Learning Kernel-Based Halfspaces with the Zero-One Loss

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

Ohad Shamir
The Hebrew University
ohadsh@cs.huji.ac.il

Karthik Sridharan
Toyota Technological Institute
karthik@tti-c.org

Abstract

We describe and analyze a new algorithm for agnostically learning kernel-based halfspaces with respect to the *zero-one* loss function. Unlike most previous formulations which rely on surrogate convex loss functions (e.g. hinge-loss in SVM and log-loss in logistic regression), we provide finite time/sample guarantees with respect to the more natural zero-one loss function. The proposed algorithm can learn kernel-based halfspaces in worst-case time $\mathrm{poly}(\exp(L\log(L/\epsilon)))$, for *any* distribution, where L is a Lipschitz constant (which can be thought of as the reciprocal of the margin), and the learned classifier is worse than the optimal halfspace by at most ϵ. We also prove a hardness result, showing that under a certain cryptographic assumption, no algorithm can learn kernel-based halfspaces in time polynomial in L.

1 Introduction

A highly important hypothesis class in machine learning theory and applications is that of halfspaces in a Reproducing Kernel Hilbert Space (RKHS). Choosing a halfspace based on empirical data is often performed using Support Vector Machines (SVMs) [25]. SVMs replace the more natural 0-1 loss function with a convex surrogate – the hinge-loss. By doing so, we can rely on convex optimization tools. However, there are no guarantees on how well the hinge-loss approximates the 0-1 loss function. There do exist some recent results on the *asymptotic* relationship between surrogate convex loss functions and the 0-1 loss function [27, 4], but these do not come with finite-sample or finite-time guarantees. In this paper, we tackle the task of learning kernel-based halfspaces with respect to the non-convex 0-1 loss function. Our goal is to derive learning algorithms and to analyze them in the finite-sample finite-time setting.

Following the standard statistical learning framework, we assume that there is an unknown distribution, $\mathcal{D}$, over the set of labeled examples, $\mathcal{X} \times \{0,1\}$, and our primary goal is to find a classifier, $h : \mathcal{X} \to \{0,1\}$, with low generalization error,

$$\mathrm{err}_{\mathcal{D}}(h) \stackrel{\mathrm{def}}{=} \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}} [|h(\mathbf{x}) - y|] \ . \tag{1}$$

The learning algorithm is allowed to sample a training set of labeled examples, $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$, where each example is sampled i.i.d. from $\mathcal{D}$, and it returns a classifier. Following the agnostic PAC learning framework [15], we say that an algorithm (ϵ, δ)-learns a concept class H of classifiers using m examples, if with probability of at least $1 - \delta$ over a random choice of m examples the algorithm returns a classifier $\hat{h}$ that satisfies

$$\mathrm{err}_{\mathcal{D}}(\hat{h}) \ \leq \ \inf_{h\in H} \mathrm{err}_{\mathcal{D}}(h) + \epsilon \ . \tag{2}$$

We note that $\hat{h}$ does not necessarily belong to H. Namely, we are concerned with *improper* learning, which is as useful as proper learning for the purpose of deriving good classifiers. A common learning paradigm is the Empirical Risk Minimization (ERM) rule, which returns a classifier that minimizes the average error over the training set,

$$\hat{h} \in \mathop{\mathrm{argmin}}_{h\in H} \frac{1}{m} \sum_{i=1}^{m} |h(\mathbf{x}_i) - y_i| \ .$$

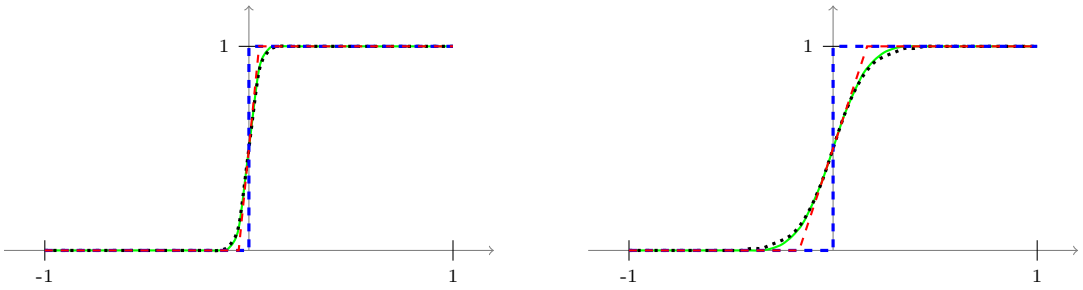

Figure 1: Illustrations of transfer functions for $L = 10$ (left) and $L = 3$ (right): the 0-1 transfer function (dashed blue line); the sigmoid transfer function (dotted black line); the erf transfer function (green line); the piece-wise linear transfer function (dashed red line).

The class of (origin centered) halfspaces is defined as follows. Let $\mathcal{X}$ be a compact subset of a RKHS, which w.l.o.g. will be taken to be the unit ball around the origin. Let $\phi_{0-1} : \mathbb{R} \to \mathbb{R}$ be the function $\phi_{0-1}(a) = \mathbf{1}(a \geq 0) = \frac{1}{2}(\mathrm{sgn}(a) + 1)$. The class of halfspaces is the set of classifiers

$$H_{\phi_{0-1}} \stackrel{\text{def}}{=} \{\mathbf{x} \mapsto \phi_{0-1}(\langle \mathbf{w}, \mathbf{x} \rangle) \,:\, \mathbf{w} \in \mathcal{X}\} \,.$$

Although we represent the halfspace using $\mathbf{w} \in \mathcal{X}$, which is a vector in the RKHS whose dimensionality can be infinite, in practice we only need a function that implements inner products in the RKHS (a.k.a. a kernel function), and one can define $\mathbf{w}$ as the coefficients of a linear combination of examples in our training set. To simplify the notation throughout the paper, we represent $\mathbf{w}$ simply as a vector in the RKHS.

It is well known that if the dimensionality of $\mathcal{X}$ is n, then the VC dimension of $H_{\phi_{0-1}}$ equals n. This implies that the number of training examples required to obtain a guarantee of the form given in Equation (2) for the class of halfspaces scales at least linearly with the dimension n [25]. Since kernel-based learning algorithms allow $\mathcal{X}$ to be an infinite dimensional inner product space, we must use a different class in order to obtain a guarantee of the form given in Equation (2).

