Electronic Colloquium on Computational Complexity, Report No. 96 (2013)

Exponentially improved algorithms and lower bounds for testing signed majorities

Dana Ron* Rocco A. Servedio†

Abstract

A *signed majority function* is a linear threshold function $f : \{+1,-1\}^n \to \{+1,-1\}$ of the form $f(x) = \mathrm{sign}(\sum_{i=1}^n \sigma_i x_i)$ where each $\sigma_i \in \{+1,-1\}$. Signed majority functions are a highly symmetrical subclass of the class of all linear threshold functions, which are functions of the form $\mathrm{sign}(\sum_{i=1}^n w_i x_i - \theta)$ for arbitrary real w_i, θ.

We study the query complexity of testing whether an unknown $f : \{+1,-1\}^n \to \{+1,-1\}$ is a signed majority function versus ϵ-far from every signed majority function. While it is known [26] that the broader class of all linear threshold functions is testable with $\mathrm{poly}(1/\epsilon)$ queries (independent of n), prior to our work the best upper bound for signed majority functions was $O(\sqrt{n}) \cdot \mathrm{poly}(1/\epsilon)$ queries (via a non-adaptive algorithm), and the best lower bound was $\Omega(\log n)$ queries for non-adaptive algorithms [27].

As our main results we exponentially improve both these prior bounds for testing signed majority functions:

- (Upper bound) We give a $\mathrm{poly}(\log n, 1/\epsilon)$-query adaptive algorithm (which is computationally efficient) for this testing problem;
- (Lower bound) We show that any non-adaptive algorithm for testing the class of signed majorities to constant accuracy must make $n^{\Omega(1)}$ queries. This directly implies a lower bound of $\Omega(\log n)$ queries for any adaptive algorithm.

Our testing algorithm performs a sequence of restrictions together with consistency checks to ensure that each successive restriction is "compatible" with the function prior to restriction. This approach is used to transform the original n-variable testing problem into a testing problem over $\mathrm{poly}(\log n, 1/\epsilon)$ variables where a simple direct method can be applied. Analysis of the degree-1 Fourier coefficients plays an important role in our proofs.

*danaron@post.tau.ac.il, Tel Aviv University. Supported by ISF grant number 246/08.

†rocco@cs.columbia.edu, Columbia University. Supported by NSF grants CCF-0915929 and CCF-1115703.

ISSN 1433-8092

1 Introduction

1.1 Background and motivation

Over the last few decades property testing has emerged as an important line of research in sublinear-time algorithms, with close connections to related research topics such as learning theory, PCPs, coding theory, computational geometry, and more. The goal in property testing is to determine whether an unknown "massive object" has a particular property while inspecting only a tiny portion of the object. Given this ambitious goal, the success criterion required of such algorithms is necessarily an approximate one: a testing algorithm should accept if the object has the property in question, and should reject if the object is far from every object with the property. Many different types of "massive objects" have been studied from this property testing perspective, including graphs, (code)words (for certain error-correcting codes), sets of points in a metric space, probability distributions, and (last but not least) Boolean functions, which are the subject of this work. For surveys see [13, 31, 32, 17].

Some of the earliest and best-known results in property testing, such as the well-studied "linearity test" of Blum et al. [6], deal with testing Boolean functions. Any property of Boolean functions can be equated with the class of all functions that have the property (for example, the linearity test for Boolean functions corresponds to the class of all parity functions). Over the past decade testing algorithms and lower bounds have been given for many natural classes of Boolean functions, such as monotone Boolean functions [18, 11, 15, 1, 22, 7], literals, conjunctions and s-term monotone DNF [30], juntas [14, 3, 4], general s-term DNF, size-s decision trees, and size-s circuits [10], small-width OBDDs [33, 16, 8], low-degree $GF(2)$ polynomials [2, 24, 23], functions with sparse or low-degree Fourier representations [20, 10] (real polynomial representations over domain $\{+1,-1\}^n$), and more.

