Learning from Noisy Data under Distributional Assumptions

Nicolò Cesa-Bianchi
Università degli Studi di Milano
cesa-bianchi@dsi.unimi.it

Shai Shalev Shwartz
The Hebrew University
shais@cs.huji.ac.il

Ohad Shamir
Microsoft Research
ohadsh@microsoft.com

Abstract

We study the framework of online learning, when individual examples are corrupted by random noise, and both examples and noise type can be chosen adversarially. Previous work has shown that without knowledge of the noise distribution, it is possible to learn using a random, potentially unbounded number of independent noisy copies of each example. Moreover, it is generally impossible to learn with just one noisy copy per example. In this paper, we explore the consequences of being given some side information on the noise distribution. We consider several settings, and show how one can learn linear and kernel-based predictors using just one or two noisy views of each example, depending on the side information provided.

1 Introduction

In a wide variety of machine learning applications, ranging from remote sensing through bioinformatics to medical tests, one has to deal with data known to be noisy and distorted. This is particularly common in physically measured data, due to factors such as sensor quality, communication constraints, or various other physical limitations. Other factors, such as privacy concerns, may force us deal with intentionally distorted data. 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 actually evaluated.

In [3], a general technique was introduced to deal with learning linear or kernel-based predictors from noisy data, where virtually nothing is known about the noise, except possibly an upper bound on its variance. Moreover, this was achieved in the challenging online framework, where an all-powerful adversary is able to choose the noise distribution, and change it on each and every round. This technique works assuming the learner is allowed to query for independent noisy copies of each example more than once. Moreover, the number of queries needed is a random quantity. Although the distribution of this random number does not depend on the scale of the problem, and the number is bounded with high probability, this is obviously a non-trivial requirement. Unfortunately, multiple queries cannot be avoided in general: in [3], it was shown that without seeing more than one noisy copy of each instance, learning becomes impossible, even in a stochastic, non-adversarial setting. However, this was under the crucial assumption that the learner has no prior knowledge about the noise distribution it needs to cope with.

While this is sometimes the case, in many applications the learner may actually know something about the noise distribution, either a-priori or based on previous measurements. In such cases, the impossibility result of [3] no longer holds. This leads to the central question dealt with in this paper: can we learn with a smaller, fixed number of queries per example, if we know something about the noise distribution?

Focusing on the squared loss, we discuss three different settings, reflecting different levels of knowledge about the noise distribution:

- **Known variance bound** (this is the setting discussed in [3]). We show that one can learn linear predictors with two independent noisy copies of each instance $\mathbf{x}_t$, and one noisy copy of each target value y_t.
- **Known covariance structure.** We show that one can learn linear predictors with only one noisy copy of $\mathbf{x}_t$ and y_t. This implies that the impossibility result [3] indeed does not hold in general, once we know the covariance structure of the noise.
- **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 copy of y_t. In this extended abstract, we focus on Gaussian kernels, but we also outline how our results can be extended to general radial kernels.

Thus, the positive learning results get stronger the more we can assume about the noise distribution. We emphasize that although the proposed algorithms are based on a similar online gradient descent approach, our results include techniques which are very different than those of [3], and are not just an extension. These techniques are discussed in Section 3, although we omit the formal proofs in this extended abstract.

2 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 $\mathbf{x} \mapsto \langle \mathbf{w}, \mathbf{x} \rangle$), 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, with associated kernel $k(\cdot, \cdot)$.

We begin by recalling the standard online learning setting, which for linear predictors and squared loss is defined as follows (with the obvious changes for kernel-based predictors): at each round t, the learner picks a predictor $\mathbf{w}_t$. The adversary then picks an example $(\mathbf{x}_t, y_t)$, the learner suffers a loss $\ell(\langle \mathbf{w}_t, \mathbf{x}_t \rangle, y_t) = (\langle \mathbf{w}_t, \mathbf{x}_t \rangle - y_t)^2$, and the example $(\mathbf{x}_t, y_t)$ is revealed to the learner. The goal of the learner is to minimize his regret with respect to a fixed convex set of hypotheses $\mathcal{W}$, namely $\sum_{t=1}^{T} \ell(\langle \mathbf{w}_t, \mathbf{x}_t \rangle, y_t) - \min_{\mathbf{w} \in \mathcal{W}} \sum_{t=1}^{T} \ell(\langle \mathbf{w}, \mathbf{x}_t \rangle, y_t)$. Typically, one wishes to find a strategy for the learner, such that no matter what is the adversary's strategy for choosing examples, the expression above is sublinear in T (implying that the average regret per round vanishes with T).

