Learning Intersections and Thresholds of Halfspaces

Adam R. Klivans*
Division of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138
klivans@eecs.harvard.edu

Ryan O'Donnell†
School of Mathematics
Institute for Advanced Study
Princeton, NJ 08540
odonnell@math.ias.edu

Rocco A. Servedio‡
Department of Computer Science
Columbia University
New York, NY 10027
rocco@cs.columbia.edu

Abstract

We give the first polynomial time algorithm to learn any function of a constant number of halfspaces under the uniform distribution on the Boolean hypercube to within any constant error parameter. We also give the first quasipolynomial time algorithm for learning any Boolean function of a polylog number of polynomial-weight halfspaces under any distribution on the Boolean hypercube. As special cases of these results we obtain algorithms for learning intersections and thresholds of halfspaces. Our uniform distribution learning algorithms involve a novel non-geometric approach to learning halfspaces; we use Fourier techniques together with a careful analysis of the noise sensitivity of functions of halfspaces. Our algorithms for learning under any distribution use techniques from real approximation theory to construct low-degree polynomial threshold functions. Finally, we also observe that any function of a constant number of polynomial-weight halfspaces can be learned in polynomial time in the model of exact learning from membership and equivalence queries.

*Supported in part by NSF grant CCR-97-01304. This work was performed while at the Department of Mathematics, MIT.

†Supported by NSF grant CCR-99-12342. This work was performed while at the Department of Mathematics, MIT.

‡Supported by an NSF Mathematical Sciences Postdoctoral Research Fellowship and by NSF grant CCR-98-77049. This work was performed while at the Division of Engineering and Applied Sciences, Harvard University.

1 Introduction

Let h be a hyperplane in $\mathbb{R}^n$, i.e. $h = \{x : \sum_{i=1}^n w_i x_i = \theta\}$. Such a hyperplane naturally induces a Boolean function $f(x) = \mathrm{sgn}(\sum_{i=1}^n w_i x_i - \theta)$ which is called a *linear threshold function* or simply a *halfspace*. Learning an unknown halfspace from labeled data is one of the oldest problems in machine learning, dating back to the 1950s [14, 45]. This problem has been intensively studied over the years, and as described in Section 2.3 efficient algorithms are now known for several different learning models.

While the problem of learning a single halfspace is fairly well understood, learning more complicated functions which depend on several halfspaces seems to be quite difficult; in particular learning an intersection of several unknown halfspaces stands as a central open problem in computational learning theory. Intersections of halfspaces are attractive for many reasons: any convex body can be expressed as an intersection of halfspaces, and several well-studied classes of Boolean functions such as DNF formulas can be naturally viewed as special cases of intersections of halfspaces over the Boolean cube. Finally, we hope that learning an intersection (AND) of halfspaces will be a first step towards learning richer and more expressive functions of halfspaces.

1.1 Previous Work

Given the apparent difficulty of learning intersections of halfspaces from random examples, several researchers have considered learning algorithms which are allowed to make *membership queries* for the value of the unknown function at points of the algorithm's choice. Building on work of Blum et. al. [17] and Baum [7], Kwek and Pitt [34] have given a membership query algorithm for learning the intersection of k halfspaces in $\mathbb{R}^n$ (with respect to any probability distribution) in time polynomial in n and k.

Progress has been much more limited for learning intersections of halfspaces from random examples only; all such results to date require that the examples be drawn from some restricted class of probability distributions. Baum [8] gave a polynomial time algorithm for learning an intersection of two origin-centered halfspaces under any "symmetric" distribution (which satisfies $\mathcal{D}(x) = \mathcal{D}(-x)$ for all $x \in \mathbb{R}^n$). His algorithm is essentially a reduction to the problem of learning a single halfspace. Building on work of Blum and Kannan [16], Vempala [51] gave a polynomial time algorithm which can learn an intersection of $\log n / \log \log n$ halfspaces under "non-concentrated" distributions on the Euclidean ball in $\mathbb{R}^n$. Vempala's algorithm uses random sampling to identify the subspace spanned by the normal vectors of the unknown halfspaces.

Finally, Auer *et al.* [6] gave an online algorithm to learn depth-two neural networks with constant fan-in at the hidden nodes, i.e. halfspaces in which each input variable is replaced by a halfspace which depends on a constant number d of variables. Their algorithm runs in time poly(n^d) but requires some restrictive assumptions on the top-level halfspace. We discuss the relation between our results and those of Auer *et al.* at the end of Section 4.1.

1.2 Our Results

We give new results for learning intersections of halfspaces, thresholds of halfspaces, and arbitrary functions of halfspaces over the Boolean cube. All of our algorithms learn from random examples only, and we obtain results for learning both from uniformly distributed examples and from examples drawn from an arbitrary probability distribution over the Boolean hypercube.

1.2.1 Uniform Distribution Learning

Our main learning result for the uniform distribution is a polynomial time algorithm for learning any function of any constant number of halfspaces to within any constant error parameter. More precisely, we prove:

Theorem 1 *Let $\mathcal{F}_k$ be the class of all Boolean functions $g : \{+1,-1\}^k \rightarrow \{+1,-1\}$, and let $\mathcal{H}_n$ be the class of all linear threshold functions (halfspaces) on $\{+1,-1\}^n$. The class of all functions $\{g(h_1(x),\ldots,h_k(x)) : g \in \mathcal{F}_k, h_i \in \mathcal{H}_n\}$ can be learned under the uniform distribution to accuracy ϵ in time $n^{O(k^2/\epsilon^2)}$, assuming $\epsilon < 1/k^2$.*

For $k = O(1)$ and $\epsilon = \Omega(1)$ this time bound is polynomial in n. We note that prior to our work no polynomial time algorithm was known which could learn even an intersection of two arbitrary halfspaces under the uniform distribution on $\{+1,-1\}^n$.

We can substantially improve the dependence on k for the special case of learning a read-once intersection or majority of halfspaces:

Theorem 2 *Let $h_1,\ldots,h_k$ be arbitrary halfspaces on $\{+1,-1\}^n$ which depend on disjoint sets of variables. The class $\{h_1(x) \wedge h_2(x) \wedge \cdots \wedge h_k(x)\}$ of read-once intersections of k halfspaces can be learned under the uniform distribution to accuracy ϵ in time $n^{O(\log(k)/\epsilon^2)}$, assuming $\epsilon < 1/\log k$.*

Theorem 3 *Let $h_1,\ldots,h_k$ be arbitrary halfspaces on $\{+1,-1\}^n$ which depend on disjoint sets of variables. The class $\{sgn(h_1(x)+h_2(x)+\cdots+h_k(x))\}$ of read-once majorities of k halfspaces can be learned under the uniform distribution to accuracy ϵ in time $n^{\tilde{O}(\log(k/\epsilon)/\epsilon^4)}$, assuming $\epsilon < 1/\log k$.*[1]

1.2.2 Learning under Arbitrary Distributions

Our algorithms for learning under arbitrary distributions use different techniques. In this scenario, our time bounds for learning depend chiefly on two parameters: the number of halfspaces k and the weight w of each halfspace (i.e. the magnitude of its integer coefficients). Our main learning result for arbitrary distributions is:

Theorem 4 *Let $\mathcal{F}_k$ be the class of all Boolean functions $g : \{+1,-1\}^k \rightarrow \{+1,-1\}$ and let $\mathcal{H}_n^w$ be the class of all weight-w halfspaces on $\{+1,-1\}^n$. The class of all functions $\{g(h_1(x),\ldots,h_k(x)) : g \in \mathcal{F}_k, h_i \in \mathcal{H}_n^w\}$ can be learned to accuracy ϵ under any distribution in time $n^{O(k^2 \log w)}/\epsilon$.*

Although this time bound can be exponentially large in general, we can learn any function of polylog(n) many halfspaces of poly(n) weight in quasipolynomial time under any probability distribution on the Boolean cube.

When we restrict attention to intersections of halfspaces, we can get better time bounds.

Theorem 5 *Let $h_1,\ldots,h_k$ be weight-w halfspaces on $\{+1,-1\}^n$. The class $\{h_1(x) \wedge h_2(x) \wedge \cdots \wedge h_k(x)\}$ of intersections of k weight-w halfspaces can be learned to accuracy ϵ under any distribution in time $n^{O(k \log k \log w)}/\epsilon$.*

This theorem does well for the intersection of a fairly small number of halfspaces of polynomial weight. For the intersection of a large number of halfspaces of very small weight, we obtain a different bound:

[1] Throughout the paper we write $\tilde{O}(\cdot)$ to hide factors which are polylogarithmic in the $\tilde{O}(\cdot)$'s argument.

Theorem 6 *Let $h_1, \ldots, h_k$ be weight-w halfspaces on $\{+1, -1\}^n$. The class $\{h_1(x) \wedge h_2(x) \wedge \cdots \wedge h_k(x)\}$ of intersections of k weight-w halfspaces can be learned to accuracy ϵ under any distribution in time $n^{O(\sqrt{w} \log k)}/\epsilon$.*

Theorems 5 and 6 exhibit a tradeoff between the number of halfspaces k and the weight w of the halfspaces; as discussed in Section 4 this tradeoff is essentially optimal given our techniques.

We can generalize the bound of Theorem 5 to thresholds of halfspaces:

Theorem 7 *Let $h_1, \ldots, h_\ell$ be weight-w halfspaces on $\{+1, -1\}^n$ and let g be a weight-k halfspace on $\{+1, -1\}^\ell$. The class $\{g(h_1(x), \ldots, h_\ell(x))\}$ of weight-k thresholds of weight-w halfspaces can be learned to accuracy ϵ under any distribution in time $n^{O(k \log k \log w)}/\epsilon$.*

Finally, we extend recent results of Klivans and Servedio [32] on learning DNF formulas (ORs of ANDs) to thresholds of ANDs:

Theorem 8 *Let $C_1, \ldots, C_\ell$ be arbitrary Boolean conjunctions over $\{+1, -1\}^n$ and let $g : \{+1, -1\}^\ell \rightarrow \{+1, -1\}$ be a weight-w halfspace. The class $\{g(C_1, \ldots, C_\ell)\}$ of weight-w thresholds of ANDs can be learned to accuracy ϵ under any distribution in time $n^{O(n^{1/3} \log w)}/\epsilon$.*

We thus achieve the same running time bound from [32] for learning DNF formulas while learning a strictly more expressive class of functions.

1.3 Learning in the Exact Model

We also give results for learning an intersection of k weight-w halfspaces in the model of *exact learning* from membership and equivalence queries [4]. Polynomial time algorithms for learning in this model are known to imply polynomial time algorithms in the PAC model augmented with membership queries [4].

We show that an intersection of a constant number of polynomial weight halfspaces can be learned in polynomial time:

Theorem 9 *Let $h_1, \ldots, h_k$ be weight-w halfspaces on $\{0, 1\}^n$. The class $\{h_1(x) \wedge h_2(x) \wedge \cdots \wedge h_k(x)\}$ of intersections of k weight-w halfspaces can be learned exactly from membership and equivalence queries in time poly(n, w^k).*

More generally, we give a polynomial time algorithm for learning any function of a constant number of halfspaces:

Theorem 10 *Let $\mathcal{F}_k$ be the class of all Boolean functions $g : \{0, 1\}^k \rightarrow \{+1, -1\}$ and let $\mathcal{H}_n^w$ be the class of all weight-w halfspaces on $\{0, 1\}^n$. The class of all functions $\{g(h_1(x), \ldots, h_k(x)) : g \in \mathcal{F}_k, h_i \in \mathcal{H}_n^w\}$ can be learned exactly from membership and equivalence queries in time poly$(n^{2^k}, w^{k \cdot 2^k})$.*

1.4 Our Approach

The techniques we use for learning under the uniform distribution are quite different from those we use for learning under an arbitrary distribution. In the uniform distribution case, we show tight concentration bounds for the Fourier spectra of functions of halfspaces; this lets us learn using a Fourier based sampling algorithm from Linial *et al.* [35]. In the arbitrary distribution case, we show how to represent functions of halfspaces as low-degree polynomial threshold functions, which lets us learn using linear programming.

1.4.1 Uniform Distribution Learning: Fourier Analysis and Noise Sensitivity

The centerpiece of our uniform-distribution learning algorithms is a new Fourier concentration bound for functions of halfspaces. Recall that a Fourier concentration bound for a class of functions C is a statement of the following form: For every function $f \in C$ on n inputs, all but an ϵ fraction of the Fourier spectrum of f is concentrated on degree up to $\alpha(\epsilon, n)$. Given such a bound, the "low-degree algorithm" of Linial *et al.* [35] provides a uniform-distribution learning algorithm for C running in time $n^{O(\alpha(\epsilon,n))}$. The main result in [35] is a Fourier concentration bound for the class of functions expressible by AC^0 circuits, with $\alpha(\epsilon, n) = \mathrm{polylog}(n/\epsilon)$. Our new concentration bound is $\alpha(\epsilon, n) = O(k\sqrt{\epsilon})$ for the class of arbitrary functions of k halfspaces. We also give tighter bounds for more restricted classes of functions of halfspaces.

Our technique for proving these concentration bounds is to study the *noise sensitivity* of halfspaces. The noise sensitivity of a function f is simply the probability that $f(x)$ differs from $f(y)$ where x is a randomly chosen point and y is a slight perturbation of x. The noise sensitivity of Boolean functions was studied extensively by Benjamini *et al.* [12]; they showed that functions with low noise sensitivity have good Fourier concentration bounds. We prove tight bounds on the noise sensitivity of halfspaces, and are thus able to get Fourier concentration bounds for various classes of functions of halfspaces.