One way to define a slightly different concept class is to approximate the non-continuous function, ϕ_{0-1}, with a Lipschitz continuous function, $\phi : \mathbb{R} \to [0, 1]$, which is often called a transfer function. For example, we can use a sigmoidal transfer function

$$\phi_{\mathrm{sig}}(a) \stackrel{\text{def}}{=} \frac{1}{1 + \exp(-4L\,a)} \,, \tag{3}$$

which is a L-Lipschitz function. Other L-Lipschitz transfer functions are the erf function and the piece-wise linear function:

$$\phi_{\mathrm{erf}}(a) \stackrel{\text{def}}{=} \tfrac{1}{2}\left(1 + \mathrm{erf}\left(\sqrt{\pi}\,L\,a\right)\right) \quad , \quad \phi_{\mathrm{pw}}(a) \stackrel{\text{def}}{=} \max\left\{\min\left\{\tfrac{1}{2} + L\,a\,,\, 1\right\}\, 0\right\} \tag{4}$$

An illustration of these transfer functions is given in Figure 1. Analogously to the definition of $H_{\phi_{0-1}}$, for a general transfer function ϕ we define H_ϕ to be the set of predictors $\mathbf{x} \mapsto \phi(\langle \mathbf{w}, \mathbf{x} \rangle)$. Since now the range of ϕ is not $\{0, 1\}$ but rather the entire interval $[0, 1]$, we interpret $\phi(\langle \mathbf{w}, \mathbf{x} \rangle)$ as the probability to output the label 1. The definition of $\mathrm{err}_{\mathcal{D}}(h)$ remains[1] as in Equation (1).

The advantage of using a Lipschitz transfer function can be seen via Rademacher generalization bounds [3]. In fact, a simple corollary of the contraction lemma implies the following:

Theorem 1 *Let $\epsilon, \delta \in (0, 1)$ and let ϕ be an L-Lipschitz transfer function. Let m be an integer satisfying*

$$m \;\geq\; \left(\frac{2L + 3\sqrt{2\ln(8/\delta)}}{\epsilon}\right)^2 \,.$$

Then, for any distribution $\mathcal{D}$ over $\mathcal{X} \times \{0, 1\}$, the ERM algorithm (ϵ, δ)-learns the concept class H_ϕ using m examples.

[1] Note that in this case $\mathrm{err}_{\mathcal{D}}(h)$ can be interpreted as $\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}, b\sim\phi(\langle \mathbf{w},\mathbf{x}\rangle)}[y \neq b]$.

The above theorem tells us that the sample complexity of learning H_ϕ is $\tilde{\Omega}(L^2/\epsilon^2)$. Crucially, the sample complexity does not depend on the dimensionality of $\mathcal{X}$, but only on the Lipschitz constant of the transfer function. This allows us to learn with kernels, when the dimensionality of $\mathcal{X}$ can even be infinite. A related analysis compares the error rate of a halfspace $\mathbf{w}$ to the number of margin mistakes $\mathbf{w}$ makes on the training set - see Section 4.1 for a comparison.

From the computational complexity point of view, the result given in Theorem 1 is problematic, since the ERM algorithm should solve the non-convex optimization problem

$$\operatorname*{argmin}_{\mathbf{w}:\|\mathbf{w}\|\leq 1} \frac{1}{m}\sum_{i=1}^{m} |\phi(\langle \mathbf{w}, \mathbf{x}_i\rangle) - y_i| \ . \tag{5}$$

Solving this problem in polynomial time is hard under reasonable assumptions (see Section 3 in which we present a formal hardness result). Adapting a technique due to [6] we show in the full version of this paper [22] that it is possible to find an ϵ-accurate solution to Equation (5) (where the transfer function is ϕ_{pw}) in time poly $\left(\exp\left(\frac{L^2}{\epsilon^2}\log(\frac{L}{\epsilon})\right)\right)$. The main contribution of this paper is the derivation and analysis of a more simple learning algorithm that (ϵ,δ)-learns the class H_{sig} using time and sample complexity of at most poly $\left(\exp\left(L\log(\frac{L}{\epsilon})\right)\right)$. That is, the runtime of our algorithm is exponentially smaller than the runtime required to solve the ERM problem using the technique described in [6]. Moreover, the algorithm of [6] performs an exhaustive search over all $(L/\epsilon)^2$ subsets of the m examples in the training set, and therefore its runtime is always order of m^{L^2/ϵ^2}. In contrast, our algorithm's runtime depends on a parameter B, which is bounded by $\exp(L)$ only under a worst-case assumption. Depending on the underlying distribution, B can be much smaller than the worst-case bound. In practice, we will cross-validate for B, and therefore the worst-case bound will often be pessimistic.

The rest of the paper is organized as follows. In Section 2 we describe our main results. Next, in Section 3 we provide a hardness result, showing that it is not likely that there exists an algorithm that learns H_{sig} or H_{pw} in time polynomial in L. We outline additional related work in Section 4. In particular, the relation between our approach and margin-based analysis is described in Section 4.1, and the relation to approaches utilizing a distributional assumption is discussed in Section 4.2. We wrap up with a discussion in Section 5.

2 Main Results

In this section we present our main result. Recall that we would like to derive an algorithm which learns the class H_{sig}. However, the ERM optimization problem associated with H_{sig} is non-convex. The main idea behind our construction is to learn a larger hypothesis class, denoted H_B, which approximately contains H_{sig}, and for which the ERM optimization problem becomes convex. The price we need to pay is that from the statistical point of view, it is more difficult to learn the class H_B than the class H_{sig}, therefore the sample complexity increases.