Following [3], we now make the following twist, which limits the information available to the learner: In each round, the adversary also picks a vector-valued random variable $\mathbf{n}_t^x$ and random variable n_t^y. Instead of receiving $(\mathbf{x}_t, y_t)$, the learner is given access to one or two independent copies of $\tilde{\mathbf{x}}_t$ and $\tilde{y}_t$, where $\tilde{\mathbf{x}}_t = \mathbf{x}_t + \mathbf{n}_t^x$, 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 change from round to round in a possibly adversarial manner. We assume throughout the paper that $\mathbf{n}_t^x$ and n_t^y are zero-mean and independent. If $\mathbb{E}[\mathbf{n}_t^x]$ and $\mathbb{E}[n_t^y]$ are not zero, and are known to the learner, we can always reduce to the zero-mean case by deducting $\mathbb{E}[\mathbf{n}_t^x]$ and $\mathbb{E}[n_t^y]$ from $\tilde{\mathbf{x}}_t$ and $\tilde{y}_t$. The independence 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.

In [3], it was assumed that nothing is known about $\mathbf{n}_t^x$ and n_t^y, other than an upper bound on the variance. In this paper, we consider settings where some side information on the distribution of $\mathbf{n}_t^x, n_t^y$ is provided to the learner.

In this framework, our goal is to minimize the expected regret in hindsight with respect to the *unperturbed* data, namely

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

where the expectation is with respect to the noise introduced into the data. Using online-to-batch conversion techniques (e.g., [1]), one can easily convert an online algorithm with a

sublinear regret in Eq. (1), to a learning algorithm in the batch statistical setting, in which the training set is corrupted by noise. Namely, the learner receives corrupted versions of examples $\{\mathbf{x}_t, y_t\}_{t=1}^T$ sampled i.i.d. from some unknown distribution, and we wish to find a predictor minimizing the expected loss $\mathbb{E}_{\mathbf{x},y}[\ell(\langle \mathbf{w}, \mathbf{x}\rangle, y)]$ over $\mathcal{W}$, in spite of the noise in the training data. Also, one can easily extend our techniques to obtain high-probability bounds on the actual regret.

3 Overview of Techniques

We begin by presenting a high-level and informal overview of the techniques we use to get our results.

A fundamental method we use in all our settings is the online gradient algorithm due to Zinkevich [4]. At its heart is the following observation: for any set of vectors $\nabla_1, \ldots, \nabla_T$, 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 scalar value. Then for any $\mathbf{u} \in \mathcal{W}$, it holds that

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

where the $O(\cdot)$ notation hides dependencies on the norms of $\mathbf{u}$ and ∇_t. In particular, suppose that we let ∇_t be the gradient of $\ell(\mathbf{w}_t, \mathbf{x}_t, y_t)$ w.r.t. $\mathbf{w}_t$. Then by convexity, the left-hand side of Eq. (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.

3.1 "Stochastic" Online Gradient Descent

In our setting of noisy data, the algorithm described above is irrelevant, because $(\mathbf{x}_t, y_t)$ is unknown and we cannot compute ∇_t. However, one can deal with it as follows. Suppose that instead of ∇_t, we pick *random* vectors $\tilde{\nabla}_t$. It turns out that based on Eq. (2), one can still show that $\mathbb{E}\left[\sum_{t=1}^T \langle \mathbf{w}_t - \mathbf{u}, \tilde{\nabla}_t\rangle\right] = O(\sqrt{T})$, where the $O(\cdot)$ notation hides dependencies on the norm of $\mathbf{u}$ and $\mathbb{E}\left[\|\tilde{\nabla}_t\|^2\right]$ for all t. In particular, in our noisy data setting, we cannot compute ∇_t, but suppose we can use the noisy data that we do have, in order to construct a random vector $\tilde{\nabla}_t$, such that $\mathbb{E}[\tilde{\nabla}_t] = \nabla_t$. In that case, the left-hand side 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 w.r.t. the noisy examples (note that $\mathbf{w}_t$ is a random vector that depends on the noisy examples). Applying the same convexity trick 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 sublinear regret bound. The only technical issue is how to construct unbiased estimates $\tilde{\nabla}_t$ using the noisy data.

