the discussion above. As to the second part, note that by (25), we have that $\left\| \tilde{\Psi}(\mathbf{x}_t) \right\|^2$ equals

$$\begin{aligned}
&\frac{p^{2N_1+2N_2+4} 2^{2N_2}}{\left(N_1! N_2! s^{2N_1+2N_2}(p-1)^2\right)^2} \left(\prod_{j=1}^{N_1} \left(\left\langle \tilde{\mathbf{x}}^{(2j-1)}, \tilde{\mathbf{x}}^{(2j)} \right\rangle \right)^2 \right) \\
&\quad \times \left(\sum_{k_1,\ldots,k_{N_2}=1}^{d} \left(\tilde{x}_{k_1}^{(2N_1+1)} \cdots \tilde{x}_{k_{N_2}}^{(2N_1+N_2)} \right)^2 \right) \\
&= \frac{p^{2N_1+2N_2+4} 2^{2N_2}}{\left(N_1! N_2! s^{2N_1+2N_2}(p-1)^2\right)^2} \left(\prod_{j=1}^{N_1} \left(\left\langle \tilde{\mathbf{x}}^{(2j-1)}, \tilde{\mathbf{x}}^{(2j)} \right\rangle \right)^2 \right) \\
&\quad \times \left(\prod_{j=1}^{N_2} \left\| \tilde{\mathbf{x}}^{(N_1+j)} \right\|^2 \right) \\
&\le \frac{p^{2N_1+2N_2+4} 2^{2N_2}}{\left(N_1! N_2! s^{2N_1+2N_2}(p-1)^2\right)^2} B_{\tilde{\mathbf{x}}}^{2N_1} B_{\tilde{\mathbf{x}}}^{N_2}.
\end{aligned}$$

The expectation of this expression over N_1, N_2 is equal to

$$\begin{aligned}
&\left(\sum_{n_1=0}^{\infty} \frac{p-1}{p^{n_1+1}} \frac{p^{2n_1+2}}{(n_1! s^{2n_1}(p-1))^2} B_{\tilde{\mathbf{x}}}^{2n_1} \right) \\
&\quad \times \left(\sum_{n_2=0}^{\infty} \frac{p-1}{p^{n_2+1}} \frac{p^{2n_2+2} 2^{2n_2}}{(n_2! s^{2n_2}(p-1))^2} B_{\tilde{\mathbf{x}}}^{n_2} \right) \\
&= \left(\frac{p}{p-1} \right)^2 \left(\sum_{n_1=0}^{\infty} \frac{(pB_{\tilde{\mathbf{x}}}^2)^{n_1}}{(n_1! s^{2n_1})^2} \right) \left(\sum_{n_2=0}^{\infty} \frac{(4p^2 B_{\tilde{\mathbf{x}}})^{n_2}}{(n_2! s^{2n_2})^2} \right) \\
&= \left(\frac{p}{p-1} \right)^2 \left(\sum_{n_1=0}^{\infty} \left(\frac{(\sqrt{p} B_{\tilde{\mathbf{x}}}/s^2)^{n_1}}{n_1!} \right)^2 \right) \\
&\quad \times \left(\sum_{n_2=0}^{\infty} \left(\frac{(2p\sqrt{B_{\tilde{\mathbf{x}}}}/s^2)^{n_2}}{n_2!} \right)^2 \right) \\
&\le \left(\frac{p}{p-1} \right)^2 \left(\sum_{n_1=0}^{\infty} \frac{(\sqrt{p} B_{\tilde{\mathbf{x}}}/s^2)^{n_1}}{n_1!} \right)^2 \\
&\quad \times \left(\sum_{n_2=0}^{\infty} \frac{(2p\sqrt{B_{\tilde{\mathbf{x}}}}/s^2)^{n_2}}{n_2!} \right)^2 \\
&= \left(\frac{p}{p-1} \right)^2 \exp\left(\frac{\sqrt{p} B_{\tilde{\mathbf{x}}} + 2p\sqrt{B_{\tilde{\mathbf{x}}}}}{s^2} \right).
\end{aligned}$$

■

After the discussion above, the pseudocode for `Map_Estimate` below should be self-explanatory.

