The Chow Parameters Problem

[Extended Abstract] *

Ryan O'Donnell
Carnegie Mellon University
Pittsburgh, PA 15213
odonnell@cs.cmu.edu

Rocco A. Servedio[†]
Columbia University
New York, NY 10027
rocco@cs.columbia.edu

ABSTRACT

In the 2nd Annual FOCS (1961), C. K. Chow proved that every Boolean threshold function is uniquely determined by its degree-0 and degree-1 Fourier coefficients. These numbers became known as the *Chow Parameters*. Providing an algorithmic version of Chow's theorem — i.e., efficiently constructing a representation of a threshold function given its Chow Parameters — has remained open ever since. This problem has received significant study in the fields of circuit complexity, game theory and the design of voting systems, and learning theory.

In this paper we effectively solve the problem, giving a randomized PTAS with the following behavior:

Theorem: Given the Chow Parameters of a Boolean threshold function f over n bits and any constant $\epsilon > 0$, the algorithm runs in time $O(n^2 \log^2 n)$ and with high probability outputs a representation of a threshold function f' which is ϵ-close to f.

Along the way we prove several new results of independent interest about Boolean threshold functions. In addition to various structural results, these include the following new algorithmic results in learning theory (where threshold functions are usually called "halfspaces"):

- An $\tilde{O}(n^2)$-time uniform distribution algorithm for learning halfspaces to constant accuracy in the "Restricted Focus of Attention" (RFA) model of Ben-David et al. [3]. This answers the main open question of [6].
- An $\tilde{O}(n^2)$-time agnostic-type learning algorithm for halfspaces under the uniform distribution. This contrasts with recent results of Guruswami and Raghavendra [21] who show that the learning problem we solve is NP-hard under general distributions.

As a special case of the latter result we obtain the fastest known algorithm for learning halfspaces to constant accuracy in the uniform distribution PAC learning model. For constant ϵ our algorithm runs in time $\tilde{O}(n^2)$, which substantially improves on previous bounds and nearly matches the $\Omega(n^2)$ bits of training data that any successful learning algorithm must use.

Categories and Subject Descriptors

F.2.2 [**Nonnumerical Algorithms and Problems**]: Computations on Discrete Structures; I.2.6 [**Learning**]: Concept Learning

General Terms

Algorithms, Theory

Keywords

Boolean function; Fourier Analysis; Threshold function; Chow parameters

1. INTRODUCTION

This paper is concerned with Boolean threshold functions:

Definition 1. *A Boolean function* $f : \{-1,1\}^n \to \{-1,1\}$ *is a* threshold function *if it is expressible as* $f(x) = \mathrm{sgn}(w_0 + w_1 x_1 + \cdots + w_n x_n)$ *for some real numbers* $w_0, w_1, \ldots, w_n$.

Boolean threshold functions are of fundamental interest in circuit complexity, game theory/voting theory, and learning theory. Early computer scientists studying "switching functions" (i.e., Boolean functions) spent an enormous amount of effort on the class of threshold functions; see for instance the books [11, 24, 34, 45, 36] on this topic. More recently, researchers in circuit complexity have worked to understand the computational power of threshold functions and shallow circuits with these functions as gates; see e.g. [19, 42, 22, 23, 20]. In game theory and social choice theory, where simple cooperative games [40] correspond to monotone Boolean functions, threshold functions (with nonnegative weights) are known as "weighted majority" games and have been extensively studied as models for voting, see e.g. [41, 25, 12, 48]. Finally, in various guises, the problem of learning an unknown threshold function ("halfspace") has arguably been the central problem in machine learning for much of the last two decades, with algorithms such as Perceptron, Weighted Majority, boosting, and support vector machines emerging as central tools in the field.

[*] A full version of this paper is available at www.cs.columbia.edu/~rocco/papers/chow.html.

[†] Supported in part by NSF award CCF-0347282, by NSF award CCF-0523664, and by a Sloan Foundation Fellowship.

STOC'08, May 17–20, 2008, Victoria, British Columbia, Canada.
Copyright 2008 ACM 978-1-60558-047-0/08/05 ...$5.00.

A beautiful result of C. K. Chow from the 2nd FOCS conference [10] gives a surprising characterization of Boolean threshold functions: among all Boolean functions, each threshold function f is uniquely determined by the "center of mass" of its positive inputs, $\text{avg}\{x : f(x) = 1\}$, and the number of positive inputs $\#\{x : f(x) = 1\}$. These $n+1$ parameters of f are equivalent, after scaling and additive shifting, to its degree-0 and degree-1 Fourier coefficients (and also, essentially, to its "influences" or "Banzhaf power indices"). We give a formal definition:

Definition 2. *Given any Boolean function $f : \{-1,1\}^n \to \{-1,1\}$, its* Chow Parameters[1] *are the rational numbers $\widehat{f}(0)$, $\widehat{f}(1)$, ..., $\widehat{f}(n)$ defined by $\widehat{f}(0) = \mathbf{E}[f(x)]$, $\widehat{f}(i) = \mathbf{E}[f(x)x_i]$, for $1 \le i \le n$. We also say the* Chow vector *of f is $\vec{\chi} = \vec{\chi}_f = (\widehat{f}(0), \widehat{f}(1), \dots, \widehat{f}(n))$.*

Throughout this paper the notation $\mathbf{E}[\cdot]$ and $\mathbf{Pr}[\cdot]$ refers to an $x \in \{-1,1\}^n$ chosen uniformly at random. (We note that this corresponds to the "Impartial Culture Assumption" in the theory of social choice, see e.g. [5].) Our notation slightly abuses the standard Fourier coefficient notation of $\widehat{f}(\emptyset)$ and $\widehat{f}(\{i\})$.

Chow's theorem implies that the following algorithmic problem is in principle solvable:

The Chow Parameters Problem: *Given the Chow Parameters $\widehat{f}(0)$, $\widehat{f}(1)$, ..., $\widehat{f}(n)$ of a Boolean threshold function f, output a representation of f as $f(x) = \text{sgn}(w_0 + w_1x_1 + \cdots w_nx_n)$.*

Unfortunately, the proof of Chow's theorem (reviewed in Section 2) is completely nonconstructive and does not suggest any algorithm, much less an efficient one. As we now briefly describe, over the past five decades the Chow Parameters problem has been considered by researchers in a range of different fields.

1.1 Background

As far back as 1960 researchers studying Boolean functions were interested in finding an efficient algorithm for the Chow Parameters problem [14]. Electrical engineers at the time faced the following problem: Given an explicit truth table, determine if it can be realized as a threshold circuit and if so, which one. The Chow Parameters are easily computed from a truth table, and Chow's theorem implies that they give a unique representation for every threshold function. Several heuristics were proposed for the Chow Parameters problem [28, 50, 27, 11], an empirical study was performed to compare various methods [52], and lookup tables were produced mapping Chow vectors into weights-based representations for each threshold function on six [37], seven [51], and eight [39] bits. Winder provides a good early survey [53]. Generalizations of Chow's theorem were given in [8, 43].

Researchers in game theory have also considered the Chow Parameters problem; Chow's theorem was independently rediscovered by the game theorist Lapidot [32] and subsequently studied in [12, 13, 48, 17]. In the realm of social choice and voting theory the Chow Parameters represent the Banzhaf power indices [41, 2] of the n voters — a measure of each one's "influence" over the outcome. Here the Chow Parameters problem is very natural: Consider designing a voting rule for, say, the European Union. Target Banzhaf power indices are given, usually in proportion to the square-root of the states' populations, and one wishes to come up with a weighted majority voting rule whose power indices are as close to the targets as possible. Researchers in voting theory have recently devoted significant attention to this problem [33, 9], calling it a "fundamental constitutional problem" [16] and in particular considering its computational complexity [46, 1].

The Chow Parameters problem also has motivation from learning theory. Ben-David and Dichterman [3] introduced the "Restricted Focus of Attention" model to formalize the idea that learning algorithms often have only partial access to each example vector. Birkendorf et al. [6] performed a comprehensive study of the RFA model and observed that the approximation version of the Chow Parameters problem (given approximate Chow Parameters, output an approximating threshold function) is equivalent to the problem of efficiently learning threshold functions under the uniform distribution in the 1-RFA model. (In the 1-RFA model the learner is only allowed to see one bit of each example string in addition to the label; we give details in Section 6.) As the main open question posed in [6], Birkendorf et al. asked whether there is an efficient uniform distribution learning algorithm for threshold functions in the 1-RFA model. This question motivated subsequent research [18, 44] which gave *information-theoretic* sample complexity upper bounds for this learning problem (see Section 3); however no computationally efficient algorithm was previously known.