The class H_B we use is a class of *linear* predictors in some other RKHS. The kernel function that implements the inner product in the newly constructed RKHS is

$$K(\mathbf{x},\mathbf{x}') \stackrel{\text{def}}{=} \frac{1}{1-\nu\langle \mathbf{x},\mathbf{x}'\rangle} \ , \tag{6}$$

where $\nu \in (0,1)$ is a parameter and $\langle \mathbf{x},\mathbf{x}'\rangle$ is the inner product in the original RKHS. As mentioned previously, $\langle \mathbf{x},\mathbf{x}'\rangle$ is usually implemented by some kernel function $K'(\mathbf{z},\mathbf{z}')$, where $\mathbf{z}$ and $\mathbf{z}'$ are the pre-images of $\mathbf{x}$ and $\mathbf{x}'$ with respect to the feature mapping induced by K'. Therefore, the kernel in Equation (6) is simply a composition with K', i.e. $K(\mathbf{z},\mathbf{z}') = 1/(1-\nu K'(\mathbf{z},\mathbf{z}'))$.

To simplify the presentation we will set $\nu = 1/2$, although in practice other choices might be more effective. It is easy to verify that K is a valid positive definite kernel function (see for example [19, 10]). Therefore, there exists some mapping $\psi : \mathcal{X} \to \mathbb{V}$, where $\mathbb{V}$ is an RKHS with $\langle \psi(\mathbf{x}), \psi(\mathbf{x}')\rangle = K(\mathbf{x},\mathbf{x}')$. The class H_B is defined to be:

$$H_B \stackrel{\text{def}}{=} \{\mathbf{x} \mapsto \langle \mathbf{v}, \psi(\mathbf{x})\rangle \ : \ \mathbf{v} \in \mathbb{V}, \ \|\mathbf{v}\|^2 \leq B\} \ . \tag{7}$$

The main result we prove in this section is the following:

Theorem 2 *Let $\epsilon,\delta \in (0,1)$ and let $L \geq 3$. Let $B = 2L^4 + \exp\left(7L\log\left(\frac{2L}{\epsilon}\right)+3\right)$ and let m be a sample size that satisfies $m \geq \frac{8B}{\epsilon^2}\left(2+9\sqrt{\ln(8/\delta)}\right)^2$. Then, for any distribution $\mathcal{D}$, with probability of at least $1-\delta$, any ERM predictor $\hat{h} \in H_B$ with respect to H_B satisfies*

$$\text{err}_{\mathcal{D}}(\hat{h}) \leq \min_{h\in H_{\text{sig}}} \text{err}_{\mathcal{D}}(h_{\text{sig}}) + \epsilon \ .$$

We note that the bound on B is far from being the tightest possible in terms of constants and second-order terms. Also, the assumption of $L \geq 3$ is rather arbitrary, and is meant to simplify the presentation of the bound.

To prove this theorem, we start with analyzing the time and sample complexity of learning H_B. The sample complexity analysis follows directly from a Rademacher generalization bound [3]. In particular, the following theorem tells us that the sample complexity of learning H_B with the ERM rule is order of B/ϵ^2 examples.

Theorem 3 *Let $\epsilon, \delta \in (0,1)$, let $B \geq 1$, and let m be a sample size that satisfies*

$$m \;\geq\; \frac{2B}{\epsilon^2}\left(2 + 9\sqrt{\ln(8/\delta)}\right)^2 \;.$$

Then, for any distribution $\mathcal{D}$, the ERM algorithm (ϵ, δ)-learns H_B.

Proof Since $K(\mathbf{x}, \mathbf{x}) \leq 2$, the Rademacher complexity of H_B is bounded by $\sqrt{2B/m}$ (see also [13]). Additionally, using Cauchy-Schwartz inequality we have that the loss is bounded, $|\langle \mathbf{v}, \psi(\mathbf{x})\rangle - y| \leq \sqrt{2B} + 1$. The result now follows directly from [3, 13]. ■

Next, we show that the ERM problem with respect to H_B can be solved in time poly(m). The ERM problem associated with H_B is

$$\min_{\mathbf{v}:\|\mathbf{v}\|^2 \leq B} \frac{1}{m}\sum_{i=1}^{m} |\langle \mathbf{v}, \psi(\mathbf{x}_i)\rangle - y_i| \;.$$

Since the objective function is defined only via inner products with $\psi(\mathbf{x}_i)$, and the constraint on $\mathbf{v}$ is defined by the ℓ_2-norm, it follows by the Representer theorem [26] that there is an optimal solution $\mathbf{v}^\star$ that can be written as $\mathbf{v}^\star = \sum_{i=1}^m \alpha_i \psi(\mathbf{x}_i)$. Therefore, instead of optimizing over $\mathbf{v}$, we can optimize over the set of weights $\alpha_1, \ldots, \alpha_m$ by solving the equivalent optimization problem

$$\min_{\alpha_1,\ldots,\alpha_m} \frac{1}{m}\sum_{i=1}^{m}\left|\sum_{j=1}^{m}\alpha_j K(\mathbf{x}_j, \mathbf{x}_i) - y_i\right| \quad \text{s.t.} \quad \sum_{i,j=1}^{m}\alpha_i\alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \leq B \;.$$

This is a convex optimization problem in $\mathbb{R}^m$ and therefore can be solved in time poly(m) using standard optimization tools.[2] We therefore obtain:

Corollary 1 *Let $\epsilon, \delta \in (0,1)$ and let $B \geq 1$. Then, for any distribution $\mathcal{D}$, it is possible to (ϵ, δ)-learn H_B in sample and time complexity of* poly$\left(\frac{B}{\epsilon}\log(1/\delta)\right)$.

It is left to understand why the class H_B approximately contains the class H_{sig}. Recall that for any transfer function, ϕ, we define the class H_ϕ to be all the predictors of the form $\mathbf{x} \mapsto \phi(\langle \mathbf{w}, \mathbf{x}\rangle)$. The first step is to show that H_B contains the union of H_ϕ over all polynomial transfer functions that satisfy a certain boundedness condition on their coefficients.