Subroutine 5 `Map_Estimate`(A_t, p)

Sample N_1 according to $\mathbb{P}(N_1 = n_1) = (p-1)/p^{n_1+1}$

Sample N_2 according to $\mathbb{P}(N_2 = n_2) = (p-1)/p^{n_2+1}$

Query A_t for $2N_1 + N_2$ times to get $\tilde{\mathbf{x}}_t^{(1)}, \ldots, \tilde{\mathbf{x}}_t^{(2N_1+N_2)}$

Return $\tilde{\mathbf{x}}_t^{(1)}, \ldots, \tilde{\mathbf{x}}_t^{(2N_1+N_2)}$ as $\tilde{\Psi}(\mathbf{x}_t)$.

We now turn to the subroutine `Prod` (Subroutine 10), which given the two elements $\tilde{\Psi}(\mathbf{x}), \tilde{\Psi}(\mathbf{x}')$ in the RKHS, returns their inner product.

Subroutine 6 `Prod`$(\tilde{\Psi}(\mathbf{x}), \tilde{\Psi}(\mathbf{x}'))$

Let $\tilde{\mathbf{x}}^{(n)}, \ldots, \tilde{\mathbf{x}}^{(2n_1+n_2)}$ be the vectors comprising $\tilde{\Psi}(\mathbf{x})$

Let $\tilde{\mathbf{x}'}^{(1)}, \ldots, \tilde{\mathbf{x}'}^{(2n_1'+n_2')}$ be the vectors comprising $\tilde{\Psi}(\mathbf{x}')$

If $n_2' \neq n_2'$ return 0, else return

$$\frac{(-1)^{n_1+n_1'} p^{n_1+n_1'+2n_2+4} 2^{2n_2}}{n_1! n_1'! (n_2!)^2 s^{2(n_1+n_1'+2n_2)} (p-1)^4} \times \left(\prod_{j=1}^{n_1} \left\langle \tilde{\mathbf{x}}^{(2j-1)}, \tilde{\mathbf{x}}^{(2j)} \right\rangle\right) \left(\prod_{j=1}^{n_1'} \left\langle \tilde{\mathbf{x}}'^{(2j-1)}, \tilde{\mathbf{x}}'^{(2j)} \right\rangle\right) \times \left(\prod_{j=1}^{n_2} \left\langle \tilde{\mathbf{x}}^{(2n_1+j)}, \tilde{\mathbf{x}}'^{(2n_1'+j)} \right\rangle\right)$$

The proof of the following lemma is a straightforward algebraic exercise, similar to the proof of Lemma 13. ∎

Lemma 15: $\mathrm{Prod}(\tilde{\Psi}(\mathbf{x}), \tilde{\Psi}(\mathbf{x}'))$ returns $\left\langle \tilde{\Psi}(\mathbf{x}), \tilde{\Psi}(\mathbf{x}') \right\rangle$.

As described in the main text, when we wish to apply our learned predictor on a given instance $\mathbf{x}'$, we also need a subroutine to compute $\left\langle \tilde{\Psi}(\mathbf{x}), \Psi(\mathbf{x}') \right\rangle$, where $\mathbf{x}'$ is an explicitly given vector. The pseudocode is described in Subroutine 11. It is very similar to Subroutine 10, and the proof is essentially the same.

Subroutine 7 `Mult`$(\tilde{\Psi}(\mathbf{x}), \mathbf{x}')$

Let $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(2n_1+n_2)}$ be the vectors comprising $\tilde{\Psi}(\mathbf{x})$

Return

$$\frac{(-1)^{n_1} p^{n_1+n_2+2} 2^{2n_2}}{n_1! (n_2!)^2 s^{2(n_1+2n_2)} (p-1)^2} \exp\left(-\frac{\|\mathbf{x}'\|^2}{s^2}\right) \times \left(\prod_{j=1}^{n_1} \left\langle \tilde{\mathbf{x}}^{(2j-1)}, \tilde{\mathbf{x}}^{(2j)} \right\rangle\right) \left(\prod_{j=1}^{n_2} \left\langle \tilde{\mathbf{x}}^{(2n_1+j)}, \mathbf{x}' \right\rangle\right).$$

