SIAM J. COMPUT.
Vol. 40, No. 6, pp. 1623–1646

LEARNING KERNEL-BASED HALFSPACES WITH THE 0-1 LOSS*

SHAI SHALEV-SHWARTZ†, OHAD SHAMIR‡, AND KARTHIK SRIDHARAN§

Abstract. We describe and analyze a new algorithm for agnostically learning kernel-based halfspaces with respect to the 0-1 loss function. Unlike most of the previous formulations, which rely on surrogate convex loss functions (e.g., hinge-loss in support vector machines (SVMs) and log-loss in logistic regression), we provide finite time/sample guarantees with respect to the more natural 0-1 loss function. The proposed algorithm can learn kernel-based halfspaces in worst-case time 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.

Key words. learning halfspaces, kernel methods, learning theory

AMS subject classifications. 68Q32, 68T05, 68Q17

DOI. 10.1137/100806126

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) [27]. 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 [29, 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 nonconvex 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,

$$\mathrm{err}_{\mathcal{D}}(h) \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}}[|h(\mathbf{x}) - y|] \ . \tag{1.1}$$

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 independent and identically

*Received by the editors August 20, 2010; accepted for publication (in revised form) September 28, 2011; published electronically December 13, 2011. This work was supported by Israeli Science Foundation grant 590-10. A preliminary version of this paper appeared in *Proceedings of the 23rd Annual Conference on Learning Theory* (COLT), A. T. Kalai and M. Mohri, eds., Omnipress, Madison, WI, 2010, pp. 441–450.

http://www.siam.org/journals/sicomp/40-6/80612.html

†School of Computer Science and Engineering, The Hebrew University, Jerusalem 91904, Israel (shais@cs.huji.ac.il).

‡School of Computer Science and Engineering, The Hebrew University, Jerusalem 91904, Israel. Current address: Microsoft Research New England, One Memorial Drive, Cambridge, MA 02142 (ohadsh@microsoft.com).

§Toyota Technological Institute, 6045 S. Kenwood Ave., Chicago, IL 60637 (karthik@ttic.edu).

distributed (i.i.d.) from $\mathcal{D}$, and it returns a classifier. Following the agnostic probably approximately correct (PAC) learning framework [17], 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

$$\mathrm{err}_{\mathcal{D}}(\hat{h}) \ \leq \ \inf_{h \in H} \mathrm{err}_{\mathcal{D}}(h) + \epsilon \ . \tag{1.2}$$

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,

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

The class of (origin centered) halfspaces is defined as follows. Let $\mathcal{X}$ be a compact subset of an 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

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

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 need only 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 Vapnik–Chervonenkis (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 (1.2) for the class of halfspaces scales at least linearly with the dimension n [27]. 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 (1.2).

One way to define a slightly different concept class is to approximate the noncontinuous 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

$$\phi_{\mathrm{sig}}(a) \ \stackrel{\mathrm{def}}{=} \ \frac{1}{1 + \exp(-4L\,a)} \ , \tag{1.3}$$

which is an L-Lipschitz function. Other L-Lipschitz transfer functions are the erf function and the piecewise linear function:

$$\phi_{\mathrm{erf}}(a) \ \stackrel{\mathrm{def}}{=} \ \frac{1}{2}\left(1 + \mathrm{erf}\left(\sqrt{\pi}\,L\,a\right)\right), \quad \phi_{\mathrm{pw}}(a) \ \stackrel{\mathrm{def}}{=} \ \max\left\{\min\left\{\frac{1}{2} + L\,a, \, 1\right\}, \, 0\right\}. \tag{1.4}$$

An illustration of these transfer functions is given in Figure 1.1.

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

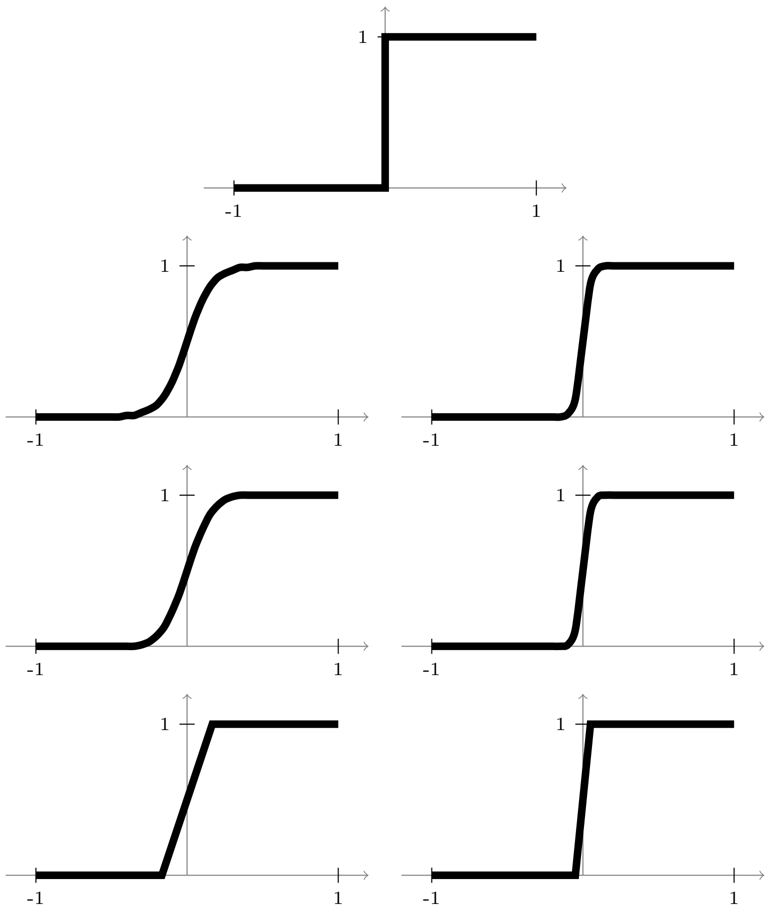

FIG. 1.1. *Transfer functions used throughout the paper. From top to bottom and left to right: The* 0-1 *transfer function; the sigmoid transfer function* ($L = 3$ *and* $L = 10$)*; the* erf *transfer function* ($L = 3$ *and* $L = 10$)*; the piecewise linear transfer function* ($L = 3$ *and* $L = 10$)*.*

rather the entire interval $[0, 1]$, we interpret $\phi(\langle \mathbf{w}, \mathbf{x}\rangle)$ as the probability of outputing the label 1. The definition of $\mathrm{err}_{\mathcal{D}}(h)$ remains[1] as in (1.1).

The advantage of using a Lipschitz transfer function can be seen via Rademacher generalization bounds [5]. In fact, a simple corollary of the so-called contraction lemma implies the following.

THEOREM 1.1. *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.*

The above theorem tells us that the sample complexity of learning H_ϕ is $\tilde{\Omega}(L^2/\epsilon^2)$.

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

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 that $\mathbf{w}$ makes on the training set; see subsection 4.1 for a comparison.

From the computational complexity point of view, the result given in Theorem 1.1 is problematic, since the ERM algorithm should solve the nonconvex optimization problem

$$\underset{\mathbf{w}:\|\mathbf{w}\|\leq 1}{\operatorname{argmin}} \ \frac{1}{m}\sum_{i=1}^{m} |\phi(\langle \mathbf{w}, \mathbf{x}_i\rangle) - y_i| \ . \tag{1.5}$$

Solving this problem in polynomial time is hard under reasonable assumptions, as we formally show in section 3. Adapting a technique due to [7] we show in Appendix A that it is possible to find an ϵ-accurate solution to (1.5) (where the transfer function is ϕ_{pw}) in time $\mathrm{poly}(\exp(\frac{L^2}{\epsilon^2}\log(\frac{L}{\epsilon})))$. 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 $\mathrm{poly}\left(\exp\left(L\log(\frac{L}{\epsilon})\right)\right)$. That is, the runtime of our algorithm is exponentially smaller than the runtime required to solve the ERM problem using the technique described in [7]. Moreover, the algorithm of [7] performs an exhaustive search over all $(L/\epsilon)^2$ subsets of the m examples in the training set, and therefore its runtime is always on the 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.