1.4.2 Learning under Arbitrary Distributions: Polynomial Threshold Functions

Our results for learning under an arbitrary distribution begin with the fact that a polynomial threshold function of degree d can be learned in time $n^{O(d)}$. Our contribution is showing that various classes of functions of halfspaces can be expressed as polynomial threshold functions of low degree. This technique has previously been used for learning by Klivans and Servedio [32], who showed that any polynomial-size DNF formula can be represented as a polynomial threshold function of degree $\tilde{O}(n^{1/3})$. They thus obtained a learning algorithm for DNF which works under any distribution and runs in time $2^{\tilde{O}(n^{1/3})}$.

We show that any function of k halfspaces where the sum of the (integer) coefficients of each halfspace is bounded by w can be represented as a polynomial threshold function of degree $O(k^2 \log w)$. We prove this using rational function approximation tools which were first used in a complexity theory context by Beigel *et al.* [9]. Roughly, we use rational functions which approximate the function sgn to closely approximate the ±1 output values of our halfspaces. Having done this we obtain a single polynomial threshold function computing an arbitrary function g of halfspaces by composing these approximations with an interpolating polynomial for g.

In certain circumstances, we can trade off the dependences on k and w by using extremal polynomials in place of rational functions. By using Chebychev polynomials (previously used by Klivans and Servedio [32] and Nisan and Szegedy [41] in a somewhat similar context), we obtain a polynomial threshold function of degree $O(w^{1/2} \log k)$ computing the intersection of k halfspaces where the sum of the coefficients of each linear threshold function is at most w.

1.5 Learning in the Exact Model

For learning in the exact model, we observe that the functions to be learned are computed by polynomial size finite automata. We then simply use known algorithms for learning finite automata in polynomial time using membership and equivalence queries (see for example [4, 10]).

2 Preliminaries

2.1 Definitions and learning model

Throughout the paper unless otherwise indicated we represent TRUE and FALSE by -1 and $+1$ respectively. A *linear threshold function* or *halfspace* on $\{+1, -1\}^n$ is a function $f : \{+1, -1\}^n \rightarrow \{+1, -1\}$, $f(x) = \mathrm{sgn}(\sum_{i=1}^n w_i x_i - \theta)$ where $w_1, \dots, w_n, \theta \in \mathbb{R}$ and $\mathrm{sgn}(z) = +1$ iff $z \geqslant 0$. We say that $w_1, \dots, w_n, \theta$ *represent* f; since the domain is the discrete Boolean cube, every halfspace f has some representation in which $w_1, \dots, w_n, \theta \in \mathbb{Z}$. The *weight* of f is the smallest integer w for which there exist a representation $w_1, \dots, w_n, \theta \in \mathbb{Z}$ with $|\theta| + \sum_{i=1}^n |w_i| = w$. The *majority* function on inputs $x_1, \dots, x_n$ is $\mathrm{MAJ}(x) = \mathrm{sgn}(\sum_{i=1}^n x_i)$.

We use Valiant's well-studied *Probably Approximately Correct* (PAC) model of learning Boolean functions from random examples [50]. In this model a *concept class* C is a collection $\cup_{n \geqslant 1} C_n$ of Boolean functions where each $c \in C_n$ is a function on n bits. In the PAC model a learning algorithm has access to an *example oracle* $EX(c, \mathcal{D})$ which, when queried, provides a labeled example $\langle x, c(x) \rangle$ where x is drawn from distribution $\mathcal{D}$ over $\{+1, -1\}^n$ and $c \in C_n$ is the unknown target concept which the algorithm is trying to learn. Given Boolean functions c, h on $\{+1, -1\}^n$ we say that h is an *ϵ-approximator for c under $\mathcal{D}$* if $\Pr_{x \in \mathcal{D}}[h(x) = c(x)] \geqslant 1 - \epsilon$; the goal of a PAC learning algorithm is to generate an ϵ-approximator for the unknown target concept c. More precisely, an algorithm A is a *PAC learning algorithm for concept class C* if the following condition holds: for all $n \geqslant 1$, all $c \in C_n$, any distribution $\mathcal{D}$ on $\{+1, -1\}^n$, and any $0 < \epsilon < \frac{1}{2}, 0 < \delta < 1$, if A is given ϵ and δ as input and has access to $EX(c, \mathcal{D})$, then with probability at least $1 - \delta$ algorithm A outputs an ϵ-approximator for c under $\mathcal{D}$. We say that *A PAC learns C in time t* if A runs for at most t time steps and outputs a hypothesis h which can be evaluated on any point $x \in \{+1, -1\}^n$ in time t.

If the above condition holds only for the uniform distribution $\mathcal{U}$ on $\{+1, -1\}^n$, we say that A is a *uniform distribution PAC learning algorithm for $\mathcal{C}$*. Uniform distribution PAC learnability of Boolean functions has been studied by many authors; see, e.g., [11, 15, 19, 20, 21, 25, 26, 27, 28, 29, 30, 31, 35, 38, 46, 47, 48, 52, 53].

All of our learning algorithms, both for the uniform distribution and for arbitrary distributions, have running time bounds with a $\log(1/\delta)$ dependence on δ, and hence we typically omit mention of this dependence.

2.2 Fourier Analysis and Uniform Distribution Learning

We briefly review some facts about Fourier analysis on the Boolean cube and its relation to uniform distribution learning. For a detailed treatment with proofs see [37].

Let the space $\{+1, -1\}^n$ be endowed with the uniform probability measure, and define an inner product on functions $f, g : \{+1, -1\}^n \rightarrow \mathbb{R}$ by $\langle f, g \rangle = \mathbf{E}[fg]$. For $S \subseteq [n]$ the parity function $\chi_S : \{+1, -1\}^n \rightarrow \{+1, -1\}$ is defined by $\chi_S(x) = \prod_{i \in S} x_i$. The set of all 2^n parity functions $\{\chi_S\}_{S \subseteq [n]}$ forms an orthonormal basis for the vector space of real-valued functions on $\{+1, -1\}^n$, and hence every real-valued function $f : \{+1, -1\}^n \rightarrow \mathbb{R}$ can be uniquely expressed as a linear combination $f(x) = \sum_{S \subseteq [n]} \hat{f}(S) \chi_S(x)$. The coefficients $\hat{f}(S)$ are called the *Fourier coefficients* of f; collectively, they are its *Fourier spectrum.* Orthonormality implies that $\hat{f}(S) = \langle f, \chi_S \rangle$ and thus $\hat{f}(S)$ measures the correlation between f and χ_S. Orthonormality also implies *Parseval's identity,* which states that for every real-valued function f we have $E[f^2] = \sum_{S \subseteq [n]} \hat{f}(S)^2$. For Boolean functions we thus have $\sum_{S \subseteq [n]} \hat{f}(S)^2 = 1$.

Let $\alpha(\epsilon, n)$ be a function $\alpha : (0, \frac{1}{2}) \times \mathbb{N} \rightarrow \mathbb{N}$. We say that a concept class C has a *Fourier* *concentration bound* of $\alpha(\epsilon, n)$ if for all $n \geqslant 1$, all $0 < \epsilon < \frac{1}{2}$, and all $f \in C_n$ we have

$$\sum_{S \subseteq [n], |S| \geqslant \alpha(\epsilon, n)} \hat{f}(S)^2 \leqslant \epsilon,$$

i.e. at most an ϵ fraction of the Fourier spectrum weight resides in coefficients of degree more than $\alpha(\epsilon, n)$.

An important connection between Fourier analysis and learning theory is the following result, implicit in [35] (see [37] for a nice exposition and proof):

Fact 11 *Let C be a concept class with a Fourier concentration bound of $\alpha(\epsilon, n)$. Then there is a uniform distribution PAC learning algorithm for C which runs in time $n^{O(\alpha(\epsilon,n))}$.*

2.3 Polynomial Threshold Functions and Learning under Arbitrary Distributions

A *polynomial threshold function* is a function $f : \{+1, -1\}^n \to \{+1, -1\}$, $f(x) = \mathrm{sgn}(p(x))$ where $p(x_1, \ldots, x_n)$ is a real-valued multivariate polynomial. Equivalently, we say that the polynomial p *sign-represents* f. The *degree* of a polynomial threshold function is the degree of the polynomial p. Note that a linear threshold function is a polynomial threshold function of degree one.

The following fact is well known (see [18]):

Fact 12 *The concept class of linear threshold functions over $\{+1, -1\}^n$ can be PAC learned under any distribution in time poly$(n)/\epsilon$.*

The algorithm of Fact 12 is based on polynomial time linear programming. We will use the following easy extension of Fact 12 (proved in e.g. [32]):

Fact 13 *Let C be a concept class over $\{+1, -1\}^n$ such that each $f \in C_n$ can be expressed as a polynomial threshold function of degree at most $d(n)$. Then C can be PAC learned under any distribution in time $n^{O(d(n))}/\epsilon$.*

Remark 14 *By results of Maass and Turan [36] the class of linear threshold functions is also known to be learnable in polynomial time in the model of* exact learning from equivalence queries*. An analogue of Fact 13 holds in this model as well, and in fact all of our distribution-independent PAC learning results hold in this model as well.*

3 Learning Functions of Halfspaces under Uniform Distributions

3.1 From Noise Sensitivity to Fourier Concentration

Let $f : \{+1, -1\}^n \to \{+1, -1\}$ be a Boolean function. We define the *noise sensitivity* of f at ϵ to be:

$$\mathrm{ns}_\epsilon(f) = \Pr_{x, noise}[f(x) \neq f(N_\epsilon(x))]$$

(cf. [12]). Here x is uniformly chosen from $\{+1, -1\}^n$, and N_ϵ is the *noise operator* which flips each bit of its input independently with probability ϵ. The following formula from [42] relates the noise sensitivity of f to its Fourier spectrum; related formulas are given in [20, 12].

Proposition 15 *For any* $f : \{+1,-1\}^n \to \{+1,-1\}$*:*

$$\mathrm{ns}_\epsilon(f) = \frac{1}{2} - \frac{1}{2}\sum_{S\subseteq[n]} (1-2\epsilon)^{|S|}\hat{f}(S)^2.$$

It follows that a strong upper bound on the noise sensitivity of f implies a good Fourier concentration bound for f:

Proposition 16 *For any* $f : \{+1,-1\}^n \to \{+1,-1\}$*,* $0 < \gamma < 1/2$*:*

$$\sum_{|S|\geqslant 1/\gamma} \hat{f}(S)^2 < 2.32\,\mathrm{ns}_\gamma(f).$$

Proof: Starting from Proposition 15, we have:

$$\begin{aligned}
2\mathrm{ns}_\gamma(f) &= 1 - \sum_{S\subseteq[n]} (1-2\gamma)^{|S|}\hat{f}(S)^2 \\
&= \sum_{S\subseteq[n]} \hat{f}(S)^2 - \sum_{S\subseteq[n]} (1-2\gamma)^{|S|}\hat{f}(S)^2 \\
&= \sum_{S\subseteq[n]} [1-(1-2\gamma)^{|S|}]\hat{f}(S)^2 \\
&\geqslant \sum_{|S|\geqslant 1/\gamma} [1-(1-2\gamma)^{|S|}]\hat{f}(S)^2 \\
&\geqslant \sum_{|S|\geqslant 1/\gamma} [1-(1-2\gamma)^{1/\gamma}]\hat{f}(S)^2 \\
&> \sum_{|S|\geqslant 1/\gamma} (1-e^{-2})\hat{f}(S)^2,
\end{aligned}$$

and the result follows since $2/(1-e^{-2}) < 2.32$. ■

Corollary 17 *Let* $f : \{+1,-1\}^n \to \{+1,-1\}$ *be any Boolean function, and let* $\alpha : [0,1/2] \to [0,1]$ *be an increasing function such that* $\mathrm{ns}_\epsilon(f) \leqslant \alpha(\epsilon)$*. Then:*

$$\sum_{|S|\geqslant m} \hat{f}(S)^2 \leqslant \epsilon \qquad \textit{for} \qquad m = \frac{1}{\alpha^{-1}(\epsilon/2.32)}.$$

In Section 3.2 we first give a detailed analysis of the noise sensitivity of a single linear threshold function. We then extend this analysis to arbitrary functions of several halfspaces and thus obtain Fourier concentration bounds and learning results using Corollary 17 and Fact 11. In Section 3.5 we give noise sensitivity bounds (and thus obtain Fourier concentration bounds and learning results) for read-once intersections and majorities of halfspaces.

3.2 Noise Sensitivity of Halfspaces and Functions of Halfspaces

3.2.1 Noise Sensitivity of a Halfspace

Benjamini *et al.* [12] were the first to analyze the noise sensitivity of linear threshold functions. They proved that every halfspace $h : \{+1,-1\}^n \to \{+1,-1\}$ has $\mathrm{ns}_\epsilon(h) \leqslant C\epsilon^{1/4}$ where C is an absolute constant (note that this bound does not depend on n). Y. Peres subsequently improved this bound to $O(\sqrt{\epsilon})$ (unpublished, [43]). Since the majority function has noise sensitivity $\Omega(\sqrt{\epsilon})$ [12, 42], this is the best bound possible that depends only on ϵ.