We are now ready to prove Theorem 7. First, regarding the expected number of queries, notice that to run Algorithm 4, we invoke `Map_Estimate` and `Grad_Length_Estimate` once at round t. `Map_Estimate` uses a random number $2B_1 + B_2$ of queries, where B_1, B_2 are independent and distributed as $\mathbb{P}(B_1 = n) = \mathbb{P}(B_2 = n) = (p-1)/p^{n+1}$. `Grad_Length_Estimate` invokes `Map_Estimate` a random number C of times, where $\mathbb{P}(C = n) = (p-1)/p^{n+1}$. The total number of queries is therefore $\sum_{j=1}^{C+1}(2B_{j,1} + B_{j,2})$, where $B_{j,1}$, $B_{j,2}$ are i.i.d. copies of B_1, B_2, respectively. The expected value of this expression, using a standard result on the expected value of a sum of a random number of random variables, is equal to $(1 + \mathbb{E}[C])(2\mathbb{E}[B_{j,1}] + \mathbb{E}[B_{j,2}])$, or $\left(1 + \frac{1}{p-1}\right)\frac{3}{p-1} = \frac{3p}{(p-1)^2}$.

In terms of running time, the analysis is completely identical to the one performed in the proof of Theorem 6, and the expected running time is the same up to constants.

The regret bound in the theorem follows from Theorem 10, with the expressions for constants following from Lemma 4, Lemma 14, and Lemma 15.

G. Proof of Examples 3 and 4

Examples 3 and 4 use the error function $\mathrm{Erf}(a)$ in order to construct analytic approximations of the hinge loss and the absolute loss (see Fig. 2). The error function is useful for our purposes, since it is analytic in all of $\mathbb{R}$, and smoothly interpolates between -1 for $a \ll 0$ and 1 for $a \gg 0$. Thus, it can be used to approximate derivative of losses which are piecewise linear, such as the hinge loss $\ell(a) = \max\{0, 1-a\}$ and the absolute loss $\ell(a) = |a|$.

To approximate the absolute loss, we use the antiderivative of $\mathrm{Erf}(sa)$. This function represents an analytic upper bound on the absolute loss, which becomes tighter as c increases. It can be verified that the antiderivative (with the constant free parameter fixed so the function has the desired behavior) is

$$\ell(a) = a\,\mathrm{Erf}(sa) + \frac{1}{c\sqrt{\pi}} \exp(-c^2 a^2).$$

While this loss function may seem to have slightly complex form, we note that our algorithm only needs to calculate the derivative of this loss function at various points (namely $\mathrm{Erf}(sa)$ for various values of a), which can be easily done.

By a Taylor expansion of the error function, we have that

$$\ell(a)' = \frac{2}{\sqrt{\pi}} \sum_{n=0}^{\infty} \frac{(-1)^n (sa)^{2n+1}}{n!(2n+1)}.$$

Therefore, $\ell'_+(a)$ in this case is at most

$$\frac{2}{\sqrt{\pi}} \sum_{n=0}^{\infty} \frac{(sa)^{2n+1}}{n!(2n+1)} \leq \frac{2}{as\sqrt{\pi}} \sum_{n=0}^{\infty} \frac{(sa)^{2(n+1)}}{(n+1)!} = \frac{2}{as\sqrt{\pi}} \exp(c^2 a^2 - 1).$$

We now turn to deal with Example 4. This time, we use the antiderivative of $(\mathrm{Erf}(c(a-1))-1)/2$. This function smoothly interpolates between -1 for $a \ll -1$ and 0 for $a \gg 0$. Therefore, its antiderivative with respect to x represents an analytic upper bound on the hinge loss, which becomes tighter as c increases. It can be verified that the antiderivative (with the constant free parameter fixed so the function has the desired behavior) is

$$\ell(a) = \frac{(a-1)(\mathrm{Erf}(c(a-1)) - 1)}{2} + \frac{1}{2\sqrt{\pi}c} \exp(-c^2 (a-1)^2).$$