Interestingly, the very same algorithm we use in this paper also recovers the complexity bound of [16] for agnostically learning halfspaces with the 0-1 transfer function, without kernels and under a distributional assumption.

The rest of the paper is organized as follows. In section 2 we describe our main positive 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 subsection 4.1, and the relation to approaches utilizing a distributional assumption is discussed in subsection 4.2. In the latter subsection, we also point out how our algorithm recovers the same complexity bound as [16]. We wrap up with a discussion in section 5.

2. 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 nonconvex. 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}, and 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

$$K(\mathbf{x}, \mathbf{x}') \ \stackrel{\mathrm{def}}{=} \ \frac{1}{1-\nu\langle \mathbf{x}, \mathbf{x}'\rangle} \ , \tag{2.1}$$

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 preimages of $\mathbf{x}$ and $\mathbf{x}'$ with respect to the feature mapping induced by K'. Therefore, the kernel in (2.1) 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, [23, 11]). 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

$$H_B \stackrel{\text{def}}{=} \{\mathbf{x} \mapsto \langle \mathbf{v}, \psi(\mathbf{x}) \rangle \, : \, \mathbf{v} \in \mathbb{V}, \; \|\mathbf{v}\|^2 \le B\} \; . \tag{2.2}$$

The main positive result we prove in this section is the following.

THEOREM 2.1. *Let $\epsilon, \delta \in (0,1)$. For any $L \ge 3$, let*

$$B = 6L^4 + \exp\left(9L \log\left(\frac{2L}{\epsilon}\right) + 5\right),$$

and let m be a sample size that satisfies $m \ge (8B/\epsilon^2)\,(2 + 9\sqrt{\ln(8/\delta)})^2$. 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

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

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 $L \ge 3$ is rather arbitrary and is meant to simplify the presentation of the bound.

2.1. Proof of Theorem 2.1. To prove this theorem, we start with analyzing the time and sample complexities of learning H_B. The sample complexity analysis follows directly from a Rademacher generalization bound [5]. In particular, the following theorem tells us that the sample complexity of learning H_B with the ERM rule is on the order of B/ϵ^2 examples.

THEOREM 2.2. *Let $\epsilon, \delta \in (0,1)$, let $B \ge 1$, and let m be a sample size that satisfies*

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

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

Proof. Since $K(\mathbf{x}, \mathbf{x}) \le 2$, the Rademacher complexity of H_B is bounded by $\sqrt{2B/m}$ (see also [15]). Additionally, using the Cauchy–Schwarz inequality, we have that the loss is bounded, $|\langle \mathbf{v}, \psi(\mathbf{x}) \rangle - y| \le \sqrt{2B} + 1$. The result now follows directly from [5, 15]. ∎

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

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

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

[28] 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

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

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

COROLLARY 2.3. *Let $\epsilon, \delta \in (0,1)$ and let $B \ge 1$. Then, for any distribution $\mathcal{D}$, it is possible to (ϵ,δ)-learn H_B in sample and time complexity of* $\mathrm{poly}\left(\frac{B}{\epsilon} \log(1/\delta)\right)$.

We now explain 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 2.4. *Let P_B be the following set of polynomials (possibly with infinite degree):*

$$P_B \stackrel{\text{def}}{=} \left\{ p(a) = \sum_{j=0}^\infty \beta_j \, a^j \ : \ \sum_{j=0}^\infty \beta_j^2 \, 2^j \le B \right\} \ . \tag{2.3}$$

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 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

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

This is exactly the kernel function defined in (2.1) (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 $k_1, \ldots, k_j \in \{1, \ldots, n\}^j, j =$

[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 2.2; see, for example, [9, 24].

$0,\ldots,\infty$). By definition of ψ and $\mathbf{v}_\mathbf{w}$, we have that

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

In addition,

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

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]). $\square$

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

LEMMA 2.5. *Let ϕ_{sig} be as defined in* (1.3), *where for simplicity we assume $L \ge 3$. For any $\epsilon > 0$, let*

$$B = 6L^4 + \exp\left(9L\log\left(\frac{2L}{\epsilon}\right) + 5\right).$$

Then there exists $p \in P_B$ such that

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

The proof of the lemma is based on a Chebyshev approximation technique and is given in Appendix B. Since the proof is rather involved, we also present a similar lemma, whose proof is simpler, for the ϕ_{erf} transfer function (see Appendix C). 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}.

Combining Theorem 2.2 and Lemma 2.4, we get that with probability of at least $1-\delta$,

$$\mathrm{err}_\mathcal{D}(\hat{h}) \le \min_{h\in H_B} \mathrm{err}_\mathcal{D}(h) + \epsilon/2 \le \min_{p\in P_B}\min_{h\in H_p} \mathrm{err}_\mathcal{D}(h) + \epsilon/2\ . \tag{2.4}$$

From Lemma 2.5 we obtain that for any $\mathbf{w} \in \mathcal{X}$, if $h(\mathbf{x}) = \phi_{\mathrm{sig}}(\langle \mathbf{w},\mathbf{x}\rangle)$, 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 $\mathrm{err}_\mathcal{D}(h') \le \mathrm{err}_\mathcal{D}(h)+$

$\epsilon/2$. Since it holds for all $\mathbf{w}$, we get that

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

Combining this with (2.4), Theorem 2.1 follows.

3. Hardness. In this section we derive a hardness result for agnostic learning of H_{sig} or H_{pw} with respect to the 0-1 loss. The hardness result relies on the hardness of standard (nonagnostic)[3] PAC learning of intersection of halfspaces given in Klivans and Sherstov [18] (see also similar arguments in [13]). 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.

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 $\{a_1\mathbf{v}_1 + \cdots + a_n\mathbf{v}_n : a_1, \ldots, a_n \in \mathcal{Z}\}$, even given the information that it is shorter by a factor of at least $\tilde{O}(n^{1.5})$ than any other nonparallel vector. This problem is believed to be hard—there are no known subexponential algorithms, and it is known to be NP-hard if $\tilde{O}(n^{1.5})$ is replaced by a small constant (see [18] for more details).

With this assumption, Klivans and Sherstov proved the following.

THEOREM 3.1 (Theorem 1.2 in Klivans and Sherstov [18]). *Let $\mathcal{X} = \{\pm 1\}^n$, let*

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

and let $H_k = \{\mathbf{x} \mapsto (h_1(\mathbf{x}) \wedge \cdots \wedge h_k(\mathbf{x})) : \text{for all } i, h_i \in H\}$. 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 3.2. *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* 3.1.

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 (nonagnostic) PAC model. In the standard PAC model, weak learning implies strong learning [22], and hence the existence of a weak learning algorithm that learns H_k will contradict Theorem 3.1.

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 $h^\star = h_1^\star \wedge \cdots \wedge h_k^\star$, 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 such that $h_j^\star(\mathbf{x}) = 0$, we can use the union bound to get that

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

[3]In the *standard* PAC model, we assume that some hypothesis in the class has $\mathrm{err}_{\mathcal{D}}(h) = 0$, while in the *agnostic* PAC model, which we study in this paper, $\mathrm{err}_{\mathcal{D}}(h)$ 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.

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,

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

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 3.1, 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 then 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 3.2 we obtain the following.

