Learning Intersections of Halfspaces with a Margin

Adam R. Klivans*[1] and Rocco A. Servedio[2]

[1] Divsion of Engineering and Applied Sciences, Harvard University
Cambridge, MA 02138
klivans@eecs.harvard.edu

[2] Department of Computer Science, Columbia University
New York, NY 10027, USA
rocco@cs.columbia.edu

Abstract. We give a new algorithm for learning intersections of halfspaces with a margin, i.e. under the assumption that no example lies too close to any separating hyperplane. Our algorithm combines random projection techniques for dimensionality reduction, polynomial threshold function constructions, and kernel methods. The algorithm is fast and simple. It learns a broader class of functions and achieves an exponential runtime improvement compared with previous work on learning intersections of halfspaces with a margin.

1 Introduction

The Perceptron algorithm and Perceptron Convergence Theorem are among the oldest and most famous results in machine learning. The Perceptron Convergence Theorem (see e.g. [10]) states that at most $4/\rho^2$ iterations of the Perceptron update rule are required in order to correctly classify any set S of examples which are consistent with some halfspace which has margin ρ on S. (Roughly speaking, this margin condition means that no example lies within distance ρ of the separating hyperplane; we give precise definitions later.)

Since halfspace learning is so widely used in machine learning algorithms and applications, it is of great interest to develop efficient algorithms for learning intersections of halfspaces and other more complex functions of halfspaces. While this problem has been intensively studied, progress to date has been quite limited; we give a brief overview of relevant previous work on learning intersections of halfspaces at the end of this section.

Our results: toward Perceptron-like performance for learning intersections of halfspaces. In this paper we take a perspective similar to that of the original Perceptron Convergence Theorem by highlighting the role of the margin; our goal is to obtain results analogous to the Perceptron Convergence Theorem for learning intersections of halfspaces with margin ρ. (Roughly speaking, an intersection of t halfspaces has margin ρ relative to a data set if each of

* Supported by an NSF Mathematical Sciences Postdoctoral Research Fellowship.

	Arriaga & Vempala [3]	This Paper
$h_1 \wedge \cdots \wedge h_t$	$n \cdot \text{poly}\left(\frac{\log t}{\rho}\right) + \left(\frac{\log t}{\rho}\right)^{\frac{t \log \frac{t}{\rho}}{\rho^2}}$	$n\left(\frac{t}{\rho}\right)^{t \log t \log \frac{1}{\rho}}$ or $n\left(\frac{\log t}{\rho}\right)^{\sqrt{\frac{1}{\rho}} \log t}$
$f(h_1, \ldots, h_t)$	————————	$n\left(\frac{t}{\rho}\right)^{t^2 \log \frac{1}{\rho}}$

Table 1. Bounds on running time for learning intersections and arbitrary functions of t halfspaces with margin ρ. Each h_i is a halfspace over $\mathbf{R}^n$; in the second line f denotes an arbitrary Boolean function (not known a priori to the learner) on t bits. In each case the target function is assumed to have margin ρ.

the defining halfspaces has margin ρ on the data set; we give precise definitions later.) The margin is a natural parameter to consider; previous work by Arriaga and Vempala [3] on learning intersections of halfspaces has explicitly studied the dependence on this parameter. Since the Perceptron algorithm learns a single halfspace in time $O(1/\rho^2)$, the ultimate goal in this framework would be an algorithm which can learn (say) an intersection of two halfspaces in time polynomial in $1/\rho$ as well.

Table 1 summarizes our main results. For any constant $t = O(1)$ number of halfspaces (in our opinion this is the most interesting case) our learning algorithm runs in $(1/\rho)^{O(\log 1/\rho)}$ time, i.e. quasipolynomial in $1/\rho$. This is an exponential improvement over Arriaga and Vempala's previous result [3] which was an algorithm that runs in $(1/\rho)^{\omega(1/\rho^2)}$ time. (Put another way, our algorithm can learn the intersection of $O(1)$ halfspaces with margin at least $1/2^{\sqrt{\log n}}$ in poly(n) time, whereas Arriaga and Vempala require the margin to be at least $\omega(1/\sqrt{\log n})$ to achieve poly(n) runtime.) In fact, we can learn any Boolean function of $t = O(1)$ halfspaces, not just an intersection of halfspaces, in $(1/\rho)^{O(\log 1/\rho)}$ time.

One can instead consider the number of halfspaces t as the relevant asymptotic parameter and view ρ as fixed at $\Theta(1)$. For this case we give an algorithm which has a $t^{O(\log \log t)}$ dependence on t; this algorithm can learn an intersection of $t = n^{1/\log \log n}$ many halfspaces in poly(n) time. In contrast, the previous algorithm of [3] has a $t^{\omega(t)}$ dependence on t and thus runs in poly(n) time only for $t = o(\frac{\log n}{\log \log n})$ many halfspaces.

As described below all our results are achieved using simple iterative algorithms (in fact using simple variants of the Perceptron algorithm!).

Our Approach. Our algorithm for learning an intersection of t halfspaces in $\mathbf{R}^n$ with margin ρ is given in Figure 1. The algorithm has three conceptual stages: (i) random projection, (ii) polynomial threshold function construction, and (iii) kernel methods used to learn polynomial threshold functions. We now give a brief overview of each of these stages.

Random Projection: Random projection for dimensionality reduction has emerged as a useful tool in many areas of CS theory. The key fact on which most of these applications are based is the Johnson-Lindenstrauss lemma [13]

Algorithm $\mathbf{A}(EX(c,\mathcal{D}))$**:**

1. Let M be a $n \times k$ random projection matrix.
2. Draw m many examples from $EX(c,\mathcal{D})$ and project them to $\mathbf{R}^k$ using M.
3. Run the kernel Perceptron algorithm using the polynomial kernel $K_d(x,y) = (x \cdot y + 1)^d$ over the projected examples until a consistent hypothesis is obtained. Let h' be the kernel Perceptron hypothesis (a mapping from $\mathbf{R}^k$ to $\{-1,1\}$).
4. Output $h : \mathbf{R}^n \to \{-1,1\}$, $h(x) = \text{sign}(h'(M^T x))$ as the final hypothesis.

Fig. 1. The algorithm is given access to a source $EX(c,\mathcal{D})$ of random labelled examples, where the target concept c is an intersection of t halfspaces over $\mathbf{R}^n$ which has margin ρ with respect to distribution $\mathcal{D}$. The values of m, k and d are given in Section 6.

which shows that a random projection of a set of m points in $\mathbf{R}^n$ into $\mathbf{R}^k$ (with $k \approx O(\frac{\log m}{\epsilon^2})$) with high probability will not change pairwise distances by more than a $(1 \pm \epsilon)$ factor. Arriaga and Vempala [3] were the first to give learning algorithms based on random projections. Their key insight was that since the geometry of a sample does not change much under random projection, one can run learning algorithms in the low dimensional space $\mathbf{R}^k$ rather than $\mathbf{R}^n$ and thus get a computational savings.

As described in Section 3, the first step of our algorithm is to perform a random projection of the sample from $\mathbf{R}^n$ into a lower dimensional space $\mathbf{R}^k$ where k has no dependence on n. After this projection, with high probability we have data points in $\mathbf{R}^k$ which are labelled according to some intersection of halfspaces with margin $\rho/2$.