We will use a refinement of Peres's bound that takes into account how "balanced" the halfspace is between outputs $+1$ and -1. To motivate our bound, let h be a halfspace with $p = \Pr[h(x) = +1] < \frac{1}{2}$. If x is an input to h and $y = N_\epsilon(x)$, then both x and y are uniformly distributed, so by a union bound $\Pr[h(x) = +1 \text{ or } h(y) = +1] \leqslant 2p$ and hence $\mathrm{ns}_\epsilon(h) \leqslant 2p$. Thus if $2p \ll \sqrt{\epsilon}$ it should be possible to get a much stronger bound upper bound on noise sensitivity which depends on p.

Our bound on the noise sensitivity of an arbitrary linear threshold function is as follows:

Theorem 18 *Let $h : \{+1,-1\}^n \to \{+1,-1\}$ be a halfspace, i.e. $h(x) = sgn(\sum_{i=1}^n w_i x_i - \theta)$ for some $w \in \mathbb{R}^n, \theta \in \mathbb{R}$. Let $p = \min\{\Pr[h = +1], \Pr[h = -1]\}$. Then:*

$$\mathrm{ns}_\epsilon(h) \leqslant \min\{2\,p, 20.5\,p\sqrt{\epsilon \ln(1/p)}\}.$$

Before proving Theorem 18 we discuss some of its implications. We devote Section 3.3 to a detailed proof and discussion of Theorem 18.

Since $p\sqrt{\ln(1/p)}$ is maximized when $p = e^{-1/2}$, and $20.5 \cdot e^{-1/2}\sqrt{\ln e^{1/2}} < 8.8$, we get:

Corollary 19 *For any halfspace $h : \{+1,-1\}^n \to \{+1,-1\}$, we have $\mathrm{ns}_\epsilon(h) \leqslant 8.8\sqrt{\epsilon}$.*

Since $(8.8 \cdot 2.32)^2 < 441$, Corollary 17 yields:

Theorem 20 *Let $f : \{+1,-1\}^n \to \{+1,-1\}$ be any linear threshold function. Then for all $0 < \epsilon < 1/2$,*

$$\sum_{|S| \geqslant (21/\epsilon)^2} \hat{f}(S)^2 \leqslant \epsilon.$$

3.2.2 Bounds on Functions of k Halfspaces

Corollary 19 immediately implies an upper bound on the noise sensitivity of any function of k halfspaces:

Theorem 21 *Let $f : \{+1,-1\}^n \to \{+1,-1\}$ be any function of k halfspaces. Then $\mathrm{ns}_\epsilon(f) \leqslant O(k\sqrt{\epsilon})$.*

Proof: Let $h_1, \ldots, h_k$ be arbitrary Boolean halfspaces, and let $g : \{+1,-1\}^k \to \{+1,-1\}$ be an arbitrary Boolean function. Write $f = g(h_1, \ldots, h_k)$. By Corollary 19, $\Pr[h_i(x) \neq h_i(N_\epsilon(x))]$ is $O(\sqrt{\epsilon})$ for each $i = 1 \ldots k$. By a union bound, the probability that at least one h_i's value changes under noise is at most $O(k\sqrt{\epsilon})$. But f's value changes only if at least one of the h_i's values changes. Hence $\Pr[f(x) \neq f(N_\epsilon(x))] \leqslant O(k\sqrt{\epsilon})$. ■

By applying Corollary 17, we get a Fourier concentration bound of $O(k^2/\epsilon^2)$ for the class of functions of k halfspaces (assuming $\epsilon < 1/k^2$). We now get our main uniform distribution learning result, Theorem 1, using Fact 11.

3.3 A Proof of Theorem 18

Recall the statement of Theorem 18:
Theorem 18: *Let* $h : \{+1,-1\}^n \to \{+1,-1\}$ *be a halfspace, i.e.* $h(x) = sgn(\sum_{i=1}^n w_i x_i - \theta)$ *for some* $w \in \mathbb{R}^n, \theta \in \mathbb{R}$. *Let* $p = \min\{\Pr[h = +1], \Pr[h = -1]\}$. *Then:*

$$\mathrm{ns}_\epsilon(h) \leqslant \min\{2\,p, 20.5\,p\sqrt{\epsilon \ln(1/p)}\}.$$

As we argued before, $\mathrm{ns}_\epsilon(h) \leqslant 2\,p$ always. Hence it suffices to show $\mathrm{ns}_\epsilon(h) \leqslant 20.5\,p\sqrt{\epsilon \ln(1/p)}$. Without loss of generality we assume that $\sum_{i=1}^n w_i x_i \neq \theta$ for all $x \in \{+1,-1\}^n$ (if not we can slightly perturb the weights without changing the Boolean function h.)

When the noise operator N_ϵ changes the sign of an input bit, we call this a *flip*. When all the flips taken together cause the value of h to change, we call this a *flop*. Let $P(k)$ be the probability of a flop, conditioned on exactly k flips occurring. We will upper bound $P(k)$ and then take an appropriate binomial average over k in the end.

So let us suppose that there are exactly $k > 0$ flips. Write $m = \lfloor n/k \rfloor$. Let $x \in \{+1,-1\}^n$ be chosen uniformly at random, and let π be a uniformly random permutation on $[n]$. Define $X_1 = \sum_{i=1}^k x_i w_{\pi(i)}$, $X_2 = \sum_{i=k+1}^{2k} x_i w_{\pi(i)}$, $\ldots$, $X_m = \sum_{i=(m-1)k+1}^{mk} x_i w_{\pi(i)}$, and finally $Z = \sum_{i=mk+1}^{n} x_i w_{\pi(i)}$. Write $S = \sum_{j=1}^m X_j + Z$. Because of the random permutation π, we can "assume that the weights of X_1 were flipped." In other words, $(S, S - 2X_1)$ has exactly the same joint distribution as $(w \cdot y, w \cdot y')$, where $y \in \{+1,-1\}^n$ is uniform and y' is y with exactly k randomly selected bits flipped.

Put $S' = S - X_1 = \sum_{j=2}^m X_j + Z$, so the "sum before flipping" is $S' + X_1$ and the "sum after flipping" is $S' - X_1$. Hence a flop occurs iff $|S' - \theta| < |X_1|$. (Note that $|S' - \theta| = |X_1|$ is impossible, by our first assumption.)

Suppose that we condition on there being *no* flop; i.e., we condition on the event $|S' - \theta| > |X_1|$. Then since $\mathrm{sgn}(X_1)$ is independent from both $|X_1|$ and S', we have that $\Pr[\mathrm{sgn}(X_1) = \mathrm{sgn}(S' - \theta)] = \Pr[\mathrm{sgn}(X_1) \neq \mathrm{sgn}(S' - \theta)] = 1/2$. But, since we are conditioning on the event $|S' - \theta| > |X_1|$, we have that $\mathrm{sgn}(S' - \theta)$ and $\mathrm{sgn}(S - \theta)$ are always the same. Therefore we may conclude that under this conditioning, $\Pr[\mathrm{sgn}(S - \theta) \neq \mathrm{sgn}(X_1)] = 1/2$; i.e.,

$$\Pr[\mathrm{sgn}(S - \theta) \neq \mathrm{sgn}(X_1) \text{ \& no flop}] = \frac{1}{2}\Pr[\text{no flop}].$$

Yet the event $[\mathrm{sgn}(S - \theta) \neq \mathrm{sgn}(X_1) \text{ \& no flop}]$ is exactly the same event as $[\mathrm{sgn}(S - \theta) \neq \mathrm{sgn}(X_1)]$, since if there *is* a flop, then $\mathrm{sgn}(S - \theta)$ must be the same as $\mathrm{sgn}(X_1)$. Hence we have $\frac{1}{2}\Pr[\text{no flop}] = \Pr[\mathrm{sgn}(S - \theta) \neq \mathrm{sgn}(X_1)]$; i.e.,

$$\frac{1}{2}(1 - P(k)) = \Pr[\mathrm{sgn}(S - \theta) \neq \mathrm{sgn}(X_1)].$$

Now note that we could have derived this statement with X_2 in place of X_1, or indeed any of $X_2, \ldots X_m$ in place of X_1, simply because once we apply the random permutation π, we could have picked any of these blocks to "be the flips." So in fact,

$$\forall i = 1 \ldots m,\ \ \frac{1}{2}(1 - P(k)) = \Pr[\mathrm{sgn}(S - \theta) \neq \mathrm{sgn}(X_i)].$$

Write τ for the random variable $\mathrm{sgn}(S - \theta)$, and σ_i for the random variable $\mathrm{sgn}(X_i)$. Converting probabilities to expectations of indicator variables, we have:

$$\forall i = 1 \ldots m,\ \ \frac{1}{2}(1 - P(k)) = \mathbf{E}[\mathbf{1}_{\tau \neq \sigma_i}].$$

Summing these equations over all i yields:

$$\frac{m}{2}(1-P(k)) \quad = \quad \mathbf{E}[\sum_{i=1}^{m} \mathbf{1}_{\tau\neq\sigma_i}] \tag{1}$$

Recall that this expectation is over the choices of x and π, which force the values of τ and σ_i.

The above arguments are due to Peres, and indeed from this point it is fairly easy to obtain an $O(\sqrt{\epsilon})$ upper bound. Some more work is required to obtain our desired bound which depends on p.

Suppose without loss of generality that $p = \Pr[h = -1]$, so $p = \Pr[\tau = -1]$. For $t = 1 \dots m$, define:

$$p_t = \Pr[\tau = -1 \mid \text{exactly } t \text{ of the } \sigma_i\text{'s are } +1]$$

Since the event $S < \theta$ is negatively correlated with the events $X_i > 0$, we conclude that:

$$1 \geqslant p_0 \geqslant p_1 \geqslant p_2 \geqslant \cdots \geqslant p_m \geqslant 0 \tag{2}$$

But also note that $\sigma_1, \dots, \sigma_m$ are all independent and uniformly distributed in $\{+1, -1\}$. Therefore:

$$p = \sum_{t=0}^{m} \frac{\binom{m}{t}}{2^m} p_t \tag{3}$$

Continuing from (1), we have:

$$\begin{aligned}
\frac{m}{2}(1-P(k)) &= \mathbf{E}[\#\text{ of } \sigma_i\text{'s differing from } \tau] \\
&= \sum_{t=0}^{m} \frac{\binom{m}{t}}{2^m} \mathbf{E}[\sum_{i=1}^{m} \mathbf{1}_{\tau\neq\sigma_i} \mid \text{exactly } t\ \sigma_i\text{'s are } +1] \\
&= \sum_{t=0}^{m} \frac{\binom{m}{t}}{2^m} \mathbf{E}[t\mathbf{1}_{\tau=-1} + (m-t)\mathbf{1}_{\tau=+1} \mid \text{exactly } t\ \sigma_i\text{'s are } +1] \\
&= \sum_{t=0}^{m} \frac{\binom{m}{t}}{2^m} (t\Pr[\tau=-1|\text{exactly } t\ \sigma_i\text{'s are } +1] + (m-t)\Pr[\tau=+1|\text{exactly } t\ \sigma_i\text{'s are } +1]) \\
&= \sum_{t=0}^{m} \frac{\binom{m}{t}}{2^m} (tp_t + (m-t)(1-p_t)).
\end{aligned}$$

So $\frac{m}{2}(1-P(k)) = \mathbf{E}_T[Tp_T + (m-T)(1-p_T)]$, where $T \sim \text{Binomial}(m, 1/2)$. Some arithmetic gives:

$$\begin{aligned}
P(k) &= \frac{2}{m}\mathbf{E}_T[T] - 1 + 2\mathbf{E}_T[p_T] - \frac{4}{m}\mathbf{E}_T[Tp_T] \\
&= 2\left(p - \frac{2}{m}\sum_{t=0}^{m} \frac{\binom{m}{t}}{2^m} tp_t\right)
\end{aligned} \tag{4}$$

We will obtain an upper bound for $P(k)$ by maximizing (4) subject to (2) and (3). This is a linear programming problem. Hence the maximum occurs at a vertex, which in this case means the maximum occurs when, for an appropriate $1 \leqslant b \leqslant m/2$, we have $p_0 = p_1 = \cdots = p_{b-1} = 1$, $p_{b+1} = p_{b+2} = \cdots = p_m = 0$, and p_b is such that (3) is tight. (We have $b \leqslant m/2$ since $p \leqslant 1/2$.)

Henceforth we let the p_t's take on these values which maximize (4) and we will reason about the value of (4).

Our goal is now to show that

$$(4) \leqslant \frac{(14.48)\, p\sqrt{\ln(1/p)}}{\sqrt{m}}. \tag{5}$$

An easy case occurs if $p \leqslant 2^{-m}$. In this case $p_t = 0$ for all $t > 0$ and hence $(4) = 2p$. But $p \leqslant 2^{-m}$ implies that $\frac{(14.48)\, p\sqrt{\ln(1/p)}}{\sqrt{m}} \leqslant (14.48)\, p$, so (5) is proved.

Now assume $p > 2^{-m}$. We claim that it suffices to show (5) in the case that $p_b = 1$; i.e., the case that $p = \sum_{t=0}^{b} \binom{m}{t} 2^{-m}$ for some $b > 0$. For suppose that (5) holds in this case; then given any $p^* > 2^{-m}$ and associated b and p_t^*'s, we may write $p^* = (1 - p_b)p^- + p_b p^+$, where $p^- = \sum_{t=0}^{b-1} \binom{m}{t} 2^{-m}$ and $p^+ = \sum_{t=0}^{b} \binom{m}{t} 2^{-m}$. One can easily check that the value of (4) associated to p^* is $(1 - p_b)(4)^- + p_b(4)^+$, where $(4)^-$ denotes the value of (4) for p^-, and similarly for $(4)^+$. But by assumption (5) holds for p^- and p^+; since $p\sqrt{\ln(1/p)}$ is a concave function of p, we have that (5) holds for p^0 as well.