COROLLARY 3.3. *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* $\mathrm{poly}(L, 1/\epsilon, 1/\delta)$ *and (ϵ, δ)-learns the class H_{pw}.*

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

THEOREM 3.4. *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* $\mathrm{poly}(L, 1/\epsilon, 1/\delta)$ *and (ϵ, δ)-learns the class H_{sig}.*

Proof. Let h be a hypothesis in the class H defined in Theorem 3.1, 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

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

This means that for any $\epsilon > 0$, if we pick $L = \frac{\log(2/\epsilon - 1)}{2}$ and define $h_{\mathrm{sig}}(\mathbf{x}) = \phi_{\mathrm{sig}}(\langle \mathbf{w}, \mathbf{x} \rangle - \theta - 1/2)$, then $|h(\mathbf{x}) - h_{\mathrm{sig}}(\mathbf{x})| \leq \epsilon/2$. 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 $h_{\mathrm{sig}}(\mathbf{x}) = \phi_{\mathrm{sig}}(\langle \mathbf{w}', \mathbf{x}' \rangle)$. Last, let us normalize $\tilde{\mathbf{w}} = \mathbf{w}'/\|\mathbf{w}'\|$, $\tilde{\mathbf{x}} = \mathbf{x}/\|\mathbf{x}'\|$ and redefine L to be

$$L = \frac{\|\mathbf{w}'\| \|\mathbf{x}'\| \log(2/\epsilon - 1)}{2} \tag{3.1}$$

so that $h_{\mathrm{sig}}(\mathbf{x}) = \phi_{\mathrm{sig}}(\langle \tilde{\mathbf{w}}, \tilde{\mathbf{x}} \rangle)$. Thus we see that if there exists an algorithm that runs in time poly$(L, 1/\epsilon, 1/\delta)$ and $(\epsilon/2, \delta)$-learns the class H_{sig}, then, since for all $h \in H$ there exists $h_{\mathrm{sig}} \in H_{\mathrm{sig}}$ such that $|h_{\mathrm{sig}}(\mathbf{x}) - h(\mathbf{x})| \leq \epsilon/2$, there also exists an algorithm that (ϵ, δ)-learns the concept class H defined in Theorem 3.1 in time polynomial in $(L, 1/\epsilon, 1/\delta)$ (for L defined in (3.1)). But by definition of L in (3.1) and the fact that $\|\mathbf{w}'\|$ and $\|\mathbf{x}'\|$ are of size 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 3.2. ∎

4. Related work. The problem of learning kernel-based halfspaces has been extensively studied before, mainly in the framework of support vector machines (SVMs) [27, 11, 23]. When the data is separable with a margin μ, it is possible to learn halfspaces in polynomial time. The learning problem becomes much more difficult when the data is not separable with a margin.

In terms of hardness results, [7] derives 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, [14] 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 holds 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 [18].

Practical algorithms such as SVMs 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, [29, 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 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

$$\mathrm{err}_{\mathcal{D},\mu}(\mathbf{w}) \;=\; \Pr[h_{\mathbf{w}}(\mathbf{x}) \neq y \vee |\langle \mathbf{w}, \mathbf{x} \rangle| \leq \mu] \;. \tag{4.1}$$

Intuitively, $\mathrm{err}_{\mathcal{D},\mu}(\mathbf{w})$ would be 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 (1.2)) and requires that the learner find a classifier h that satisfies

$$\mathrm{err}_{\mathcal{D}}(h) \;\leq\; \min_{\mathbf{w}:\|\mathbf{w}\|=1} \mathrm{err}_{\mathcal{D},\mu}(\mathbf{w}) + \epsilon \;. \tag{4.2}$$

Bounds of the above form are called margin-based bounds and are widely used in the statistical analysis of SVMs and AdaBoost. It was shown [5, 21] that $m = \Theta(\log(1/\delta)/(\mu\,\epsilon)^2)$ examples are sufficient (and necessary) to learn a classifier for which (4.2) holds with probability of at least $1 - \delta$. Note that, as in the sample complexity bound we gave in Theorem 1.1, 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 (4.2) 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 (4.2) holds.

As a direct corollary of the above discussion we obtain that it is possible to learn a vector $\mathbf{w}$ that guarantees (4.2) in time $\mathrm{poly}(\exp(\tilde{O}(1/\mu)))$.

A computational complexity analysis under margin assumptions was first carried out in [7] (see also the hierarchical worst-case analysis recently proposed in [6]). The technique used in [7] 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 subsequences 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 on the order of $1/(\mu\epsilon)^2$, the runtime of the method described by [7] 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 [7] 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 and low-degree approaches. The idea of approximating the 0-1 transfer function with a polynomial was first proposed by Kalai et al. [16] who studied the problem of agnostically learning halfspaces without kernels in $\mathbb{R}^n$ under a distributional assumption. In particular, they showed that if the data distribution is uniform over $\mathcal{X}$, where $\mathcal{X}$ is the unit ball, then it is possible to agnostically learn $H_{\phi_{0-1}}$ in time $\mathrm{poly}(n^{1/\epsilon^4})$. Their approach is based on approximating the 0-1 transfer function with a low-degree polynomial, and then explicitly learning the $O(n^d)$ coefficients in the polynomial expansion, where d is the polynomial degree. This approach was further generalized by Blais, O'Donnell, and Wimmer [8], who showed that similar bounds hold for product distributions.

Besides distributional assumptions, these works are characterized by strong dependence on the dimensionality n 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. In contrast, our algorithm requires only the coefficients, not the degree, of the polynomials to be bounded, and no explicit handling of polynomial coefficients is required. 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 (see, e.g., [3]).

However, one can still ask how these approaches compare in the regime where n is considered a constant. Indeed, the proof of our main theorem, Theorem 2.1, is based on a certain approximating polynomial which in fact has finite degree. In principle, one could work explicitly with the polynomial expansion corresponding to this polynomial, but this does not seem to lead to improved sample complexity or time complexity guarantees. Moreover, this results in a rather inelegant algorithm, with guarantees which hold only with respect to that particular approximating polynomial. In contrast, our algorithm learns with respect to the much larger class H_B, which includes *all* polynomials with an appropriate coefficient bound (see Lemma 2.4), without the need to explicitly specify an approximating polynomial.

Finally, and quite interestingly, it turns out that the very same algorithm we use in this paper recovers the same complexity bound of [16]. To show this, note that although the ϕ_{0-1} transfer function cannot be expressed as a polynomial in P_B for any finite B, it can still be approximated by a polynomial in P_B. In particular, the following lemma shows that by imposing a uniform distribution assumption on the marginal distribution over $\mathcal{X}$, one can approximate the ϕ_{0-1} transfer function by a polynomial. In fact, to obtain the approximation, we use exactly the same Hermite polynomials construction as in [16]. However, while [16] shows that the ϕ_{0-1} transfer function can be approximated by a low-degree polynomial, we are concerned with polynomials having bounded coefficients. By showing that the approximating polynomial has bounded coefficients, we are able to rederive the results in [16] with a different algorithm.

LEMMA 4.1. *Let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \{0,1\}$, where $\mathcal{X}$ is the unit ball in $\mathbb{R}^n$ and the marginal distribution of $\mathcal{D}$ on $\mathcal{X}$ is uniform. For any $\epsilon \in (0,1)$, if $B = \mathrm{poly}(n^{1/\epsilon^4})$, then there exists $p \in P_B$ such that*

$$\mathbb{E}[|p(\langle \mathbf{w}, \mathbf{x} \rangle) - y|] \;\leq\; \mathbb{E}[|\phi_{0-1}(\langle \mathbf{w}, \mathbf{x} \rangle) - y|] + \epsilon\ .$$

The proof of the lemma is provided in Appendix D. As a direct corollary, using Theorem 2.2, we obtain the following.

COROLLARY 4.2. *Assume that the conditions of Lemma* 4.1 *hold. Let $\epsilon, \delta \in (0,1)$, and let $B = \mathrm{poly}(n^{1/\epsilon^4})$. Then the ERM predictor with respect to H_B (as described in section* 2*) (ϵ, δ)-learns $H_{\phi_{0-1}}$ in time and sample complexity* $\mathrm{poly}(B \log(1/\delta))$.

As mentioned earlier, this result matches the complexity bound of [16] up to second-order terms. We note that [16, 8] also obtained results under more general families of distributions, but our focus in this paper is different, and therefore we made no attempt to recover all of their results.

5. Discussion. In this paper we described and analyzed a new technique for agnostically learning kernel-based halfspaces with the 0-1 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 whether 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 leads to faster decrease of the error rate as a function of the number of examples (see, for example, [10, 26, 25]). Studying the computational complexity of learning under these conditions is left to future work.

Appendix A. Solving the ERM problem given in (1.5). In this section we show how to approximately solve (1.5) when the transfer function is ϕ_{pw}. The technique we use is similar to the covering technique described in [7].