Polynomial Threshold Functions: Constructions of polynomial threshold functions (PTFs) have recently proved quite useful in computational learning theory; for example the DNF learning algorithm of [16] has at its heart the fact that any DNF formula can be expressed as a low degree thresholded polynomial $\text{sign}(p(x))$. The second conceptual step of our algorithm is to construct a polynomial threshold function for an intersection of halfspaces over $\mathbf{R}^k$. We show in Section 4 that any intersection of halfspaces with margin $\rho/2$ over $\mathbf{R}^k$ can be expressed as a low-degree polynomial threshold function p over $\mathbf{R}^k$. Moreover, unlike previous analyses (which only gave degree bounds) we show that this PTF p has nonnegligible *PTF margin* (we define PTF margin in Section 2.2). We can thus view our projected data in $\mathbf{R}^k$ as being labelled according to some degree-d PTF over $\mathbf{R}^k$ which has nonnegligible PTF margin. (We emphasize that this is only a conceptual rather than an algorithmic step – the learning algorithm itself does not have to do anything at this stage!)

Kernel Methods: The third step is to learn the low-degree polynomial threshold function over $\mathbf{R}^k$. As shown in Section 5 we do this using the Perceptron algorithm with the standard polynomial kernel $K_d(x,y) = (1 + x \cdot y)^d$. The kernel Perceptron algorithm learns an implicit representation of a halfspace over an expanded feature space; here the expanded space has a feature for each monomial of degree up to d, and thus each example in $\mathbf{R}^k$ corresponds to a point in $\mathbf{R}^{\binom{k+d}{d}}$. We show that since there is a polynomial threshold function which correctly classifies the data in $\mathbf{R}^k$ with some PTF margin, there must be a halfspace over $\mathbf{R}^{\binom{k+d}{d}}$ which correctly classifies the expanded data with a margin, and thus we can use kernel Perceptron to learn.

Comparison with Previous Work. Many researchers have considered the problem of learning intersections of halfspaces. Efficient algorithms are known for learning intersections of halfspaces under the uniform distribution on the unit ball [7, 22] and on the Boolean cube [15], but less is known about learning under more general probability distributions. Baum [4] gave an algorithm which learns an intersection of two origin-centered halfspaces under any symmetric distribution $\mathcal{D}$ (which satisfies $\mathcal{D}(x) = \mathcal{D}(-x)$ for all $x \in \mathbf{R}^n$), and Klivans *et al.* [15] gave a PTF-based algorithm which learns an intersection of $O(1)$ many poly(n)-weight halfspaces over $\{0,1\}^n$ in $n^{O(\log n)}$ time under any distribution.

The most closely related previous work is that of Arriaga and Vempala [3] who gave an algorithm for learning an intersection of halfspaces with margin ρ; see Table 1 for a comparison with their results. Their algorithm uses random projection to reduce dimensionality and then uses a brute-force search over all (combinatorially distinct) halfspaces over the sample data. In contrast, our algorithm combines polynomial threshold functions and kernel methods with random projections, and is able to achieve an exponential runtime savings over [3].

2 Preliminaries

2.1 Concepts and Margins

A *concept* is simply a Boolean function $c : \mathbf{R}^n \to \{-1, +1\}$. A *halfspace* over $\mathbf{R}^n$ is a Boolean function $h : \mathbf{R}^n \to \{-1, 1\}$ defined by a vector $w \in \mathbf{R}^n$ and a value $\theta \in \mathbf{R}$; given an input $x \in \mathbf{R}^n$, the value of $h(x)$ is $\mathrm{sign}(w \cdot x - \theta)$, i.e. $h(x) = +1$ if $w \cdot x \geq \theta$ and $h(x) = -1$ if $w \cdot x < \theta$. An *intersection of t halfspaces* $h_1, \ldots, h_t$ is the Boolean AND of these halfspaces, i.e. the value is 1 if $h_i(x) = 1$ for all $i = 1, \ldots, t$ and is -1 otherwise.

For two vectors $x, y \in \mathbf{R}^n$ we write $\|x - y\|$ to denote the Euclidean distance between x and y and we write S^{n-1} for the unit ball in $\mathbf{R}^n$. We have:

Definition 1. *Given $X \subset \mathbf{R}^n$ and a concept c over $\mathbf{R}^n$, write $\|X\|$ to denote* $\max_{z \in X} \|z\|$. *We say that c has* (geometric) margin ρ with respect to X *if*

$$\rho = \min\{\|z - y\| : z \in X, y \in \mathbf{R}^n, c(z) \neq c(y)\}/\|X\|.$$

Our definition of the geometric margin is similar to the notion of robustness defined in Arriaga and Vempala [3]; the difference is that we normalize by dividing by the radius of the data set $\|X\|$. In the case where $\|X\| = 1$ these notions coincide and the condition is simply that for every $z \in X$, every point within a ball of radius ρ around z has the same label as z under c.

For $\mathcal{D}$ a probability distribution over $\mathbf{R}^n$ we write $\mathrm{Supp}(\mathcal{D})$ to denote the set $\{x \in \mathbf{R}^n : \mathcal{D}(x) > 0\}$. We say that c has *margin ρ with respect to distribution $\mathcal{D}$* if

c has margin ρ on Supp($\mathcal{D}$). Thus, for $\mathcal{D}$ a distribution where Supp($\mathcal{D}$) $\subset S^{n-1}$, an intersection of t halfspaces has margin ρ with respect to $\mathcal{D}$ if every point in Supp($\mathcal{D}$) lies at least distance ρ away from each of the t separating hyperplanes.

Throughout this paper we assume that: (i) All halfspaces in our intersection of halfspaces learning problem are origin-centered, i.e. of the form $\text{sign}(w \cdot x - \theta)$ with $\theta = 0$ – this can be achieved by adding an $(n+1)$st coordinate to each example. (ii) All examples lie on the unit ball S^{n-1} – this can be achieved by adding a new coordinate so that all examples have the same norm and rescaling.

2.2 Polynomial Threshold Functions and PTF Margins

Let $f : \mathbf{R}^n \to \{-1, 1\}$ be a Boolean function and X be a subset of $\mathbf{R}^n$. A real polynomial p in n variables is said to be a *polynomial threshold function (PTF) for f over X* if $\text{sign}(p(x)) = f(x)$ for all $x \in X$. The *degree* of a polynomial threshold function p is simply the degree of the polynomial p. Polynomial threshold functions are well studied in the case where $X = \{0,1\}^n$ or $\{-1,1\}^n$ (see e.g. [5, 16, 18, 20]) but we will consider other more general subsets X.

For $S \subseteq \{x_1, \ldots, x_n\}$ a multiset of variables, we write x_S to denote the monomial $\prod_{i \in S} x_i$. For $p(x) = \sum_S c_S x_S$ a polynomial, we write $\|p\|$ to denote $\sqrt{\sum_S c_S^2}$, i.e. the L_2 norm of the vector of coefficients of p. Given a PTF p over X, we define the *PTF margin of p over X* to be $\min\{|p(z)| : z \in X\}/\|p\|$. Note that if $p(x) = w \cdot x$ is a degree-1 polynomial which has $\|p\| = \sqrt{w_1^2 + \cdots + w_n^2} = 1$, then the PTF margin of p over X is equal to the geometric margin of $\text{sign}(p(x))$ over X (up to scaling by $\|X\|$). However in general for polynomials of degree greater than 1 these two notions are not equivalent.