This is the basic learning mechanism in the first two settings we consider, in Section 4. This technique already appears in [3] (as well as previous work in other settings, e.g., [2]), and our main contribution for these two settings is the observation that it can be done with one or two noisy copies of each example, under appropriate distributional assumptions. The third setting we consider, for kernel-based predictors (Section 5), is where the main technical novelty of this paper lies, as it requires a rather different approach than that of [3]. This approach is discussed below.

3.2 "Parallel Worlds" Online Gradient Descent

With kernels, one assumes that the predictors lie in some reproducing kernel Hilbert space (RKHS), in which inner products can be efficiently computed using a kernel function. Instances are mapped into that space using a feature mapping Ψ. In [3], learning was achieved by constructing an unbiased estimate $\tilde{\nabla}_t$ of ∇_t in the RKHS, but that required a random number of noisy copies of each example. Here, we take the following very different approach:

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

$$\mathbb{E}\left[\langle\hat{\Psi}(\mathbf{a}),\hat{\Psi}(\tilde{\mathbf{x}}_t)\rangle\right]=\langle\Psi(\mathbf{a}),\Psi(\mathbf{x}_t)\rangle \tag{3}$$

where the expectation is with respect to the noise. Thus, "noisy" inner products in the surrogate RKHS correspond (in expectation) to "clean" 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 Zinkevich's algorithm 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 $\hat{\mathbf{u}}$, it holds that $\mathbb{E}\left[\sum_{t=1}^{T}\langle\hat{\mathbf{w}}_t-\hat{\mathbf{u}},\hat{\nabla}_t\rangle\right]=\mathbb{E}\left[\sum_{t=1}^{T}\langle\mathbf{w}_t-\mathbf{u},\nabla_t\rangle\right]$, where $\mathbf{w}_t$ and $\mathbf{u}$ are certain mappings of each $\hat{\mathbf{w}}_t$ and $\hat{\mathbf{u}}$ to the RKHS of Ψ, and ∇_t are the gradients w.r.t. the unperturbed examples $(\mathbf{x}_t,y_t)$. But the left-hand side is $O(\sqrt{T})$, since we applied Zinkevich's algorithm in the surrogate RKHS. 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 approach used for the other settings, $\mathbb{E}[\hat{\nabla}_t]$ *is not equal* to ∇_t. Indeed, they live in different mathematical spaces! Thus, this is a different approach rather than an extension of the technique discussed in Subsec. 3.1.

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.

4 Setting 1+2: Knowledge on the Covariance

We begin with the simplest setting, which is when we only know that $\mathbb{E}[\|\tilde{\mathbf{x}}_t\|^2]\leq B_{\tilde{\mathbf{x}}}^2$ and $\mathbb{E}\left[\tilde{y}_t^2\right]\leq B_{\tilde{y}}^2$ for some parameters[1] $B_{\tilde{\mathbf{x}}},B_{\tilde{y}}$. While this is a similar setting to that discussed in [3], the results there focus on learning with general kernels and loss functions, using a random number of independent noisy copies of each example. Here, 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 3, the algorithm is based on an adaptation of the online convex optimization algorithm due to [4], and the main requirement is to construct an unbiased estimate of the gradient ∇_t. The following theorem provides a regret bound for Algorithm 1 (left).

Algorithm 1 Linear Regression with Noise.