For each i, let $b_i = 2(y_i - 1/2)$. It is easy to verify that the objective of (1.5) can be rewritten as

$$\frac{1}{m}\sum_{i=1}^{m} f(b_i\langle \mathbf{w}, \mathbf{x}_i\rangle), \quad \text{where} \;\; f(a) = \min\left\{1, \max\left\{0, \frac{1}{2} - L\,a\right\}\right\} \;. \tag{A.1}$$

Let $g(a) = \max\{0, 1/2 - L\,a\}$. Note that g is a convex function, $g(a) \geq f(a)$ for every a, and equality holds whenever $a \geq -1/2L$.

Let $\mathbf{w}^\star$ be a minimizer of (A.1) over the unit ball. We partition the set $[m]$ into

$$I_1 = \{i \in [m] : g(b_i\langle \mathbf{w}^\star, \mathbf{x}_i\rangle) = f(b_i\langle \mathbf{w}^\star, \mathbf{x}_i\rangle)\}, \quad I_2 = [m] \setminus I_1 \;.$$

Now, let $\hat{\mathbf{w}}$ be a vector that satisfies

$$\sum_{i \in I_1} g(b_i\langle \hat{\mathbf{w}}, \mathbf{x}_i\rangle) \;\leq\; \min_{\mathbf{w}:\|\mathbf{w}\|\leq 1} \sum_{i\in I_1} g(b_i\langle \mathbf{w}, \mathbf{x}_i\rangle) + \epsilon\, m \;. \tag{A.2}$$

Clearly, we have

$$\begin{aligned}
\sum_{i=1}^{m} f(b_i\langle \hat{\mathbf{w}}, \mathbf{x}_i\rangle) &\leq \sum_{i\in I_1} g(b_i\langle \hat{\mathbf{w}}, \mathbf{x}_i\rangle) + \sum_{i \in I_2} f(b_i\langle \hat{\mathbf{w}}, \mathbf{x}_i\rangle) \\
&\leq \sum_{i\in I_1} g(b_i\langle \hat{\mathbf{w}}, \mathbf{x}_i\rangle) + |I_2| \\
&\leq \sum_{i\in I_1} g(b_i\langle \mathbf{w}^\star, \mathbf{x}_i\rangle) + \epsilon\, m + |I_2| \\
&= \sum_{i=1}^{m} f(b_i\langle \mathbf{w}^\star, \mathbf{x}_i\rangle) + \epsilon\, m \;.
\end{aligned}$$

Dividing the two sides of the above by m we obtain that $\hat{\mathbf{w}}$ is an ϵ-accurate solution to (A.1). Therefore, it suffices to show a method that finds a vector $\hat{\mathbf{w}}$ that satisfies (A.2). To do so, we use a standard generalization bound (based on Rademacher complexity) as follows.

LEMMA A.1. *Let us sample $i_1, \ldots, i_k$ i.i.d. according to the uniform distribution over I_1. Let $\hat{\mathbf{w}}$ be a minimizer of $\sum_{j=1}^{k} g(b_{i_j}\langle \mathbf{w}, \mathbf{x}_{i_j}\rangle)$ over $\mathbf{w}$ in the unit ball. Then,*

$$\mathbb{E}\left[\frac{1}{|I_1|}\sum_{i\in I_1} g(b_i\langle \hat{\mathbf{w}}, \mathbf{x}_i\rangle) - \min_{\mathbf{w}:\|\mathbf{w}\|\leq 1} \frac{1}{|I_1|}\sum_{i \in I_1} g(b_i\langle \mathbf{w}, \mathbf{x}_i\rangle)\right] \;\leq\; \frac{2L}{\sqrt{k}} \;,$$

where expectation is over the choice of $i_1, \ldots, i_k$.

Proof. Simply note that g is L-Lipschitz, and then apply a Rademacher generalization bound with the contraction lemma (see [5]). ∎

The above lemma immediately implies that if $k \geq 4L^2/\epsilon^2$, then there exist $i_1, \ldots, i_k$ in I_1 such that if

$$\hat{\mathbf{w}} \in \operatorname*{argmin}_{\mathbf{w}:\|\mathbf{w}\|\leq 1} \sum_{j=1}^{k} g(b_{i_j}\langle \mathbf{w}, \mathbf{x}_{i_j}\rangle), \tag{A.3}$$

then $\hat{\mathbf{w}}$ satisfies (A.2) and therefore it is an ϵ-accurate solution of (A.1). The algorithm will simply perform an exhaustive search over all $i_1, \ldots, i_k$ in $[m]$; for each such sequence the procedure will find $\hat{\mathbf{w}}$ as in (A.3) in polynomial time. Finally, the procedure will output the $\hat{\mathbf{w}}$ that minimizes the objective of (A.1). The total runtime of the procedure is therefore $\mathrm{poly}(m^k)$. Plugging in the value of $k = \lceil 4L^2/\epsilon^2 \rceil$ and the value of m according to the sample complexity bound given in Theorem 1.1, we obtain the total runtime of

$$\mathrm{poly}\left(\left(\frac{L}{\epsilon}\right)^{L^2/\epsilon^2}\right) = \mathrm{poly}\left(\exp\left(\frac{L^2}{\epsilon^2}\log\left(\frac{L}{\epsilon}\right)\right)\right).$$

Appendix B. Proof of Lemma 2.5. In order to approximate ϕ_{sig} with a polynomial, we will use the technique of *Chebyshev approximation* (cf. [20]). One can write any continuous function on $[-1,+1]$ as a Chebyshev expansion $\sum_{n=0}^{\infty} \alpha_n T_n(\cdot)$, where each $T_n(\cdot)$ is a particular nth degree polynomial denoted as the nth Chebyshev polynomial (of the first kind). These polynomials are defined as $T_0(x) = 1, T_1(x) = x$ and then recursively via $T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$. For any n, $T_n(\cdot)$ is bounded in $[-1,+1]$. The coefficients in the Chebyshev expansion of ϕ_{sig} are equal to

$$\alpha_n = \frac{1 + \mathbf{1}(n > 0)}{\pi} \int_{x=-1}^{1} \frac{\phi_{\mathrm{sig}}(x) T_n(x)}{\sqrt{1-x^2}} dx. \tag{B.1}$$

Truncating the series after some threshold $n = N$ provides an Nth degree polynomial which approximates the original function.

Before we start, we note that there has been much work and strong theorems about the required polynomial *degree* as a function of the desired approximation (e.g., Jackson-type inequalities [2]). However, we do not know how to apply these theorems here, since we need a bound on the required *coefficient sizes* as a function of the desired approximation. This is the reason for the explicit and rather laborious calculation below.

In order to obtain a bound on B, we need to understand the behavior of the coefficients in the Chebyshev approximation. These are determined in turn by the behavior of α_n as well as the coefficients of each Chebyshev polynomial $T_n(\cdot)$. The following two lemmas provide the necessary bounds.

LEMMA B.1. *For any $n > 1$, $|\alpha_n|$ in the Chebyshev expansion of ϕ_{sig} on $[-1,+1]$ is upper bounded as follows:*

$$|\alpha_n| \leq \frac{1/L + 2/\pi}{(1+\pi/4L)^n}.$$

Also, we have $|\alpha_0| \leq 1$, $|\alpha_1| \leq 2$.

Proof. The coefficients α_n, $n = 1, \ldots,$ in the Chebyshev series are given explicitly by

$$\alpha_n = \frac{2}{\pi} \int_{x=-1}^{1} \frac{\phi_{\mathrm{sig}}(x) T_n(x)}{\sqrt{1-x^2}} dx. \tag{B.2}$$

For α_0, the same equality holds with $2/\pi$ replaced by $1/\pi$, so α_0 equals

$$\frac{1}{\pi}\int_{x=-1}^{1}\frac{\phi_{\rm sig}(x)}{\sqrt{1-x^2}}dx,$$

which by definition of $\phi_{\rm sig}(x)$ is at most $(1/\pi)\int_{x=-1}^{1}\left(\sqrt{1-x^2}\right)^{-1}dx = 1$. As for α_1, it equals

$$\frac{2}{\pi}\int_{x=-1}^{1}\frac{\phi_{\rm sig}(x)x}{\sqrt{1-x^2}}dx,$$

whose absolute value is at most $(2/\pi)\int_{x=-1}^{1}\left(\sqrt{1-x^2}\right)^{-1}dx = 2$.