Lemma 1 *Let P_B be the following set of polynomials (possibly with infinite degree)*

$$P_B \stackrel{\text{def}}{=} \left\{ p(a) = \sum_{j=0}^{\infty}\beta_j a^j \;:\; \sum_{j=0}^{\infty}\beta_j^2 2^j \leq B \right\} \;. \tag{8}$$

Then,

$$\bigcup_{p \in P_B} H_p \;\subset\; H_B \;.$$

Proof To simplify the proof, we first assume that $\mathcal{X}$ is simply the unit ball in $\mathbb{R}^n$, for an arbitrarily large but finite n. Consider the mapping $\psi : \mathcal{X} \to \mathbb{R}^{\mathbb{N}}$ defined as follows: for any $\mathbf{x} \in \mathcal{X}$, we let $\psi(\mathbf{x})$ be an infinite vector, indexed by $k_1 \ldots, k_j$ for all $(k_1, \ldots, k_j) \in \{1, \ldots, n\}^j$ and $j = 0 \ldots \infty$, where

[2] In fact, using stochastic gradient descent, we can (ϵ, δ)-learn H_B in time $O(m^2)$, where m is as defined in Theorem 3 —See for example [8, 20].

the entry at index $k_1 \ldots, k_j$ equals $2^{-j/2} x_{k_1} \cdot x_{k_2} \cdots x_{k_j}$. The inner-product between $\psi(\mathbf{x})$ and $\psi(\mathbf{x}')$ for any $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ can be calculated as follows,

$$\langle \psi(\mathbf{x}), \psi(\mathbf{x}') \rangle \;=\; \sum_{j=0}^{\infty} \sum_{(k_1,\ldots,k_j)\in\{1,\ldots,n\}^j} 2^{-j} x_{k_1} x'_{k_1} \cdots x_{k_j} x'_{k_j} \;=\; \sum_{j=0}^{\infty} 2^{-j} (\langle \mathbf{x}, \mathbf{x}' \rangle)^j \;=\; \frac{1}{1 - \frac{1}{2}\langle \mathbf{x}, \mathbf{x}' \rangle}.$$

This is exactly the kernel function defined in Equation (6) (recall that we set $\nu = 1/2$) and therefore ψ maps to the RKHS defined by K. Consider any polynomial $p(a) = \sum_{j=0}^{\infty} \beta_j a^j$ in P_B, and any $\mathbf{w} \in \mathcal{X}$. Let $\mathbf{v}_{\mathbf{w}}$ be an element in $\mathbb{R}^{\mathbb{N}}$ explicitly defined as being equal to $\beta_j 2^{j/2} w_{k_1} \cdots w_{k_j}$ at index $k_1, \ldots, k_j$ (for all $k_1, \ldots, k_j \in \{1, \ldots, n\}^j, j = 0 \ldots \infty$). By definition of ψ and $\mathbf{v}_{\mathbf{w}}$, we have that

$$\langle \mathbf{v}_{\mathbf{w}}, \psi(\mathbf{x}) \rangle = \sum_{j=0}^{\infty} \sum_{k_1,\ldots,k_j} 2^{-j/2} \beta_j 2^{j/2} w_{k_1} \cdots w_{k_j} x_{k_1} \cdot \cdots \cdot x_{k_j} = \sum_{j=0}^{\infty} \beta_j (\langle \mathbf{w}, \mathbf{x} \rangle)^j = p(\langle \mathbf{w}, \mathbf{x} \rangle) \ .$$

In addition,

$$\|\mathbf{v}_{\mathbf{w}}\|^2 = \sum_{j=0}^{\infty} \sum_{k_1,\ldots,k_j} \beta_j^2 2^j w_{k_1}^2 \cdots w_{k_j}^2 = \sum_{j=0}^{\infty} \beta_j^2 2^j \sum_{k_1} w_{k_1}^2 \sum_{k_2} w_{k_2}^2 \cdots \sum_{k_j} w_{k_j}^2 = \sum_{j=0}^{\infty} \beta_j^2 2^j \left(\|\mathbf{w}\|^2\right)^j \leq B.$$

Thus, the predictor $\mathbf{x} \mapsto \langle \mathbf{v}_{\mathbf{w}}, \psi(\mathbf{x}) \rangle$ belongs to H_B and is the same as the predictor $\mathbf{x} \mapsto p(\langle \mathbf{w}, \mathbf{x} \rangle)$. This proves that $H_p \subset H_B$ for all $p \in P_B$ as required. Finally, if $\mathcal{X}$ is an infinite dimensional RKHS, the only technicality is that in order to represent $\mathbf{x}$ as a (possibly infinite) vector, we need to show that our RKHS has a countable basis. This holds since the inner product $\langle \mathbf{x}, \mathbf{x}' \rangle$ over $\mathcal{X}$ is continuous and bounded (see [1]). ■

Finally, the following lemma states that with a sufficiently large B, there exists a polynomial in P_B which approximately equals to ϕ_{sig}. This implies that H_B approximately contains H_{sig}.

Lemma 2 *Let ϕ_{sig} be as defined in Equation (3), where for simplicity we assume $L \geq 3$. For any $\epsilon > 0$, let*

$$B = 2L^4 + \exp\left(7L \log\left(\tfrac{2L}{\epsilon}\right) + 3\right).$$

Then there exists $p \in P_B$ such that

$$\forall \mathbf{x}, \mathbf{w} \in \mathcal{X}, \quad |p(\langle \mathbf{w}, \mathbf{x} \rangle) - \phi_{\text{sig}}(\langle \mathbf{w}, \mathbf{x} \rangle)| \leq \epsilon \ .$$

The proof of the lemma is based on a Chebyshev approximation technique and is given in the full version of our paper [22]. Since the proof is rather involved, we also present a similar lemma, whose proof is simpler, for the ϕ_{erf} transfer function (see [22]). It is interesting to note that ϕ_{erf} actually *belongs* to P_B for a sufficiently large B, since it can be defined via its infinite-degree Taylor expansion. However, the bound for ϕ_{erf} depends on $\exp(L^2)$, rather than $\exp(L)$ for the sigmoid transfer function ϕ_{sig}.

Finally, Theorem 2 is obtained as follows: Combining Theorem 3 and Lemma 1 we get that with probability of at least $1 - \delta$,

$$\text{err}_{\mathcal{D}}(\hat{h}) \;\leq\; \min_{h \in H_B} \text{err}_{\mathcal{D}}(h) + \epsilon/2 \leq \min_{p \in P_B} \min_{h \in H_p} \text{err}_{\mathcal{D}}(h) + \epsilon/2 \ . \tag{9}$$

From Lemma 2 we obtain that for any $\mathbf{w} \in \mathcal{X}$, if $h(\mathbf{x}) = \phi_{\text{sig}}(\langle \mathbf{w}, \mathbf{x} \rangle)$ then there exists a polynomial $p_0 \in P_B$ such that if $h'(\mathbf{x}) = p_0(\langle \mathbf{w}, \mathbf{x} \rangle)$ then $\text{err}_{\mathcal{D}}(h') \leq \text{err}_{\mathcal{D}}(h) + \epsilon/2$. Since it holds for all $\mathbf{w}$, we get that

$$\min_{p \in P_B} \min_{h \in H_p} \text{err}_{\mathcal{D}}(h) \leq \min_{h \in H_{\text{sig}}} \text{err}_{\mathcal{D}}(h) + \epsilon/2 \ .$$

Combining this with Equation (9), Theorem 2 follows.