Upper bound on variance	**Known covariance**
PARAMETERS: $\eta, B_{\mathbf{w}}$	PARAMETERS: $\eta, B_{\mathbf{w}}$
INITIALIZE: $\mathbf{w}_1=\mathbf{0}$	INITIALIZE: $\mathbf{w}_1=\mathbf{0}$
For $t=1,2,\ldots,T$	For $t=1,2,\ldots,T$
Receive $(\tilde{\mathbf{x}}_t,\tilde{y}_t)$	Receive $(\tilde{\mathbf{x}}_t,\tilde{y}_t)$
Receive another independent copy $\tilde{\mathbf{x}}'_t$	
$\tilde{\nabla}_t=2(\langle\mathbf{w}_t,\tilde{\mathbf{x}}_t\rangle-\tilde{y}_t)\tilde{\mathbf{x}}'_t$	$\tilde{\nabla}_t=2(\langle\mathbf{w}_t,\tilde{\mathbf{x}}_t\rangle-\tilde{y}_t)\tilde{\mathbf{x}}_t-\Sigma_t\mathbf{w}_t$
$\mathbf{w}'=\mathbf{w}_t-\eta\tilde{\nabla}_t$	$\mathbf{w}'=\mathbf{w}_t-\eta\tilde{\nabla}_t$
$\mathbf{w}_{t+1}=\min\{1,B_{\mathbf{w}}/\|\mathbf{w}'\|\}\,\mathbf{w}'$	$\mathbf{w}_{t+1}=\min\{1,B_{\mathbf{w}}/\|\mathbf{w}'\|\}\,\mathbf{w}'$

Theorem 1 *Assume that* $\mathbb{E}[\|\tilde{\mathbf{x}}_t\|^2]\leq B_{\tilde{\mathbf{x}}}^2$, $\mathbb{E}\left[\tilde{y}_t^2\right]\leq B_{\tilde{y}}^2$, *and* $\tilde{\mathbf{x}}_t,\tilde{\mathbf{x}}'_t,\tilde{y}_t$ *are mutually independent. Then if we run Algorithm 1 (left) with parameters* $B_{\mathbf{w}}$, $\eta=B_{\mathbf{w}}/\sqrt{GT}$, *where* $G=4(B_{\mathbf{w}}^2B_{\tilde{\mathbf{x}}}^2+B_{\tilde{y}}^2)B_{\tilde{\mathbf{x}}}^2$, *the expected regret w.r.t. any* $\mathbf{w}$ *with norm not greater than* $B_{\mathbf{w}}$ *is at most* $B_{\mathbf{w}}\sqrt{GT}$.

[1] Strictly speaking, this is an upper bound on the raw second moment, not the variance. However, it is easily implied by assuming a bound on the noise variance, and the norm of the unperturbed data.

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 $\mathbb{E}[\tilde{y}_t^2]$ (with our algorithmic approach, knowing $\mathbb{E}[\tilde{y}_t^2]$ 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 earlier setting, 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 (left) to work. See Algorithm 1 (right) for the pseudocode. The regret bound obtained is similar in flavor to Theorem 1, and is skipped for brevity.

5 Setting 3: Gaussian Distribution

The final and most complex setting we consider 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 the previous section 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.

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 (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, but note that our techniques can also be extended 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 section, 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. As to $\tilde{y}_t$, similar to the previous settings, we will only need to assume that $\mathbb{E}[\tilde{y}_t^2] \leq B_{\tilde{y}}^2$ for some parameter $B_{\tilde{y}}$.

The algorithm that we present (Algorithm 2) 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 $\tilde{\mathbf{y}}_t$. As in the linear case, the learning algorithm that we use is based on online gradient descent, with the main twist being that instead of using a Gaussian kernel of width s^2, we use a surrogate kernel, as discussed in Section 3. This surrogate is defined as

$$\hat{k}(\mathbf{x}, \mathbf{x}') = R_{\Sigma,s,d}^2 \exp\left(-\sum_{i=1}^{d} \frac{(x_i - x_i')^2}{s^2 - 2\sigma_i^2}\right), \quad \text{where} \quad R_{\Sigma,s,d} = \left(\prod_{i=1}^{d} \frac{s^2}{s^2 - 2\sigma_i^2}\right)^{1/4}. \tag{4}$$

This can be shown to be a kernel by standard results. Note that we generally assume that $2\|\Sigma\| = 2\max_i \sigma_i^2 < s^2$ for all i. Moreover, $R_{\Sigma,s,d}$ 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 instances $\mathbf{x}$ provided are of order $\Theta(1)$.