2.3 The Perceptron Algorithm and Kernel Perceptron

Perceptron is a simple iterative algorithm which finds a linear separator for a labelled data set $X \subset \mathbf{R}^n$ if such a separator exists. The algorithm maintains a weight vector $w \in \mathbf{R}^n$ and a bias $\theta \in \mathbf{R}$ and updates these parameters additively after each example; see e.g. Chapter 2 of [10] for details. The Perceptron Convergence Theorem bounds the number of updates in terms of the maximum margin of any halfspace (the following is adapted from Theorem 2.3 of [10]):

Theorem 1. *Let $X \subset \mathbf{R}^n$ be a set of labelled examples such that there is some halfspace h (which need not be origin-centered) which has margin ρ over X. Then the Perceptron algorithm makes at most $\frac{4}{\rho^2}$ mistakes on X.*

Let $\phi : \mathbf{R}^n \to \mathbf{R}^N$ be a function which we call a *feature expansion*. We refer to $\mathbf{R}^n$ as the original feature space and $\mathbf{R}^N$ as the expanded feature space. The *kernel* corresponding to ϕ is the function $K(x, y) = \phi(x) \cdot \phi(y)$. The use of kernels in machine learning has received much research attention in recent years (see e.g. [10, 12] and references therein).

Given a data set $X \subset \mathbf{R}^n$, it is well known (see e.g. [11]) that the Perceptron algorithm can be simulated over $\phi(X)$ in the expanded feature space $\mathbf{R}^N$ using the kernel function $K(x, y)$ to yield an implicit representation of a halfspace in $\mathbf{R}^N$. If evaluating $K(x, y)$ takes time T and the Perceptron algorithm is simulated until M mistakes are made on a data set X with $|X| = m$, the time required is $O(mTM^2)$ (see e.g. [12, 14]).

3 Random Projections

We say that an $n \times k$ matrix M is a *random projection matrix* if each entry of M is chosen independently and uniformly from $\{-1, 1\}$. We will use the following lemma from Arriaga and Vempala [3] (see Achlioptas [1] for similar results):

Lemma 1. *[3] Fix $\rho < 1$, $0 < c < \frac{1}{2}$ and $w \in \mathbf{R}^n$ with $\|w\| = 1$. Let M be an $n \times k$ random projection matrix. For any $x \in \mathbf{R}^n$ we have*

$$\Pr[w \cdot x - 2c \leq (M^T w) \cdot (M^T x) \leq w \cdot x + 2c] \geq 1 - 6e^{-(c^2 - c^3)k/4} \geq 1 - 6e^{-c^2 k/8}.$$

With this lemma in hand we can establish the main theorem on random projection which we will use:

Theorem 2. *Let X be a set of m points on S^{n-1} and let $h = sign(w \cdot x)$ be a halfspace which has margin ρ on X. Let $k \geq \frac{2048}{\rho^2} \log(\frac{18m}{\delta})$ and let M be a $n \times k$ random projection matrix. Let $M(X) \subset \mathbf{R}^k$ denote the projection of X under M and let $h' : \mathbf{R}^k \to \{-1, +1\}$ denote the function $h'(y) = sign((M^T w) \cdot y)$. Then with probability $1 - \delta$, the halfspace h' correctly classifies $M(X)$ with margin at least $\frac{\rho}{2}$ and we have $\frac{1}{2} \leq \|M(X)\| \leq 2$.*

Proof. We may assume that $\|w\| = 1$. After applying M to the points in X, we need to verify that Definition 1 is satisfied for h' with respect to the points in $M(X)$. Setting $c = \frac{\rho}{16}$ and setting k as above, taking $x = w$ in Lemma 1 we have that with probability at least $1 - \frac{\delta}{3m}$, $\|M^T w\|^2 \leq \|w\|^2 + \frac{\rho}{8} = 1 + \frac{\rho}{8}$, so $\|M^T w\| \leq 1 + \frac{\rho}{16}$.

Now for each point $z \in X$, applying Lemma 1 with $x = z$, with probability at least $1 - \frac{\delta}{3m}$ we have $(w \cdot z) - \frac{\rho}{8} \leq (M^T w) \cdot (M^T z) \leq (w \cdot z) + \frac{\rho}{8}$. Since $|(w \cdot z)| \geq \rho$, this gives $|(M^T w) \cdot (M^T z)| \geq \frac{7\rho}{8}$. Hence with probability at least $1 - \frac{\delta}{2}$ we have $\min\{\|z' - y\| : z' \in M(X), y \in \mathbf{R}^k, h'(z') \neq h'(y)\} \geq \min_{z \in X} |(M^T w) \cdot (M^T z)| / \|M^T w\| \geq \frac{7\rho/8}{1 + \rho/16} \geq \frac{3\rho}{4}$. Lemma 1 similarly implies that $1 - \frac{\rho}{8} \leq \|M(X)\| \leq 1 + \frac{\rho}{16}$ with probability at least $1 - \frac{\delta}{2}$. Thus with probability $1 - \delta$, h' has margin at least $\frac{\rho}{2}$ on $M(X)$ and $\frac{1}{2} \leq \|M(x)\| \leq 2$. □

A union bound yields the following corollary:

Corollary 1. *Let X be a set of m points on S^{n-1} and let $H = \bigwedge_{i=1}^t h_i = sign(w^1 \cdot x) \wedge \ldots \wedge sign(w^t \cdot x)$ be an intersection of t halfspaces which has margin ρ on X. Let $k \geq \frac{2048}{\rho^2} \cdot \log(\frac{18mt}{\delta})$ and let M be a $n \times k$ random projection matrix. Let $M(X) \subset \mathbf{R}^k$ denote the projection of X under M and let $H' = \bigwedge_{i=1}^t sign((M^T w^i) \cdot y)$. Then with probability $1 - \delta$, the intersection of halfspaces H' correctly classifies $M(X)$ with margin at least $\frac{\rho}{2}$ and $\frac{1}{2} \leq \|M(X)\| \leq 2$.*

Thus with high probability the projected set of examples in $\mathbf{R}^k$ is classified by an intersection of halfspaces with margin $\frac{\rho}{2}$. It is easy to see that the corollary in fact holds for any Boolean function (not just intersections) of t halfspaces.

4 Polynomial Threshold Functions for Intersections of Halfspaces with a Margin

In this section we give several constructions of polynomial threshold functions for intersections of halfspaces with a margin. In each case we give a PTF and also a lower bound on the PTF margin of the polynomial threshold function which we construct. These PTF margin lower bounds will be useful when we analyze the performance of kernel methods for learning polynomial threshold functions.

In order to lower bound the PTF margin of a polynomial p we must upper bound $\|p\|$. Fact 3 (proved in Appendix A) helps obtain such upper bounds:

Fact 3 *1. For $i = 1, \ldots, \ell$ let $q_i(x) = \sum_S c_{i,S} x_S$ be a polynomial of degree at most d over $x_1, \ldots, x_k$ with $\|q_i\|^2 \leq M$. Then $\|q_1(x) \ldots q_\ell(x)\|^2 \leq K^\ell M^\ell$ where $K = \binom{k+d}{d}$.*

2. Let $q_1, \ldots, q_\ell$ be polynomials with $\|q_i\|^2 \leq M_i$. Then $\|q_1 + \cdots + q_\ell\|^2 \leq \ell(M_1 + \cdots + M_\ell)$.