To summarize, we believe that the range of different contexts in which the Chow Parameters Problem has arisen is evidence of its fundamental status.

1.2 The Chow Parameters Problem Reformulated as an Approximation Problem

It is unlikely that the Chow Parameters Problem can be solved exactly in polynomial time — note that even checking the correctness of a candidate solution is #P-complete, because computing $\widehat{f}(0)$ is equivalent to counting feasible 0-1 knapsack solutions. Thus, as is implicitly proposed in [6, 1], it is natural to look for a polynomial-time approximation scheme (PTAS). Here we mean an approximation in the following sense:

Definition 3. *The* distance *between two Boolean functions $f, g : \{-1,1\}^n \to \{-1,1\}$ is $\text{dist}(f,g) \overset{def}{=} \mathbf{Pr}[f(x) \neq g(x)]$. If $\text{dist}(f,g) \le \epsilon$ we say that f and g are* ϵ-close.

We would like a PTAS which, given a value ϵ and the Chow Parameters of f, outputs a threshold function f' that is ϵ-close to f. With this relaxed goal of approximating f, one may even tolerate only an approximation of the Chow Parameters of f; this gives us the variant of the problem that Birkendorf et al. considered. (Note that, as we discuss in Section 3, it is in no way obvious that approximate Chow Parameters even *information-theoretically* specify an approximator to f.) In particular we will consider the following notion of "approximate" Chow Parameters:

Definition 4. *Let $f, g : \{-1,1\}^n \to \{-1,1\}$. We define $d_{\text{Chow}}(f,g) \overset{def}{=} \sqrt{\sum_{j=0}^n (\widehat{f}(j) - \widehat{g}(j))^2}$ to be the* Chow distance *between f and g.*

[1] Chow's theorem was proven simultaneously by Tannenbaum [47], but the terminology "Chow Parameters" has stuck.

1.3 Our Results

Our main result is an efficient PTAS $\mathcal{A}$ for the Chow Parameters problem which succeeds given approximations to the Chow Parameters. We prove:

Main Theorem. *There is a function* $\kappa(\epsilon) = 2^{-\tilde{O}(1/\epsilon^2)}$ *such that the following holds.*

Let $f : \{-1,1\}^n \to \{-1,1\}$ *be a threshold function and let* $0 < \epsilon < 1/2$. *Write* $\vec{\chi}$ *for the Chow vector of* f *and assume that* $\vec{\alpha}$ *is a vector satisfying* $\|\vec{\alpha} - \vec{\chi}\| \leq \kappa(\epsilon)$.

Then given as input $\vec{\alpha}$ *and* ϵ *the randomized algorithm* $\mathcal{A}$ *performs* $2^{\mathrm{poly}(1/\kappa(\epsilon))} \cdot n^2 \cdot \log n \cdot \log(\frac{n}{\delta})$ *bit operations and outputs the (weights-based) representation of a threshold function* f^* *which with probability at least* $1 - \delta$ *satisfies* $\mathrm{dist}(f, f^*) \leq \epsilon$.

Although the running time dependence on ϵ is doubly-exponential, we emphasize that the polynomial dependence on n is quadratic, independent of ϵ; i.e., $\mathcal{A}$ is an "EPTAS". Some of our learning applications have only singly-exponential dependence on ϵ.

1.4 Our Approach

We briefly describe the two main ingredients of our approach and explain how we combine them to obtain the efficient algorithm $\mathcal{A}$.

First ingredient: small Chow distance from a threshold function implies small distance. An immediate question that arises when thinking about the Chow Parameters problem is how to recognize whether a candidate solution is a good one. If we are given the Chow vector $\vec{\chi}_f$ of an unknown threshold function f and we have a candidate threshold function g, we can approximate the Chow vector $\vec{\chi}_g$ of g by sampling. The following Proposition is easily proved via Fourier analysis:

PROPOSITION 5. $d_{\mathrm{Chow}}(f,g) \leq 2\sqrt{\mathrm{dist}(f,g)}$.

This means that if $d_{\mathrm{Chow}}(f,g)$ is large then f and g are far apart. But if $d_{\mathrm{Chow}}(f,g)$ is small, does this necessarily mean that f and g are close?

This question has been studied in the learning theory community by [6] (for threshold functions with small integer weights), [18], and [44]. In Section 3 we show that the answer is yes by proving the following "robust" version of Chow's theorem:

THEOREM 6. *Let* $f : \{-1,1\}^n \to \{-1,1\}$ *be any threshold function and let* $g : \{-1,1\}^n \to \{-1,1\}$ *be any function such that* $d_{\mathrm{Chow}}(f,g) \leq \epsilon$. *Then* $\mathrm{dist}(f,g) \leq \tilde{O}\left(1/\sqrt{\log(1/\epsilon)}\right)$.

This is the first result of this nature that is completely independent of n. A key ingredient in the proof of Theorem 6 is a new result showing that every threshold function f is extremely close to a threshold function f' for which only a very small fraction of points have small margin (see Section 3.3 for a precise statement). We feel that this and Theorem 6 have independent interest as structural results about threshold functions.

Second ingredient: using the Chow Parameters as weights. Next, we establish a result, Theorem 15, whose Corollary 16 is the following:

> Let $f(x) = \mathrm{sgn}(w_0 + w_1 x_1 + \cdots + w_n x_n)$ be any threshold function, and let H be the set of $\mathrm{poly}(1/\epsilon)$ indices i for which $|w_i|$ (equivalently, $|\widehat{f}(i)|$) is largest. Then there exists a threshold function $f'(x) = \mathrm{sgn}(v_0 + v_1 x_1 + \cdots + v_n x_n)$ with $\mathrm{dist}(f, f') \leq \epsilon$ in which the weights v_i for $i \in [n] \setminus H$ *are the Chow Parameters* $\widehat{f}(i)$ *themselves.*

The heuristic of using the Chow Parameters as possible weights was considered by several researchers in the early '60s (see [53]); however no theorem on the efficacy of this approach was previously known. Our proof of Theorem 15 and its robust version Theorem 18 rely in part on recent work on Property Testing for threshold functions [35].

The algorithm and intuitive explanation. Given these two ingredients, our PTAS $\mathcal{A}$ for the approximate Chow Parameters problem works by constructing a "small" (depending only on ϵ) number of candidate threshold functions. It enumerates "all" (in some sense) possible weight settings for the indices in H, and for each one produces a candidate threshold function by setting the remaining weights equal to the given Chow Parameters. The second ingredient tells us that at least one of these candidate threshold functions must be close to to the unknown threshold function f, and thus must have small Chow distance to f, by Proposition 5. Now the first ingredient tells us that *any* threshold function whose Chow distance to the target Chow vector is small must itself be close to the target. So the algorithm can estimate each of the candidates' Chow vectors (this takes $\tilde{O}(n^2)$ time) and output any candidate whose Chow distance to the target vector is small.

Consequences in learning theory. As we describe in Section 6, our approach yields a range of new algorithmic results in learning theory.

2. PRELIMINARIES

We assume familiarity with the basic elements of Fourier analysis over the Boolean cube $\{-1,1\}^n$.

Let us introduce a notion of "margin" for threshold functions:

DEFINITION 7. *Let* $f : \{-1,1\}^n \to \{-1,1\}$ *be a Boolean threshold function,* $f(x) = \mathrm{sgn}(w_0 + w_1 x_1 + \cdots + w_n x_n)$, *where the weights are scaled so that* $\sum_{j \geq 0} w_j^2 = 1$. *Given a particular input* $x \in \{-1,1\}^n$ *we define* $\mathrm{marg}(f,x) = |w_0 + w_1 x_1 + \cdots + w_n x_n|$.[2]

REMARK 8. *The usual notion of "margin" from learning theory also involves scaling the* data points x *so that* $\|x\| \leq 1$ *for all* x. *Thus we have* $\text{learning-theoretic-margin}(f,x) = \mathrm{marg}(f,x)/\sqrt{n}$.

We now present a proof of Chow's 1961 theorem:

THEOREM 9. *Let* $f : \{-1,1\}^n \to \{-1,1\}$ *be a Boolean threshold function and let* $g : \{-1,1\}^n \to \{-1,1\}$ *be a Boolean function such that* $\widehat{g}(j) = \widehat{f}(j)$ *for all* $0 \leq j \leq n$. *Then* $g = f$.

Note that another way of phrasing this is: "If f is a Boolean threshold function, g is a Boolean function, and $d_{\mathrm{Chow}}(f,g) = 0$, then $\mathrm{dist}(f,g) = 0$." Our Theorem 6 gives a "robust" version of this statement.