To get a closed-form bound on the integral in (B.2) for general n and L, we will need to use some tools from complex analysis. The calculation closely follows [12].[4]

Let us consider $\phi_{\rm sig}(x)$ at some point x, and think of x as a complex number in the two-dimensional complex plain. A basic result in complex analysis is Cauchy's integral formula, which states that we can rewrite the value of a function at a given point by an integral over some closed path which "circles" that point in the complex plane. More precisely, we can rewrite $\phi_{\rm sig}(x)$ as

$$\phi_{\rm sig}(x) = \frac{1}{2\pi i}\oint_C \frac{\phi_{\rm sig}(z)}{z-x}\,dz, \tag{B.3}$$

where C is some closed path around x (with the integration performed counterclockwise). For this to be valid, we must assume that $\phi_{\rm sig}$ is *holomorphic* in the domain bounded by C, namely, that it is (complex) differentiable there. Substituting this into (B.2), we get that

$$\alpha_n = \frac{1}{\pi^2 i}\oint_C \phi_{\rm sig}(z)\left(\int_{x=-1}^{1}\frac{T_n(x)}{\sqrt{1-x^2}(z-x)}dx\right)dz. \tag{B.4}$$

Performing the variable change $x=\cos(\theta)$, and using the well-known fact that $T_n(\cos(\theta)) = \cos(n\theta)$, it is easily verified that

$$\int_{x=-1}^{1}\frac{T_n(x)}{\sqrt{1-x^2}(z-x)}dx = \frac{\pi}{\sqrt{z^2-1}(z\pm\sqrt{z^2-1})^n},$$

where the sign in $\pm$ is chosen so that $|z\pm\sqrt{z^2-1}|>1$. Substituting this back into (B.4), we get

$$\alpha_n = \frac{1}{\pi i}\oint_C \frac{\phi_{\rm sig}(z)dz}{\sqrt{z^2-1}(z\pm\sqrt{z^2-1})^n}dz, \tag{B.5}$$

where C is a closed path which contains the interval $[-1,+1]$.

This equation is valid whenever $\phi_{\rm sig}(z)$ is holomorphic in the domain bounded by C. In such domains, we can change the path C in whichever way we want, without changing the value of the integral. However, $\phi_{\rm sig}$ is not holomorphic everywhere.

[4]We note that such calculations also appear in standard textbooks on the subject, but they are usually carried out under asymptotic assumptions and disregarding coefficients which are important for our purposes.

Recalling that $\phi_{\mathrm{sig}}(z) = 1/(1+\exp(-4Lz))$ and using the closure properties of holomorphic functions, $\phi_{\mathrm{sig}}(z)$ is holomorphic at z if and only if $1+\exp(-4Lz) \neq 0$. Thus, the singular points are $z_k = i(\pi + 2\pi k)/4L$ for any $k = 0, \pm 1, \pm 2, \ldots$. Note that this forms a discrete set of isolated points. Functions of this type are called *meromorphic* functions. The fact that ϕ_{sig} is "well behaved" in this sense allows us to perform the analysis below.

If C contains any of these problematic points, then (B.5) is not valid. However, by the well-known residue theorem from complex analysis, this can be remedied by augmenting C with additional path integrals, which go in a small clockwise circle around these points, and then taking the radius of these circles to zero. Intuitively, these additional paths "cut off" the singular points from the domain bounded by C. This leads to additional terms in (B.5), one for any singular point z_k, which can be written as

$$\lim_{z \to z_k} -2(z - z_k) \frac{\phi_{\mathrm{sig}}(z)}{\sqrt{z_k^2 - 1}\left(z_k \pm \sqrt{z_k^2 - 1}\right)^n},$$

assuming the limit exists (this is known as the *residue* of the function we integrate in (B.5)). This limit for $z_0 = i\pi/4L$ equals

$$-\frac{2}{\sqrt{z_0^2 - 1}\left(z_0 \pm \sqrt{z_0^2 - 1}\right)^n} \lim_{z \to z_0} (z - z_0)\phi_{\mathrm{sig}}(z).$$

Plugging in the expression for ϕ_{sig}, and performing a variable change, the limit in the expression above equals

$$\lim_{z \to 0} \frac{z}{1 + e^{-i\pi - 4Lz}} = \lim_{z \to 0} \frac{z}{1 - e^{-4Lz}} = \lim_{z \to 0} \frac{1}{4Le^{-4Lz}} = 1/4L,$$

where we used l'Hôpital's rule to calculate the limit. Thus, we get that the residue term corresponding to z_0 is

$$-\frac{1/2L}{\sqrt{z_0^2 - 1}\left(z_0 \pm \sqrt{z_0^2 - 1}\right)^n}.$$

Performing a similar calculation for the other singular points, we get that the residue term for z_k is

$$-\frac{1/2L}{\sqrt{z_k^2 - 1}\left(z_k \pm \sqrt{z_k^2 - 1}\right)^n}.$$

Overall, we get that for well-behaved curves C, which do not cross any of the singular points,

$$\alpha_n = \frac{1}{\pi i} \oint_C \frac{\phi_{\mathrm{sig}}(z)dz}{\sqrt{z^2 - 1}(z \pm \sqrt{z^2 - 1})^n} dz - \sum_{k \in K_C} \frac{1/2L}{\sqrt{z_k^2 - 1}\left(z_k \pm \sqrt{z_k^2 - 1}\right)^n}, \tag{B.6}$$

where $k \in K_C$ if and only if the singular point z_k is inside the domain bounded by C.

It now remains to pick C appropriately. For some parameter $\rho > 1$, we pick C to be an ellipse such that any point z on it satisfies $|z \pm \sqrt{z^2 - 1}| = \rho$. We assume that ρ is such that the ellipse is uniformly bounded away from the singular points of our function. This is possible because the singular points constitute a discrete, well-spaced set of points along a line. We then let $\rho \to \infty$.

Since we picked ρ so that the ellipse is bounded away from the singular points, it follows that $|\phi_{\mathrm{sig}}(z)|$ is uniformly bounded along the ellipse. From that it is easy to verify that as $\rho \to \infty$, the integral

$$\frac{1}{\pi i}\oint_C \frac{\phi_{\mathrm{sig}}(z)dz}{\sqrt{z^2-1}(z \pm \sqrt{z^2-1})^n}dz = \frac{1}{\pi i}\oint_C \frac{\phi_{\mathrm{sig}}(z)dz}{\sqrt{z^2-1}\rho^n}dz$$

tends to zero. Also, as $\rho \to \infty$, all singular points eventually get inside the domain bounded by C, and it follows that (B.6) can be rewritten as

$$\alpha_n = -\sum_{k=-\infty}^{\infty} \frac{1/2L}{\sqrt{z_k^2-1}\left(z_k \pm \sqrt{z_k^2-1}\right)^n}.$$

Substituting the values of z_k and performing a routine simplification leads to the following:[5]

$$\alpha_n = \sum_{k=-\infty}^{\infty} \frac{-1/2L}{i^{n+1}\sqrt{((\pi+2\pi k)/4L)^2+1}\left((\pi+2\pi k)/4L \pm \sqrt{((\pi+2\pi k)/4L)^2+1}\right)^n}.$$

Recall that $\pm$ was chosen such that the absolute value of the relevant terms is as large as possible. Therefore,

$$\begin{aligned}|\alpha_n| &\le \sum_{k=-\infty}^{\infty} \frac{1/2L}{\sqrt{((\pi+2\pi k)/4L)^2+1}\left(|\pi+2\pi k|/4L + \sqrt{((\pi+2\pi k)/4L)^2+1}\right)^n}\\ &\le \sum_{k=-\infty}^{\infty} \frac{1/2L}{(|\pi+2\pi k|1/4L+1)^n} \le \frac{1/2L}{(1+\pi/4L)^n} + 2\sum_{k=1}^{\infty}\frac{1/2L}{(1+\pi(1+2k)/4L)^n}\\ &\le \frac{1/2L}{(1+\pi/4L)^n} + \int_{k=0}^{\infty}\frac{1/L}{(1+\pi(1+2k)/4L)^n}dk.\end{aligned}$$

Solving the integral and simplifying gives us

$$|\alpha_n| \le \frac{1}{(1+\pi/4L)^n}\left(1/4L + \frac{2+\pi/2L}{\pi(n-1)}\right).$$

Since $n \ge 2$, the result in the lemma follows. □

LEMMA B.2. *For any nonnegative integer n and $j = 0, 1, \ldots, n$, let $t_{n,j}$ be the coefficient of x^j in $T_n(x)$. Then $t_{n,j} = 0$ for any j with a different parity than n, and for any $j > 0$,*

$$|t_{n,j}| \le \frac{e^{n+j}}{\sqrt{2\pi}}.$$

[5] On first look, it might appear that α_n takes imaginary values for even n, due to the i^{n+1} factor, despite α_n being equal to a real-valued integral. However, it can be shown that $\alpha_n = 0$ for even n. This additional analysis can also be used to slightly tighten our final results in terms of constants in the exponent, but it was not included for simplicity.