4.1 Constructions based on Rational Functions

Recall that a *rational function* is a quotient of two real polynomials, i.e. $Q(x) = a(x)/b(x)$. The *degree* of Q is defined as $\deg(a) + \deg(b)$. Building on earlier results of Newman [17] on rational functions which approximate the absolute value function $|x|$, in [6] Beigel *et al.* gave a construction of a low-degree rational function which closely approximates the function $\operatorname{sgn}(x)$. We will use the following lemma (Lemma 9 of [6]):

Lemma 2. *[6] For all integers $r, \ell \geq 1$ there is a univariate rational function $P_\ell^r(x) = \frac{a(x)}{b(x)}$ of degree $O(\ell \log r)$ with the following properties: (i) $P_\ell^r(x) \in [1, 1 + \frac{1}{r}]$ for all $x \in [1, 2^\ell]$; (ii) $P_\ell^r(x) \in [-1 - \frac{1}{r}, -1]$ for all $x \in [-2^\ell, -1]$; and (iii) Each coefficient of $a(x), b(x)$ has magnitude at most $2^{O(\ell^2 \log r)}$.*

The following theorem generalizes Theorem 24 in [15], which addresses the special case of intersections of low-weight halfspaces over the space $X = \{0,1\}^n$:

Theorem 4. *Let X be a subset of $\mathbf{R}^k$ with $\frac{1}{2} \leq \|X\| \leq 2$ and $c : \mathbf{R}^k \to \{-1, 1\}$ be an intersection of t origin-centered halfspaces $h_1, \ldots, h_t$. If c has margin ρ on X then there is a polynomial threshold function of degree $d = O(t \log t \log \frac{1}{\rho})$ for c on X. If $d \leq k$ then this PTF has PTF margin $(\rho/k)^{O(t \log t \log 1/\rho)}$ on X.*

Proof. We must exhibit a polynomial $p(x)$ of the claimed degree such that for any $z \in X$ we have $\operatorname{sign}(p(z)) = c(z)$ and $\frac{|p(z)|}{\|p\|} \geq (k/\rho)^{O(t \log t \log 1/\rho)}$.

Let $w^1 \cdot x = 0, \ldots, w^t \cdot x = 0$ be the t hyperplanes which define halfspaces $h_1, \ldots, h_t$; we may assume without loss of generality that each $\|w^i\| = 1$. Now consider the sum of rational functions

$$Q(x) = P^{2t}_{\log 4/\rho}(2(w^1 \cdot x)/\rho) + \cdots + P^{2t}_{\log 4/\rho}(2(w^t \cdot x)/\rho) - t + 1/2.$$

Fix any $z \in X$. Since c has margin ρ on X and $\frac{1}{2} \le \|X\| \le 2$, for each $i = 1, \ldots, t$ we have $\frac{\rho}{2} \le \rho\|X\| \le |w^i \cdot z| \le \|w^i\| \cdot \|X\| \le 2$ and hence $|2(w^i \cdot z)/\rho| \in [1, \frac{4}{\rho}]$. Consequently $P^{2t}_{\log 4/\rho}(\frac{2(w^i \cdot z)}{\rho})$ lies in $[1, 1+\frac{1}{2t}]$ if $h_i(z) = 1$ and lies in $[-1-\frac{1}{2t}, -1]$ if $h_i(z) = -1$. Thus if $h_i(z) = 1$ for all i we have $Q(z) \ge t - t + \frac{1}{2} = \frac{1}{2}$, and if $h_i(z) = -1$ for some i we have $Q(z) < -1 + (t-1) + \frac{(t-1)}{2t} - t + \frac{1}{2} < -\frac{1}{2}$. So $\operatorname{sign}(Q(z)) = c(z)$ for all $z \in X$.

Since $Q(x)$ is a sum of t rational functions of degree $O(\log t \log \frac{1}{\rho})$, we can clear denominators and re-express $Q(x)$ as a single rational function $A(x)/B(x)$ of degree $O(t \log t \log \frac{1}{\rho})$. It follows that the function $p(x) = A(x)B(x)$, which is a polynomial of degree $O(t \log t \log \frac{1}{\rho})$, has $\operatorname{sign}(p(z)) = \operatorname{sign}(Q(z))$ as desired.

Now we must bound $\|p\|$. We have $\|\frac{2w^i \cdot x}{\rho}\|^2 = \frac{4}{\rho^2}$ so by part (1) of Fact 3 we have that $\|(\frac{2w^i \cdot x}{\rho})^j\|^2 \le (\frac{4k}{\rho^2})^j$ for all j. By Lemma 2 we have that $P^{2t}_{\log 4/\rho}(x) = \frac{a(x)}{b(x)}$ where $a(x), b(x)$ are polynomials of degree $O(\log t \log \frac{1}{\rho})$ with coefficients of magnitude at most $2^{O((\log \frac{1}{\rho})^2 \log t)} = (\frac{1}{\rho})^{O(\log t \log 1/\rho)}$. It follows from part (2) of Fact 3 that $\|a(\frac{2w^i \cdot x}{\rho})\|^2 \le (\frac{k}{\rho})^{O(\log t \log 1/\rho)} \cdot (\frac{1}{\rho})^{O(\log t \log 1/\rho)}$ which equals $(\frac{k}{\rho})^{O(\log t \log 1/\rho)}$, and the same holds for $\|b(\frac{2w^i \cdot x}{\rho})\|^2$. Expressing $Q(x)$ as a rational function $A(x)/B(x)$, we have that $B(x) = \prod_{i=1}^t b(\frac{2w^i \cdot x}{\rho})$, so since $d \le k$ part (1) of Fact 3 implies that $\|B(x)\|^2 \le k^{O(t \log t \log 1/\rho)}(\frac{k}{\rho})^{O(t \log t \log 1/\rho)} = (\frac{k}{\rho})^{O(t \log t \log 1/\rho)}$. Simple calculations using part (1) of Fact 3 show that $\|A(x)\|^2$ and $\|p(x)\| = \|A(x)B(x)\|$ are also $(k/\rho)^{O(t \log t \log 1/\rho)}$, and we are done. □

By modifying this construction, we get a polynomial threshold function for any Boolean function of t halfspaces rather than just an intersection (at a relatively small cost in degree and PTF margin):

Theorem 5. *Let $f : \{-1,1\}^t \to \{-1,1\}$ be any Boolean function on t bits. Let X be a subset of $\mathbf{R}^k$ with $\frac{1}{2} \le \|X\| \le 2$ and $c : \mathbf{R}^k \to \{-1,1\}$ be the function $f(h_1, \ldots, h_t)$ where $h_1, \ldots, h_t$ are origin-centered halfspaces in $\mathbf{R}^k$. If c has margin ρ on X then there is a PTF of degree $d = O(t^2 \log \frac{1}{\rho})$ for c on X. If $d \le k$ then this PTF has PTF margin $(\rho/k)^{O(t^2 \log 1/\rho)}$ on X.*

Proof. As before, we give a polynomial $p(x)$ of the claimed degree such that for any $z \in X$ we have $\operatorname{sign}(p(z)) = c(z)$ and $\frac{|p(z)|}{\|p\|} \ge (k/\rho)^{O(t^2 \log 1/\rho)}$.

Again let $w^1 \cdot x = 0, \ldots, w^t \cdot x = 0$ be the hyperplanes for halfspaces $h_1, \ldots, h_t$, where each w^i is a unit vector. For each $i = 1, \ldots, t$ consider the rational function

$$Q_i(x) = P^{2^{3t}}_{\log 4/\rho}\left(2(w^i \cdot x)/\rho\right).$$

Fix any $z \in X$. As before we have that $|2(w^i \cdot z)/\rho| \in [1, \frac{4}{\rho}]$, so by Lemma 2 the value of $Q_i(z)$ differs from the ± 1 value $h_i(z) = \text{sign}(w^i \cdot z)$ by at most $\frac{1}{2^{3t}}$. Since f is a Boolean function on t inputs, it is expressible as a multilinear polynomial $\tilde{f}$ of degree t, with coefficients of the form $i/2^t$ where i is an integer in $[-2^t, 2^t]$. (The polynomial $\tilde{f}$ is just the Fourier representation of f.) Multiply $\tilde{f}$ by 2^t, so now $\tilde{f} : \{+1, -1\}^t \to \{+2^t, -2^t\}$, and $\tilde{f}$ has integer coefficients which are at most 2^t in absolute value.