[2] This notation is slightly informal since it doesn't show the dependence on the *representation* of f.

PROOF. Write $f(x) = \mathrm{sgn}(w_0+w_1x_1+\cdots+w_nx_n)$, where the weights are scaled so that $\sum_{j=0}^{n} w_j^2 = 1$. We may assume without loss of generality that $\mathrm{marg}(f,x) \neq 0$ for all x. (Otherwise, first perturb the weights slightly without changing f.) Now we have

$$\begin{aligned} 0 &= \sum_{j=0}^{n} w_j(\widehat{f}(j) - \widehat{g}(j)) = \mathbf{E}[(w_0 + \sum_{i=1}^{n} w_i x_i)(f(x) - g(x))] \\ &= \mathbf{E}[\mathbf{1}_{\{f(x)\neq g(x)\}} \cdot 2\mathrm{marg}(f,x)]. \end{aligned}$$

The first equality is by the assumption that $\widehat{f}(j) = \widehat{g}(j)$ for all $0 \leq j \leq n$, the second equality is Plancherel's identity, and the third equality uses the fact that $f(x) = \mathrm{sgn}(w_0 + w_1x_1 + \cdots + w_nx_n)$. But since $\mathrm{marg}(f,x)$ is always strictly positive, we must have $\mathbf{Pr}[f(x) \neq g(x)] = 0$ as claimed. □

3. FIRST INGREDIENT: SMALL CHOW DISTANCE IMPLIES SMALL DISTANCE

Our main result in this section is the following.

Theorem 6 (restated) *Let* $f : \{-1,1\}^n \to \{-1,1\}$ *be any threshold function and let* $g : \{-1,1\}^n \to \{-1,1\}$ *be any Boolean function such that* $d_{\mathrm{Chow}}(f,g) \leq \epsilon$. *Then* $\mathrm{dist}(f,g) \leq \tilde{O}\left(1/\sqrt{\log(1/\epsilon)}\right)$.

Let us compare this with some recent results with a similar qualitative flavor. The main result of [18] is a proof that for any threshold function f and any Boolean function g, if $|\widehat{f}(j) - \widehat{g}(j)| \leq (\epsilon/n)^{O(\log(n/\epsilon)\log(1/\epsilon))}$ for all $0 \leq j \leq n$, then $\mathrm{dist}(f,g) \leq \epsilon$. Note that the condition of Goldberg's theorem requires that $d_{\mathrm{Chow}}(f,g) \leq n^{-O(\log n)}$. Subsequently Servedio [44] showed that to obtain $\mathrm{dist}(f,g) \leq \epsilon$ it suffices to have $|\widehat{f}(j) - \widehat{g}(j)| \leq 1/(2^{\tilde{O}(1/\epsilon^2)} \cdot n)$ for all $0 \leq j \leq n$. This is a worse requirement in terms of ϵ but a better one in terms of n; however it still requires that $d_{\mathrm{Chow}}(f,g) \leq 1/\sqrt{n}$. In contrast, Theorem 6 allows the Chow distance between f and g to be an absolute constant *independent* of n. This independence of n will be crucial later on when we use Theorem 6 to obtain a computationally efficient algorithm for the Chow Parameters problem.

At a high level, we prove Theorem 6 by giving a "robust" version of the proof of Chow's Theorem (Theorem 9). A first obvious approach to making the argument robust is to try to show that every threshold function has margin $\Omega(1)$ (independent of n) on every x. However this is well known to be badly false. A next attempt might be to show that every threshold function has a representation with margin $\Omega(1)$ on *almost* every x. This too turns out to be impossible (cf. our discussion after the statement of Lemma 12 below). The key to getting an "n-independent" margin lower bound is to also very slightly alter the threshold function. Specifically, in Theorem 13 we show that any threshold function f is very close to another threshold function f' satisfying $\mathrm{marg}(f',x) \geq \Omega(1)$ for almost all x. This is the key structural result for threshold functions that allows us to "robustify" the proof of Theorem 9.

3.1 The Critical Index and Anticoncentration

Fix a representation $f(x) = \mathrm{sgn}(w_0 + w_1x_1 + \cdots + w_nx_n)$ of a threshold function. Throughout Section 3 we adopt the convention that $|w_1| \geq \cdots \geq |w_n| > 0$ (this is without loss of generality for our results on margin and Chow distance, by permuting indices).

The notion of the "critical index" of the sequence of weights $w_1,\ldots,w_n$ will be useful for us. Roughly speaking, it allows us to approximately decompose any linear form $w_0 + w_1x_1+\cdots+w_nx_n$ over random ±1 x_i's into a short dominant "head", $w_0 + w_1x_1 + \cdots + w_{\mathrm{small}}x_{\mathrm{small}}$, and a long remaining "tail" which acts like a Gaussian random variable. The "τ-critical index" of $w_1,\ldots,w_n$ is essentially the least index ℓ for which the random variable $w_\ell x_\ell + \cdots + w_nx_n$ behaves like a Gaussian up to error τ. The notion of a critical index was (implicitly) introduced and used in [44].

To prove a margin lower bound for f, we need to show some kind of "anticoncentration" for the random variable $w_0+w_1x_1+\cdots+w_nx_n$; we want it to rarely be near 0. Let us describe intuitively how analyzing the critical index helps us show this. If the critical index of $w_1,\ldots,w_n$ is large, then it must be the case that the initial weights $w_1, w_2, \ldots$ up to the critical index are rapidly decreasing (roughly speaking, if the weights $w_i, w_{i+1}, \ldots$ stayed about the same for a long stretch this would cause $w_ix_i+\cdots+w_nx_n$ to behave like a Gaussian). This rapid decrease can in turn be shown to imply that the the "head" part $w_0 + w_1x_1 + \cdots + w_{\mathrm{small}}x_{\mathrm{small}}$ is not too concentrated around any particular value; see Theorem 11 below. On the other hand, if the critical index ℓ is small, then the random variable $w_\ell x_\ell + \cdots + w_nx_n$ behaves like a Gaussian. Since Gaussians have good anticoncentration, the overall linear form $w_0 + w_1x_1 + \cdots + w_nx_n$ will have good anticoncentration, regardless of the head part's value. We need to alter f slightly to make these two cases go through, but having done so, we are able to bound the fraction of inputs x for which $\mathrm{marg}(f,x)$ is very small. As described, this margin bound can then be used to prove Theorem 6.

We now give precise definitions. For $1 \leq k \leq n$ we write σ_k to denote the 2-norm of the "tail weights" starting from k; i.e. $\sigma_k \stackrel{\mathrm{def}}{=} \sqrt{\sum_{i\geq k}^{n} w_i^2}$.

DEFINITION 10. *Fix a parameter* $0 < \tau < 1/2$. *We define the* τ-critical index *of the weight vector* w *to be the least index* ℓ *such that* w_ℓ *is "small" relative to* σ_ℓ *in the following sense:*

$$\frac{|w_\ell|}{\sigma_\ell} \leq \tau. \qquad (1)$$

(If no index $1 \leq \ell \leq n$ satisfies (1), as is the case for $(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots, \frac{1}{2^n})$ for example, then we say that the τ-critical index is $+\infty$.) The connection between Equation (1) and behaving like a Gaussian up to error τ is given by the Berry-Esseen Theorem.

The following anticoncentration result shows that if the critical index is large, then the random variable $w_1x_1+\cdots+w_nx_n$ does not put much probability mass close to any particular value:

THEOREM 11. *Let* $0 < \tau < 1/2$ *and* $t \geq 1$ *be parameters, and define* $k = \left\lceil O(1)\frac{t}{\tau^2}\ln\left(\frac{t}{\tau}\right)\right\rceil$. *If the* τ*-critical index* ℓ *for* $w_1,\ldots,w_n$ *satisfies* $\ell \geq k$, *then we have*

$$\mathop{\mathbf{Pr}}_{x}[|w_0 + w_1x_1 + \cdots + w_nx_n| \leq \sqrt{t}\cdot\sigma_k] \leq O(2^{-t}).$$

A similar result was established in [44]. We prove Theorem 11 in the full version of the paper.

3.2 Approximating Threshold Functions Using Not-Too-Large Head Weights

The following lemma roughly says that any threshold function f can be approximated by a threshold function f' in which the 2-norm of the tail weights, σ_k, is at least an $\Omega(1)$ fraction of the head weights. This is important so that the Gaussian random variable to which the tail part is close has $\Omega(1)$ variance and thus sufficiently good anticoncentration.