Proof. We have the standard facts that $t_{n,j} = 0$ for j, n with different parities, and that $|t_{n,0}| \leq 1$. Using an explicit formula from the literature for the coefficients of Chebyshev polynomials (see [20, p. 24]), as well as Stirling approximation, we have that

$$\begin{aligned} |t_{n,j}| &= 2^{n-(n-j)-1} \frac{n}{n - \frac{n-j}{2}} \binom{n - \frac{n-j}{2}}{\frac{n-j}{2}} = \frac{2^j n}{n+j} \frac{\left(\frac{n+j}{2}\right)!}{\left(\frac{n-j}{2}\right)! j!} \\ &\leq \frac{2^j n}{j!(n+j)} \left(\frac{n+j}{2}\right)^j = \frac{n(n+j)^j}{(n+j)j!} \leq \frac{n(n+j)^j}{(n+j)\sqrt{2\pi j}(j/e)^j} \\ &= \frac{ne^j}{(n+j)\sqrt{2\pi j}} \left(1 + \frac{n}{j}\right)^j \leq \frac{ne^j}{(n+j)\sqrt{2\pi j}} e^n, \end{aligned}$$

from which the lemma follows. $\square$

We are now in a position to prove a bound on B. As discussed earlier, $\phi_{\mathrm{sig}}(x)$ in the domain $[-1, +1]$ equals the expansion $\sum_{n=0}^{\infty} \alpha_n T_x$. The error resulting from truncating the Chebyshev expanding at index N, for any $x \in [-1, +1]$, equals

$$\left| \phi_{\mathrm{sig}}(x) - \sum_{n=0}^{N} \alpha_n T_n(x) \right| = \left| \sum_{n=N+1}^{\infty} \alpha_n T_n(x) \right| \leq \sum_{n=N+1}^{\infty} |\alpha_n|,$$

where in the last transition we used the fact that $|T_n(x)| \leq 1$. Using Lemma B.1 and assuming $N > 0$, this is at most

$$\sum_{n=N+1}^{\infty} \frac{1/L + 2/\pi}{(1 + \pi/4L)^n} = \frac{4 + 8L/\pi}{\pi(1 + \pi/4L)^N}.$$

In order to achieve an accuracy of less than ϵ in the approximation, we need to equate this to ϵ and solve for N, i.e.,

$$N = \left\lceil \log_{1+\pi/4L} \left(\frac{4 + 8L/\pi}{\pi\epsilon} \right) \right\rceil. \tag{B.7}$$

The series left after truncation is $\sum_{n=0}^{N} \alpha_n T_n(x)$, which we can write as a polynomial $\sum_{j=0}^{N} \beta_j x^j$. Using Lemmas B.1 and B.2, the absolute value of the coefficient β_j for $j > 1$ can be upper bounded by

$$\begin{aligned} &\sum_{n=j\ldots N, n=j \mod 2} |a_n||t_{n,j}| \leq \sum_{n=j\ldots N, n=j \mod 2} \frac{1/L + 2/\pi}{(1 + \pi/4L)^n} \frac{e^{n+j}}{\sqrt{2\pi}} \\ &= \frac{(1/L + 2/\pi)e^j}{\sqrt{2\pi}} \sum_{n=j\ldots N, n=j \mod 2} \left(\frac{e}{1 + \pi/4L} \right)^n \\ &= \frac{(1/L + 2/\pi)e^j}{\sqrt{2\pi}} \left(\frac{e}{1 + \pi/4L} \right)^j \sum_{n=0}^{\lfloor \frac{N-j}{2} \rfloor} \left(\frac{e}{1 + \pi/4L} \right)^{2n} \\ &\leq \frac{(1/L + 2/\pi)e^j}{\sqrt{2\pi}} \left(\frac{e}{1 + \pi/4L} \right)^j \frac{(e/(1 + \pi/4L))^{N-j+2} - 1}{(e/(1 + \pi/4L))^2 - 1}. \end{aligned}$$

Since we assume $L \geq 3$, we have in particular $e/(1+\pi/4L) > 1$, so we can upper bound the expression above by dropping the 1 in the numerator, to get

$$\frac{1/L + 2/\pi}{\sqrt{2\pi}((e/(1+\pi/4L))^2 - 1)} \left(\frac{e}{1+\pi/4L}\right)^{N+2} e^j.$$

The cases β_0, β_1 need to be treated separately, due to the different forms of the bounds on α_0, α_1. Repeating a similar analysis (using the fact that $|t_{n,1}| = n$ for any odd n, and $|t_{n,0}| = 1$ for any even n), we get

$$\beta_0 \leq 1 + \frac{2}{\pi} + \frac{4L}{\pi^2},$$
$$\beta_1 \leq 2 + \frac{3(1+2L/\pi)(4L+\pi)}{\pi^2}.$$

Now that we have a bound on the β_j, we can plug it into the bound on B and get that B is upper bounded by

$$\begin{aligned}
\sum_{j=0}^{N} 2^j \beta_j^2 &\leq \beta_0^2 + 2\beta_1^2 + \sum_{j=2}^{N} \left(\frac{1/L + 2/\pi}{\sqrt{2\pi}((e/(1+\pi/4L))^2 - 1)}\right)^2 \left(\frac{e}{1+\pi/4L}\right)^{2N+4} (2e^2)^j \\
&\leq \beta_0^2 + 2\beta_1^2 + \left(\frac{1/L + 2/\pi}{\sqrt{2\pi}((e/(1+\pi/4L))^2 - 1)}\right)^2 \left(\frac{e}{1+\pi/4L}\right)^{2N+4} \frac{(2e^2)^{N+1}}{e^2 - 1} \\
&= \beta_0^2 + 2\beta_1^2 + \frac{(1/L + 2/\pi)^2 e^6}{(e^2-1)\pi((e/(1+\pi/4L))^2 - 1)^2 (1+\pi/4L)^4} \left(\frac{\sqrt{2}e^2}{1+\pi/4L}\right)^{2N}.
\end{aligned}$$

Using the assumption $L \geq 3$, a straightforward numerical calculation allows us to upper bound the above by

$$6L^4 + 0.56\left(\frac{\sqrt{2}e^2}{1+\pi/4L}\right)^{2N} \leq 6L^4 + 0.56(2e^4)^N.$$

Combining this with (B.7), we get that this is upper bounded by

$$6L^4 + 0.56(2e^4)^{\log_{1+\pi/4L}\left(\frac{4+8L/\pi}{\pi\epsilon}\right)+1},$$

which can be rewritten as

$$6L^4 + 1.12\exp\left(\frac{\log(2e^4)\log\left(\frac{2+4L/\pi}{\pi\epsilon}\right)}{\log(1+\pi/4L)} + 4\right). \tag{B.8}$$

Using the fact that $\log(1+x) \geq x(1-x)$ for $x \geq 0$, and the assumption that $L \geq 3$, we can bound the exponent by

$$\frac{\log(2e^4)\log\left(\frac{2+4L/\pi}{\pi\epsilon}\right)}{\frac{\pi}{4L}\left(1-\frac{\pi}{4L}\right)} + 4 \leq \frac{4L\log(2e^4)\log\left(\frac{2+4L/\pi}{\pi\epsilon}\right)}{\pi(1-\frac{\pi}{12})} + 4 \leq 9\log(2L/\epsilon)L + 4.$$

Substituting back into (B.8), and upper bounding 1.12 by e for readability, we get an overall bound on B of the form

$$6L^4 + \exp\left(9\log(2L/\epsilon)L + 5\right).$$

Appendix C. The $\phi_{\mathbf{erf}}(\cdot)$ function. In this section, we prove a result analogous to Lemma 2.5, using the $\phi_{\mathrm{erf}}(\cdot)$ transfer function. In a certain sense, it is stronger, because we can show that $\phi_{\mathrm{erf}}(\cdot)$ actually belongs to P_B for sufficiently large B. However, the resulting bound is worse than Lemma 2.5, as it depends on $\exp(L^2)$ rather than $\exp(L)$. However, the proof is much simpler, which helps to illustrate the technique.