3 Hardness

In this section we derive a hardness result for agnostic learning of H_{sig} or H_{pw} with respect to the zero-one loss. The hardness result relies on the hardness of standard (non-agnostic)[3] PAC learning of intersection of halfspaces given in Klivans and Sherstov [16] (see also similar arguments in [11]). The hardness result is representation-independent —it makes no restrictions on the learning algorithm and in particular also holds for improper learning algorithms. The hardness result is based on the following cryptographic assumption:

[3] In the *standard* PAC model, we assume that some hypothesis in the class has $\text{err}_{\mathcal{D}}(h) = 0$, while in the *agnostic* PAC model, which we study in this paper, $\text{err}_{\mathcal{D}}(h)$ might be strictly greater than zero for all $h \in H$. Note that our definition of (ϵ, δ)-learning in this paper is in the agnostic model.

Assumption 1 *There is no polynomial time solution to the $\tilde{O}(n^{1.5})$-unique-Shortest-Vector-Problem.*

In a nutshell, given a basis $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathbb{R}^n$, the $\tilde{O}(n^{1.5})$-unique-Shortest-Vector-Problem consists of finding the shortest nonzero vector in $\{a_1\mathbf{v}_1 + \ldots + a_n\mathbf{v}_n : a_1, \ldots, a_n \in \mathcal{Z}\}$, even given the information that it is shorter by a factor of at least $\tilde{O}(n^{1.5})$ than any other non-parallel vector. This problem is believed to be hard - there are no known sub-exponential algorithms, and it is known to be NP-hard if $\tilde{O}(n^{1.5})$ is replaced by a small constant (see [16] for more details).

With this assumption, Klivans and Sherstov proved the following:

Theorem 4 (Theorem 1.2 in Klivans and Sherstov [16]) *Let $\mathcal{X} = \{\pm 1\}^n$, let*

$$H = \{\mathbf{x} \mapsto \phi_{0,1}(\langle \mathbf{w}, \mathbf{x} \rangle - \theta - 1/2) : \theta \in \mathbb{N}, \mathbf{w} \in \mathbb{N}^n, |\theta| + \|\mathbf{w}\|_1 \leq poly(n)\} \ ,$$

and let $H_k = \{\mathbf{x} \mapsto (h_1(\mathbf{x}) \wedge \ldots \wedge h_k(\mathbf{x})) : \forall i, h_i \in H\}$. Then, based on Assumption 1, H_k is not efficiently learnable in the standard PAC model for any $k = n^\rho$ where $\rho > 0$ is a constant.

The above theorem implies the following.

Lemma 3 *Based on Assumption 1, there is no algorithm that runs in time* $\mathrm{poly}(n, 1/\epsilon, 1/\delta)$ *and (ϵ, δ)-learns the class H defined in Theorem 4.*

Proof To prove the lemma we show that if there is a polynomial time algorithm that learns H in the *agnostic* model, then there exists a weak learning algorithm (with a polynomial edge) that learns H_k in the standard (non-agnostic) PAC model. In the standard PAC model, weak learning implies strong learning [18], hence the existence of a weak learning algorithm that learns H_k will contradict Theorem 4.

Indeed, let $\mathcal{D}$ be any distribution such that there exists $h^\star \in H_k$ with $\mathrm{err}_{\mathcal{D}}(h^\star) = 0$. Let us rewrite $h^\star = h_1^\star \wedge \ldots \wedge h_k^\star$ where for all i, $h_i^\star \in H$. To show that there exists a weak learner, we first show that there exists some $h \in H$ with $\mathrm{err}_{\mathcal{D}}(h) \leq 1/2 - 1/2k^2$.

Since for each $\mathbf{x}$ if $h^\star(\mathbf{x}) = 0$ then there exists j s.t. $h_j^\star(\mathbf{x}) = 0$, we can use the union bound to get that

$$1 = \mathbb{P}[\exists j : h_j^\star(\mathbf{x}) = 0 | h^\star(\mathbf{x}) = 0] \leq \sum_j \mathbb{P}[h_j^\star(\mathbf{x}) = 0 | h^\star(\mathbf{x}) = 0] \leq k \max_j \mathbb{P}[h_j^\star(\mathbf{x}) = 0 | h^\star(\mathbf{x}) = 0] \ .$$

So, for j that maximizes $\mathbb{P}[h_j^\star(\mathbf{x}) = 0 | h^\star(\mathbf{x}) = 0]$ we get that $\mathbb{P}[h_j^\star(\mathbf{x}) = 0 | h^\star(\mathbf{x}) = 0] \geq 1/k$. Therefore,

$$\begin{aligned} \mathrm{err}_{\mathcal{D}}(h_j^\star) &= \mathbb{P}[h_j^\star(\mathbf{x}) = 1 \wedge h^\star(\mathbf{x}) = 0] = \mathbb{P}[h^\star(\mathbf{x}) = 0] \ \mathbb{P}[h_j^\star(\mathbf{x}) = 1 | h^\star(\mathbf{x}) = 0] \\ &= \mathbb{P}[h^\star(\mathbf{x}) = 0] \left(1 - \mathbb{P}[h_j^\star(\mathbf{x}) = 0 | h^\star(\mathbf{x}) = 0]\right) \leq \mathbb{P}[h^\star(\mathbf{x}) = 0] \left(1 - 1/k\right) \ . \end{aligned}$$

Now, if $\mathbb{P}[h^\star(\mathbf{x}) = 0] \leq 1/2 + 1/k^2$ then the above gives

$$\mathrm{err}_{\mathcal{D}}(h_j^\star) \leq (1/2 + 1/k^2)(1 - 1/k) \leq 1/2 - 1/2k^2 \ ,$$

where the inequality holds for any positive integer k. Otherwise, if $\mathbb{P}[h^\star(\mathbf{x}) = 0] > 1/2 + 1/k^2$, then the constant predictor $h(\mathbf{x}) = 0$ has $\mathrm{err}_{\mathcal{D}}(h) < 1/2 - 1/k^2$. In both cases we have shown that there exists a predictor in H with error of at most $1/2 - 1/2k^2$.