So we may assume that $p = \sum_{t=0}^{b} \binom{m}{t} 2^{-m}$, i.e., $p = \Pr[T \leqslant b]$, where as before $T \sim \text{Binomial}(m, 1/2)$. Now we can rewrite (4) as:

$$2\left(p - \frac{2}{m} p\mathbf{E}[T \mid T \leqslant b]\right) = \frac{4p}{m}\mathbf{E}\left[\frac{m}{2} - T \mid T \leqslant b\right].$$

Thus showing (5) amounts to showing that

$$\mathbf{E}\left[\frac{m}{2} - T \mid T \leqslant b\right] \leqslant ((14.48)/4)\sqrt{\ln(1/p)}\sqrt{m}, \tag{6}$$

where $p = \Pr[T \leqslant b]$. Equation (6) follows from the following lemma (whose proof we defer until later):

Lemma 22 *Let $n \geqslant 1$, let $X \sim$ Binomial$(n, 1/2)$, let $0 \leqslant b \leqslant n/2$, and let $p = \Pr[X \leqslant b]$. Then:*

$$\mathbf{E}\left[\frac{n}{2} - X \mid X \leqslant b\right] < 3.62\,\sqrt{\ln(1/p)}\sqrt{n}.$$

We have therefore shown that $P(k) \leqslant (14.48)\, p\sqrt{\ln(1/p)}\sqrt{m}$ for any $k > 0$.

It remains only to take the appropriate binomial average over k to obtain our final bound on $\mathrm{ns}_\epsilon(h)$. Since $m = \lfloor n/k \rfloor$, we have

$$\begin{aligned}
\Pr[\text{flop}] &= \mathbf{E}_k[P(k)] \\
&\leqslant (14.48)\, p\sqrt{\ln(1/p)}\mathbf{E}[\sqrt{1/\lfloor n/k \rfloor}] \\
&\leqslant (14.48)\, p\sqrt{\ln(1/p)}\sqrt{\mathbf{E}[1/\lfloor n/k \rfloor]} \\
&\leqslant (14.48)\, p\sqrt{\ln(1/p)}\sqrt{(2/n)\mathbf{E}[k]} \\
&= (14.48)\sqrt{2}p\sqrt{\epsilon \ln(1/p)} \\
&< 20.5\, p\sqrt{\epsilon \ln(1/p)}
\end{aligned}$$

and Theorem 18 is proved. ■

Proof of Lemma 22: We will make use of the Chernoff bound $\Pr\left[X \leqslant \frac{n}{2} - \delta\frac{n}{2}\right] < \exp(-n\delta^2/4)$ which holds for $0 \leqslant \delta \leqslant 1$. This immediately yields

$$b \geqslant \frac{n}{2} - \sqrt{\ln(1/p)}\sqrt{n}. \tag{7}$$

The Chernoff bound also tells us that $\Pr\left[X < \frac{n}{2} - 2\sqrt{\ln(1/p)}\sqrt{n}\right] < p^4$. Hence:

$$\Pr\left[X < \frac{n}{2} - 2\sqrt{\ln(1/p)}\sqrt{n} \mid X \leqslant \frac{n}{2}\right] < 2\,p^4.$$

We write d for $2\sqrt{\ln(1/p)}\sqrt{n}$ and α for $2\,p^4$, so we have $\Pr\left[X < \frac{n}{2} - d \mid X \leqslant \frac{n}{2}\right] < \alpha$. It follows from the log-concavity of the binomial distribution[2] that

$$\Pr\left[X < u - d \mid X \leqslant u\right] < \alpha$$

holds for every $u \leqslant n/2$. In particular this holds for $u = b, u = b - d, u = b - 2\,d, u = b - 3\,d, \ldots,$ whence:

$$\begin{aligned}
\mathbf{E}\left[b - X \mid X \leqslant b\right] &\leqslant d + \alpha(2\,d) + \alpha^2(3\,d) + \alpha^3(4\,d) + \cdots \\
&< \frac{d}{(1-\alpha)^2} \\
&= \frac{2\sqrt{\ln(1/p)}\sqrt{n}}{(1-2\,p^4)^2} \\
&\leqslant \frac{2\sqrt{\ln(1/p)}\sqrt{n}}{(1-2\,(1/2)^4)^2} \qquad (8) \\
&< 2.62\sqrt{\ln(1/p)}\sqrt{n}. \qquad (9)
\end{aligned}$$

So:

$$\begin{aligned}
\mathbf{E}\left[\frac{n}{2} - X \mid X \leqslant b\right] &= \frac{n}{2} - b + \mathbf{E}\left[b - X \mid X \leqslant b\right] \\
&< 3.62\sqrt{\ln(1/p)}\sqrt{n},
\end{aligned}$$

by (7) and (9). (Lemma 22) ■

3.4 Tightness of the Fourier Concentration Bound for Halfspaces

In this section we prove the following theorem which shows that our Fourier concentration bound for halfspaces (Theorem 20) is tight up to a constant factor:

Theorem 23 *For any $0 < \epsilon < 1/2$, the following inequality holds for all sufficiently large odd n:*

$$\sum_{\substack{S \subseteq [n] \\ |S| > \alpha}} \widehat{\mathrm{MAJ}_n}(S)^2 > \epsilon,$$

whenever $\alpha < \frac{8}{\pi^3\epsilon^2} - 2$. For sufficiently small ϵ, $\frac{1}{4\epsilon^2} < \frac{8}{\pi^3\epsilon^2} - 2$, and so we may use this simpler bound.

[2] In particular, log-concavity implies the "new-is-better-than-used" property of mathematical reliability, which is exactly what we use here. See, e.g., [2].

The following proposition will be useful:

Proposition 24 *Let d be an odd positive integer. Then:*

$$\lim_{\substack{n\to\infty \\ n \text{ odd}}} \sum_{\substack{S\subseteq[n] \\ |S|=d}} \widehat{\mathrm{MAJ}_n}(S)^2 = \frac{2}{\pi d}\frac{\binom{d-1}{(d-1)/2}}{2^{d-1}}.$$

Proof: We will use Bernasconi's formula for the Fourier coefficients of MAJ_n, which appears in [13]. Write $M = \mathrm{MAJ}_n$, where n is odd; then for all $S \subseteq [n]$ with $|S| = d$,

$$\widehat{M}(S) = \begin{cases} (-1)^{\frac{d-1}{2}} \frac{2}{2^n} \frac{\binom{d-1}{\frac{d-1}{2}}\binom{n-d}{\frac{n-d}{2}}}{\binom{\frac{n-1}{2}}{\frac{d-1}{2}}} & \text{if } d \text{ is odd,} \\ 0 & \text{if } d \text{ is even.} \end{cases}$$

It follows that for odd d:

$$\sum_{\substack{S\subseteq[n] \\ |S|=d}} \widehat{M}(S)^2 = \frac{4}{2^{2n}}\binom{n}{d}\frac{\binom{d-1}{\frac{d-1}{2}}^2\binom{n-d}{\frac{n-d}{2}}^2}{\binom{\frac{n-1}{2}}{\frac{d-1}{2}}^2}. \tag{10}$$

We write $f(n) \sim g(n)$ when $f(n)/g(n) \to 1$ as $n \to \infty$.

The factor $\binom{n-d}{\frac{n-d}{2}}^2$ is easy to handle; by Stirling's formula,

$$\binom{n-d}{\frac{n-d}{2}} \sim \sqrt{2/\pi}\,(n-d)^{-1/2}\,2^{n-d}. \tag{11}$$

Now we consider the two other binomial factors in (10) involving n; i.e., the ratio $\binom{n}{d}/\binom{\frac{n-1}{2}}{\frac{d-1}{2}}^2$. Using the fact that $\binom{m}{k} = \frac{m^k}{k!}(1-1/m)(1-2/m)\cdots(1-(k+1)/m)$, we get:

$$\begin{aligned} \frac{\binom{n}{d}}{\binom{\frac{n-1}{2}}{\frac{d-1}{2}}^2} &= \frac{n^d\left(\frac{d-1}{2}\right)!^2}{d!\left(\frac{n-1}{2}\right)^{d-1}} \cdot \frac{(1-1/n)(1-2/n)\cdots(1-(d-1)/n)}{(1-2/(n-1))^2(1-4/(n-1))^2\cdots(1-(d-3)/(n-1))^2} \\ &\sim \frac{n^d\left(\frac{d-1}{2}\right)!^2}{d!\left(\frac{n-1}{2}\right)^{d-1}} \qquad (12) \\ &= n2^{d-1}\frac{\left(\frac{d-1}{2}\right)!^2}{d!}\left(1+\frac{1}{n-1}\right)^{d-1} \\ &\sim n2^{d-1}\frac{\left(\frac{d-1}{2}\right)!^2}{d!}, \qquad (13) \end{aligned}$$

where (12) and (13) hold because d is fixed.

Plugging (11) and (13) into (10), we get:

$$\begin{aligned} \sum_{\substack{S\subseteq[n] \\ |S|=d}} \widehat{M}(S)^2 &\sim \frac{4}{2^{2n}}n2^{d-1}\frac{\left(\frac{d-1}{2}\right)!^2}{d!}\binom{d-1}{\frac{d-1}{2}}^2\frac{2}{\pi}\frac{1}{n-d}\,2^{2(n-d)} \\ &= \frac{4}{\pi}\frac{n}{n-d}2^{-d}\frac{\left(\frac{d-1}{2}\right)!^2}{d!}\frac{(d-1)!^2}{\left(\frac{d-1}{2}\right)!^4} \\ &\sim \frac{2}{\pi d}\frac{\binom{d-1}{(d-1)/2}}{2^{d-1}}. \end{aligned}$$

■

We now note that $\frac{2}{\pi d}\frac{\binom{d-1}{(d-1)/2}}{2^{d-1}}$ is exactly $\frac{2}{\pi}$ times the coefficient on x^d in the Taylor expansion of $\arcsin x$ [1]. Indeed, since the Taylor expansion of $\arcsin x$ has zero coefficients on even powers, we in fact have:

Corollary 25 *Let d be any positive integer. Then:*

$$\lim_{\substack{n\to\infty \\ n \text{ odd}}} \sum_{\substack{S\subseteq[n] \\ |S|=d}} \widehat{\mathrm{MAJ}_n}(S)^2 = [x^d]\left(\frac{2}{\pi}\arcsin x\right),$$

where the right hand side denotes the coefficient on x^d in the Taylor expansion of $\frac{2}{\pi}\arcsin x$.

It is interesting to compare this with the way in which arcsin arises in [42].

Now we can prove Theorem 23.

Proof of Theorem 23: Given ϵ, let $\alpha = \alpha(\epsilon)$ be a fixed odd integer to be specified later, depending only on ϵ. By Proposition 25 we know that for each $0 < d \leqslant \alpha$,

$$\lim_{\substack{n\to\infty \\ n \text{ odd}}} \sum_{\substack{S\subseteq[n] \\ |S|=d}} \widehat{\mathrm{MAJ}_n}(S)^2 = [x^d]\left(\frac{2}{\pi}\arcsin x\right).$$

Since we are only concerned about this fact for a finite number of d's, independent of n, for any $\delta > 0$ we have that for sufficiently large n,

$$\sum_{\substack{S\subseteq[n] \\ |S|=d}} \widehat{\mathrm{MAJ}_n}(S)^2 \leqslant (1+\delta)\left([x^d]\left(\frac{2}{\pi}\arcsin x\right)\right)$$

for each $0 < d \leqslant \alpha$. So let n be any odd number which is sufficiently large, and write $M = \mathrm{MAJ}_n$. Then:

$$\sum_{|S|\leqslant\alpha} \widehat{M}(S)^2 \leqslant (1+\delta)\sum_{d=1}^{\alpha}\left([x^d]\left(\frac{2}{\pi}\arcsin x\right)\right).$$

(Since M is balanced, $\widehat{M}(\varnothing) = 0$.)

Using $\frac{2}{\pi}\arcsin 1 = 1$, we conclude that $\sum_{d=1}^{\infty}\left([x^d]\left(\frac{2}{\pi}\arcsin x\right)\right) = 1$. Now for all odd d,

$$\frac{2}{\pi d}\frac{\binom{d-1}{(d-1)/2}}{2^{d-1}} > \left(\frac{2}{\pi d}\right)^{3/2},$$

by Stirling's formula. Hence:

$$\begin{aligned}
\sum_{d=1}^{\alpha}\left([x^d]\left(\frac{2}{\pi}\arcsin x\right)\right) &= 1-\sum_{d>\alpha}\left([x^d]\left(\frac{2}{\pi}\arcsin x\right)\right)\\
&= 1-\sum_{\substack{d>\alpha\\ d \text{ odd}}}\frac{2}{\pi d}\frac{\binom{d-1}{(d-1)/2}}{2^{d-1}}\\
&< 1-\sum_{\substack{d>\alpha\\ d \text{ odd}}}\left(\frac{2}{\pi d}\right)^{3/2}\\
&\leqslant 1-\frac{1}{2}\int_{\alpha+2}^{\infty}\left(\frac{2}{\pi d}\right)^{3/2}\mathrm{d}d\\
&= 1-\frac{1}{2}\left[\left(\frac{2}{\pi}\right)^{3/2}\frac{2}{d^{1/2}}\right]_{d=\alpha+2}\\
&= 1-\left(\frac{2}{\pi}\right)^{3/2}\frac{1}{\sqrt{\alpha+2}},
\end{aligned}$$

so,

$$\sum_{|S|\leqslant\alpha}\widehat{M}(S)^2 \leqslant (1+\delta)\left(1-\left(\frac{2}{\pi}\right)^{3/2}\frac{1}{\sqrt{\alpha+2}}\right).$$

To make this quantity smaller than $1-\epsilon$, it suffices to take $\alpha < \frac{8}{\pi^3\epsilon^2}-2$ and δ sufficiently small. ■

3.5 Extensions: Learning Read-Once Functions of Halfspaces

Theorem 21 bounds $\mathrm{ns}_\epsilon(f)$ by $O(k\sqrt{\epsilon})$ for f an arbitrary function of k halfspaces. Can stronger bounds be obtained if f is restricted to be a simple function (such as intersection) of k halfspaces?