Now we would like to argue that $\tilde{f}(Q_1(z), \ldots, Q_t(z))$ has the same sign as $f(h_1(z), \ldots, h_t(z))$. To do this we show that the "error" of each $Q_i(z)$ relative to the ± 1 value $h_i(z)$ (which error is at most $\frac{1}{2^{3t}}$) does not cause $\tilde{f}$ to have the wrong sign. The polynomial $\tilde{f}$ has at most 2^t terms, each of which is the product of an integer coefficient of magnitude at most 2^t and up to t of the Q_i's. The product of the Q_i's incurs error at most $O(t2^{-3t})$ relative to the corresponding product of the h_i's, and thus the error of any given term (including the integer coefficient) is at most $O(t2^{-2t})$. Since we add up at most 2^t terms, the overall error is at most $O(t2^{-t})$ error, which is much less than what we could tolerate (we could tolerate error 2^t; recall that $\tilde{f}$ takes value $\pm 2^t$ on ± 1 inputs). Thus $\tilde{f}(Q_1(z), \ldots, Q_t(z))$ has the same sign as $f(h_1(z), \ldots, h_t(z))$ for all $z \in X$.

Now $\tilde{f}$ is a multilinear polynomial of degree t, and each Q_i is a rational function of degree $O(t \log w)$. We can bring $\tilde{f}(Q_1, \ldots, Q_t)$; to a common denominator (which is the product of the denominators of the Q_i's) of degree $O(t^2 \log w)$. Hence we have a single multivariate rational function $A(x)/B(x)$ which takes the right sign on z, and we can convert this rational function to a polynomial threshold function $p(x) = A(x)B(x)$ as in the proof of Theorem 4.

Now we must bound $\|p\|$. Let $Q_i(x) = \frac{a_i(x)}{b_i(x)}$. The analysis from the previous proof implies that $\|a_i(x)\|^2$ and $\|b_i(x)\|^2$ are both at most $(\frac{k}{\rho})^{O(t \log 1/\rho)}$. Now consider a monomial (in the "variables" $Q_1(x), \ldots, Q_t(x)$) in the polynomial $\tilde{f}(Q_1(x), \ldots, Q_t(x))$. Since the numerator $\alpha(x)$ of such a monomial is the product of at most t of the $a_i(x)$'s, and each $a_i(x)$ has degree at most $O(\log t \log \frac{1}{\rho})$, the fact that $d \le k$ and part (1) of Fact 3 together give $\|\alpha(x)\|^2 \le k^{O(t \log t \log 1/\rho)} (\frac{k}{\rho})^{O(t^2 \log 1/\rho)}$ which equals $(\frac{k}{\rho})^{O(t^2 \log 1/\rho)}$. The same holds for the denominator $\beta(x)$ of such a monomial. Since the common denomiator for $\tilde{f}(Q_1, \ldots, Q_t)$ is the product of the denominators of the Q_i's, clearing all denominators we have that $\tilde{f}(Q_1, \ldots, Q_t) = A(x)/B(x)$ with $\|A(x)\|^2$ and $\|B(x)\|^2$ both at most $(\frac{k}{\rho})^{O(t^2 \log 1/\rho)}$. We thus have $\|p(x)\|^2 = \|A(x)B(x)\|^2 = (\frac{k}{\rho})^{O(t^2 \log 1/\rho)}$ and the theorem is proved. □

4.2 Constructions using Extremal Polynomials

The bounds from the previous section are quite strong when t is relatively small. If t is large but ρ is also quite large, then the following bounds based on Chebyshev polynomials are better.

The r-th Chebyshev polynomial of the first kind, $T_r(x)$, is a univariate degree-r polynomial with the following properties [9]:

Lemma 3. *The polynomial* $T_r(x) = \sum_{i=0}^{r} a_i x^i$ *satisfies: (i)* $|T_r(x)| \leq 1$ *for* $|x| \leq 1$ *with* $T_r(1) = 1$*; (ii)* $T_r'(x) \geq r^2$ *for* $x > 1$ *with* $T_r'(1) = r^2$*; and (iii) For* $i = 0, \ldots, r$ *each* a_i *is an integer with* $|a_i| \leq 2^r$.

The following theorem generalizes results in [16]:

Theorem 6. *Let* X *be a subset of* $\mathbf{R}^k$ *with* $\frac{1}{2} \leq \|X\| \leq 2$ *and let* $c : \mathbf{R}^k \to \{-1, 1\}$ *be an intersection of* t *origin-centered halfspaces* $h_1, \ldots, h_t$*. If* c *has margin* ρ *on* X *then there is a PTF of degree* $d = O(\sqrt{1/\rho} \log t)$ *for* c *on* X*. If* $d \leq k$ *then this PTF has PTF margin* $1/k^{O(\sqrt{1/\rho} \log t)}$ *on* X.

Proof: As in the previous proofs we must exhibit a polynomial $p(x)$ such that for any $z \in X$ we have $\operatorname{sign}(p(z)) = c(z)$ and $\frac{|p(z)|}{\|p\|} \geq 1/k^{O(\sqrt{1/\rho} \log t)}$.

Let $w^1 \cdot x = 0, \ldots, w^t \cdot x = 0$ be the t hyperplanes for halfspaces $h_1, \ldots, h_t$ where each $\|w^i\| = 1$. Let P be the univariate polynomial $P(x) = T_r(1-x)$ where $r = \lceil \sqrt{2/\rho} \rceil$. The first part of Lemma 3 implies that $|P(x)| \leq 1$ for $x \in [0, 2]$, and the second part implies that $P(x) \geq 2$ for $x \leq \frac{-\rho}{2}$. Now consider the polynomial threshold function $\operatorname{sign}(p(x))$ where

$$p(x) = t + \frac{1}{2} - \sum_{i=1}^{t} (P(w^i \cdot x))^{\lceil \log 2t \rceil}.$$

Since P is a polynomial of degree $r = \lceil \sqrt{2/\rho} \rceil$ and $w^i \cdot x$ is a polynomial of degree 1, this polynomial threshold function has degree $d = \lceil \sqrt{2/\rho} \rceil \cdot \lceil \log 2t \rceil$. We now show that $p(x)$ has the desired properties described above.

We first show that for any $z \in X$ the polynomial p takes the right sign and has magnitude at least $\frac{1}{2}$. Fix any $z \in X$. For each $i = 1, \ldots, t$ we have $\frac{\rho}{2} \leq \rho \|X\| \leq |w^i \cdot z| \leq \|w^i\| \cdot \|X\| \leq 2$.

- If $c(z) = 1$ then for each i we have $\frac{\rho}{2} \leq w^i \cdot z \leq 2$ and hence we have that $P(w^i \cdot z)$ (and also $P(w^i \cdot z)^{\lceil \log 2t \rceil}$ lies in $[-1, 1]$. Consequently we have that $p(z) \geq t + \frac{1}{2} - t \geq \frac{1}{2}$ so $\operatorname{sign}(p(z)) = c(z) = 1$.
- If $c(z) = -1$ then for some i we have $w^i \cdot z \in [-2, -\frac{\rho}{2}]$, so consequently $P(w^i \cdot z) \geq 2$ and $P(w^i \cdot z)^{\lceil \log 2t \rceil} \geq 2t$. Since $P(w^j \cdot z)^{\lceil \log 2t \rceil} \geq -1$ for all j, we have $p(z) \leq t + \frac{1}{2} - 2t + (t-1) = -\frac{1}{2}$ so $\operatorname{sign}(p(z)) = c(z) = -1$.