Finally, if we can agnostically learn H in time $\mathrm{poly}(n, 1/\epsilon, 1/\delta)$, then we can find h' with $\mathrm{err}_{\mathcal{D}}(h') \leq \min_{h \in H} \mathrm{err}_{\mathcal{D}}(h) + \epsilon \leq 1/2 - 1/2k^2 + \epsilon$ in time $\mathrm{poly}(n, 1/\epsilon, 1/\delta)$ (recall that $k = n^\rho$ for some $\rho > 0$). This means that we can have a weak learner that runs in polynomial time, and this concludes our proof. ■

Let h be a hypothesis in the class H defined in Theorem 4 and take any $\mathbf{x} \in \{\pm 1\}^n$. Then, there exist an integer θ and a vector of integers $\mathbf{w}$ such that $h(\mathbf{x}) = \phi_{0,1}(\langle \mathbf{w}, \mathbf{x} \rangle - \theta - 1/2)$. But since $\langle \mathbf{w}, \mathbf{x} \rangle - \theta$ is also an integer, if we let $L = 1$ this means that $h(\mathbf{x}) = \phi_{\mathrm{pw}}(\langle \mathbf{w}, \mathbf{x} \rangle - \theta - 1/2)$ as well. Furthermore, letting $\mathbf{x}' \in \mathbb{R}^{n+1}$ denote the concatenation of $\mathbf{x}$ with the constant 1 and letting $\mathbf{w}' \in \mathbb{R}^{n+1}$ denote the concatenation of $\mathbf{w}$ with the scalar $(-\theta - 1/2)$ we obtain that $h(\mathbf{x}) = \phi_{\mathrm{pw}}(\langle \mathbf{w}', \mathbf{x}' \rangle)$. Last, let us normalize $\tilde{\mathbf{w}} = \mathbf{w}'/\|\mathbf{w}'\|$, $\tilde{\mathbf{x}} = \mathbf{x}/\|\mathbf{x}'\|$, and redefine L to be $\|\mathbf{w}'\| \, \|\mathbf{x}'\|$, we get that $h(\mathbf{x}) = \phi_{\mathrm{pw}}(\langle \tilde{\mathbf{w}}, \tilde{\mathbf{x}} \rangle)$. That is, we have shown that H is contained in a class of the form H_{pw} with a Lipschitz constant bounded by $\mathrm{poly}(n)$. Combining the above with Lemma 3 we obtain the following:

Corollary 2 *Let L be a Lipschitz constant and let H_{pw} be the class defined by the L-Lipschitz transfer function ϕ_{pw}. Then, based on Assumption 1, there is no algorithm that runs in time* $\text{poly}(L, 1/\epsilon, 1/\delta)$ *and (ϵ, δ)-learns the class H_{pw}.*

A similar argument leads to the hardness of learning H_{sig}.

Theorem 5 *Let L be a Lipschitz constant and let H_{sig} be the class defined by the L-Lipschitz transfer function ϕ_{sig}. Then, based on Assumption 1, there is no algorithm that runs in time* $\text{poly}(L, 1/\epsilon, 1/\delta)$ *and (ϵ, δ)-learns the class H_{sig}.*

Proof Let h be a hypothesis in the class H defined in Theorem 4 and take any $\mathbf{x} \in \{\pm 1\}^n$. Then, there exist an integer θ and a vector of integers $\mathbf{w}$ such that $h(\mathbf{x}) = \phi_{0,1}(\langle \mathbf{w}, \mathbf{x} \rangle - \theta - 1/2)$. However, since $\langle \mathbf{w}, \mathbf{x} \rangle - \theta$ is also an integer, we see that

$$|\phi_{0,1}(\langle \mathbf{w}, \mathbf{x} \rangle - \theta - 1/2) - \phi_{\text{sig}}(\langle \mathbf{w}, \mathbf{x} \rangle - \theta - 1/2)| \leq \frac{1}{1 + \exp(2L)} \ .$$

This means that for any $\epsilon > 0$, if we pick $L = \frac{\log(2/\epsilon - 1)}{2}$ and define $h_{\text{sig}}(\mathbf{x}) = \phi_{\text{sig}}(\langle \mathbf{w}, \mathbf{x} \rangle - \theta - 1/2)$, then $|h(\mathbf{x}) - h_{\text{sig}}(\mathbf{x})| \leq \epsilon/2$. Furthermore, letting $\mathbf{x}' \in \mathbb{R}^{n+1}$ denote the concatenation of $\mathbf{x}$ with the constant 1 and letting $\mathbf{w}' \in \mathbb{R}^{n+1}$ denote the concatenation of $\mathbf{w}$ with the scalar $(-\theta - 1/2)$ we obtain that $h_{\text{sig}}(\mathbf{x}) = \phi_{\text{sig}}(\langle \mathbf{w}', \mathbf{x}' \rangle)$. Last, let us normalize $\tilde{\mathbf{w}} = \mathbf{w}'/\|\mathbf{w}'\|$, $\tilde{\mathbf{x}} = \mathbf{x}/\|\mathbf{x}'\|$, and redefine L to be

$$L = \frac{\|\mathbf{w}'\| \|\mathbf{x}'\| \log(2/\epsilon - 1)}{2} \tag{10}$$

so that $h_{\text{sig}}(\mathbf{x}) = \phi_{\text{sig}}(\langle \tilde{\mathbf{w}}, \tilde{\mathbf{x}} \rangle)$. Thus we see that if there exists an algorithm that runs in time $\text{poly}(L, 1/\epsilon, 1/\delta)$ and $(\epsilon/2, \delta)$-learns the class H_{sig}, then since for all $h \in H$ exists $h_{\text{sig}} \in H_{\text{sig}}$ such that $|h_{\text{sig}}(\mathbf{x}) - h(\mathbf{x})| \leq \epsilon/2$, there also exists an algorithm that (ϵ, δ)-learns the concept class H defined in Theorem 4 in time polynomial in $(L, 1/\epsilon, 1/\delta)$ (for L defined in Equation 10). But by definition of L in Equation 10 and the fact that $\|\mathbf{w}'\|$ and $\|\mathbf{x}'\|$ are of size $\text{poly}(n)$, this means that there is an algorithm that runs in time polynomial in $(n, 1/\epsilon, 1/\delta)$ and (ϵ, δ)-learns the class H, which contradicts Lemma 3. ■

4 Related work

The problem of learning kernel-based halfspaces has been extensively studied before, mainly in the framework of SVM [25, 10, 19]. When the data is separable with a margin μ, it is possible to learn a halfspaces in polynomial time. The learning problem becomes much more difficult when the data is not separable with margin.