LEMMA 12. *Let $f : \{-1,1\}^n \to \{-1,1\}$ be any threshold function, $f(x) = \operatorname{sgn}(w_0 + w_1x_1 + \cdots + w_nx_n)$ (recall that we assume $|w_1| \geq |w_2| \geq \cdots \geq |w_n|$). Let $0 < \epsilon < 1/2$ and $1 \leq k \leq n$ be parameters, and write $\sigma_k \overset{def}{=} \sqrt{\sum_{j\geq k} w_j^2}$.*

Assuming $\sigma_k > 0$, there are numbers $v_0, \ldots, v_{k-1}$ satisfying

$$|v_i| \leq k^{(k+1)/2} \cdot \sqrt{3\ln(2/\epsilon)} \cdot \sigma_k \qquad (2)$$

such that the threshold function $f' : \{-1,1\}^n \to \{-1,1\}$ defined by

$$f'(x) = \operatorname{sgn}(v_0 + v_1x_1 + \cdots + v_{k-1}x_{k-1} + w_kx_k + \cdots + w_nx_n)$$

satisfies $\operatorname{dist}(f, f') \leq \epsilon$. One may further ensure that $|v_1| \geq |v_2| \geq \cdots \geq |v_{k-1}| \geq |w_k|$ and that $\operatorname{sgn}(v_i) = \operatorname{sgn}(w_i)$ for all i.

We prove this lemma in the full version. To illustrate the lemma, consider the threshold function

$$f(x) = \operatorname{sgn}(nx_1 + nx_2 + x_3 + \cdots + x_n), \qquad (3)$$

with $k = 3$. The tail weights here have $\sigma_3 = \sqrt{n-2}$, which of course is not a constant fraction of the two head weights, n. Further, this cannot be fixed just by choosing a different weights-based representation of the same function f. What Lemma 12 shows here is that we can shrink the head weights from n all the way down to $\Theta(\sqrt{\ln(1/\epsilon)})\sqrt{n}$ without changing the function on more than an ϵ fraction of points (this heavily uses the fact that the tail acts like a Gaussian with standard deviation $\sqrt{n-2}$). Then indeed σ_3 is an $\Omega(1)$ fraction of the head weights, as desired.

3.3 Every Threshold Function is Close to a Threshold Function for which Few Points have Small Margin

In this subsection we describe how Lemma 12 and Theorem 11 can be used to establish our main structural results about margins:

THEOREM 13. *Let $f : \{-1,1\}^n \to \{-1,1\}$ be any threshold function and let $0 < \tau < 1/2$. Then there is a threshold function $f' : \{-1,1\}^n \to \{-1,1\}$ with $\operatorname{dist}(f, f') \leq \epsilon$ satisfying $\mathbf{Pr}_x[\operatorname{marg}(f', x) \leq \rho] \leq O(\tau)$, where*

$$\epsilon = \epsilon(\tau) = 2^{-2^{O(\log^3(1/\tau)/\tau^2)}} \text{ and } \rho = \rho(\tau) = 2^{-O(\log^3(1/\tau)/\tau^2)}.$$

We remark that although we only get a margin bound ρ which is exponentially small in the fraction τ of points which fail it, the amount ϵ by which we have to change f is extremely small: doubly-exponential.

We may rephrase the above result as follows:

COROLLARY 14. *Let $f : \{-1,1\}^n \to \{-1,1\}$ be any threshold function and let $\rho > 0$ be sufficiently small. Then there is a threshold function $f' : \{-1,1\}^n \to \{-1,1\}$ with $\operatorname{dist}(f, f') \leq 2^{-1/\rho}$ satisfying*

$$\mathop{\mathbf{Pr}}_x[\operatorname{marg}(f', x) \leq \rho] \leq \tilde{O}\left(1/\sqrt{\log(1/\rho)}\right).$$

The plan for the proof of Theorem 13 follows the intuition from Section 3.1. We consider the location of the τ-critical index of f. Case 1 is that it occurs quite early. In that case, the resulting tail acts like a Gaussian (up to error τ), and hence we can get a good anticoncentration bound so long as the tail's variance is large enough. To ensure this, we alter f at the beginning of the argument using Lemma 12, which yields tail weights with $\Omega(1)$ total variance. Case 2 is that the critical index occurs late. In this case we get anticoncentration by appealing to Theorem 11. We again use Lemma 12 so that the σ_k parameter is not too small.

We now give the formal proof.

PROOF OF THEOREM 13. We intend to apply Theorem 11 in Case 2 with its t parameter set to $\log(1/\tau)$, so that the anticoncentration is $O(\tau)$. Thus we will need to ensure the τ-critical index parameter ℓ is at least

$$k \overset{\text{def}}{=} \left\lceil O(1)\frac{\log(1/\tau)}{\tau^2}\ln\left(\frac{\log(1/\tau)}{\tau}\right)\right\rceil. \qquad (4)$$

To that end, fix a weights-based representation of f,

$$f(x) = \operatorname{sgn}(w_0 + w_1x_1 + \cdots + w_nx_n),$$

where we may assume that $|w_1| \geq |w_2| \geq \cdots \geq |w_n| > 0$. Write $\sigma_k = \sqrt{\sum_{j\geq k} w_j^2}$, and observe that $\sigma_k > 0$ since each $w_i \neq 0$. Now apply Lemma 12, with its parameter ϵ set to $2^{-k^{O(k)}}$. This yields a new threshold function

$$f'(x) = \operatorname{sgn}(v_0 + v_1x_1 + \cdots + v_{k-1}x_{k-1} + w_kx_k + \cdots w_nx_n),$$

where each v_i satisfies

$$|v_i| \leq k^{O(k)} \cdot \sigma_k, \qquad (5)$$

and also $|v_1| \geq |v_2| \geq \cdots \geq |v_{k-1}| \geq |w_k|$.

To analyze $\operatorname{marg}(f', x)$, let us normalize the weights of f', writing

$$f'(x) = \operatorname{sgn}(u_0 + u_1x_1 + \cdots + u_{k-1}x_{k-1} + u_kx_k + \cdots u_nx_n),$$

where $\sum_{j\geq 0} u_j^2 = 1$. Letting σ'_i denote $\sqrt{\sum_{j\geq i} u_j^2}$, it is easy to see that (5) implies

$$\sigma'_k \geq k^{-O(k)}. \qquad (6)$$

Recalling that we still have $|u_1| \geq |u_2| \geq \cdots \geq |u_n| > 0$, let ℓ be the τ-critical index for $u_1, \ldots, u_n$, and consider two cases:

Case 1: $\ell < k$. In this case, consider any fixed choice for $x_1, \ldots, x_{\ell-1}$ and write $h = u_0 + u_1x_1 + \cdots + u_{\ell-1}x_{\ell-1}$. Using the definition of τ-critical index and applying the Berry-Esseen theorem to $u_\ell x_\ell + \cdots + u_nx_n$, we get

$$\mathop{\mathbf{Pr}}_{x_\ell,\ldots,x_n}[-h - \gamma \leq u_\ell x_\ell + \cdots + u_nx_n \leq -h + \gamma] \leq \frac{2\gamma}{\sigma'_\ell} + 2\tau,$$

for any choice of $\gamma \geq 0$. Taking $\gamma = \tau\sigma'_\ell \geq \tau\sigma'_k$ we conclude

$$\mathop{\mathbf{Pr}}_x[\operatorname{marg}(f', x) \leq \tau\sigma'_k] \leq 4\tau.$$

Case 2: $\ell \geq k$. In this case we apply Theorem 11, with its parameter t set to $\log(1/\tau)$, as described at the beginning of the proof. With k defined as in (4), we conclude

$$\Pr_x[\mathrm{marg}(f', x) \leq \sqrt{\log(1/\tau)} \cdot \sigma'_k] \leq O(\tau).$$

Combining the results of the two cases and using $\sigma'_k \geq k^{-O(k)}$ from (6), we conclude that we always have

$$\Pr_x[\mathrm{marg}(f', x) \leq \tau k^{-O(k)}] \leq O(\tau).$$

Now it only remains to observe that by definition (4) of k,

$$k^{-O(k)} = 2^{-O(\log^3(1/\tau)/\tau^2)}.$$

Hence we have that

$$\mathrm{dist}(f, f') \leq 2^{-k^{O(k)}} \leq \epsilon(\tau)$$

and

$$\tau k^{-O(k)} \geq \tau 2^{-O(\log^3(1/\tau)/\tau^2)} \geq \rho(\tau).$$

□

3.4 Proof of Theorem 6

We prove Theorem 6 using essentially the same simple argument used in the proof of Theorem 9, but now applied to the approximator f' which has the margin property asserted in Corollary 14.