By a Taylor expansion of the error function, we have that

$$\ell(a)' = -\frac{1}{2} + \frac{1}{\sqrt{\pi}} \sum_{n=0}^{\infty} \frac{(-1)^n (c(a-1))^{2n+1}}{n!(2n+1)}.$$

Thus, $\ell'_+(a)$ in this case can be upper bounded by

$$\frac{1}{2} + \frac{1}{\sqrt{\pi}} \sum_{n=0}^{\infty} \frac{(sa)^{2n+1}}{n!(2n+1)} \leq \frac{1}{2} + \frac{1}{as\sqrt{\pi}} \sum_{n=0}^{\infty} \frac{(sa)^{2(n+1)}}{(n+1)!} \leq \frac{1}{2} + \frac{1}{as\sqrt{\pi}} \left(\exp(c^2 a^2) - 1\right).$$

H. Proof of Theorem 8

Fix a large enough $B_{\mathbf{w}} \geq 1$ to be specified later. Let $\mathbf{x} = (1, 0, \ldots, 0)$ and let $\mathcal{D}$ to be the uniform distribution on $\{3\mathbf{x}, -\mathbf{x}\}$. To prove the result then we just need to show that

$$\operatorname*{argmin}_{w\,:\,|w|^2 \leq B_{\mathbf{w}}} \ell(3w,1) + \ell(-w,1) \quad \text{and} \quad \operatorname*{argmin}_{w\,:\,|w|^2 \leq B_{\mathbf{w}}} \ell(w,1) \tag{26}$$

are disjoint, for some appropriately chosen $B_{\mathbf{w}}$.

First, we show that the first set above is a subset of $\{w : |w|^2 \leq R\}$ for some fixed R which does not depend on $B_{\mathbf{w}}$. We do a case-by-case analysis, depending on how $\ell(\cdot, 1)$ looks like.

1) $\ell(\cdot,1)$ monotonically increases in $\mathbb{R}$. Impossible by assumption (2).
2) $\ell(\cdot,1)$ monotonically decreases in $\mathbb{R}$. First, recall that since $\ell(\cdot,1)$ is convex, it is differentiable almost anywhere, and its derivative is monotonically increasing. Now, since $\ell(\cdot,1)$ is convex and bounded from below, $\ell'(w,1)$ must tend to 0 as $w \to \infty$ (wherever $\ell(\cdot,1)$ is differentiable, which is almost everywhere by convexity). Moreover, by assumption (2), $\ell'(w,1)$ is upper bounded by a strictly negative constant for any $w < 0$. As a result, the gradient of $\ell(3w,1) + \ell(-w,1)$, which equals $3\ell'(3w,1) - \ell'(-w,1)$, must be positive for large enough $w > 0$, and negative for large enough $w < 0$, so the minimizers of $\ell(3w,1) + \ell(-w,1)$ are in some bounded subset of $\mathbb{R}$.
3) There is some $s \in \mathbb{R}$ such that $\ell(\cdot,1)$ monotonically decreases in $(-\infty, s)$ and monotonically increases in (s, ∞). If the function is constant in (s,∞) or in $(-\infty, s)$, we are back to one of the two previous cases. Otherwise, by convexity of $\ell(\cdot)$, we must have some a, b, $a \leq s \leq b$, such that $\ell(\cdot,1)$ is strictly decreasing at $(-\infty, a)$, and strictly increasing at (b,∞). In that case, it is not hard to see that $\ell(3w,1) + \ell(-w,1)$ must be strictly increasing for any $w > \max\{|a|,|b|\}$, and strictly decreasing for any $w < -\max\{|a|,|b|\}$. So again, the minimizers of $\ell(3w,1) + \ell(-w,1)$ are in some bounded subset of $\mathbb{R}$.