In terms of hardness results, [6] derive hardness results for proper learning with sufficiently small margins. There are also strong hardness of approximation results for *proper* learning *without* margin (see for example [12] and the references therein). We emphasize that we allow improper learning, which is just as useful for the purpose of learning good classifiers, and thus these hardness results do not apply. Instead, the hardness result we derived in Section 3 hold for improper learning as well. As mentioned before, the main tool we rely on for deriving the hardness result is the representation independent hardness result for learning intersections of halfspaces given in [16].

Practical algorithms such as SVM often replace the 0-1 error function with a convex surrogate, and then apply convex optimization tools. However, there are no guarantees on how well the surrogate function approximates the 0-1 error function. Recently, [27, 4] studied the *asymptotic* relationship between surrogate convex loss functions and the 0-1 error function. In contrast, in this paper we show that even with a finite sample, surrogate convex loss functions can be competitive with the 0-1 error function as long as we replace inner-products with the kernel $K(\mathbf{x}, \mathbf{x}') = 1/(1 - 0.5\langle \mathbf{x}, \mathbf{x}' \rangle)$.

4.1 Margin analysis

Recall that we circumvented the dependence of the VC dimension of $H_{\phi_{0-1}}$ on the dimensionality of $\mathcal{X}$ by replacing ϕ_{0-1} with a Lipschitz transfer function. Another common approach is to require that the learned classifier will be competitive with the *margin* error rate of the optimal halfspace. Formally, the μ-margin error rate of a halfspace of the form $h_{\mathbf{w}}(\mathbf{x}) = \mathbf{1}(\langle \mathbf{w}, \mathbf{x} \rangle > 0)$ is defined as:

$$\text{err}_{\mathcal{D},\mu}(\mathbf{w}) \ = \ \Pr[h_{\mathbf{w}}(\mathbf{x}) \neq y \vee |\langle \mathbf{w}, \mathbf{x} \rangle| \leq \mu] \ . \tag{11}$$

Intuitively, $\mathrm{err}_{\mathcal{D},\mu}(\mathbf{w})$ is the error rate of $h_{\mathbf{w}}$ had we μ-shifted each point in the worst possible way. Margin based analysis restates the goal of the learner (as given in Equation (2)) and requires that the learner will find a classifier h that satisfies:

$$\mathrm{err}_{\mathcal{D}}(h) \;\leq\; \min_{\mathbf{w}:\|\mathbf{w}\|=1} \mathrm{err}_{\mathcal{D},\mu}(\mathbf{w}) + \epsilon \;. \tag{12}$$

Bounds of the above form are called margin-based bounds and are widely used in the statistical analysis of Support Vector Machines and AdaBoost. It was shown [3, 17] that $m = \Theta(\log(1/\delta)/(\mu\,\epsilon)^2)$ examples are sufficient (and necessary) to learn a classifier for which Equation (12) holds with probability of at least $1-\delta$. Note that as in the sample complexity bound we gave in Theorem 1, the margin based sample complexity bound also does not depend on the dimension.

In fact, the Lipschitz approach used in this paper and the margin-based approach are closely related. First, it is easy to verify that if we set $L = 1/(2\mu)$, then for any $\mathbf{w}$ the hypothesis $h(\mathbf{x}) = \phi_{\mathrm{pw}}(\langle \mathbf{w}, \mathbf{x}\rangle)$ satisfies $\mathrm{err}_{\mathcal{D}}(h) \leq \mathrm{err}_{\mathcal{D},\mu}(\mathbf{w})$. Therefore, an algorithm that (ϵ,δ)-learns H_{pw} also guarantees that Equation (12) holds. Second, it is also easy to verify that if we set $L = \frac{1}{4\mu}\log\left(\frac{2-\epsilon}{\epsilon}\right)$ then for any $\mathbf{w}$ the hypothesis $h(\mathbf{x}) = \phi_{\mathrm{sig}}(\langle \mathbf{w}, \mathbf{x}\rangle)$ satisfies $\mathrm{err}_{\mathcal{D}}(h) \leq \mathrm{err}_{\mathcal{D},\mu}(\mathbf{w}) + \epsilon/2$. Therefore, an algorithm that $(\epsilon/2,\delta)$-learns H_{sig} also guarantees that Equation (12) holds.

As a direct corollary of the above discussion we obtain that it is possible to learn a vector $\mathbf{w}$ that guarantees Equation (12) in time $\mathrm{poly}(\exp(\tilde{O}(1/\mu)))$.

A computational complexity analysis under margin assumptions was first carried out in [6] (see also the hierarchical worst-case analysis recently proposed in [5]). The technique used in [6] is based on the observation that in the noise-free case, an optimal halfspace can be expressed as a linear sum of at most $1/\mu^2$ examples. Therefore, one can perform an exhaustive search over all sub-sequences of $1/\mu^2$ examples, and choose the optimal halfspace. Note that this algorithm will always run in time m^{1/μ^2}. Since the sample complexity bound requires that m will be order of $1/(\mu\epsilon)^2$, the runtime of the method described by [6] becomes $\mathrm{poly}(\exp(\tilde{O}(1/\mu^2)))$. In comparison, our algorithm achieves a better runtime of $\mathrm{poly}(\exp(\tilde{O}(1/\mu)))$. Moreover, while the algorithm of [6] performs an exhaustive search, our algorithm's runtime depends on the parameter B, which is $\mathrm{poly}(\exp(\tilde{O}(1/\mu)))$ only under a worst-case assumption. Since in practice we will cross-validate for B, it is plausible that in many real-world scenarios the runtime of our algorithm will be much smaller.