Denote $\hat{\Psi}$ to be the feature mapping corresponding to this RKHS. The pseudocode of our algorithm is presented below. Formally speaking, it is just applying Zinkevich's algorithm, 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 2 we need an auxiliary definition. Suppose that $\mathbf{u}$ 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 $a_1, \ldots, a_m \geq 0$ (for example, this includes $\operatorname{argmin}_{\mathbf{w}:\|\mathbf{w}\| \leq B_{\mathbf{w}}} \sum_{t=1}^{T} \ell(\langle \mathbf{w}_t, \Psi(\mathbf{x}_t) \rangle, y_t)$ for any $B_{\mathbf{w}} > 0$ by the representer theorem). Define $\beta_{\mathbf{u}}$ 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, 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 $\beta_{\mathbf{u}}$ to be $\pi/2$. The main theorem is the following.

Algorithm 2 Kernel Learning Algorithm with Gaussian Noise $\mathcal{N}(0,\Sigma)$

PARAMETERS: W, η
INITIALIZE: $\alpha_i := 0$ for all $i = 1 \dots T$
For $t = 1, \dots, T$:
  Define $\hat{\mathbf{w}}_t = \sum_{i=1}^{t-1} \alpha_i \hat{\Psi}(\tilde{\mathbf{x}}_i)$, and 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 w.r.t. $\hat{\Psi}(\tilde{\mathbf{x}}'_t)$ 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 $r_t > W^2$, let $\alpha_i := \alpha_i \frac{W}{\sqrt{r_t}}$ for all $i = 1, \dots, t$

Theorem 2 *Assume that the noise introduced into $\mathbf{x}_t$ has a known distribution $\mathcal{N}(0,\Sigma)$, with Σ diagonal and $2\|\Sigma\| \le s^2$ (where s^2 is the kernel width), and that the noise introduced into y_t is independent with $\mathbb{E}[\tilde{y}_t^2] \le B_{\tilde{y}}^2$. Let $B_{\mathbf{w}} > 0, \beta \in (0, \pi/2]$ be fixed. Then if we run Algorithm 2 with any $W \ge \frac{\sqrt{5} B_{\mathbf{w}} R_{\Sigma,s,d}}{\sin(\beta)}$, and $\eta = W/2R\sqrt{(W^2 R_{\Sigma,s,d}^2 + B_{\tilde{y}}^2)T}$, then for all $\mathbf{u}$ with $\|\mathbf{u}\| \le B_{\mathbf{w}}, \beta_{\mathbf{u}} \ge \beta$, it holds that the expected regret w.r.t. $\mathbf{u}$ is at most $2WR_{\Sigma,s,d}\sqrt{(W^2 R_{\Sigma,s,d}^2 + B_{\tilde{y}}^2)T}$. In particular, if $s^2 = \Omega(d)$, $\|\Sigma\| = O(1)$, and $B_{\tilde{y}} = O(1)$, then the expected regret is at most $O(W^2\sqrt{T})$.*

The intuition for $\beta_{\mathbf{u}}$ 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, $\beta_{\mathbf{u}}$ 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, $\beta_{\mathbf{u}}$ 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 $\mathbf{u}$ with $\beta_{\mathbf{u}}$ not too small is a rather mild assumption, if something interesting can be learned even on the unperturbed data.

Finally, we note that the techniques we use here can be extended, in a certain sense, to the large family of *radial kernels*, that is kernels on $\mathbf{x}, \mathbf{x}'$ which can be written as a function of $\|\mathbf{x} - \mathbf{x}'\|$, such as $\exp(-\|\mathbf{x} - \mathbf{x}'\|/s)$ and $(1 + \|\mathbf{x} - \mathbf{x}'\|^2/m)^{-\alpha}$ for appropriate s, m, α. The key idea is to reduce the problem to the Gaussian kernel case, using Schoenberg's theorem, which implies that any radial kernel can be written as an integral over Gaussian kernels. Constructing a surrogate kernel for each of these Gaussian kernels (as in Eq. (4)) and integrating leads to a surrogate kernel which enjoys the properties required for Algorithm 2 to work with the desired radial kernel. We skip the details in this extended abstract.

References

[1] N. Cesa-Bianchi, A. Conconi, and C. Gentile. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50(9):2050–2057, September 2004.

[2] A. Flaxman, A. Tauman Kalai, and H. McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of SODA*, pages 385–394, 2005.

[3] Shai Shalev-Shwartz Nicolò Cesa-Bianchi and Ohad Shamir. Online learning of noisy data with kernels. In *Proceedings of COLT*, 2010.

[4] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on Machine Learning*, 2003.