The relevant lemma is the following.

LEMMA C.1. *Let $\phi_{\mathrm{erf}}(\cdot)$ be as defined in* (1.4)*, where for simplicity we assume $L \geq 3$. For any $\epsilon > 0$, let*

$$B \leq \frac{1}{4} + 2L^2\left(1 + 3\pi e L^2 e^{4\pi L^2}\right).$$

Then $\phi_{\mathrm{erf}}(\cdot) \in P_B$.

Proof. By a standard fact, $\phi_{\mathrm{erf}}(\cdot)$ is equal to its infinite Taylor series expansion at any point, and this series equals

$$\phi_{\mathrm{erf}}(a) = \frac{1}{2} + \frac{1}{\sqrt{\pi}}\sum_{n=0}^{\infty}\frac{(-1)^n(\sqrt{\pi}La)^{2n+1}}{n!(2n+1)}.$$

Luckily, this is an infinite degree polynomial, and it is only left to calculate for which values of B does it belong to P_B. Plugging in the coefficients in the bound on B, we get that

$$\begin{aligned}
B &\leq \frac{1}{4} + \frac{1}{\pi}\sum_{n=0}^{\infty}\frac{(2\pi L^2)^{2n+1}}{(n!)^2(2n+1)^2} \leq \frac{1}{4} + \frac{1}{\pi}\sum_{n=0}^{\infty}\frac{(2\pi L^2)^{2n+1}}{(n!)^2} \\
&= \frac{1}{4} + 2L^2\left(1 + \sum_{n=1}^{\infty}\frac{(2\pi L^2)^{2n}}{(n!)^2}\right) \leq \frac{1}{4} + 2L^2\left(1 + \sum_{n=1}^{\infty}\frac{(2\pi L^2)^{2n}}{(n/e)^{2n}}\right) \\
&= \frac{1}{4} + 2L^2\left(1 + \sum_{n=1}^{\infty}\left(\frac{2\pi e L^2}{n}\right)^{2n}\right).
\end{aligned}$$

Thinking of $(2\pi e L^2/n)^{2n}$ as a continuous function of n, a simple derivative exercise shows that it is maximized for $n = 2\pi L^2$, with value $e^{4\pi L^2}$. Therefore, we can upper bound the series in the expression above as follows:

$$\begin{aligned}
\sum_{n=1}^{\infty}\left(\frac{2\pi e L^2}{n}\right)^{2n} &= \sum_{n=1}^{\lfloor 2\sqrt{2}\pi e L^2\rfloor}\left(\frac{2\pi e L^2}{n}\right)^{2n} + \sum_{n=\lceil 2\sqrt{2}\pi e L^2\rceil}^{\infty}\left(\frac{2\pi e L^2}{n}\right)^{2n} \\
&\leq 2\sqrt{2}\pi e L^2 e^{4\pi L^2} + \sum_{n=\lceil 2\sqrt{2}\pi e L^2\rceil}^{\infty}\left(\frac{1}{2}\right)^n \leq 3\pi e L^2 e^{4\pi L^2},
\end{aligned}$$

where the last transition is by the assumption that $L \geq 3$. Substituting into the bound on B, we get the result stated in the lemma. ∎

Appendix D. Proof of Lemma 4.1. Our proof technique is closely related to the one in [16]. In particular, we use the same kind of approximating polynomials (based on Hermite polynomials). The main difference is that while in [16] the degree of the approximating polynomial was the dominating factor, for our algorithm the dominating factor is the size of the coefficients in the polynomial. We note that we have made no special attempt to optimize the proof or the choice of polynomials to our algorithm, and it is likely that the result below can be substantially improved. To maintain uniformity with the rest of the paper, we will assume that the halfspace with which we compete passes through the origin, although the analysis below can be easily extended when we relax this assumption.

For the proof, we will need two auxiliary lemmas. The first provides a polynomial approximation to ϕ_{0-1}, which is an L_2 approximation to ϕ_{0-1} under a Gaussian-like weighting, using *Hermite polynomials*. The second lemma shows how to transform this L_2 approximating polynomial into a new L_1 approximating polynomial.

LEMMA D.1. *For any $d > 0$, there is a degree-d univariate polynomial $p_d(x) = \sum_{j=0}^{d} \beta_j x^j$ such that*

$$\int_{-\infty}^{\infty} (p_d(x) - \operatorname{sgn}(x))^2 \frac{\exp(-x^2)}{\sqrt{\pi}} dx = O\left(\frac{1}{\sqrt{d}}\right). \tag{D.1}$$

Moreover, it holds that $|\beta_j| \leq O(2^{(j+d)/2})$.

Proof. Our proof closely follows that of Theorem 6 in [16]. In that theorem, a certain polynomial is constructed, and it is proved there that it satisfies (D.1). Thus, to prove the lemma it is enough to show the bound on the coefficients of that polynomial. The polynomial is defined there as

$$p_d(x) = \sum_{i=0}^{d} c_i \bar{H}_i(x),$$

where $\bar{H}_i(x) = H_i(x)/\sqrt{2^i i!}$, $H_i(x)$ is the ith Hermite polynomial, and

$$c_i = \int_{-\infty}^{\infty} \operatorname{sgn}(x) \bar{H}_i(x) \frac{\exp(-x^2)}{\sqrt{\pi}} dx \ .$$

In the proof of Theorem 6 in [16], it is shown that $|c_i| \leq C i^{-3/4}$, where $C > 0$ is an absolute constant. Letting β_j be the coefficient of x^j in $p_d(x)$, and letting $h_{n,j}$ be the coefficient of x^j in $H_n(x)$, we have

$$|\beta_j| = \left| \sum_{n=j}^{d} c_n \frac{h_{n,j}}{\sqrt{2^n n!}} \right| \leq C \sum_{n=j}^{d} \frac{|h_{n,j}|}{\sqrt{2^n n!}}. \tag{D.2}$$

Now, using a standard formula for $h_{n,j}$ (cf. [19]),

$$|h_{n,j}| = 2^j \frac{n!}{j! \left(\frac{n-j}{2}\right)!}$$

whenever $n = j \mod 2$; otherwise $h_{n,j} = 0$. Therefore, we have that for any n, j,

$$\frac{|h_{n,j}|}{\sqrt{2^n n!}} \leq 2^{j-n/2} \sqrt{\frac{n!}{(j!)^2 \left(\left(\frac{n-j}{2}\right)!\right)^2}}. \tag{D.3}$$

Now, we claim that $(((n-j)/2)!)^2 \geq (n-j)! 2^{j-n}$. This follows from $(((n-j)/2)!)^2$

being equal to

$$\prod_{i=0}^{\frac{n-j}{2}-1}\left(\frac{n-j-2i}{2}\right)\left(\frac{n-j-2i}{2}\right)$$
$$\geq \prod_{i=0}^{\frac{n-j}{2}-1}\left(\frac{n-j-2i}{2}\right)\left(\frac{n-j-2i-1}{2}\right) = 2^{j-n}(n-j)!.$$

Plugging this into (D.3), we get that $|h_{n,j}|/\sqrt{2^n n!}$ is at most

$$2^{j/2}\sqrt{\frac{n!}{(j!)^2(n-j)!}} \leq 2^{j/2}\sqrt{\frac{n!}{j!(n-j)!}} = 2^{j/2}\sqrt{\binom{n}{j}} \leq 2^{j/2}2^{n/2}.$$