To finish the proof it remains to bound $\|p\|$. Since $\|w^i \cdot x\|^2 = 1$ for all i, by part 2 of Fact 3 we have $\|1 - w^i \cdot x\|^2 \leq 4$ so by part 1 of Fact 3 we have that $\|(1 - w^i \cdot x)^j\| \leq (4k)^j$ for $j = 0, \ldots, r$. Since (by Lemma 3) $T_r(x) = \sum_{j=0}^{r} a_j x^j$ where each $|a_j| \leq 2^r$, for each $j = 0, \ldots, r$ we have $\|a_j(1 - w^i \cdot x)^j\|^2 \leq 2^{2r}(4k)^r$. By part 2 of Fact 3 we obtain $\|T_r(1 - w^i \cdot x)\|^2 \leq (r+1)^2(16k)^r$, and now part 1 implies that $(P(w^i \cdot x))^{\lceil \log 2t \rceil} = k^{O(r \log t)}$. Using part 2 again we obtain that $\|p\| \leq (t+1)^2 k^{O(r \log t)} = k^{O(r \log t)}$, and the theorem is proved. □

As Arriaga and Vempala observed in [3], DNF formulas can be viewed as unions of halfspaces. If we rescale the cube so that it is a subset of S^{k-1}, it is easy to check that a Boolean function $f : \{-1, 1\}^k \to \{-1, 1\}$ has margin ρ with respect to $X \subseteq \{-1,1\}^k$ if for every $z \in X$ we have that every Boolean string z' which differs from z in at most a $\frac{\rho^2}{4}$ fraction of bits has $f(z') = f(z)$.

Since any DNF formula with t terms can be expressed as a union of t halfspaces, we have the following corollary of Theorem 6:

Corollary 2. *Let $X \subset \{-1,1\}^k$ and let c be a t-term DNF formula on k variables. If c has margin ρ on X then there is a polynomial threshold function of degree $O(\sqrt{1/\rho}\log t)$ for c on X which has PTF margin $1/k^{O(\sqrt{1/\rho}\log t)}$ on X. If $d \le k$ then this PTF has PTF margin $(1/k)^{O(\sqrt{1/\rho}\log t)}$ on X.*

A similar corollary for DNF formulas also follows from Theorem 4 but we are most interested in DNFs with t =poly(n) terms so we focus on Theorem 6.

5 Kernel Perceptron for learning PTFs with PTF margin

In this section we first define a new kernel, the Complete Symmetric Kernel, which arises naturally in the context of polynomial threshold functions. We give an efficient algorithm for computing this kernel (which may be of independent interest), and indeed all results of the paper could be proved using this new kernel. To make our overall algorithm simpler, however, we ultimately use the standard polynomial kernel which we discuss later in this section.

Let $\phi_d : \mathbf{R}^k \to \mathbf{R}^{\binom{k+d}{d}}$ be a feature expansion which maps $(x_1, \ldots, x_k)$ to the vector $(1, x_1, \ldots, x_k, x_1^2, x_1x_2, \ldots)$ containing all monomials of degree up to d. Let $K_d(x,y) = \phi_d(x) \cdot \phi_d(y)$ be the kernel corresponding to ϕ_d. We refer to $K_d(x,y)$ as the *complete symmetric kernel* since as explained in Appendix B the value $K_d(x,y)$ equals the sum of certain complete symmetric polynomials.

For a data set $X \subset \mathbf{R}^k$ we write $\phi_d(X)$ to denote the expanded data set of points in $\mathbf{R}^{\binom{k+d}{d}}$. The following lemma gives a mistake bound for the Perceptron algorithm using the complete symmetric kernel:

Lemma 4. *Let $X \subset \mathbf{R}^k$ be a set of labelled examples such that there is some degree-d polynomial threshold function $p(x)$ which correctly classifies X and has PTF margin ρ over X. Then the Perceptron algorithm (run on $\phi_d(X)$ using the complete symmetric kernel K_d) makes at most $\frac{4\|\phi_d(X)\|^2}{\rho^2}$ mistakes on X.*

Proof. The vector $W \in \mathbf{R}^{\binom{k+d}{d}}$ whose coordinates are the coefficients of p has margin $\frac{\min_{z\in X}|W\cdot\phi_d(z)|}{\|W\|\cdot\|\phi_d(X)\|}$ over $\phi_d(X)$. Since $W \cdot \phi_d(z) = p(z)$ and $\|W\| = \|p\|$, the lemma follows by from the definition of the PTF margin of p and the Perceptron Convergence Theorem (Theorem 1). $\square$

In Appendix B we give a polynomial time algorithm for computing $K_d(x,y)$, but this algorithm is somewhat cumbersome. With the aim of obtaining a faster and simpler overall algorithm, we now describe an alternate approach based on the well known polynomial kernel.

As in [10], we define the degree-d polynomial kernel $K'_d : \mathbf{R}^k \times \mathbf{R}^k \to \mathbf{R}$ as $K'_d(x,y) = (1 + x \cdot y)^d$. It is clear that $K'_d(x,y)$ can be computed efficiently. Let $\phi'_d : \mathbf{R}^k \to \mathbf{R}^{\binom{k+d}{d}}$ be the feature expansion such that $K'_d(x,y) = \phi'_d(x) \cdot \phi'_d(y)$; note that $\phi'_d(x)$ differs from $\phi_d(x)$ defined above because of the coefficients that arise in the expansion of $(1 + x \cdot y)^d$.

We have the following polynomial kernel analogue of Lemma 4:

Lemma 5. *Let $X \subset \mathbf{R}^k$ be a set of labelled examples such that there is some degree-d polynomial threshold function $p(x)$ which correctly classifies X and has PTF margin ρ over X. Then the Perceptron algorithm (run on $\phi'_d(X)$ using the polynomial kernel K'_d) makes at most $\frac{4(1+\|X\|^2)^d}{\rho^2}$ mistakes on X.*

Proof. We view $\phi'_d(x)$ as a vector $(a_S x_S)$ of monomials with coefficients. By inspection of the coefficients of $(1 + x \cdot y)^d$ it is clear that each $a_S \geq 1$. Let W' be the vector in $\mathbf{R}^{\binom{k+d}{d}}$ such that $W' \cdot \phi'_d(x) = p(x)$ as a formal polynomial. For each monomial x_S in $p(x)$, the W'_S coordinate of W' equals $W_S/a_S \leq W_S$ where W is defined as in the proof of Lemma 4 so we have $\|W'\| \leq \|W\|$.

The vector W' has margin $\frac{\min_{z \in X} |W' \cdot \phi'_d(z)|}{\|W'\| \cdot \|\phi'_d(X)\|} = \frac{\min_{z \in X} |p(z)|}{\|W'\| \cdot \|\phi'_d(X)\|} \geq \frac{\min_{z \in X} |p(z)|}{\|W\| \cdot \|\phi'_d(X)\|}$ over $\phi'_d(X)$. It is easy to verify that $\|\phi'_d(X)\| \leq (1 + \|X\|^2)^{d/2}$, so W' has margin at least $\frac{\min_{z \in X} |p(z)|}{\|W\| \cdot (1+\|X\|^2)^{d/2}} = \frac{\rho}{(1+\|X\|^2)^{d/2}}$. The lemma now follows from the Perceptron Convergence Theorem. □

The output hypothesis of this kernel Perceptron is an (implicit representation of a) halfspace over $\mathbf{R}^{\binom{k+d}{d}}$ which can be viewed as a polynomial threshold function of degree d over $\mathbf{R}^k$.