Proof. Given f, apply Corollary 14 with its parameter ρ set (with foresight) to $\rho = \sqrt{\epsilon \log(1/\epsilon)}$. This yields a threshold function $f'(x) = \mathrm{sgn}(u_0 + u_1 x_1 + \cdots + u_n x_n)$, with $\sum_{j=0}^n u_j^2 = 1$ satisfying

$$\mathrm{dist}(f, f') \leq 2^{-1/\rho} \ll \epsilon$$

and

$$\begin{aligned}\Pr_x[\mathrm{marg}(f', x) \leq \rho] \leq \tau \quad &\overset{\mathrm{def}}{=} \quad \tilde{O}\left(1/\sqrt{\log(1/\rho)}\right) \\ &= \quad \frac{\mathrm{poly}\log\log(1/\epsilon)}{\sqrt{\log(1/\epsilon)}}. \qquad (7)\end{aligned}$$

Since $\mathrm{dist}(f, f') \leq \epsilon$, Proposition 5 gives $d_{\mathrm{Chow}}(f, f') \leq 2\sqrt{\epsilon}$ and thus $d_{\mathrm{Chow}}(f', g) \leq 3\sqrt{\epsilon}$ by the triangle inequality. We now follow the proof of Chow's Theorem 9:

$$\begin{aligned} 3\sqrt{\epsilon} \geq d_{\mathrm{Chow}}(f', g) &= \sqrt{\sum_{j=0}^n u_j^2} \cdot \sqrt{\sum_{j=0}^n (\widehat{f'}(j) - \widehat{g}(j))^2} \\ \geq \sum_{j=0}^n u_j(\widehat{f'}(j) - \widehat{g}(j)) &= \mathbf{E}[\mathbf{1}_{\{f'(x) \neq g(x)\}} \cdot 2\mathrm{marg}(f', x)], \end{aligned} \qquad (8)$$

where the second inequality is Cauchy-Schwarz.

Now suppose that $\Pr[f'(x) \neq g(x)] \geq 2\tau$. Then by (7) we must have that for at least a τ fraction of x's, both $f'(x) \neq g(x)$ and $\mathrm{marg}(f', x) > \rho$. This gives a contribution exceeding $\tau\rho$ to (8). But

$$\tau\rho = \sqrt{\epsilon} \cdot \mathrm{poly}\log\log(1/\epsilon) > 3\sqrt{\epsilon},$$

a contradiction. Thus $\mathrm{dist}(f', g) \leq 2\tau$ and so $\mathrm{dist}(f, g)$ is at most

$$\mathrm{dist}(f, f') + \mathrm{dist}(f', g) \leq \epsilon + 2\tau = \tilde{O}\left(1/\sqrt{\log(1/\epsilon)}\right). \quad \square$$

4. SECOND INGREDIENT: USING CHOW PARAMETERS AS WEIGHTS FOR TAIL VARIABLES

We begin this section with some informal motivation for and description of our "second ingredient."

Since every threshold function is unate, the magnitude of the Fourier coefficient $|\hat{f}(i)|$ is equal to the *influence* of the variable x_i on f; i.e. $\Pr[f(x) \neq f(y)]$ where x is drawn uniformly from $\{-1, 1\}^n$ and y is x with the ith bit flipped. As done in the "first ingredient", it is natural to group together the high-influence variables, forming the "head" indices of f. We refer to the remaining indices as the "tail" indices. Note that an algorithm for the Chow Parameters problem can do this grouping, since it is given the $\hat{f}(i)$'s.

The following theorem states that any threshold function f is either already close to a junta over the head indices (i.e. a Boolean function that depends only on the head indices) or is close to a threshold function obtained by replacing the tail weights with (suitably scaled versions of) the tail Chow Parameters. (We have made no effort to optimize the precise polynomial dependence of $\tau(\epsilon)$ on ϵ.)

Theorem 15. *There is a function $\tau(\epsilon) = \mathrm{poly}(\epsilon)$ such that the following holds:*
Let f be a Boolean threshold function over head indices H and tail indices T,

$$f(x) = \mathrm{sgn}\left(v_0 + \sum_{i \in H} v_i x_i + \sum_{i \in T} w_i x_i\right),$$

and let $0 < \epsilon < 1/2$. Assume that H contains all indices i such that $|\widehat{f}(i)| \geq \tau(\epsilon)^2$. Then one of the following holds:

(i) f is $O(\epsilon)$-close to a junta over H (which is a threshold function); or,

(ii) we can normalize the weights so that $\sum_{i \in T} w_i^2 = 1$, in which case f is $O(\epsilon)$-close to the Boolean threshold function

$$f'(x) = \mathrm{sgn}\left(v_0 + \sum_{i \in H} v_i x_i + \sum_{i \in T} \frac{\widehat{f}(i)}{\sigma} x_i\right),$$

where σ denotes $\sqrt{\sum_{i \in T} \widehat{f}(i)^2}$.

We remark that by Parseval's identity, one can take the set $H \subset [n]$ to be the $1/\tau(\epsilon)^4 = \mathrm{poly}(1/\epsilon)$ indices for which $|w_i|$ is largest. Theorem 15 has the following immediate corollary:

Corollary 16. *Under the hypotheses of Theorem 15, there exists a threshold function $f'(x) = \mathrm{sgn}(v_0 + v_1 x_1 + \cdots + v_n x_n)$ which is $O(\epsilon)$-close to f in which $v_i = \widehat{f}(i)$ for all $i \notin H$.*

Proof. In case (i) we can clearly put the junta f' over H into the desired format by scaling the weights $\{v_i\}_{i \in H}$ so large that the weights $\{v_i = \widehat{f}(i)\}_{i \notin H}$ are collectively irrelevant. Otherwise, we are in case (ii) and we can scale all weights by σ. □

In the full version of this paper, we will show that statement (ii) of Theorem 15 in fact *always* holds (assuming $\sigma \neq 0$), even when f is close to a junta.

Theorem 15 suggests an approach to constructing a "small" list of candidate threshold functions for the Chow Parameters problem. We take H to be all indices with Chow Parameter of magnitude at least $\tau(\epsilon)^2$; as mentioned, there are at most $1/\tau(\epsilon)^4$ such indices. If f is close to a junta over H (case (i)), we can construct a list of candidates that will contain such a close-to-f junta by simply enumerating all junta threshold functions over H; intuitively this is a "small" number of candidates since $|H|$ is "small." On the other hand, if we are in case (ii) then simply using the Chow Parameters as the tail weights almost gives us a threshold function which is ϵ-close to f — it remains only to fill in the $|H|$ unknown head weights.

We deal with the unknown head weights via the following extension of Theorem 15, which shows that it is enough to consider head weights with bounded precision within a bounded range:

Theorem 17. *Statement (ii) in Theorem 15 can be replaced by the following:*

(ii) f is $O(\epsilon)$-close to a Boolean threshold function f' of the form

$$f'(x) = \operatorname{sgn}\left(u_0 + \sum_{i\in H} u_i x_i + \sum_{i \in T} \frac{\widehat{f}(i)}{\sigma} x_i\right),$$

where the weights u_i are integer multiples of $\sqrt{\tau(\epsilon)}/|H|$ with magnitude at most $2^{O(|H|\log|H|)}\sqrt{\ln(1/\tau(\epsilon))}$.

Theorem 17 is sufficient if we are given the exact values of the Chow Parameters, but as described earlier we consider the more difficult scenario in which we are only given approximations to the Chow Parameters (this is the scenario required for 1-RFA learning). Thus we want an extension of Theorem 17 which requires only that the input vector be close to the Chow Parameters of f. We prove the following:

Theorem 18. *Theorem 17 continues to hold if, instead of using the vector $\vec{\gamma} = [\widehat{f}(i)]_{i\in T}$ for the (pre-scaled) tail weights, we instead used a vector $\vec{\alpha}$ satisfying*

$$\|\vec{\alpha} - \vec{\gamma}\| \le O(\epsilon^4). \tag{9}$$

Since Theorem 18 is our ultimate goal we prove it directly. We require the following definition:

Definition 19. *Two vectors $\vec{\beta}$ and $\vec{\gamma}$ are η-*approximately parallel *if*

$$\|\vec{\beta}\| \cdot \|\vec{\gamma}\| - \vec{\beta}\cdot\vec{\gamma} \le \eta. \tag{10}$$

Our proof of Theorem 18, given in the full version of the paper, builds on ideas developed in the proof of correctness of the poly$(1/\epsilon)$-query testing algorithm for the class of threshold functions given by [35]. Here is a sketch: Together with geometric arguments that we develop in the full version, the "completeness" analysis of [35] helps us show that if f is far from a junta over H, then all restrictions of the head indices give rise to Chow vectors (of the different restrictions of f) that are mutually approximately parallel. (The completeness argument of [35] also gives us that there is a set of weights for the head indices lying in the required range and with the required precision, that are compatible in a certain technical sense with all the restrictions of the head.) Additional geometric arguments show that the *average* of the Chow vectors of the restrictions — which equals the tail of the Chow vector of f itself — is a "long" vector which is itself approximately parallel to the Chow vectors of the restrictions. Next, these properties, along with the "soundness" analysis of [35], are used to show that replacing the tail weights with the tail Chows of f causes very little error for each restriction to the head indices. Finally, the "compatible" head weights from above are used to obtain an overall high-accuracy approximator for f whose head weights have the stated bounded magnitude and granularity and whose tail weights are the tail Chow parameters of f.