Plugging this into (D.2) and simplifying, the second part of the lemma follows. ∎

LEMMA D.2. *For any positive integer d, define the polynomial*

$$Q'_d(x) = p_d\left(\sqrt{\frac{n-3}{2}}x\right),$$

where $p_d(\cdot)$ is defined as in Lemma D.1. *Let $\mathcal{U}$ denote the uniform distribution on S^{n-1}. Then for any $\mathbf{w} \in S^{n-1}$,*

$$\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{U}}[(Q'_d(\mathbf{w}\cdot\mathbf{x}) - \operatorname{sgn}(\mathbf{w}\cdot\mathbf{x}))^2] \leq O(1/\sqrt{d}).$$

As a result, if we define $Q_d(x) = Q'_d/2 + 1/2$, we get

$$\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{U}}[(Q_d(\mathbf{w}\cdot\mathbf{x}) - \phi_{0-1}(\mathbf{w}\cdot\mathbf{x}))^2] \leq O(1/\sqrt{d}).$$

The first part of this lemma is identical (up to notation) to Theorem 6 in [16], and we refer the reader to it for the proof. The second part is an immediate corollary.

With these lemmas at hand, we are now ready to prove the main result. Using the polynomial $Q_d(\cdot)$ from Lemma D.2, we know it belongs to P_B for

$$B = \sum_{j=0}^{d} 2^j\left(\left(\sqrt{\frac{n}{2}}\right)^j \beta_j\right)^2 \leq O\left(\sum_{j=0}^{d} n^j 2^j 2^d\right) = O\left((4n)^d\right). \tag{D.4}$$

Now, recall by Theorem 2.2 that if we run our algorithm with these parameters, then the returned hypothesis $\tilde{f}$ satisfies the following with probability at least $1-\delta$:

$$\operatorname{err}(\tilde{f}) \leq \mathbb{E}[|Q_D(\langle\mathbf{w}^*,\mathbf{x}\rangle) - y|] + O\left(\sqrt{\frac{B\log(1/\delta)}{m}}\right). \tag{D.5}$$

Using Lemma D.2, we have that

$$\left|\mathbb{E}[|Q(\langle\mathbf{w}^*,\mathbf{x}\rangle) - y|] - \mathbb{E}[|\phi_{0-1}(\langle\mathbf{w}^*,\mathbf{x}\rangle) - y|]\right| \leq \mathbb{E}[|Q(\langle\mathbf{w}^*,\mathbf{x}\rangle) - \phi_{0-1}(\langle\mathbf{w}^*,\mathbf{x}\rangle)|]$$
$$\leq \sqrt{\mathbb{E}[(Q(\langle\mathbf{w}^*,\mathbf{x}\rangle) - \phi_{0-1}(\langle\mathbf{w}^*,\mathbf{x}\rangle))^2]} \leq O(d^{-1/4}).$$

Plugging this back into (D.5) and choosing $d = \Theta(1/\epsilon^4)$, the result follows.

Acknowledgments. We would like to thank Adam Klivans for helping with the hardness results, as well as the anonymous reviewers for their detailed and helpful comments.

REFERENCES

[1] C. Thomas-Agnan and A. Berlinet, *Reproducing Kernel Hilbert Spaces in Probability and Statistics*, Springer, Berlin, New York, 2003.
[2] N. I. Achieser, *Theory of Approximation*, Frederick Ungar Publishing, New York, 1956.
[3] 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, MIT Press, Cambridge, MA, 1997, pp. 134–140.
[4] P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe, *Convexity, classification, and risk bounds*, J. Amer. Statist. Assoc., 101 (2006), pp. 138–156.
[5] P. L. Bartlett and S. Mendelson, *Rademacher and Gaussian complexities: Risk bounds and structural results*, J. Mach. Learn. Res., 3 (2002), pp. 463–482.
[6] S. Ben-David, *Alternative measures of computational complexity with applications to agnostic learning*, in Theory and Applications of Models of Computation, Lecture Notes in Comput. Sci. 3959, Springer, Berlin, 2006, pp. 231–235.
[7] S. Ben-David and H. Simon, *Efficient learning of linear perceptrons*, in Advances in Neural Information Processing Systems 13, MIT Press, Cambridge, MA, 2000, pp. 189–195.
[8] E. Blais, R. O'Donnell, and K. Wimmer, *Polynomial regression under arbitrary product distributions*, in Proceedings of the 21st Annual Conference on Learning Theory, R. Servedio and T. Zhang, eds., Omnipress, Madison, WI, 2008, pp. 193–204.
[9] L. Bottou and O. Bousquet, *The tradeoffs of large scale learning*, in Advances in Neural Information Processing Systems 20, MIT Press, Cambridge, MA, 2008, pp. 161–168.
[10] O. Bousquet, *Concentration Inequalities and Empirical Processes Theory Applied to the Analysis of Learning Algorithms*, Ph.D. thesis, École Polytechnique, Palaiseau, France, 2002.
[11] N. Cristianini and J. Shawe-Taylor, *Kernel Methods for Pattern Analysis*, Cambridge University Press, Cambridge, UK, 2004.
[12] D. Elliot, *The evaluation and estimation of the coefficients in the Chebyshev series expansion of a function*, Math. Comp., 18 (1964), pp. 274–284.
[13] V. Feldman, P. Gopalan, S. Khot, and A. K. Ponnuswami, *New results for learning noisy parities and halfspaces*, in Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science, IEEE Press, Piscataway, NJ, 2006, pp. 563–574.
[14] V. Guruswami and P. Raghavendra, *Hardness of learning halfspaces with noise*, in Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS), IEEE Press, Piscataway, NJ, 2006, pp. 543–552.
[15] S. Kakade, K. Sridharan, and A. Tewari, *On the complexity of linear prediction: Risk bounds, margin bounds, and regularization*, in Advances in Neural Information Processing 21, MIT Press, Cambridge, MA, 2009, pp. 793–800.
[16] A. Kalai, A. R. Klivans, Y. Mansour, and R. Servedio, *Agnostically learning halfspaces*, in Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS), IEEE Press, Piscataway, NJ, 2005, pp. 11–20.
[17] M. Kearns, R. Schapire, and L. Sellie, *Toward efficient agnostic learning*, Machine Learning, 17 (1994), pp. 115–141.
[18] A. Klivans and A. Sherstov, *Cryptographic hardness for learning intersections of halfspaces*, in Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science, IEEE Press, Piscataway, NJ, 2006, pp. 553–562.
[19] N. N. Lebedev, *Special Functions and Their Applications*, Dover, New York, 1972.
[20] J. Mason, *Chebyshev Polynomials*, CRC Press, Boca Raton, FL, 2003.
[21] D. McAllester, *Simplified PAC-Bayesian margin bounds*, in Proceedings of the 16th Annual Conference on Computational Learning Theory (COLT), Lecture Notes in Comput. Sci. 2777, Springer, Berlin, New York, 2003, pp. 203–215.
[22] R. Schapire, *The strength of weak learnability*, Machine Learning, 5 (1990), pp. 197–227.
[23] B. Schölkopf and A. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*, MIT Press, Cambridge, MA, 2002.
[24] S. Shalev-Shwartz and N. Srebro, *SVM optimization: Inverse dependence on training set size*, in Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 2008, pp. 928–935.
[25] I. Steinwart and C. Scovel, *Fast rates for support vector machines using Gaussian kernels*, Ann. Statist., 35 (2007), pp. 575–607.

[26] A. Tsybakov, *Optimal aggregation of classifiers in statistical learning*, Ann. Statist., 32 (2004), pp. 135–166.
[27] V. N. Vapnik, *Statistical Learning Theory*, John Wiley, New York, 1998.
[28] G. Wahba, *Spline Models for Observational Data*, CBMS-NSF Reg. Conf. Ser. Appl. Math. 59, SIAM, Philadelphia, 1990.
[29] T. Zhang, *Statistical behavior and consistency of classification methods based on convex risk minimization*, Ann. Statist., 32 (2004), pp. 56–85.