6 The Main Results

In this section we give our main learning results by bounding the running time of algorithm A and proving that it outputs an accurate hypothesis.

Our first theorem gives a good bound for the case where t is relatively small:

Theorem 7. *Algorithm A learns any ρ-margin intersection of t halfspaces over $\mathbf{R}^n$ in at most $\frac{n}{\epsilon} \cdot (\frac{t}{\rho} \log \frac{1}{\delta\epsilon})^{O(t \log t \log 1/\rho)}$ time steps.*

Proof. Let c be an intersection of t origin-centered halfspaces over $\mathbf{R}^n$ which has margin ρ with respect to distribution $\mathcal{D}$ where $\mathrm{Supp}(\mathcal{D}) \subset S^{n-1}$. Let m equal the number of examples our algorithm draws from $EX(c, \mathcal{D})$; we defer specifying m until the end of the proof. Let $k = O(\frac{1}{\rho^2} \cdot \log \frac{mt}{\delta})$, and $d = O(t \log t \log \frac{1}{\rho})$. Let X be the set of m examples in $\mathbf{R}^n$, and let $M(X)$ be the projected set of m examples in $\mathbf{R}^k$. Note that it takes nkm time steps to construct the set $M(X)$.

By Corollary 1, with probability $1 - \delta$ we have that $\frac{1}{2} \leq \|M(X)\| \leq 2$ and there is an intersection of t origin-centered halfspaces in $\mathbf{R}^k$ which has margin at least $\frac{\rho}{2}$ on $M(X)$. By Theorem 4 there is a polynomial threshold function over $\mathbf{R}^k$ of degree $d = O(t \log t \log \frac{1}{\rho})$ which has PTF margin $(\frac{\rho}{k})^{O(d)}$ with respect to $M(X)$. By Lemma 5 the degree-d polynomial kernel Perceptron algorithm makes at most $(\frac{k}{\rho})^{O(d)}$ mistakes when run on $M(X)$, and thus once $M(X)$ is obtained the algorithm runs for at most $m \cdot (\frac{k}{\rho})^{O(d)} = (\frac{k}{\rho})^{O(d)}/\epsilon$ time steps.

Now we show that with probability $1-\delta$ algorithm A outputs an ϵ-accurate hypothesis for c relative to $\mathcal{D}$. Since the output hypothesis $h(x) = \mathrm{sign}(p(Mx))$ is computed by first projecting $x \in \mathbf{R}^n$ down to $\mathbf{R}^k$ via M and then evaluating the k-variable PTF p, it suffices to show that p is a good hypothesis under the distribution $M(\mathcal{D})$ obtained by projecting $\mathcal{D}$ down to $\mathbf{R}^k$ via M. It is well known (see e.g. [2]) that the VC dimension of the class of degree-d PTFs over k real variables is $\binom{k+d}{d}$. Thus by the VC theorem [8] in order to learn to accuracy ϵ and confidence δ it suffices to take $m = O(\frac{k^d}{\epsilon}\log\frac{1}{\epsilon} + \frac{1}{\epsilon}\log\frac{1}{\delta})$. It is straightforward to verify that $k = (\frac{d}{\rho}\log\frac{1}{\delta\epsilon})^{O(1)}$, $m = \frac{1}{\epsilon}\cdot(\frac{d}{\rho}\log\frac{1}{\delta\epsilon})^{O(d)}$ satisfy the above conditions on m and k. Since $d = O(t\log t\log\frac{1}{\rho})$ we have $k = (\frac{t}{\rho}\log\frac{1}{\delta\epsilon})^{O(1)}$ and $m = \frac{1}{\epsilon}\cdot(\frac{t}{\rho}\log\frac{1}{\delta\epsilon})^{O(t\log t\log 1/\rho)}$ which proves the theorem. □

Note that for a constant $t = O(1)$ number of halfspaces Algorithm A has a quasipolynomial $((\frac{1}{\rho})^{O(\log 1/\rho)})$ runtime dependence on the margin ρ, in contrast with the exponential $((\frac{1}{\rho})^{O(\log\frac{1}{\rho})/\rho^2})$ dependence of [3].

The proof of Theorem 7 used the polynomial threshold function construction of Theorem 4. We can instead use the construction of Theorem 6 to obtain:

Theorem 8. *Algorithm A learns any ρ-margin intersection of t halfspaces over $\mathbf{R}^n$ in at most $\frac{n}{\epsilon}\cdot(\frac{\log t}{\rho}\log\frac{1}{\delta\epsilon})^{O(\sqrt{1/\rho}\log t)}$ time steps.*

For a constant $\rho = \Theta(1)$ margin Algorithm A has an almost polynomial $((t^{O(\log\log t)})$ runtime dependence on t, in contrast with the exponential $(t^{\omega(t)})$ dependence of [3]. By Corollary 2 the above bound holds for learning t-term DNF with margin ρ as well.

Finally, we can use the construction of Theorem 5 to obtain:

Theorem 9. *Algorithm A learns any Boolean function of t halfspaces with margin ρ in at most $\frac{n}{\epsilon}\cdot(\frac{t}{\rho}\log\frac{1}{\delta\epsilon})^{O(t^2\log 1/\rho)}$ time steps.*

7 Discussion

Is Random Projection Necessary? A natural question is whether our quantitative results could be achieved simply by using kernel Perceptron (or a Support Vector Machine) without first performing random projection. Given a data set X in $\mathbf{R}^n$ classified by an intersection of $t = 2$ halfspaces with margin ρ, Theorem 4 implies the existence of a polynomial threshold function for X of degree $d = O(\log(1/\rho))$ with PTF margin $(\rho/n)^{O(\log(1/\rho))}$. Using either the degree-d polynomial kernel or the Complete Symmetric Kernel, we obtain a halfspace over $\mathbf{R}^{\binom{n+d}{d}}$ which classifies the expanded data set $\phi(X)$ with geometric margin $(\rho/n)^{O(\log(1/\rho))}$.[3] Thus it appears that without the initial projection step, the required sample complexity for either kernel Perceptron or an SVM will be $(n/\rho)^{\Omega(\log(1/\rho))}$, as opposed to the bounds in Section 6 which do not depend on n; so random projection does indeed seem to provide a gain in efficiency.

Lower Bounds on Polynomial Threshold Functions. The main result of O'Donnell and Servedio in [19], if suitably interpreted, proves that there exists a set $X \subset \mathbf{R}^2$ labelled according to the intersection of two halfspaces with margin ρ for which any PTF correctly classifying X must have degree $\Omega(\frac{\log(1/\rho)}{\log\log(1/\rho)})$. This lower bound implies that our choice of d in the proof of Theorem 7 is essentially optimal with respect to ρ. For a discussion of other lower bounds on PTF constructions see Klivans et al. [15].