4.2 Distributional Assumptions

The idea of approximating the zero-one transfer function with a polynomial was first proposed by [14] who studied the problem of agnostically learning halfspaces without kernels in $\mathbb{R}^n$ under distributional assumption. In particular, they showed that if the distribution over $\mathcal{X}$ is uniform over the unit ball, then it is possible to agnostically learn $H_{\phi_{0-1}}$ in time $\mathrm{poly}(n^{1/\epsilon^4})$. This was further generalized by [7], who showed that similar bounds hold for product distributions.

Beside distributional assumptions, these works are characterized by explicit dependence on the dimension of $\mathcal{X}$, and therefore are not adequate for the kernel-based setting we consider in this paper, in which the dimensionality of $\mathcal{X}$ can even be infinite. More precisely, while [14] try to approximate the zero-one transfer function with a low-degree polynomial, we require instead that the coefficients of the polynomials are bounded. The principle that when learning in high dimensions "the size of the parameters is more important than their number" was one of the main advantages in the analysis of the statistical properties of several learning algorithms (e.g. [2]).

Interestingly, in [21] we show that the very same algorithm we use in this paper recover the same complexity bound of [14].

5 Discussion

In this paper we described and analyzed a new technique for agnostically learning kernel-based halfspaces with the zero-one loss function. The bound we derive has an exponential dependence on L, the Lipschitz coefficient of the transfer function. While we prove that (under a certain cryptographic assumption) no algorithm can have a polynomial dependence on L, the immediate open question is whether the dependence on L can be further improved.

A perhaps surprising property of our analysis is that we propose a single algorithm, returning a single classifier, which is simultaneously competitive against *all* transfer functions $p \in P_B$. In particular, it learns with respect to the "optimal" transfer function, where by optimal we mean the one which attains the smallest error rate, $\mathbb{E}[|p(\langle \mathbf{w}, \mathbf{x}\rangle) - y|]$, over the distribution $\mathcal{D}$.

Our algorithm boils down to linear regression with the absolute loss function and while composing a particular kernel function over our original RKHS. It is possible to show that solving the vanilla

SVM, with the hinge-loss, and composing again our particular kernel over the desired kernel, can also give similar guarantees. It is therefore interesting to study if there is something special about the kernel we propose or maybe other kernel functions (e.g. the Gaussian kernel) can give similar guarantees.

Another possible direction is to consider other types of margin-based analysis or transfer functions. For example, in the statistical learning literature, there are several definitions of "noise" conditions, some of them are related to margin, which lead to faster decrease of the error rate as a function of the number of examples (see for example [9, 24, 23]). Studying the computational complexity of learning under these conditions is left to future work.

Acknowledgments

We would like to thank Adam Klivans for helping with the Hardness results. This work was partially supported by a Google Faculty Research Grant.

References

[1] C. Thomas-Agnan A. Berlinet. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Springer, 2003.

[2] P. L. Bartlett. For valid generalization, the size of the weights is more important than the size of the network. In *Advances in Neural Information Processing Systems 9*, 1997.

[3] P. L. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

[4] P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101:138–156, 2006.

[5] S. Ben-David. Alternative measures of computational complexity. In *TAMC*, 2006.

[6] S. Ben-David and H. Simon. Efficient learning of linear perceptrons. In *NIPS*, 2000.

[7] E. Blais, R. O'Donnell, and K Wimmer. Polynomial regression under arbitrary product distributions. In *COLT*, 2008.

[8] L. Bottou and O. Bousquet. The tradeoffs of large scale learning. In *NIPS*, pages 161–168, 2008.

[9] O. Bousquet. *Concentration Inequalities and Empirical Processes Theory Applied to the Analysis of Learning Algorithms*. PhD thesis, Ecole Polytechnique, 2002.

[10] N. Cristianini and J. Shawe-Taylor. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[11] V. Feldman, P. Gopalan, S. Khot, and A.K. Ponnuswami. New results for learning noisy parities and halfspaces. In *In Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, 2006.

[12] V. Guruswami and P. Raghavendra. Hardness of learning halfspaces with noise. In *Proceedings of the 47th Foundations of Computer Science (FOCS)*, 2006.

[13] S.M. Kakade, K. Sridharan, and A. Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In *NIPS*, 2008.

[14] A. Kalai, A.R. Klivans, Y. Mansour, and R. Servedio. Agnostically learning halfspaces. In *Proceedings of the 46th Foundations of Computer Science (FOCS)*, 2005.

[15] M. J. Kearns, R. E. Schapire, and L. M. Sellie. Toward efficient agnostic learning. In *COLT*, pages 341–352, July 1992. To appear, *Machine Learning*.

[16] Adam R. Klivans and Alexander A. Sherstov. Cryptographic hardness for learning intersections of halfspaces. In *FOCS*, 2006.

[17] D. A. McAllester. Simplified PAC-Bayesian margin bounds. In *COLT*, pages 203–215, 2003.

[18] R.E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

[19] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, 2002.

[20] S. Shalev-Shwartz and N. Srebro. SVM optimization: Inverse dependence on training set size. In *International Conference on Machine Learning*, pages 928–935, 2008.

[21] S. Shalev-Shwartz, O. Shamir, and K. Sridharan. Agnostically learning halfspaces with margin errors. Technical report, Toyota Technological Institute, 2009.

[22] S. Shalev-Shwartz, O. Shamir, and K. Sridharan. Learning kernel-based halfspaces with the zero-one loss, 2010. Technical Report, available at arXiv:1005.3681.

[23] I. Steinwart and C. Scovel. Fast rates for support vector machines using gaussian kernels. *Annals of Statistics*, 35:575, 2007.

[24] A. Tsybakov. Optimal aggregation of classifiers in statistical learning. *Annals of Statistics*, 32: 135–166, 2004.

[25] V. N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[26] G. Wahba. *Spline Models for Observational Data*. SIAM, 1990.

[27] T. Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *The Annals of Statistics*, 32:56–85, 2004.