5. PROOF OF THE MAIN THEOREM

We now combine the two ingredients to prove our main result.

Theorem 20. *[Main Theorem restated.] There is a randomized algorithm $\mathcal{A}$ and a function $\kappa(\epsilon) = 2^{-\tilde{O}(1/\epsilon^2)}$ such that the following holds.*

Let $f : \{-1,1\}^n \to \{-1,1\}$ be a threshold function and let $0 < \epsilon < 1/2$. Write $\vec{\chi}$ for the Chow vector of f and assume that $\vec{\alpha}$ is a vector satisfying

$$\|\vec{\alpha} - \vec{\chi}\| \le \kappa(\epsilon). \tag{11}$$

Then given as input $\vec{\alpha}$ and ϵ, algorithm $\mathcal{A}$ performs $2^{\mathrm{poly}(1/\kappa(\epsilon))} \cdot n^2 \cdot \log n \cdot \log(\frac{n}{\delta})$ bit operations and outputs the (weights-based) representation of a threshold function f^* which with probability at least $1-\delta$ satisfies $\mathrm{dist}(f, f^*) \le \epsilon$.

Proof of Main Theorem. We first present a high-level description of the entire algorithm. We then give a more detailed explanation of how the algorithm performs its main step, Step 1, and prove correctness of the algorithm. Finally we analyze the running time.

High-level description of $\mathcal{A}$. Algorithm $\mathcal{A}$ is given $\epsilon > 0$ and the vector $\vec{\alpha}$ as input. The algorithm executes the following steps:

Step 0: Truncate each $\vec{\alpha}(i)$ to an additive accuracy of $\pm\sqrt{\kappa(\epsilon)/(n+1)}$. (Note that this changes the location of $\vec{\alpha}$ by distance at most $\kappa(\epsilon)$, so absorbing the factor of 2 into the definition of $\kappa(\epsilon)$ we have that (11) still holds for the new $\vec{\alpha}$.)

Step 1: Generate a list of $2^{\mathrm{poly}(1/\kappa(\epsilon))}$ "candidate" threshold functions f'. (Details below.)

Step 2: Let $\epsilon_0 = 2^{-\tilde{O}(1/\epsilon^2)}$ be such that in an application of Theorem 6, having $d_{\mathrm{Chow}}(f, f^*) \le 6\sqrt{\epsilon_0}$ implies $\mathrm{dist}(f, f^*) \le \epsilon$. Estimate each of the candidates' Chow vectors to within distance $\sqrt{\epsilon_0}$, and output any f^* whose Chow vector estimate has distance at most $4\sqrt{\epsilon_0}$ from $\vec{\alpha}$.

Detailed explanation of Step 1 and proof of correctness. The way that $\mathcal{A}$ generates the $2^{\mathrm{poly}(1/\kappa(\epsilon))}$ "candidate" threshold functions in Step 1 is based on Theorem 18. Let τ_0 denote $\tau(\epsilon_0)$. The set H in Theorem 18 is taken to be the set of all indices $1 \le i \le n$ for which $|\vec{\alpha}(i)| \ge \tau_0^2/2$. If we now fix $\kappa(\epsilon) = \tau_0^2/2$ (which is indeed $2^{-\tilde{O}(1/\epsilon^2)}$), we are assured that H contains all indices i for which $|\vec{\chi}(i)| = |\widehat{f}(i)| \ge \tau_0^2$, since if H were missing even one such index this would cause $\|\vec{\alpha} - \vec{\chi}\| > \kappa(\epsilon)$ contrary to (11). Note also that $|H| \le O(1/\tau_0^4) = \mathrm{poly}(1/\kappa(\epsilon))$, since $\sum \vec{\alpha}(i)^2 \approx \sum \widehat{f}(i)^2 \le 1$.

Algorithm $\mathcal{A}$ performs Step 1 by generating two sets of candidate threshold functions, corresponding to the two cases in Theorem 18. The first set simply consists of all threshold functions which are juntas over H. Recalling the classic fact [38] that every threshold function over $|H|$ Boolean variables can be represented using integer weights each of magnitude $2^{O(|H|\log|H|)}$, algorithm $\mathcal{A}$ can construct all candidate threshold functions in the first set in time $2^{O(|H|^2\log|H|)} = 2^{\mathrm{poly}(1/\kappa(\epsilon))}$ by simply creating a candidate from each possible vector of integer weights in this range. The second set of candidates consists of all threshold functions whose "head weights" (for indices in H) are integer multiples of $\sqrt{\tau_0}/|H|$ with magnitude at most $2^{O(|H|\log|H|)}\sqrt{\ln(1/\tau_0)}$ and whose "tail weights" (for indices in $T = [n] \setminus H$) are given by $\vec{\alpha}/\|\vec{\alpha}\|$. It is not difficult to see that there are again at most $2^{\mathrm{poly}(1/\kappa(\epsilon))}$ such candidates.

By Theorem 18, at least one of the two sets of candidates contains a threshold function f' which has $\mathrm{dist}(f, f') \le \epsilon_0$. (This uses the fact that as required by statement (ii) of Theorem 18, we indeed have $\|\vec{\alpha}-\vec{\chi}\| \le \kappa(\epsilon) \le \Omega(\epsilon_0^4)$.) By Proposition 5 this f' also satisfies $d_{\mathrm{Chow}}(f, f') \le 2\sqrt{\epsilon_0}$; writing $\vec{\chi}'$ for the Chow vector of f', the triangle inequality implies

$$\|\vec{\alpha} - \vec{\chi}'\| \le \|\vec{\alpha} - \vec{\chi}\| + \|\vec{\chi} - \vec{\chi}'\| \le 3\sqrt{\epsilon_0}$$

(this uses the fact that $\kappa(\epsilon)$ is smaller than $\sqrt{\epsilon_0}$).

To conclude the proof of correctness, we now observe that since Step 2 estimates the Chow vector of each candidate to within distance $\sqrt{\epsilon_0}$, there must indeed be at least one candidate f^* whose Chow vector estimate has distance at most $4\sqrt{\epsilon_0}$ from $\vec{\alpha}$. So f^*'s true Chow vector has distance at most $5\sqrt{\epsilon_0}$ from $\vec{\alpha}$, and the triangle inequality implies $d_{\mathrm{Chow}}(f, f^*) \le 6\sqrt{\epsilon_0}$ (again using $\kappa(\epsilon) \le \sqrt{\epsilon_0}$). Now Theorem 6 implies $\mathrm{dist}(f, f^*) \le \epsilon$, as desired. This concludes the proof of correctness.

Because of space constraints we give the formal running time analysis in the full paper and confine ourselves here to a few words of intuition for the running time bound. One exponential factor in $1/\epsilon$ comes from the quantitative loss incurred by going from closeness of Chow parameters to closeness of functions via Theorem 6: in order to be sure that a candidate g has $\mathrm{dist}(f, g) \le \epsilon$ we must have $d_{\mathrm{Chow}}(f, g) \le \kappa(\epsilon)$. But in order for Theorem 18 to ensure that some candidate g has $d_{\mathrm{Chow}}(f, g) \le \kappa(\epsilon)$, we must take $|H| = \mathrm{poly}(1/\kappa(\epsilon))$, and consequently there are $2^{\mathrm{poly}(1/\kappa(\epsilon))}$ many candidate settings of weights for the variables in H. This is how the doubly-exponential dependence in $1/\epsilon$ arises. The quadratic dependence on n is because for each candidate, there are $n + 1$ Chow parameters that must each be estimated to additive accuracy $\pm 1/\sqrt{n}$ (ignoring the dependence on ϵ). □

6. APPLICATIONS TO LEARNING THEORY

As we now explain, our main theorem has a range of interesting consequences in learning theory.