Alternative Algorithms. We note that after random projection, in Step 3 of Algorithm A there are several other algorithms that could be used instead of kernel Perceptron. For example, we could run a support vector machine over $\mathbf{R}^k$ with the same degree d polynomial kernel to find the maximum margin hyperplane in $\mathbf{R}^{\binom{k+d}{d}}$; alternatively we could even explicitly expand each projected example $M(x) \in \mathbf{R}^k$ into $\phi'_d(M(x)) \in \mathbf{R}^{\binom{k+d}{d}}$ and explicitly run Perceptron (or indeed any algorithm for solving linear programs such as the Ellipsoid algorithm) to learn a single halfspace in $\mathbf{R}^{\binom{k+d}{d}}$. It can be verified that each of these approaches gives the same asymptotic runtime and sample complexity as our kernel Perceptron approach. We use kernel Perceptron both for its simplicity and for its ability to take advantage of the actual margin if it is better than the worst-case bounds presented here.

Future Work and Implications for Practice. We feel that our results give some theoretical justification for the effectiveness of the polynomial kernel in practice, as kernel Perceptron takes direct advantage of the representational power of polynomial threshold functions. We are working on experimentally assessing the algorithm's performance.

Acknowledgements. We thank Santosh Vempala for helpful discussions.

References

[1] D. Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66(4):671–687, 2003.

[2] M. Anthony. Classification by polynomial surfaces. *Discrete Applied Mathematics*, 61:91–103, 1995.

[3] In Arriaga and Vempala [3] it is claimed that if the geometric margin of a degree-d PTF p in $\mathbf{R}^n$ is ρ then the margin of the corresponding halfspace in $\mathbf{R}^{\binom{n+d}{d}}$ is at least ρ^d, but this claim is in error [23]; to bound the margin of the halfspace in $\mathbf{R}^{\binom{n+d}{d}}$ one must analyze the PTF margin of p rather than its geometric margin.

[3] R. Arriaga and S. Vempala. An algorithmic theory of learning: Robust concepts and random projection. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 616–623, 1999.

[4] E. Baum. A polynomial time algorithm that learns two hidden unit nets. *Neural Computation*, 2:510–522, 1991.

[5] R. Beigel. When do extra majority gates help? polylog(n) majority gates are equivalent to one. *Computational Complexity*, 4:314–324, 1994.

[6] R. Beigel, N. Reingold, and D. Spielman. PP is closed under intersection. *Journal of Computer and System Sciences*, 50(2):191–202, 1995.

[7] A. Blum and R. Kannan. Learning an intersection of a constant number of halfspaces under a uniform distribution. *Journal of Computer and System Sciences*, 54(2):371–380, 1997.

[8] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989.

[9] E. Cheney. *Introduction to Approximation Theory*. McGraw-Hill, New York, New York, 1966.

[10] N. Cristianini and J. Shawe-Taylor. *An introduction to Support Vector Machines (and other kernel-based learning methods)*. Cambridge University Press, 2000.

[11] Y. Freund and R. Schapire. Large margin classification using the Perceptron algorithm. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, pages 209–217, 1998.

[12] R. Herbrich. *Learning Kernel Classifiers*. MIT Press, 2002.

[13] W. Johnson and J. Lindenstrauss. Extensions of Lipshitz mapping into Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

[14] R. Khardon, D. Roth, and R. Servedio. Efficiency versus Convergence of Boolean Kernels for On-Line Learning Algorithms. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.

[15] A. Klivans, R. O'Donnell, and R. Servedio. Learning intersections and thresholds of halfspaces. In *Proceedings of the Forty-Third Annual Symposium on Foundations of Computer Science*, pages 177–186, 2002.

[16] A. Klivans and R. Servedio. Learning DNF in time $2^{\tilde{O}(n^{1/3})}$. In *Proceedings of the Thirty-Third Annual Symposium on Theory of Computing*, pages 258–265, 2001.

[17] D. J. Newman. Rational approximation to $|x|$. *Michigan Mathematical Journal*, 11:11–14, 1964.

[18] R. O'Donnell and R. Servedio. Extremal properties of polynomial threshold functions. In *Proceedings of the Eighteenth Annual Conference on Computational Complexity*, pages 3–12, 2003.

[19] R. O'Donnell and R. Servedio. New degree bounds for polynomial threshold functions. In *Proceedings of the 35th ACM Symposium on Theory of Computing*, pages 325–334, 2003.

[20] M. Saks. *Slicing the hypercube*, pages 211–257. London Mathematical Society Lecture Note Series 187, 1993.

[21] A. Shpilka. *Lower Bounds for Small Depth Arithmetic and Boolean Circuits*. PhD thesis, Hebrew University, 2001.

[22] S. Vempala. A random sampling based algorithm for learning the intersection of halfspaces. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science*, pages 508–513, 1997.

[23] S. Vempala. Personal communication, 2004.

[24] J. Zhou. Introduction to symmetric polynomials and symmetric functions. Lecture notes for course at Tsinghua University, available at cms.zju.edu.cn/course/cn/SymmetricF.pdf, 2003.

A Proof of Fact 3

For (1), we have $q_1(x)\dots q_\ell(x) = \sum_{S_1,\dots,S_\ell} c_{1,S_1}\dots c_{\ell,S_\ell} x_{S_1}\dots x_{S_\ell}$ from which it follows that $\|q_1(x)\dots q_\ell(x)\|^2$ is at most

$$\left(\sum_{S_1,\dots,S_\ell} |c_{1,S_1}\dots c_{\ell,S_\ell}|\right)^2 \le K^\ell \sum_{S_1,\dots,S_\ell} (c_{S_1}\dots c_{S_\ell})^2 = K^\ell \prod_{i=1}^{\ell}\left(\sum_{S_i} c_{S_i}^2\right) \le K^\ell M^\ell$$

where the second inequality follows from Cauchy-Schwarz using the fact that each $q_i(x)$ has at most $K = \binom{k+d}{d}$ monomials (so the first sum has at most K^ℓ summands). For (2), we have $q_i(x) = \sum_S c_{i,S} x_S$ so by Cauchy-Schwarz we have

$$\|q_1(x)+\dots+q_\ell(x)\|^2 = \sum_S (c_{1,S}+\dots+c_{\ell,S})^2 \le \ell\left(\sum_S c_{1,S}^2 + \dots + \sum_S c_{\ell,S}^2\right)$$

which is at most $\ell(M_1+\dots+M_\ell)$. □

B Computing the Complete Symmetric Kernel

Recall that $\phi_d : \mathbf{R}^k \to \mathbf{R}^{\binom{k+d}{d}}$ maps $(x_1,\dots,x_k)$ to the vector $(1, x_1,\dots,x_k, x_1^2, x_1x_2,\dots)$ of all monomials of degree up to d, and $K_d(x,y) = \phi_d(x)\cdot\phi_d(y)$.

Lemma 6. *There is a poly(k,d) time algorithm for computing $K_d(x,y)$.*

Proof. Writing z_i for x_iy_i, it is easy to see that $K_d(x,y) = \sum_{\ell=0}^{d} h_\ell(z_1,\dots,z_k)$ where $h_\ell(z_1,\dots,z_k) = \sum_{d_1+\dots+d_k=\ell} z_1^{d_1}\cdots z_k^{d_k}$ is the ℓ-th complete symmetric polynomial (the sum of all monomials of degree exactly ℓ). Let $e_\ell(z_1,\dots,z_k)$ denote the ℓ-th elementary symmetric polynomial (the sum of all multilinear monomials of degree exactly ℓ). As shown in [24] (Equation (8), p. 15) the identity $h_\ell = \det(E)$ is known to hold where E is the $\ell\times\ell$ matrix whose (i,j) entry is e_{1-i+j} (interpreting e_r as 0 for $r<0$). Thus computing $K_d(x,y)$ reduces to computing the polynomials e_ℓ; these polynomials can be computed efficiently via polynomial interpolation (see e.g. Section 2.5 of [21]). □