We are now ready to show that the two sets in (26) must be disjoint. Suppose we pick $B_{\mathbf{w}}$ large enough so that the first set in (26) is strictly inside $\{w : |w|^2 \leq B_{\mathbf{w}}\}$. Assume on the contrary that there is some w, $|w|^2 < B_{\mathbf{w}}$, which belongs to both sets in (26). By assumption (2) and the fact that w minimizes $\ell(w,1)$, we may assume $w > 0$. Therefore, $0 \in \partial\ell(w,1)$ as well as $0 \in \partial(\ell(3w,1) + \ell(-w,1))$, where ∂f is the (closed and convex) subgradient set of a convex function f. By subgradient calculus, this means there is some $a/3 \in \partial\ell(3w,1)$ and $b \in \partial\ell(-w,1)$ such that $a/3 - b = 0$. This implies that $\partial\ell(3w,1) \cap \partial\ell(-w,1) \neq \emptyset$. Now, suppose that $\max \partial\ell(-w,1) < 0$. This would mean that $\min \partial\ell(3w,1) < 0$. But then $\ell(\cdot,1)$ is strictly decreasing at $(w, 3w)$, and in particular $\ell(w,1) > \ell(3w,1)$, contradicting the assumption that w minimizes $\ell(\cdot,1)$. So we must have $\max \partial\ell(-w,1) \geq 0$. Moreover, $\min \partial\ell(-w,1) \leq 0$ (because w minimizes $\ell(\cdot,1)$ and $-w < w$). Since the subgradient set is closed and convex, it follows that $0 \in \partial\ell(-w,1)$. Therefore, both w and $-w$ minimize $\ell(\cdot,1)$. But this means that $\ell'(0) = 0$, in contradiction to assumption (2).

I. Proof of Theorem 9

Let $\mathcal{D}$ be a distribution which satisfies (9). The idea of the proof is that the learner cannot know if $\mathcal{D}$ is the real distribution (on which regret is measured) or the distribution which includes noise. Specifically, consider the following two adversary strategies:

1) At each round, draw an example from $\mathcal{D}$, and present it to the learner (with the label 1) without adding noise.
2) At each round, pick the example $\mathbb{E}_{\mathcal{D}}[\mathbf{x}]$, add to it zero-mean noise sampled from $Z - \mathbb{E}_{\mathcal{D}}[\mathbf{x}]$, where Z is distributed according to $\mathcal{D}$, and present the noisy example (with the label 1) to the learner.

In both cases the examples presented to a learner appear to come from the same distribution $\mathcal{D}$. Hence, any learner observing one copy of each example cannot know which of the two strategies is played by the adversary. Since (9) implies that the set of optimal learner strategies for each of the two adversary strategies are disjoint, by picking an appropriate strategy the adversary can force a constant regret.

To formalize this argument, fix any learning algorithm that observes one copy of each example and let $\mathbf{w}_1, \mathbf{w}_2, \ldots$ be the sequence of generated predictors. Then it is sufficient to show that at least one of the following two holds

$$\limsup_{T\to\infty} \max_{\mathbf{w}\in\mathcal{W}} \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} \ell(\langle \mathbf{w}_t, \mathbf{x}_t\rangle, 1) - \ell(\langle \mathbf{w}, \mathbf{x}_t\rangle, 1)\right] > 0 \tag{27}$$

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \ell(\langle \mathbf{w}_t, \mathbb{E}[\mathbf{x}]\rangle, 1) - \min_{\mathbf{w}\in\mathcal{W}} \ell\big(\langle \mathbf{w}, \mathbb{E}[\mathbf{x}]\rangle, 1\big) > 0 \tag{28}$$

with probability 1, where in both cases the expectation is with respect to $\mathcal{D}$ and "w.p. 1" refers to the randomness of the noise. First note that (27) is implied by