6.1 Learning Threshold Functions in the 1-RFA Model

We briefly recall the 1-RFA model that was introduced by Ben-David and Dichterman [3] to model the phenomenon of a learner having incomplete access to examples. In this model there is a target function f and a distribution $\mathcal{D}$ over n-bit examples. Each time the learner is about to receive a labeled example she specifies an index $1 \le i \le n$, then an n-bit string x is drawn from the distribution $\mathcal{D}$ and the learner is given $(x_i, f(x))$, i.e. she is only shown the i-th bit of the example along with the label. It is not difficult to show [6] that it is information-theoretically impossible to learn threshold functions in the 1-RFA model if the distribution $\mathcal{D}$ is allowed to be arbitrary. Thus, attention shifted to the uniform distribution setting in which $\mathcal{D}$ is uniform over $\{-1, 1\}^n$.

[6] showed that a sample of $O(nW^2\log(\frac{n}{\delta})/\epsilon^2)$ many examples is information-theoretically sufficient for learning an unknown threshold function with integer weights w_i that satisfy $\sum_i |w_i| \le W$. For constant ϵ, the results of Goldberg [18] and Servedio [44] mentioned in Section 3 respectively yield $n^{O(\log n)}$ and $\mathrm{poly}(n)$ sample complexity bounds for learning arbitrary threshold functions. However, no efficient algorithms were proposed to accompany any of these information-theoretic bounds.

[6] asked whether there is an efficient uniform-distribution 1-RFA learning algorithm for threshold functions.[3] For constant ϵ, our Main Theorem gives an affirmative answer: each of the $n+1$ Chow Parameters ($\mathbf{E}[f(x)x_i]$ or $\mathbf{E}[f(x)]$) can be empirically estimated in the 1-RFA model, so it is straightforward to construct an approximation $\vec{\alpha}$ to the Chow vector $\vec{\chi}_f$ of f as required by our Main Theorem. Since the running time of the algorithm $\mathcal{A}$ dominates the time required to construct $\vec{\alpha}$, we have:

THEOREM 21. *There is an algorithm which properly learns threshold functions to accuracy ϵ and confidence $1 - \delta$ in the uniform distribution 1-RFA model. The algorithm performs $2^{2^{\tilde{O}(1/\epsilon^2)}} \cdot n^2 \cdot \log n \cdot \log(\frac{n}{\delta})$ bit operations.*

6.2 A Fast Agnostic-Type Learning Algorithm for Halfspaces Under the Uniform Distribution

The agnostic learning model was introduced by Kearns *et al.* in 1994 [29], but quite recently there has been considerable progress in both positive and negative results on agnostically learning threshold functions. Let $\mathcal{D}$ be a distribution over $\{-1, 1\}^n$ and let $g : \{-1, 1\}^n \to \{-1, 1\}$ be an arbitrary Boolean function. We write opt to denote the optimal error rate of any threshold function for approximating g with respect to $\mathcal{D}$, i.e.

$$\mathsf{opt} \overset{\mathrm{def}}{=} \min_f \mathbf{Pr}_{x\sim D}[f(x) \neq g(x)]$$

where the min is taken over all threshold functions f. An algorithm which, for any g and any $\mathcal{D}$, constructs a hypothesis h which has

$$\mathbf{Pr}_{x\sim D}[h(x) \neq g(x)] \le \mathsf{opt} + \epsilon \tag{12}$$

is said to be an *agnostic* learning algorithm for threshold functions.

Positive results. Kalai *et al.* gave a uniform distribution agnostic learning algorithm for threshold functions [26]: if $\mathcal{D}$ is the uniform distribution over $\{-1, 1\}^n$, their algorithm outputs a hypothesis h which satisfies (12) as desired. However, the hypothesis that the algorithm constructs is of the form $\mathrm{sgn}(p(x))$ where $p(x)$ is a polynomial of degree $O(1/\epsilon^4)$, so the algorithm is not proper since it does not

[3] More precisely, they explicitly asked whether there is a *proper* learning algorithm, i.e. one which constructs a threshold function as its hypothesis; our algorithm is of course proper.

output a threshold function. Perhaps more significantly, the running time of their algorithm is $n^{O(1/\epsilon^4)}$.

Negative results. Results of Klivans and Sherstov [30] and Feldman *et al.* [15] show that under plausible cryptographic hardness assumptions, there is no polynomial-time algorithm that can agnostically learn threshold functions under arbitrary distributions. [15] also showed that complexity-theoretic assumptions rule out even a very weak form of *proper* agnostic learning for threshold functions. More precisely, they showed that for any constant $\epsilon > 0$, if $P \neq NP$ then there is no algorithm which, given a data set of labeled examples (x, y) (where each $x \in \mathbb{Q}^n$) that has $\mathsf{opt} = 1 - \epsilon$, outputs a threshold function hypothesis that agrees with $\frac{1}{2} + \epsilon$ fraction of the labeled examples. Guruswami and Raghavendra [21] proved that this result holds even if the data points x each belong to the Boolean cube $\{-1, 1\}^n$.

Our results. As we now show, the tools we have developed quite directly yield a very fast agnostic-type uniform distribution learning algorithm for threshold functions. We call our algorithm "agnostic-type" instead of agnostic because the hypothesis it constructs is guaranteed to have error at most $O(\mathsf{opt}^{\Omega(1)}) + \epsilon$ instead of $\mathsf{opt} + \epsilon$.[4] However, our algorithm has some significant advantages to offset this drawback: chief among these is its running time, which is $\tilde{O}(n^2)$ for any constant ϵ. So for example, if $\mathsf{opt} > 0$ is a sufficiently small constant then our algorithm can construct a hypothesis with error rate 0.01 in time $\tilde{O}(n^2)$, while to construct a similarly accurate hypothesis the [26] algorithm would need running time something like n^{10^8}. We also note that our algorithm constructs a threshold function hypothesis and hence is *proper*; this is in contrast with the [26] algorithm. Indeed, it is interesting to observe that the result of [21] shows that (assuming P ≠ NP) no analogue of our algorithm with a similar performance guarantee can exist for learning under arbitrary distributions $\mathcal{D}$ over $\{-1, 1\}^n$.

Theorem 22. *There is an algorithm $\mathcal{B}$ with the following performance guarantee:*

Let g be any Boolean function and let $\mathsf{opt} = \min_f \mathbf{Pr}[f(x) \neq g(x)]$ where the min is over all threshold functions and the probability is uniform over $\{-1, 1\}^n$.

Given an input parameter $\epsilon > 0$ and access to independent uniform examples $(x, g(x))$, algorithm $\mathcal{B}$ outputs the (weights-based) representation of a threshold function f^* which with probability at least $1 - \delta$ satisfies $\mathbf{Pr}[h(x) \neq g(x)] \leq O(\mathsf{opt}^{\Omega(1)}) + \epsilon$. The algorithm performs

$$\text{poly}(1/\epsilon) \cdot n^2 \cdot \log(\tfrac{n}{\delta}) \;+\; 2^{\text{poly}(1/\epsilon)} \cdot n \cdot \log n \cdot \log(\tfrac{1}{\delta})$$

bit operations.

The algorithm and analysis (given in the full paper) are similar to Algorithm $\mathcal{A}$ from Section 5, but slightly simpler since we do not need to estimate Chow Parameters and use Theorem 6 to gauge the accuracy of each candidate – instead we can just directly estimate the empirical accuracy of each candidate using random examples. This is what enables the algorithm to save an exponential in the dependence on ϵ compared with the running time of Algorithm $\mathcal{A}$, and also a $\log n$ factor since we do not have to take a union bound over all $n + 1$ estimated Chow Parameters of each candidate.

6.3 A Fast Uniform-Distribution PAC Learning Algorithm for Halfspaces

The usual (noise-free) uniform distribution PAC learning model corresponds to the special case of the agnostic model in which the target function g is required to actually be a threshold function, i.e. $\mathsf{opt} = 0$. Theorem 22 thus immediately gives us an algorithm that can PAC learn threshold functions in the usual (noise-free) uniform distribution model in the stated time bound.

We observe that for constant ϵ the running time of this algorithm is close to optimal even in this noise-free scenario. Known information-theoretic lower bounds [4, 31] imply that *any* algorithm that learns threshold functions to fixed constant accuracy (say $\epsilon = 0.01$) under the uniform distribution must use $\Omega(n)$ labeled examples; this is true even if the algorithm is allowed to make membership queries. Thus the information-theoretic minimum input length that is required for this problem is $\Omega(n^2)$ bits — this is very close to the $O(n^2 \log n)$ bit operations our algorithm performs. As far as we are aware, the previous fastest known algorithm for learning threshold functions under the uniform distribution on $\{-1, 1\}^n$ would require using linear programming and require [49] at least $O(n^{4.5})$ bit operations (more precisely, $\tilde{O}(n^{3.5})$ arithmetic operations on $\tilde{O}(n)$-bit operands).