While we do not know the answer to this question, we can give substantially improved bounds if f is a *read-once* intersection or majority of halfspaces, i.e. each variable x_i occurs as input to at most one halfspace.

For read-once intersections of halfspaces we have the following noise sensitivity bound:

Theorem 26 *Let $f:\{+1,-1\}^n\to\{+1,-1\}$, $f(x)=h_1(x)\wedge h_2(x)\wedge\cdots\wedge h_k(x)$ where $h_1,\ldots,h_k$ are halfspaces on disjoint sets of variables. Then $\mathrm{ns}_\epsilon(f)\leqslant 2c\sqrt{\epsilon\log k}$, where $c=20.5$ is the universal constant from Theorem 18.*

Proof: By Corollary 19, $\mathrm{ns}_\epsilon(f)\leqslant 8.54k\sqrt{\epsilon}$. For $k\leqslant 6$ this is smaller than $2c\sqrt{\epsilon\ln k}$, so we assume $k\geqslant 7$.

Since we will be dealing with logical AND we write T and F for the outputs of the linear threshold functions $h_1,\ldots,h_k$. Write $p_i=\Pr[h_i=T]$ and $\eta_i=\mathrm{ns}_\epsilon(h_i)$. Note that for every Boolean function g, $\mathrm{ns}_\epsilon(g)=2\Pr[g(x)=T]\Pr[g(N_\epsilon(x))=F\mid g(x)=T]=2\Pr[g(x)=F]\Pr[g(N_\epsilon(x))=$

$T \mid g(x) = F]$, since x and $N_\epsilon(x)$ are both uniformly distributed. Hence in our case, we have:

$$\begin{aligned}
\frac{1}{2}\mathrm{ns}_\epsilon(h) &= \Pr[f(x)=T]\ \Pr[f(N_\epsilon(x))=F \mid f(x)=T] \\
&= \left(\prod_{i=1}^{k}\Pr[h_i(x)=T]\right)\left(1-\prod_{i=1}^{k}\Pr[h_i(N_\epsilon(x))=T \mid h_i(x)=T]\right) \\
&= \left(\prod_{i=1}^{k} p_i\right)\left(1-\prod_{i=1}^{k}(1-\eta_i/2p_i)\right) \\
&\leqslant \left(\prod_{i=1}^{k} p_i\right)\left(\sum_{i=1}^{k}\frac{\eta_i}{2p_i}\right), \qquad (14)
\end{aligned}$$

where we have used the fact that $\prod_{i=1}^{k}(1-x_i) \geqslant 1-\sum_{i=1}^{k} x_i$ for any $x_1,\ldots,x_k \in [0,1]$ (note that $\frac{\eta_i}{2p_i} \leqslant 1$ since $\eta_i \leqslant 2p_i$ as we've seen before).

We would now like to break up (14) into parts depending on the various values of the p_i's. For $1 \leqslant s \leqslant t \leqslant k$, let us write:

$$U_{s\ldots t} = \left(\prod_{i=s}^{t} p_i\right)\left(\sum_{i=s}^{t}\frac{\eta_i}{2p_i}\right).$$

Our goal is to show $U_{1\ldots k} \leqslant c\sqrt{\epsilon \ln k}$. We begin by claiming that it suffices to assume all of the p_i's are at least $\frac{2}{3}$. For suppose that, say, $p_1 < \frac{2}{3}$. Then:

$$U_{1\ldots k} = p_1 U_{2\ldots k} + \left(\prod_{i=1}^{k} p_i\right)\left(\frac{\eta_1}{2p_1}\right) \leqslant \frac{2}{3}U_{2\ldots k} + \frac{1}{2}\eta_1 \leqslant \frac{2}{3}U_{2\ldots k} + \frac{9}{2}\sqrt{\epsilon},$$

where the last step uses Corollary 19. Hence if we can show $U_{2\ldots k} \leqslant c\sqrt{\epsilon \ln k}$, then $U_{1\ldots k} \leqslant \frac{2}{3}c\sqrt{\epsilon\ln k} + \frac{9}{2}\sqrt{\epsilon} \leqslant c\sqrt{\epsilon \ln k}$, since $\frac{1}{3}c\sqrt{\ln k} \geqslant \frac{1}{3}c\sqrt{\ln 7} \geqslant \frac{9}{2}$. We can repeat this argument for any $p_i < \frac{2}{3}$; thus in showing $U_{1\ldots k} \leqslant C\ (\epsilon \ln k)^{\frac{1}{2}}$ it suffices to assume $p_i \geqslant \frac{2}{3}$ for all i.

Without loss of generality, we may reorder indices so that $p_1,\ldots,p_\ell < 1-\frac{1}{10k}$ and $p_{\ell+1},\ldots,p_k \geqslant 1-\frac{1}{10k}$ for some $0 \leqslant \ell \leqslant k$. Using the fact that $p_i \leqslant 1$ for all i, it is easy to see that $U_{1\ldots k} \leqslant U_{1\ldots \ell} + U_{\ell+1\ldots k}$. We now upper-bound each of these terms individually.

$$\begin{aligned}
U_{1\ldots\ell} &= \left(\prod_{i=1}^{\ell} p_i\right)\left(\sum_{i=1}^{\ell}\frac{\eta_i}{2p_i}\right) \\
&\leqslant \left(\prod_{i=1}^{\ell} p_i\right)\left(\sum_{i=1}^{\ell} c\sqrt{\epsilon}\,\frac{1-p_i}{p_i}\sqrt{\ln\frac{1}{1-p_i}}\right) \qquad \text{(Theorem 18)} \\
&\leqslant \tfrac{3}{2}c\sqrt{\epsilon \ln 10k}\left(\prod_{i=1}^{\ell} p_i\right)\left(\sum_{i=1}^{\ell}(1-p_i)\right) \qquad \text{(using } \tfrac{2}{3} \leqslant p_i \leqslant 1-\tfrac{1}{10k}\text{)} \\
&\leqslant \tfrac{3}{2}c\sqrt{\epsilon \ln 10k}\ \mu^\ell \ell(1-\mu) \quad \text{(where } \mu = \sqrt[\ell]{p_1\cdots p_\ell}\text{, by the AM-GM inequality)} \\
&\leqslant \tfrac{3}{2}c\sqrt{\epsilon \ln 10k}\ e^{-1} \qquad \text{(by elementary calculus, maximizing over } \mu \in [0,1]\text{)}.
\end{aligned}$$

$$\begin{aligned}
U_{\ell+1\ldots k} &\leqslant \left(\prod_{i=\ell+1}^{k} p_i\right)\left(\sum_{i=\ell+1}^{k} c\ \sqrt{\epsilon}\ \frac{1-p_i}{p_i}\sqrt{\ln\frac{1}{1-p_i}}\right) \qquad \text{(as before)} \\
&\leqslant (k-\ell)c\ \sqrt{\epsilon}\ \frac{\frac{1}{10k}}{1-\frac{1}{10k}}\sqrt{\ln 10k} \qquad \text{(since } \tfrac{1-p}{p}\sqrt{\ln\tfrac{1}{1-p}} \text{ is decreasing on } [1-\tfrac{1}{10k},1]\text{)} \\
&\leqslant (10-\tfrac{1}{7})^{-1}c\ \sqrt{\epsilon \ln 10k} \qquad \text{(using } k-\ell \leqslant k,\ k \geqslant 7\text{)}
\end{aligned}$$

Thus $U_{1\ldots k} \leqslant (\frac{3}{2e} + (10 - \frac{1}{7})^{-1})c\sqrt{\epsilon \ln 10k}$ and this is less than $c\sqrt{\epsilon \ln k}$ for $k \geqslant 7$, as desired. ■

Note that $k \leqslant n$ for any read-once intersection of halfspaces. By using Corollary 17 and Fact 11, we get Theorem 2.

In earlier work Hancock *et al.* [23] gave a polynomial time algorithm for learning a read-once intersection of majorities (i.e. each weight w_i of each halfspace is 1) under the uniform distribution. We emphasize that we are learning a much richer class of functions since our threshold functions can have arbitrary (even exponentially large) weights.

For the more expressive class of read-once majority of halfspaces we have the following noise sensitivity bound:

Theorem 27 *Let $f(x) = sgn(h_1(x) + \cdots + h_k(x) - \theta)$ where $h_1, \ldots, h_k$ are halfspaces on disjoint sets of variables. Then* $\mathrm{ns}_\epsilon(f) = \tilde{O}((\epsilon \log \frac{k}{\epsilon})^{1/4})$.

The proof of Theorem 27 is somewhat technical, so for the sake of exposition we leave it in Appendix A. We get Theorem 3 by applying Corollary 19 and Fact 11.

4 Learning Intersections of Halfspaces under Arbitrary Distributions

For our distribution independent results we use a set of completely different techniques. For our first set of results we use rational function approximations to the sgn function. For our second set of results, we use the extremal properties of the Chebyshev polynomials.

Throughout this section for convenience we let +1 denote TRUE and −1 denote FALSE.

4.1 Bounding PTF degree via rational function approximation

We use the rational function approximation to the sgn function introduced by Beigel *et al.* [9], building on work of Newman [40]. The motivation of Beigel *et al.* was the study of the complexity class PP. Later, Siu *et al.* [49] used the techniques for proving lower bounds against small-weight threshold circuits. The main technical theorem we need is:

Theorem 28 *[9] For every $\ell, t \geqslant 1$ there exists a rational function P_t^ℓ in one variable with the following properties:*

- *For any $x \in [1, 2^t]$ the value $P_t^\ell(x) \in [1, 1 + 1/\ell]$.*
- *For any $x \in [-2^t, -1]$ the value $P_t^\ell(x) \in [-1 - 1/\ell, -1]$.*
- *$P_t^\ell(x)$ has degree $O(t \log \ell)$.*

Here the degree of a rational function is the sum of the degrees of its numerator and denominator. Using this tool, we can get a polynomial threshold function for the intersection of k halfspaces:

Theorem 29 *Let $h_1, \ldots, h_k$ be linear threshold functions $\{+1, -1\}^n \to \{+1, -1\}$ each of weight at most w. The function $h_1(x) \wedge \cdots \wedge h_k(x)$ can be expressed as a polynomial threshold function of degree $O(k \log k \log w)$.*

Proof: For $i = 1, \ldots, k$ let $H_i(x) = \sum_{j=1}^n w^i_j x_j - \theta_i$ be a minimum weight representation of $h_i(x)$, so w^i_j, θ_j are integers and $h_i(x) = \operatorname{sgn}(H_i(x))$. Without loss of generality we may suppose that $|H_i(x)| \geqslant 1$ for all $x \in \{+1, -1\}^n$, since this can be achieved by doubling each w^i_j and replacing θ_i by $2\theta_i - 1$.

Let $\ell = 2k$ and $t = \log w$ and consider the sum of rational functions

$$I = P^{2k}_{\log w}(H_1(x)) + \cdots + P^{2k}_{\log w}(H_k(x)).$$

Now on input x, if $h_i(x) = 1$ (TRUE) for every i, then $I \geqslant k$. On the other hand, if for some i we have $h_i(x) = -1$ (FALSE), then $I \leqslant k - 1$. Thus $sgn(I - k)$ computes $h_1(x) \wedge \cdots \wedge h_k(x)$ on all inputs. Now $I - k$ is the sum of k rational functions each of degree $O(\log k \log w)$. Hence if we bring these to a common denominator, we can write $I - k$ as a single rational function $Q(x)/R(x)$ of degree $O(k \log k \log w)$, and $Q(x)/R(x)$ sign-represents $h_1(x) \wedge \cdots \wedge h_k(x)$. Hence so does $Q(x)R(x)$, which is a polynomial threshold function of degree $O(k \log k \log w)$. ■

Using this theorem and Fact 13, we get Theorem 5. We note that for $k = w^2$ the $O(k \log k \log w)$ bound of Theorem 29 is nearly optimal; Minsky and Papert have shown that the "one-in-a-box" function requires polynomial threshold function degree at least k, and the one-in-a-box function can be expressed as an AND of k halfspaces of weight k^2 (in fact the halfspaces are ORs of fan-in k^2).