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \ell(\langle \mathbf{w}_t, \mathbf{x}_t\rangle, 1) - \min_{\mathbf{w}\in\mathcal{W}} \mathbb{E}\Big[\ell(\langle \mathbf{w}, \mathbf{x}\rangle, 1)\Big] > 0 \tag{29}$$

with probability 1. Since $\mathcal{W}$ is compact, $\mathcal{D}$ is assumed to be supported on a compact subset, and ℓ is convex and hence continuous, then $\ell(\langle \mathbf{w}, \mathbf{x}\rangle, 1)$ is almost surely bounded. So by Azuma's inequality

$$\sum_{T=1}^{\infty} \mathbb{P}\left(\frac{1}{T}\sum_{t=1}^{T}\Big(\mathbb{E}_t\big[\ell(\langle \mathbf{w}_t, \mathbf{x}\rangle, 1)\big] - \ell(\langle \mathbf{w}_t, \mathbf{x}_t\rangle, 1)\Big) \geq \epsilon\right)$$

is finite for all $\epsilon > 0$, where the expectation $\mathbb{E}_t[\cdot]$ is conditioned on the randomness in the previous rounds. Letting $\bar{\mathbf{w}}_t = \frac{1}{t}\sum_{s=1}^{t} \mathbf{w}_s$ (which belongs to $\mathcal{W}$ for all t since it is a convex set), we have

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T} \ell(\langle \mathbf{w}_t, \mathbf{x}_t\rangle, 1) &\geq \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_t[\ell(\langle \mathbf{w}_t, \mathbf{x}\rangle, 1)] \\ &\geq \mathbb{E}\Big[\ell\big(\langle \bar{\mathbf{w}}_T, \mathbf{x}\rangle, 1\big)\Big]\end{aligned}$$

where the first inequality holds with probability 1 as $T \to \infty$ by the Borel-Cantelli lemma, and the second one holds for every T because ℓ is convex.

Similarly,

$$\frac{1}{T}\sum_{t=1}^{T} \ell(\langle \mathbf{w}_t, \mathbb{E}[\mathbf{x}]\rangle, 1) \geq \ell\big(\langle \bar{\mathbf{w}}_T, \mathbb{E}[\mathbf{x}]\rangle, 1\big).$$

Hence (28)–(29) are obtained if we show that no single sequence of predictors $\bar{\mathbf{w}}_1, \bar{\mathbf{w}}_2, \ldots$ simultaneously satisfies

$$\limsup_{T\to\infty} F_1(\bar{\mathbf{w}}_T) \leq 0 \qquad \text{and} \qquad \limsup_{T\to\infty} F_2(\bar{\mathbf{w}}_T) \leq 0 \tag{30}$$

where

$$F_1(\mathbf{w}_T) = \mathbb{E}\Big[\ell\big(\langle \bar{\mathbf{w}}_T, \mathbf{x}\rangle, 1\big)\Big] - \min_{\mathbf{w}\in\mathcal{W}} \mathbb{E}\big[\ell\left(\langle \mathbf{w}, \mathbf{x}\rangle, 1\right)\big]$$

and

$$F_2(\mathbf{w}_T) = \ell\big(\langle \bar{\mathbf{w}}_T, \mathbb{E}[\mathbf{x}]\rangle, 1\big) - \min_{\mathbf{w}\in\mathcal{W}} \ell\big(\langle \mathbf{w}, \mathbb{E}[\mathbf{x}]\rangle, 1\big).$$

Suppose on the contrary that there was such a sequence. Since $\bar{\mathbf{w}}_T \in \mathcal{W}$ for all T, and $\mathcal{W}$ is compact, the sequence $\bar{\mathbf{w}}_1, \bar{\mathbf{w}}_2, \ldots$ has at least a cluster point $\bar{\mathbf{w}} \in \mathcal{W}$. Moreover, it is easy to verify that the functions F_1 and F_2 are continuous. Indeed, $\ell(\langle \cdot, \mathbb{E}[\mathbf{x}]\rangle, 1)$ is continuous by convexity of ℓ and $\mathbb{E}[\ell(\langle \cdot, \mathbf{x}\rangle, 1)]$ is continuous by the compactness assumptions. Hence, any cluster point of $\bar{\mathbf{w}}_1, \bar{\mathbf{w}}_2, \ldots$ is also a cluster point of both F_1 and F_2. Since $F_1, F_2 \geq 0$ by construction, and we are assuming that neither $F_1(\bar{\mathbf{w}}) > 0$ nor $F_1(\bar{\mathbf{w}}) > 0$ for any cluster point $\bar{\mathbf{w}}$, we must have $F_1(\bar{\mathbf{w}}) = F_2(\bar{\mathbf{w}}) = 0$. But this means that $\bar{\mathbf{w}}$ belongs to both sets appearing in (9), in contradiction to the assumption they are disjoint. Thus, no sequence of predictors satisfies (30), as desired.