7. REFERENCES

[1] H. Aziz, M. Paterson, and D. Leech. Efficient algorithm for designing weighted voting games. Technical Report CS-RR-434, University of Warwick, 2007.

[2] J. Banzhaf. Weighted voting doesn't work: A mathematical analysis. *Rutgers Law Review*, 19:317–343, 1965.

[3] S. Ben-David and E. Dichterman. Learning with restricted focus of attention. *Journal of Computer and System Sciences*, 56(3):277–298, 1998.

[4] G. Benedek and A. Itai. Learnability with respecct to fixed distributions. *Theor. Comput. Sci.*, 86(2):377–390, 1991.

[5] S. Berg and D. Lepelley. On probability models in voting theory. *Statistica Neerlandica*, 48:133–146, 1994.

[6] A. Birkendorf, E. Dichterman, J. Jackson, N. Klasner, and H. Simon. On restricted-focus-of-attention learnability of Boolean functions. *Machine Learning*, 30:89–123, 1998.

[7] A. Blum, A. Kalai, and H. Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, 2003.

[8] J. Bruck. Harmonic analysis of polynomial threshold functions. *SIAM Journal on Discrete Mathematics*, 3(2):168–177, 1990.

[9] F. Carreras. On the design of voting games. *Mathematical Methods of Operations Research*, 59(3):503–515, 2004.

[10] C. Chow. On the characterization of threshold functions. In *Proceedings of the Symposium on Switching Circuit Theory and Logical Design (FOCS)*, pages 34–38, 1961.

[11] M. Dertouzos. *Threshold logic: a synthesis approach.* MIT Press, Cambridge, MA, 1965.

[12] P. Dubey and L. Shapley. Mathematical properties of the banzhaf power index. *Mathematics of Operations Research*, 4:99–131, 1979.

[4] We remark here that [26] in fact show that achieving $\mathsf{opt}+\epsilon$ accuracy in time $n^{1/\epsilon^{2-\kappa}}$ for any constant $\kappa > 0$ would imply a very substantial improvement in the fastest known algorithms for the challenging problem of learning parity with noise: in particular, this would give an algorithm running in time $2^{n^{1-\kappa'}}$, improving on the current $2^{n/\log n}$ runtime of [7]. We feel that this motivates research into algorithms which, like the one we present, have higher error rates but faster runtimes.

[13] E. Einy and E. Lehrer. Regular simple games. *International Journal of Game Theory*, 18:195–207, 1989.

[14] C. Elgot. Truth functions realizable by single threshold organs. In *Proceedings of the Symposium on Switching Circuit Theory and Logical Design (FOCS)*, pages 225–245, 1960.

[15] V. Feldman, P. Gopalan, S. Khot, and A. Ponnuswami. New results for learning noisy parities and halfspaces. In *Proceedings of the 47th Annual Symposium on Foundations of Computer Science*, pages 563–576, 2006.

[16] D. Felsenthal and M. Machover. A priori voting power: what is it all about? *Political Studies Review*, 2(1):1–23, 2004.

[17] J. Freixas. Different ways to represent weighted majority games. *Top (Journal of the Spanish Society of Statistics and Operations Research)*, 5(2):201–212, 1997.

[18] P. Goldberg. A Bound on the Precision Required to Estimate a Boolean Perceptron from its Average Satisfying Assignment. *SIAM Journal on Discrete Mathematics*, 20:328–343, 2006.

[19] M. Goldmann, J. Håstad, and A. Razborov. Majority gates vs. general weighted threshold gates. *Computational Complexity*, 2:277–300, 1992.

[20] M. Goldmann and M. Karpinski. Simulating threshold circuits by majority circuits. *SIAM Journal on Computing*, 27(1):230–246, 1998.

[21] V. Guruswami and P. Raghavendra. Hardness of learning halfspaces with noise. In *FOCS 2006*. IEEE, Oct. 2006.

[22] J. Håstad. On the size of weights for threshold gates. *SIAM Journal on Discrete Mathematics*, 7(3):484–492, 1994.

[23] T. Hofmeister. A note on the simulation of exponential threshold weights. In *Computing and Combinatorics, Second Annual International Conference (COCOON)*, pages 136–141, 1996.

[24] S. Hu. *Threshold Logic*. University of California Press, 1965.

[25] J. Isbell. A Counterexample in Weighted Majority Games. *Proceedings of the AMS*, 20(2):590–592, 1969.

[26] A. Kalai, A. Klivans, Y. Mansour, and R. Servedio. Agnostically learning halfspaces. In *Proceedings of the 46th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 11–20, 2005.

[27] K. Kaplan and R. Winder. Chebyshev approximation and threshold functions. *IEEE Trans. Electronic Computers*, EC-14:315–325, 1965.

[28] P. Kaszerman. A geometric test-synthesis procedure for a threshold device. *Information and Control*, 6(4):381–398, 1963.

[29] M. Kearns, R. Schapire, and L. Sellie. Toward Efficient Agnostic Learning. *Machine Learning*, 17(2/3):115–141, 1994.

[30] A. R. Klivans and A. A. Sherstov. Cryptographic hardness for learning intersections of halfspaces. In *FOCS*, pages 553–562. IEEE Computer Society, 2006.

[31] S. Kulkarni, S. Mitter, and J. Tsitsiklis. Active learning using arbitrary binary valued queries. *Machine Learning*, 11:23–35, 1993.

[32] E. Lapidot. The counting vector of a simple game. *Proceedings of the AMS*, 31:228–231, 1972.

[33] D. Leech. Power indices as an aid to institutional design: the generalised apportionment problem. In M. Holler, H.Kliemt, D. Schmidtchen, and M. Streit, editors, *Yearbook on New Political Economy*, 2003.

[34] P. Lewis and C. Coates. *Threshold Logic*. New York, Wiley, 1967.

[35] K. Matulef, R. O'Donnell, R. Rubinfeld, and R. Servedio. Testing Halfspaces. Manuscript. Available at http://www.cs.columbia.edu/~/rocco/papers/testltf.html. Submitted to the ECCC on November 9, 2007.

[36] S. Muroga. *Threshold logic and its applications.* Wiley-Interscience, New York, 1971.

[37] S. Muroga, I. Toda, and M. Kondo. Majority decision functions of up to six variables. *Math. Comput.*, 16:459–472, 1962.

[38] S. Muroga, I. Toda, and S. Takasu. Theory of majority switching elements. *J. Franklin Institute*, 271:376–418, 1961.

[39] S. Muroga, T. Tsuboi, and C. Baugh. Enumeration of threshold functions of eight variables. Technical Report 245, Univ. of Illinois, Urbana, 1967.

[40] J. V. Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

[41] L. Penrose. The elementary statistics of majority voting. *Journal of the Royal Statistical Society*, 109(1):53–57, 1946.

[42] A. Razborov. On small depth threshold circuits. In *Proceedings of the Third Scandinavian Workshop on Algorithm Theory (SWAT)*, pages 42–52, 1992.

[43] V. Roychowdhury, K.-Y. Siu, A. Orlitsky, and T. Kailath. Vector analysis of threshold functions. *Information and Computation*, 120(1):22–31, 1995.

[44] R. Servedio. Every linear threshold function has a low-weight approximator. In *Proceedings of the 21st Conference on Computational Complexity (CCC)*, pages 18–30, 2006.

[45] Q. Sheng. *Threshold Logic*. London, New York, Academic Press, 1969.

[46] K. Takamiya and A. Tanaka. Computational complexity in the design of voting games. Technical Report 653, The Institute of Social and Economic Research, Osaka University, 2006.

[47] M. Tannenbaum. The establishment of a unique representation for a linearly separable function. Technical report, Lockheed Missiles and Space Co., 1961. Threshold Switching Techniques Note 20, pp. 1-5.

[48] A. Taylor and W. Zwicker. A Characterization of Weighted Voting. *Proceedings of the AMS*, 115(4):1089–1094, 1992.

[49] P. Vaidya. A new algorithm for minimizing convex functions over convex sets. In *Proceedings of the Thirtheth Symposium on Foundations of Computer Science*, pages 338–343, 1989.

[50] R. Winder. Threshold logic in artificial intelligence. *Artificial Intelligence*, IEEE Publication S-142:107–128, 1963.

[51] R. Winder. Threshold functions through $n = 7$. Technical Report 7, Air Force Cambridge Research Laboratories, 1964.

[52] R. Winder. Threshold gate approximations based on chow parameters. *IEEE Transactions on Computers*, pages 372–375, 1969.

[53] R. Winder. Chow parameters in threshold logic. *Journal of the ACM*, 18(2):265–289, 1971.