By a virtually identical proof we can generalize Theorem 29 as follows:

Theorem 30 *Let $h_1, \ldots, h_\ell$ be linear threshold functions $\{+1, -1\}^n \to \{+1, -1\}$ each of weight at most w. Let $f : \{+1, -1\}^\ell \to \{+1, -1\}$ be a weight k threshold function. Then $f(h_1, \ldots, h_\ell)$ can be expressed as a polynomial threshold function of degree $O(k \log k \log w)$.*

Again, combining this with Fact 13 yields Theorem 7. We note that the degree bound given by Theorem 30 is nearly optimal for $k = w$. Hajnal *et al.* [24] show how to compute the parity function on k variables as a weight-$O(k)$ threshold of weight-$O(k)$ halfspaces, but it is well known that any polynomial threshold function computing parity on k variables requires degree at least k (see [5, 39]).

These rational function techniques can also be used to give learning algorithms for arbitrary functions of halfspaces; we get Theorem 4 from:

Theorem 31 *Let $h_1, \ldots, h_k$ be linear threshold functions $\{+1, -1\}^n \to \{+1, -1\}$ each of weight at most w. Let $f : \{+1, -1\}^k \to \{+1, -1\}$ be any function. Then $f(h_1, \ldots, h_k)$ can be expressed as a polynomial threshold function of degree $O(k^2 \log w)$.*

Proof: Let $H_i(x)$ be as in the proof of Theorem 29 and let $g_i = P^\ell_{\log w}(H_i(x))$, where $\ell = 2^{3k}$. Note that g_i is a rational function of degree $O(k \log w)$, and for any input x we have $g_i(x) \in [\operatorname{sgn}(h_i(x)) - 2^{-3k}, \operatorname{sgn}(h_i(x)) + 2^{-3k}]$. Since f is a Boolean function on k inputs, it is expressible as a multilinear polynomial $\tilde{f}$ of degree k, with coefficients of the form $i/2^k$ where i is an integer in $[-2^k, 2^k]$. (The polynomial $\tilde{f}$ is just the Fourier representation of f.) Multiply $\tilde{f}$ by 2^k, so now $\tilde{f} : \{+1, -1\}^k \to \{+2^k, -2^k\}$, and $\tilde{f}$ has integer coefficients which are at most 2^k in absolute value.

Now we would like to argue that $\tilde{f}(g_1, \ldots, g_k)$ sign-represents $f(h_1, \ldots, h_k)$. To do this, we simply show that the fact that $g_i = +1 \pm 2^{-3k}$ or -1 ± 2^{-3k} rather than $+1$ or -1 does not incur too much error. As a polynomial, $\tilde{f}$ has at most 2^k terms. Each term is the product of an integer coefficient of magnitude at most 2^k and up to k g_i's. We get at most $O(k2^{-3k})$ error on the product of the g_i's, and hence at most $O(k2^{-2k})$ error on any given term. Since we are adding up at most

2^k terms, we get at most $O(k2^{-k})$ error overall, which is much less than the error we could tolerate (namely 2^k). Hence $\tilde{f}(g_1, \ldots, g_k)$ sign-represents $f(h_1, \ldots, h_k)$.

Now $\tilde{f}$ is a multilinear polynomial of degree k, and each g_i is a rational function of degree $O(k \log w)$. Consider now $\tilde{f}(g_1, \ldots, g_k)$; we can bring it to a common denominator (namely, the product of the denominators of the g_i's) with degree only $O(k^2 \log w)$. Hence we have a single rational function of degree $O(k^2 \log w)$ which sign-represents our function, and we can convert it to a polynomial threshold function as in the proof of Theorem 29. ■

Remark. As mentioned earlier, Auer *et al.* [6] gave an n^d time learning algorithm to learn depth-two neural networks with fan-in d at the bottom level, i.e. functions of the form $f(h_1, \ldots, h_N)$ where f is a halfspace and each h_i is a halfspaces which depends on d variables. Since every function of d variables has an interpolating polynomial of degree at most d, from our polynomial threshold function vantage point it is immediately clear that every function $f(g_1, \ldots, g_N)$ (where g_i is an arbitrary function of d variables not necessarily a halfspace) has a polynomial threshold function of degree d, and thus can be learned in time n^d.

4.2 Bounding PTF Degree via Chebychev Polynomials

4.2.1 Learning Intersections of Small-Weight Halfspaces

The degree bounds obtained in the previous section are most interesting for cases when the number of underlying halfspaces is small (say a constant) and each halfspace is of weight $n^{O(1)}$. In this section we give an alternate construction of polynomial threshold functions based on Chebychev polynomials and obtain improved degree bounds for cases involving a polynomial number of small weight (e.g. weight $O(1)$) halfspaces:

Theorem 32 *Let $h_1, \ldots, h_k$ be linear threshold functions $\{+1, -1\}^n \to \{+1, -1\}$ each of weight at most w. The function $h_1(x) \wedge \cdots \wedge h_k(x)$ can be expressed as a polynomial threshold function of degree $O(\sqrt{w} \log k)$.*

The main tools we use to prove Theorem 32 are the Chebyshev polynomials of the first kind. The d-th Chebyshev polynomial of the first kind, $T_d(x)$, is a univariate degree-d polynomial with the following properties [22]:

Lemma 33 *The polynomial $T_d(x)$ satisfies*

- $|T_d(x)| \leqslant 1$ *for* $|x| \leqslant 1$ *with* $T_d(1) = 1$;
- $T'_d(x) \geqslant d^2$ *for* $x > 1$ *with* $T'_d(1) = d^2$.

Proof of Theorem 32: The proof is a generalization of the proof of Theorem 1 in [32]. Let $f = h_1 \wedge h_2 \wedge \cdots \wedge h_k$ be an intersection of k halfspaces, each of weight at most w. Since 1 represents TRUE we have $f(x) = 1$ iff each $h_i(x) = 1$. Let

$$p(y) = T_d\left(y\left(1 + \frac{1}{w}\right)\right)$$

where $d = \lceil \sqrt{w} \rceil$. Lemma 33 implies that for $y \geqslant 1$ we have $p(y) \geqslant 2$ but $|p(y)| \leqslant 1$ for $y \in [-1 + \frac{1}{w}, 1 - \frac{1}{w}]$.

Let $w_i \leqslant w$ denote the weight of the i-th halfspace h_i, so $h_i(x) = \operatorname{sgn}(w^i \cdot x - \theta_i)$ where $|\theta_i| + \sum_{j=1}^n |w^i_j| = w_i$ and w^i_j, θ_i are all integers. (Without loss of generality we may assume that each w_i and w are at least 2.) Let

$$Q_i(x) = p\left(1 - \frac{w^i \cdot x - \theta_i + 1}{w_i}\right).$$

Consider the polynomial threshold function $\operatorname{sgn}(P(x) + k + \frac{1}{2})$ where

$$P(x) = -\sum_{i=1}^{k} Q_i(x)^{\lceil \log 2k \rceil}.$$

Since p is a polynomial of degree $d = \lceil \sqrt{w} \rceil$ and $w^i \cdot x$ is a polynomial of degree 1, this polynomial threshold function has degree $\lceil \sqrt{w} \rceil \lceil \log 2k \rceil$. We now show that this polynomial threshold function computes f exactly on $\{+1, -1\}^n$.

Fix any element $x \in \{+1, -1\}^n$.

- If $f(x) = 1$ then for each i we have $0 \leqslant w^i \cdot x - \theta_i \leqslant w_i$ and hence $-1 + \frac{1}{w} \leqslant \frac{-1}{w_i} \leqslant 1 - \frac{w^i \cdot x - \theta_i + 1}{w_i} \leqslant 1 - \frac{1}{w}$. Consequently for each i we have $|Q_i(x)| \leqslant 1$ and hence $-Q_i(x)^{\lceil \log 2k \rceil} \geqslant -1$ so $P(x) \geqslant -k$.
- If $f(x) = 0$ then for some i we have $w^i \cdot x - \theta_i \leqslant -1$ and hence $1 - \frac{w^i \cdot x - \theta_i + 1}{w_i} \geqslant 1$. Consequently $Q_i(x) \geqslant 2$ and hence $-Q_i(x)^{\lceil \log 2k \rceil} \leqslant -2k$. Since $-Q_i(x)^{\lceil \log 2k \rceil} \leqslant 1$ for all i, we have $P(x) \leqslant -2k + (k-1) = -k - 1$. □

4.2.2 Learning a Threshold of ANDs

In the last section we used Chebychev polynomials to obtain polynomial threshold function degree bounds for ANDs of thresholds; we now use them to obtain bounds for thresholds of ANDs. The following theorem generalizes the bound of Klivans and Servedio (Their Theorem 1) for OR of ANDs [32]:

Theorem 34 *Let $C_1, \ldots, C_\ell$ be Boolean conjunctions, each over at most k literals, and let g be a weight-w threshold over $\{+1, -1\}^\ell$. The function $g(C_1, \ldots, C_\ell)$ can be expressed as a polynomial threshold function of degree $O(\sqrt{k} \log w)$.*

Proof: The proof is a slight modification of Theorem 32. Since literals take value ±1, it is easily seen that each conjunction $C_i(x)$ over $k_i \leqslant k$ literals can be expressed as a threshold $\operatorname{sgn}(w^i \cdot x - \theta_i)$ with the following properties: if $C_i(x) = 1$ (TRUE) then $w^i \cdot x - \theta_i = 1$, while if $C_i = -1$ (FALSE) then $-(2k_i - 1) \leqslant w^i \cdot x - \theta_i \leqslant -1$. Let $p(y) = T_d(y(1 + \frac{1}{k}))$ where $d = \lceil \sqrt{k} \rceil$ and let

$$Q_i(x) = p\left(1 - \frac{1 - (w^i x - \theta)}{2k_i}\right).$$

If $C_i(x) = 1$ then $Q_i(x) = p(1) = T_d(1 + \frac{1}{k}) \geqslant 2$ while if $C_i(x) = -1$ then $|Q_i(x)| \leqslant 1$. Consequently, if we take

$$\tilde{Q}_i(x) = 2\left(\frac{Q_i(x)}{p(1)}\right)^{\log 4w} - 1,$$

we have $\tilde{Q}_i(x) = 1$ if $C_i(x) = 1$ and $\tilde{Q}_i(x) \in [-1 - \frac{1}{2w}, -1 + \frac{1}{2w}]$ if $C_i(x) = -1$. Thus the error of each $\tilde{Q}_i$ is at most $\frac{1}{2w}$, so we may substitute the polynomials $\tilde{Q}_i(x)$ for the inputs of the weight-w threshold function g to obtain a polynomial threshold function which correctly computes $g(C_1, \ldots, C_\ell)$. ■

Using Theorem 34 in place of Klivans and Servedio's Theorem 1, the arguments of [32] can be used to prove the following strengthened version of Klivans and Servedio's main structural result for DNF formulas (the proofs are unchanged so we do not repeat them):

Theorem 35 *Let $f : \{+1, -1\}^n \to \{+1, -1\}$ be a weight-w threshold of ANDs. Then f can be a expressed as a polynomial threshold function of degree $O(n^{1/3} \log w)$.*

Applying Fact 13 we obtain Theorem 8 which is a strict generalization of the main learning result of [32].

5 Learning in the Exact Model

The model of exact learning from membership and equivalence queries was introduced by Angluin [4]. In a *membership query*, the learning algorithm submits an input $x \in \{0, 1\}^n$ to the membership oracle and receives the value $f(x)$ in response. In an *equivalence query*, the learning algorithm submits a representation of a hypothesis $h : \{0, 1\}^n \to \{0, 1\}$ to the equivalence oracle. If this hypothesis is not logically equivalent to f then the oracle returns a point $x \in \{0, 1\}^n$ such that $h(x) \neq f(x)$. The learner is required to output a hypothesis which is logically equivalent to f. For more details on this model, see [4]. Known reductions [4] imply that if a concept class is exactly learnable in time t, then there is a PAC plus membership query algorithm for the concept class which learns to accuracy ϵ in time $\text{poly}(t, 1/\epsilon)$.

We show that the intersection (or indeed any Boolean function) of a constant number of polynomial-weight halfspaces can be learned exactly in polynomial time. These results contrast with Theorems 4 and 5 which state that the intersection (or any Boolean function) of a constant number of polynomial-weight halfspaces can be learned without membership queries in *quasipolynomial* time.

5.1 Reduction to Automata

Our approach is to show that an intersection of low weight halfspaces can be computed by a relatively small automaton. A similar approach is used by Kushilevitz [33] to give a simple proof that $O(\log n)$ term DNF are exactly learnable in polynomial time.

Lemma 36 *Let $h_1, \ldots, h_k$ be weight-w halfspaces on $\{0, 1\}^n$ and let $f(x) = h_1(x) \wedge h_2(x) \wedge \cdots \wedge h_k(x)$. There is a finite automaton of size poly(n, w^k) which accepts only those strings of length n such that $f(x) = 1$.*

Proof: Consider first a single halfspace $h(x) = \text{sgn}(v_1 x_1 + \cdots + v_n x_n - \theta)$ of weight w. We describe an automaton A_h that accepts only those strings of length n such that $h(x) = 1$. The states of A_h are denoted by pairs $[u, i]$ where $u \in \{-w, -w + 1, \ldots, w - 1, w\}$ and $i \in \{0, 1, \ldots, n\}$. The initial state is $[0, 0]$. The automata is constructed in such a way that string $x_1 \ldots x_i \in \{0, 1\}^i$ will reach the state $[u, i]$ where $u = v_1 x_1 + \cdots + v_i x_i$. (This is easily achieved: the state $[u, i - 1]$ has a 0-transition to $[u, i]$ and a 1-transition to $[u + v_i, i]$.) The set of accepting states are the states $[u, n]$ where $u \geqslant \theta$.