Appendix
Alternative Notions of Regret

In the online setting, one may consider notions of regret other than (1). One choice is

$$\sum_{t=1}^{T} \ell(\langle \mathbf{w}_t, \Psi(\tilde{\mathbf{x}}_t)\rangle, y_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} \ell(\langle \mathbf{w}, \Psi(\tilde{\mathbf{x}}_t)\rangle, y_t)$$

but this is too easy, as it reduces to standard online learning with respect to examples which happen to be noisy. Another kind of regret we may want to minimize is

$$\sum_{t=1}^{T} \ell(\langle \mathbf{w}_t, \Psi(\tilde{\mathbf{x}}_t)\rangle, y_t) - \min_{\mathbf{w}\in\mathcal{W}} \ell(\langle \mathbf{w}, \Psi(\mathbf{x}_t)\rangle, y_t). \tag{31}$$

This is the kind of regret which is relevant for actually predicting the values y_t well based on the noisy instances. Unfortunately, in general this is too much to hope for. To see why, assume we deal with a linear kernel (so that $\Psi(\mathbf{x}) = \mathbf{x}$), and assume $\ell(\mathbf{w}, \mathbf{x}, y) = (\langle \mathbf{w}, \mathbf{x}\rangle - y)^2$. Now, suppose that the adversary picks some $\mathbf{w}^* \neq \mathbf{0}$ in $\mathcal{W}$, which might be even known to the learner, and at each round t provides the example $\left(\mathbf{w}^*/\left\|\mathbf{w}^*\right\|^2, 1\right)$. It is easy to verify that (31) in this case equals

$$\sum_{t=1}^{T} \left(\langle \mathbf{w}_t, \tilde{\mathbf{x}}_t\rangle - 1\right)^2 - \mathbf{0}.$$

Recall that the learner chooses $\mathbf{w}_t$ before $\tilde{\mathbf{x}}_t$ is revealed. Therefore, if the noise which leads to $\tilde{\mathbf{x}}_t$ has positive variance, it will generally be impossible for the learner to choose $\mathbf{w}_t$ such that $\langle \mathbf{w}_t, \tilde{\mathbf{x}}_t\rangle$ is arbitrarily close to 1. Therefore, the equation above cannot grow sublinearly with T.

Acknowledgment

The authors would like to thank A. Argyriou for turning our attention to Schoenberg's theorem and its applicability in our setting. This publication only reflects the authors' views.

References

[1] S. Chawla, C. Dwork, F. McSherry, A. Smith, and H. Wee, "Toward privacy in public databases," in *Proc. Theory of Cryptogr. Conf. (TCC), LNCS*, 2005, vol. 2.

[2] D. Nettleton, A. Orriols-Puig, and A. Fornells, "A study of the effect of different types of noise on the precision of supervised learning techniques," *Artif. Intell. Rev.*, 2010.

[3] N. Bshouty, J. Jackson, and C. Tamon, "Uniform-distribution attribute noise learnability," *Inf. Computat.*, vol. 187, no. 2, pp. 277–290, 2003.

[4] S. Goldman and R. Sloan, "Can pac learning algorithms tolerate random attribute noise?," *Algorithmica*, vol. 14, no. 1, pp. 70–84, 1995.

[5] M. Kearns and M. Li, "Learning in the presence of malicious errors," *SIAM J. Comput.*, vol. 22, no. 4, pp. 807–837, 1993.

[6] N. Cesa-Bianchi, E. Dichterman, P. Fischer, E. Shamir, and H. Simon, "Sample-efficient strategies for learning in the presence of noise," *J. ACM*, vol. 46, no. 5, pp. 684–719, 1999.

[7] N. Littlestone, "Redundant noisy attributes, attribute errors, and linear threshold learning using Winnow," in *Proc. COLT*, 1991, pp. 147–156.

[8] B. Schölkopf and A. Smola, *Learning with Kernels*. Cambridge: MIT Press, 2002.

[9] N. Cesa-Bianchi, A. Conconi, and C. Gentile, "On the generalization ability of on-line learning algorithms," *IEEE Trans. Inf. Theory*, vol. 50, no. 9, pp. 2050–2057, Sep. 2004.

[10] I. Steinwart and A. Christmann, *Support Vector Machines*. New York: Springer, 2008.

[11] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proc. ICML*, 2003, pp. 928–936.

[12] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*. Cambridge, U.K.: Cambridge Univ. Press, 2006.

[13] A. Flaxman, A. Kalai, and H. McMahan, "Online convex optimization in the bandit setting: Gradient descent without a gradient," in *Proc. SODA*, 2005, pp. 385–394.

[14] J. Abernethy, E. Hazan, and A. Rakhlin, "Competing in the dark: An efficient algorithm for bandit linear optimization," in *COLT*, 2008, pp. 263–274.

[15] L. Paninski, "Estimation of entropy and mutual information," *Neural Computat.*, vol. 15, no. 6, pp. 1191–1253, 2003.

[16] R. Singh, "Existence of unbiased estimates," *Sankhyā: The Indian J. Statist.*, vol. 26, no. 1, pp. 93–96, 1964.

[17] S. Bhandari and A. Bose, "Existence of unbiased estimators in sequential binomial experiments," *Sankhyā: The Indian J. Statist.*, vol. 52, no. 1, pp. 127–130, 1990.

[18] I. Schoenberg, "Metric spaces and completely monotone functions," *Ann. Math.*, vol. 39, no. 4, pp. 811–841, Oct. 1938.

[19] P. Ressel, "A short proof of Schoenberg's theorem," *Proc. AMS*, vol. 57, no. 1, pp. 66–68, May 1976.

[20] C. Micchelli, Y. Xu, and H. Zhang, "Universal kernels," *J. Mach. Learn. Res.*, vol. 7, pp. 2651–2667, 2006.

[21] C. Scovel, D. Hush, I. Steinwart, and J. Theiler, Radial Kernels and Their Reproducing Kernel Hilbert Spaces Los Alamos Nat. Lab., 2010, Tech. Rep. LA-UR-09-01900.
[22] W. Rudin, *Fourier Analysis on Groups (reprint edition)*. New York: Wiley Classics Library, 1994.
[23] P. Bartlett, M. Jordan, and J. McAuliffe, "Convexity, classification and risk bounds," *J. Amer. Statist. Assoc.*, vol. 101, no. 473, pp. 138–156, Mar. 2006.

Nicoló Cesa-Bianchi is professor of Computer Science at the University of Milano, Italy. His main research interests include statistical learning theory, game-theoretic learning, and pattern analysis. He is coauthor, with Gabor Lugosi, of the monography "Prediction, Learning, and Games" (Cambridge University Press, 2006).

Dr. Cesa-Bianchi was President of the Association for Computational Learning (2006–2009). He is a member of the steering committee of the EC-funded Network of Excellence PASCAL2, action editor for the IEEE TRANSACTIONS ON INFORMATION THEORY and for the *Journal of Machine Learning Research*. He is a recipient of a Google Research Award (2010).

Shai Shalev-Shwartz is on the faculty of the Department of Computer Science and Engineering, The Hebrew University of Jerusalem, Israel. He is interested in learning theory, online prediction, optimization, and practical algorithms.

Ohad Shamir received the Ph.D. degree from the Department of Computer Science and Engineering, The Hebrew University of Jerusalem.

He is a Postdoctoral Researcher at Microsoft Research New England, Cambridge, MA. He is broadly interested in statistical and online machine learning, with emphasis on novel algorithms which combine practical applicability and theoretical insight.