The automaton for an intersection of k weight-w halfspaces is constructed similarly. It has states $[u_1, \ldots, u_k, i]$ where each $u_j \in \{-w, w\}$ and $i \in \{0, \ldots, n\}$. The state $[u_1, \ldots, u_k, i - 1]$ has a

0-transition to $[u_1, \ldots, u_k, i]$ and a 1-transition to $[u_1 + v_i^1, \ldots, u_k + v_i^k, i]$ where v_i^j is the coefficient of x_i in halfspace h_j. The set of accepting states are the states $[u_1, \ldots, u_k, n]$ where each u_j is at least as large as the corresponding threshold θ_j. ■

We can prove a similar lemma for the case of an arbitrary function of halfspaces:

Lemma 37 *Let $h_1, \ldots, h_k$ be weight-w halfspaces on $\{0,1\}^n$, let g be any Boolean function mapping $\{0,1\}^k$ to $\{0,1\}$, and let $f(x) = g(h_1(x), \ldots, h_k(x))$. There is an automaton of size poly$(n^{2^k}, w^{k \cdot 2^k})$ which accepts only those strings of length n such that $f(x) = 1$.*

Recall that any Boolean function on k variables can be computed by a DNF, i.e. by a 2^k-way OR of k-way ANDs. Lemma 37 follows from Lemma 36 and the fact that the size of an automaton computing the union of automata A_1 and A_2 is at most the product of the sizes of A_1 and A_2.

Now we can apply the following result originally due to Angluin [3] (see [10] for extensions and improvements):

Theorem 38 *[3] Let A be an automaton of size t. Then A can be learned in the exact model from membership and equivalence queries in time polynomial in t and in the length of the longest counterexample.*

It is easily seen that the longest counterexample for our automata will be of length n. Theorems 9 and 10 now follow directly from Lemma 36, Lemma 37 and Theorem 38.

References

[1] M. Abramowitz and I. Stegun. *Handbook of Mathematical Functions.* Dover, 1972.

[2] M. Y. An. Log-concave probability distributions: Theory and statistical testing. Technical Report Economics Working Paper Archive at WUSTL, Washington University at St. Louis, 1995.

[3] D. Angluin. Learning regular sets from queries and counterexamples. *Information and Computation*, 75(2):87–106, 1987.

[4] D. Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, 1988.

[5] J. Aspnes, R. Beigel, M. Furst, and S. Rudich. The expressive power of voting polynomials. *Combinatorica*, 14(2):1–14, 1994.

[6] Peter Auer, Stephen Kwek, Wolfgang Maass, and Manfred K. Warmuth. Learning of depth two neural networks with constant fan-in at the hidden nodes (extended abstract). In *Proceedings of the Ninth Annual Conference on Computational Learning Theory*, pages 333–343. ACM Press, 1996.

[7] E. Baum. Neural net algorithms that learn in polynomial time from examples and queries. *IEEE Trans. on Neural Networks*, 2:5–19, 1991.

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

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

[10] A. Beimel, F. Bergadano, N. Bshouty, E. Kushilevitz, and S. Varricchio. On the applications of multiplicity automata in learning. In *Proceedings of the Thirty-Seventh Annual Symposium on Foundations of Computer Science*, pages 349–358, 1996.

[11] M. Bellare. A technique for upper bounding the spectral norm with applications to learning. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 62–70, 1992.

[12] I. Benjamini, G. Kalai, and O. Schramm. Noise sensitivity of Boolean functions and applications to percolation. *Inst. Hautes Études Sci. Publ. Math.*, 90:5–43, 1999.

[13] A. Bernasconi. *Mathematical techniques for the analysis of Boolean functions.* PhD thesis, Institute for Computational Mathematics, Pisa, 1998.

[14] H. Block. The perceptron: a model for brain functioning. *Reviews of Modern Physics*, 34:123–135, 1962.

[15] A. Blum, M. Furst, J. Jackson, M. Kearns, Y. Mansour, and S. Rudich. Weakly learning DNF and characterizing statistical query learning using Fourier analysis. In *Proceedings of the Twenty-Sixth Annual Symposium on Theory of Computing*, pages 253–262, 1994.

[16] 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.

[17] Avrim Blum, Prasad Chalasani, Sally A. Goldman, and Donna K. Slonim. Learning with unreliable boundary queries. *Journal of Computer and System Sciences*, 56(2):209–222, 1998.

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

[19] D. Boneh and R. Lipton. Amplification of weak learning over the uniform distribution. In *Proceedings of the Sixth Annual Workshop on Computational Learning Theory*, pages 347–351, 1993.

[20] N. Bshouty, J. Jackson, and C. Tamon. More efficient PAC learning of DNF with membership queries under the uniform distribution. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory*, pages 286–295, 1999.

[21] N. Bshouty and C. Tamon. On the Fourier spectrum of monotone functions. *Journal of the ACM*, 43(4):747–770, 1996.

[22] E. Cheney. *Introduction to approximation theory.* McGraw-Hill, New York, New York, 1966.

[23] M. Golea, M. Marchand, and T. Hancock. On learning μ-perceptron networks on the uniform distribution. *Neural Networks*, 9:67–82, 1994.

[24] A. Hajnal, W. Maass, P. Pudlak, M. Szegedy, and G. Turan. Threshold circuits of bounded depth. *Journal of Computer and System Sciences*, 46:129–154, 1993.

[25] T. Hancock. *The complexity of learning formulas and decision trees that have restricted reads.* PhD thesis, Harvard University, 1992.

[26] T. Hancock and Y. Mansour. Learning monotone k-μ DNF formulas on product distributions. In *Proceedings of the Fourth Annual Conference on Computational Learning Theory*, pages 179–193, 1991.

[27] J. Jackson. An efficient membership-query algorithm for learning DNF with respect to the uniform distribution. *Journal of Computer and System Sciences*, 55:414–440, 1997.

[28] J. Jackson, A. Klivans, and R. Servedio. Learnability beyond AC^0. In *Proceedings of the 34th ACM Symposium on Theory of Computing*, 2002.

[29] R. Khardon. On using the Fourier transform to learn disjoint DNF. *Information Processing Letters*, 49:219–222, 1994.

[30] M. Kharitonov. Cryptographic hardness of distribution-specific learning. In *Proceedings of the Twenty-Fifth Annual Symposium on Theory of Computing*, pages 372–381, 1993.

[31] A. Klivans and R. Servedio. Boosting and hard-core sets. In *Proceedings of the Fortieth Annual Symposium on Foundations of Computer Science*, pages 624–633, 1999.

[32] 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.

[33] E. Kushilevitz. A simple algorithm for learning $O(\log n)$-term DNF. *Information Processing Letters*, 61(6):289–292, 1997.

[34] S. Kwek and L. Pitt. PAC learning intersections of halfspaces with membership queries. *Algorithmica*, 22(1/2):53–75, 1998.

[35] N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, Fourier transform and learnability. *Journal of the ACM*, 40(3):607–620, 1993.

[36] W. Maass and G. Turan. *How fast can a threshold gate learn?*, pages 381–414. MIT Press, 1994.

[37] Y. Mansour. *Learning Boolean functions via the Fourier transform*, pages 391–424. 1994.

[38] Y. Mansour. An $O(n^{\log \log n})$ learning algorithm for DNF under the uniform distribution. *Journal of Computer and System Sciences*, 50:543–550, 1995.

[39] M. Minsky and S. Papert. *Perceptrons: an introduction to computational geometry.* MIT Press, Cambridge, MA, 1968.

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

[41] N. Nisan and M. Szegedy. On the degree of Boolean functions as real polynomials. In *Proceedings of the Twenty-Fourth Annual Symposium on Theory of Computing*, pages 462–467, 1992.

[42] R. O'Donnell. Hardness amplification within NP. In *Proceedings of the 34th ACM Symposium on Theory of Computing*, 2002.

[43] Y. Peres. personal communication, 2001.

[44] V. V. Petrov. *Limit theorems of probability theory.* Oxford Science Publications, Oxford, England, 1995.

[45] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–407, 1958.

[46] Y. Sakai and A. Maruoka. Learning monotone log-term DNF formulas under the uniform distribution. *Theory of Computing Systems*, 33:17–33, 2000.

[47] R. Servedio. On PAC learning using Winnow, Perceptron, and a Perceptron-like algorithm. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory*, pages 296–307, 1999.

[48] R. Servedio. On learning monotone DNF under product distributions. In *Proceedings of the Fourteenth Annual Conference on Computational Learning Theory*, pages 473–489, 2001.

[49] K.-Y. Siu, V.P. Roychowdhury, and T. Kailath. Rational approximation techniques for analysis of neural networks. *IEEE Transactions on Information Theory*, 40(2):445–474, 1994.

[50] L. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[51] 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.

[52] K. Verbeurgt. Learning DNF under the uniform distribution in quasi-polynomial time. In *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pages 314–326, 1990.

[53] K. Verbeurgt. Learning sub-classes of monotone DNF on the uniform distribution. In *Proceedings of the Ninth Conference on Algorithmic Learning Theory*, pages 385–399, 1998.

A Noise Sensitivity for Read-once Majority of Thresholds

In this section we prove Theorem 27:

Theorem 27 *Let $f(x) = sgn(h_1(x) + \cdots + h_k(x) - \theta)$ where $h_1, \ldots, h_k$ are linear threshold functions on disjoint sets of variables. Then* $\mathrm{ns}_\epsilon(f) = \tilde{O}((\epsilon \log \frac{k}{\epsilon})^{1/4})$.

Proof: Let $p_i = \Pr[h_i(x) = -1]$ and $q_i = \min\{p_i, 1 - p_i\}$. Let ϵ_i denote $\mathrm{ns}_\epsilon(h_i)$. Let $c = 20.5$ denote the universal constant from Theorem 18.

We begin by reducing to the case in which each q_i is not too small, and each $\epsilon_i \leqslant q_i$. Let $s_i \in \{+1, -1\}$ be the less likely value for $h_i(x)$, so s_i is such that $\Pr[h_i(x) = s_i] = q_i$. Let:

$$\alpha = \sqrt{c}\,(\epsilon \ln k)^{\frac{1}{4}},$$

and let $S_\alpha = \{i \in [k] \colon q_i \leqslant \alpha/k\}$. Recalling that both x and $N_\epsilon(x)$ are both uniformly distributed, a union bound tells us that the probability there exists an $i \in S_\alpha$ such that $h(x) = s_i$ or $h(N_\epsilon(x)) = s_i$ is at most $\sum_{i \in S_\alpha} 2q_i \leqslant \sum_{i \in S_\alpha} 2\alpha/k \leqslant 2\alpha$. Since an additive $\alpha = O(1)(\epsilon \log k)^{\frac{1}{4}}$ does not affect the bound we are trying to prove, we may assume without loss of generality that $h_i(x) = h_i(N_\epsilon(x)) = -s_i$ for all $i \in S_\alpha$. In this case these halfspaces are irrelevant to the noise sensitivity calculation, and we can therefore assume without loss of generality that $S_\alpha = \varnothing$; i.e., that $q_i > \alpha/k$ for all i.

Next, note that each $\epsilon_i \leqslant c\ q_i\sqrt{\epsilon \ln \frac{1}{q_i}}$ by Theorem 18. We claim that we may assume $c\ q_i\sqrt{\epsilon \ln \frac{1}{q_i}} \leqslant q_i$ and hence $\epsilon_i \leqslant q_i$. The claimed inequality holds if and only if $q_i \geqslant \exp\left(-\frac{1}{c^2\epsilon}\right)$; since each $q_i > \alpha/k = \sqrt{c}\ \epsilon^{\frac{1}{4}}\left(\frac{\ln^{\frac{1}{4}} k}{k}\right)$, it suffices to establish $\sqrt{c}\ \epsilon^{\frac{1}{4}}\frac{\ln^{\frac{1}{4}} k}{k} \geqslant \exp\left(-\frac{1}{c^2\epsilon}\right)$. It is easy to check that this holds so long as $k < \exp(\frac{1}{c^2\epsilon})$ because $\frac{\ln^{\frac{1}{4}} k}{k}$ decreases for $k \geqslant 2$. But this holds without loss of generality, for otherwise $\epsilon \ln k \geqslant \frac{1}{c^2} = \Omega(1)$ and the theorem is trivially true.

To summarize, we may assume without loss of generality:

$$q_i \geqslant \alpha/k, \qquad\qquad q_i \geqslant \epsilon_i. \tag{15}$$

Let F_i denote the random variable $h_i(N_\epsilon(x)) - h_i(x)$. Since $\Pr[h_i(x) = +1, h_i(N_\epsilon(x)) = -1] = \Pr[h_i(x) = -1, h_i(N_\epsilon(x)) = +1] = \epsilon_i/2$, we have that $F_i = \pm 2$ with probability $\epsilon_i/2$ each, and $F_i = 0$ with probability $1 - \epsilon_i$. Let H denote $\sum_{i=1}^k h_i(x)$ and let F denote $\sum_{i=1}^k F_i$; thus F denotes the amount by which $h_1(x) + \cdots + h_k(x)$ and $h_1(N_\epsilon(x)) + \cdots + h_k(N_\epsilon(x))$ differ. The proof of the present theorem centers around the observation that if $f(x) \neq f(N_\epsilon(x))$ then we must have $|F| \geqslant |H - \theta|$. Hence:

$$\mathrm{ns}_\epsilon(f) \leqslant \Pr[|F| \geqslant |H - \theta|],$$

and we shall proceed by upper-bounding the probability that $|F| \geqslant |H - \theta|$.

While the random variables H and F are clearly not independent, each is a sum of independent Bernoulli random variables, and this allows us to bring to bear a number of standard estimates. Let us write σ_H^2 and σ_F^2 for the variances of H and F respectively. To upper bound the probability that H is close to the threshold θ we shall use the following result, whose proof we defer to the end of this section:

Lemma 39 *Let $H_1, \ldots, H_k$ be independent ± 1-valued random variables, let $H = \sum_{i=1}^k H_i$, and let σ_H^2 denote Var$[H]$. Then for every $\theta \in \mathbb{R}$ and every $\lambda \geqslant 1$ we have:*

$$\Pr[|H - \theta| \leqslant \lambda] \leqslant O(1)\frac{\lambda}{\sigma_H}.$$

To bound the size of $|F|$ in terms of H's deviation from θ, we will use an estimate for σ_F/σ_H. We claim:

$$\sigma_F/\sigma_H \leqslant O(1)\left(\epsilon \log \frac{k}{\alpha}\right)^{\frac{1}{4}}. \tag{16}$$

To see this, let us compute:

$$\sigma_F^2 = \sum_{i=1}^k \mathrm{Var}[F_i] = 4\sum_{i=1}^k \epsilon_i \leqslant 8C\sum_{i=1}^k q_i\left(\epsilon \ln \frac{1}{q_i}\right)^{\frac{1}{2}} \leqslant O(1)\left(\epsilon \log \frac{k}{\alpha}\right)^{\frac{1}{2}} \sum_{i=1}^k q_i,$$

where the first inequality follows from Theorem 18, and the second because $q_i \geqslant \alpha/k$ using (15). As for the variance of H:

$$\sigma_H^2 = \sum_{i=1}^k \mathrm{Var}[H_i(x)] = \sum_{i=1}^k (1 - (1 - 2p_i)^2) = \sum_{i=1}^k 4p_i(1 - p_i) = 4\sum_{i=1}^k 4q_i(1 - q_i) \geqslant 2\sum_{i=1}^k q_i,$$

where the inequality uses $q_i \leqslant \frac{1}{2}$. The inequality claimed in (16) now follows.

The proof now splits into two cases, depending on whether $\sigma_F \geqslant 1$ or $\sigma_F < 1$.

Case 1: $\mathbf{\sigma_F^2 \geqslant 1}$. In this case, we shall use a simple tail bound to show that F is unlikely to exceed a moderate quantity times σ_F in magnitude. Recall that F is the sum of the independent random variables F_i, and each one has mean zero and satisfies $|F_i| \leqslant 2$. We now apply Bernstein's inequality (see, e.g., Section 2.2 of [44]):

Bernstein's Inequality: Let $F_1, \ldots, F_k$ be independent random variables with mean 0 and bounded range $|F_i| \leqslant M$, and let F be their sum. Then for all $\eta > 0$ we have $\Pr[|F| \geqslant \eta] \leqslant 2\exp\left[-\eta^2/2(\sigma_F^2 + M\eta)\right]$.

In our case, we have that for any $\tau > 0$:

$$\Pr[|F| \geqslant \tau\sigma_F] \leqslant 2\exp\left[-\frac{(\tau\sigma_F)^2}{2(\sigma_F^2 + 2\tau\sigma_F)}\right] \leqslant 2\exp\left[-\frac{\tau^2}{2 + 4\tau/\sigma_F}\right] \leqslant 2\exp\left[-\frac{\tau^2}{2 + 4\tau}\right],$$

where the last inequality uses $\sigma_F \geqslant 1$.

As for H, we shall take $\tau \geqslant 1$, so $\tau\sigma_F \geqslant 1$ and we can apply Lemma 39 to conclude:

$$\mathsf{P}[|H - \theta| \leqslant \tau\sigma_F] \leqslant O(1)\frac{\tau\sigma_F}{\sigma_H} \leqslant O(1)\tau\left(\epsilon \log\frac{k}{\alpha}\right)^{\frac{1}{4}},$$

where the second inequality is from (16). Hence:

$$\begin{aligned}\Pr[|F| \geqslant |H - \theta|] &\leqslant \mathsf{P}[|F| \geqslant \tau\sigma_F] + \mathsf{P}[|H - \theta| \leqslant \tau\sigma_F] \\ &\leqslant 2\exp\left[-\frac{\tau^2}{2 + 4\tau}\right] + O(1)\tau\left(\epsilon \log\frac{k}{\alpha}\right)^{\frac{1}{4}}\end{aligned}$$

Taking $\tau = A\log(1/(\epsilon\log\frac{k}{\alpha}))$ with A a suitably large constant we get:

$$\mathrm{ns}_\epsilon(f) \leqslant O(1)\left(\epsilon \log\frac{k}{\alpha}\right)^{\frac{1}{4}}\log(1/(\epsilon\log\tfrac{k}{\alpha})).$$

Case 2: $\mathbf{\sigma_F^2 < 1}$. In this case, let S denote the random set $\{i : h_i(x) \neq h_i(N_\epsilon(x))\}$; we refer to S as the *flip set.* We now consider the random variable H *conditioned on S being the flip set.* As conditional random variables, we have $(H|S) = \sum_{i=1}^k (h_i(x)|S)$, and the random variables $(h_i(x)|S)$ are still independent. It is easily verified that for $i \in S$, $(h_i(x)|S)$ takes the values ± 1 with equal probability; whereas, for $i \notin S$, $(h_i(x)|S) = -1$ with probability $\frac{p_i - \epsilon_i/2}{1-\epsilon_i}$ and $(h_i(x)|S) = +1$ with probability $\frac{1-p_i-\epsilon_i/2}{1-\epsilon_i}$. Thus for $i \in S$, $\mathrm{Var}[h_i(x)|S] = 1$, and for $i \notin S$:

$$\mathrm{Var}[h_i(x)|S] = 1 - \left(\frac{1-2p_i}{1-\epsilon_i}\right)^2 = 1 - \frac{(1-2q_i)^2}{(1-\epsilon_i)^2} \geqslant \frac{(1-2q_i)^2}{(1-q_i)^2} \geqslant 2q_i(1-q_i),$$

where the first inequality uses $\epsilon_i \leqslant q_i$ from (15) and the second is elementary for $q_i \in [0, \frac{1}{2}]$. Recalling that $\mathrm{Var}[h_i(x)] = 4q_i(1-q_i)$ we see that regardless of whether $i \in S$ or not, $\mathrm{Var}[h_i(x)|S] \geqslant \frac{1}{2}\mathrm{Var}[h_i(x)]$. Hence $\mathrm{Var}[H|S] \geqslant \frac{1}{2}\sigma_H^2$. It thus follows from Lemma 39 that for every $\theta \in \mathbb{R}$ and $\lambda \geqslant 1$, $\mathsf{P}[|H - \theta| \leqslant \lambda \mid S] \leqslant O(1)\sqrt{2}\frac{\lambda}{\sigma_H}$, independently of S. Since $|F| \leqslant 2|S|$ is immediate, we conclude:

$$\begin{aligned}
\mathrm{ns}_\epsilon(f) &\leqslant \Pr[|H-\theta| \leqslant 2|S|] \\
&= \sum_{S\subseteq[k]} \Pr[\text{flip set is } S] \cdot \Pr[|H-\theta| \leqslant 2|S| \mid \text{flip set is } S] \\
&\leqslant \sum_{S\subseteq[k]} \Pr[\text{flip set is } S] \cdot O(1)\frac{|S|}{\sigma_H} \\
&= O(1)\mathbf{E}[|S|]/\sigma_H \\
&= O(1)\left(\sum_{i=1}^{k} \epsilon_i\right)/\sigma_H \\
&= O(1)(\sigma_F^2/\sigma_H) \\
&\leqslant O(1)(\sigma_F/\sigma_H),
\end{aligned}$$

where the last inequality is since $\sigma_F \leqslant 1$ in Case 2. Hence we have $\mathrm{ns}_\epsilon(f) \leqslant O(1)(\sigma_F/\sigma_H) \leqslant O(1)\left(\epsilon \log \frac{k}{\alpha}\right)^{\frac{1}{4}}$ by (16).

In conclusion, in both Cases 1 and 2 we have:

$$\mathrm{ns}_\epsilon(f) \leqslant O(1)\left(\epsilon \log \frac{k}{\alpha}\right)^{\frac{1}{4}} \log(1/(\epsilon \log \tfrac{k}{\alpha})).$$

Since $\alpha = O(1)(\epsilon \ln k)^{\frac{1}{4}}$, the proof is complete. ■

A.1 Proof of Lemma 39

We prove the following:

Proposition 40 *Let $X_1, \dots, X_n$ be independent ± 1-valued random variables where $\Pr[X_k = -1] = p_k$ and let $x = \sum_{k=1}^n X_k$. Then for every $\theta \in \mathbb{R}$,*

$$\Pr[|x-\theta| \leqslant 1] \leqslant \frac{O(1)}{\sqrt{\sum_{k=1}^n p_k(1-p_k)}}.$$

Lemma 39 as stated earlier follows easily from this via a union bound over a suitably chosen sequence of values for θ. The proof given below is based on similar arguments in Petrov's work [44].
Proof: Define:

$$p(x) = \frac{2(1-\cos x)}{x^2} \geqslant 0 \qquad \text{and} \qquad h(t) = \begin{cases} 1-|t|, & |t| \leqslant 1 \\ 0, & \text{else} \end{cases}.$$

Elementary integration by parts shows that $p(x)$ is the inverse Fourier transform of $h(t)$; i.e.,

$$p(x) = \int_{-\infty}^{\infty} e^{-itx} h(t)\, dt.$$

By considering the Taylor expansion of $\cos x$, we get that $p(x) \geqslant \frac{11}{12}$ on the interval $[-1, 1]$. Hence:

$$\begin{aligned}
\Pr[|x-\theta| \leqslant 1] &= \mathop{\mathbf{E}}_{x}[\mathbf{1}_{x\in[\theta-1,\theta+1]}] \\
&\leqslant \tfrac{12}{11}\mathbf{E}[p(x-\theta)] \\
&= \tfrac{12}{11}\mathbf{E}\Big[\int_{-\infty}^{\infty} e^{-it(x-\theta)} h(t)\, dt\Big] \\
&= \frac{12}{11}\int_{-\infty}^{\infty} \mathbf{E}[e^{-itx}e^{it\theta}h(t)]\, dt \\
&= \frac{12}{11}\Big|\int_{-\infty}^{\infty} e^{it\theta}h(t)\mathbf{E}[e^{-itx}]\, dt\Big| \qquad (17) \\
&\leqslant \frac{12}{11}\int_{-1}^{1} \big|\mathbf{E}[e^{-itx}]\big|\, dt, \qquad (18)
\end{aligned}$$

with (17) following because the quantity is already real and nonnegative, and (18) following because $|e^{it\theta}| \leqslant 1$, $h(t) = 0$ outside $[-1, 1]$, and $|h(t)| \leqslant 1$ otherwise.

Now,

$$\begin{aligned}
\mathop{\mathbf{E}}_{x}[e^{-itx}] &= \mathop{\mathbf{E}}_{x_k \leftarrow X_k}[\exp(-it\sum_{k=1}^{n} x_k)] \\
&= \mathop{\mathbf{E}}_{x_k \leftarrow X_k}\Big[\prod_{k=1}^{n} \exp(-itx_k)\Big] \\
&= \prod_{k=1}^{n} \mathop{\mathbf{E}}_{x_k \leftarrow X_k}[\exp(-itx_k)] \qquad \text{(by independence)} \\
&= \prod_{k=1}^{n} (p_k \exp(it) + (1-p_k)\exp(-it)) \\
&= \prod_{k=1}^{n} (\cos t + i(2p_k - 1)\sin t).
\end{aligned}$$

By comparing Taylor expansions one can establish that

$$|\cos t + i(2p-1)\sin t| \leqslant \exp(-\frac{11}{24}p(1-p)t^2).$$

for $p \in [0,1], t \in [-1,1]$. We may conclude that:

$$\begin{aligned}
\Pr[|x-\theta| \leqslant 1] &\leqslant \frac{12}{11}\int_{-1}^{1}\prod_{k=1}^{n}\exp(-\tfrac{11}{24}p_k(1-p_k)t^2)\,dt \\
&= \frac{12}{11}\int_{-1}^{1}\exp\left(-\frac{11}{24}\left[\sum_{k=1}^{n}p_k(1-p_k)\right]t^2\right)\,dt \\
&= \frac{12}{11}\int_{-1}^{1}\exp\left(-\frac{t^2}{2\left(\sqrt{\frac{12}{11}}\left(\sum p_k(1-p_k)\right)^{-1/2}\right)^2}\right)\,dt \\
&\leqslant \frac{12}{11}\int_{-\infty}^{\infty}\exp\left(-\frac{t^2}{2\left(\sqrt{\frac{12}{11}}\left(\sum p_k(1-p_k)\right)^{-1/2}\right)^2}\right)\,dt \\
&= \sqrt{2\pi}(\tfrac{12}{11})^{3/2}\left(\sum p_k(1-p_k)\right)^{-1/2},
\end{aligned}$$

since $(\sqrt{2\pi}\sigma)^{-1}\exp(-t^2/2\sigma^2)$ is a probability density function for every positive σ. ■