Most of the classes described above have either a "logical/combinatorial" flavor (such as DNF formulas and decision trees) or an "algebraic" flavor (such as low-degree $GF(2)$ polynomials and functions with low-degree or sparse Fourier representations). An exception is the work of [26], which gives a testing algorithm for a class of Boolean functions with a natural *geometric* definition, namely the class of all *linear threshold functions* (LTFs). An LTF is a function $f : \{+1,-1\}^n \to \{+1,-1\}$ defined by $f(x) = \mathrm{sign}(\sum_{i=1}^n w_i x_i - \theta)$ for some real values $w_1, \dots, w_n, \theta$, i.e. its output is determined by whether the input $x \in \{+1,-1\}^n$ lies on one side of a hyperplane in $\mathbb{R}^n$. The main result of [26] is a $\mathrm{poly}(1/\epsilon)$-query algorithm (independent of n) that tests whether any $f : \{+1,-1\}^n \to \{+1,-1\}$ is an LTF versus ϵ-far from every LTF.

Subsequent to [26], [27] studied the testability of a natural subclass of LTFs, namely the class of *signed majority functions*. These are LTFs in which each weight w_1 is either $+1$ or -1 and the threshold θ is 0. Equivalently, a signed majority function is computed by a Maj gate with n inputs where the i-th input is either the literal x_i or $\neg x_i$. Signed majority functions are of interest for several reasons: for one thing, they are arguably the simplest and most symmetrical LTFs that depend on all n input variables. They have the largest total influence (equivalently, edge boundary) of all LTFs (or even of all unate functions), and they have a natural interpretation as fair voting systems where each voter has equal weight and one of two opposing orientations (such as Liberal/Conservative).

Perhaps surprisingly, the class of signed majority functions is provably harder to test than the general class of all LTFs. [27] gave an $O(\sqrt{n}) \cdot \mathrm{poly}(1/\epsilon)$-query non-adaptive testing algorithm for the class of signed majorities, and proved an $\Omega(\log n)$-query lower bound for non-adaptive algorithms. Thus, while [27] showed that signed majorities are indeed harder to test than general LTFs, their upper and lower bounds for this class were exponentially far apart.

1.2 Our results: Near-optimal algorithms and lower bounds for testing signed majorities

Our main positive result improves the query complexity of the [27] algorithm by an exponential factor (by utilizing adaptivity), and our main negative result improves the [26] lower bound by an exponential factor. We thus give upper and lower bounds for testing signed majority functions which are tight up to polynomial factors (in terms of the dependence on n).

In more detail, our main positive result is the following:

Theorem 1 *There is a* $\mathrm{poly}(\log n, 1/\epsilon)$*-query adaptive algorithm for testing whether an arbitrary and unknown* $f : \{+1,-1\}^n \rightarrow \{+1,-1\}$ *is a signed majority function versus* ϵ*-far from every signed majority function. The algorithm runs in poly*$(n, 1/\epsilon)$ *time.*

Our main lower bound result shows that any nonadaptive algorithm must use exponentially more queries (as a function of n) than our adaptive algorithm:

Theorem 2 *Any non-adaptive algorithm for testing whether an unknown black-box* $f : \{+1,-1\}^n \rightarrow \{+1,-1\}$ *is a signed majority function versus* $\Theta(1)$*-far from every signed majority function must make* $n^{\Omega(1)}$ *queries.*

It is well-known that that the nonadaptive query complexity of any testing problem for Boolean functions is at most exponentially larger than the adaptive query complexity (since a nonadaptive algorithm can simply carry out all possible executions that an adaptive algorithm would perform given any possible sequences of answers to its queries). Consequently Theorem 2 implies that any adaptive algorithm for $\Theta(1)$-testing $\mathcal{SMAJ}$ must make $\Omega(\log n)$ queries, and hence the $\mathrm{poly}(\log n, 1/\epsilon)$ query complexity of Theorem 1 is optimal up to polynomial factors.

1.3 A high-level discussion of the ideas behind our algorithm

We first briefly discuss the approach taken in [27] for testing signed majority functions, and then present the techniques we apply, which give us an exponential improvement in the dependence on n.

It is well known that if $f : \{+1,-1\}^n \rightarrow \{+1,-1\}$ is a signed majority function over n variables, then all degree-1 Fourier coefficients of f, which we denote by $\widehat{f}(1), \ldots, \widehat{f}(n)$ have the same absolute value, which we denote by $\widehat{M}(n) = \Theta(1/\sqrt{n})$ (see Section 2.2). On the other hand, Matulef et al. [27] show that if f is ϵ-far from every signed majority function then at least an $\Omega(\epsilon^2)$-fraction of the degree-1 Fourier coefficients of f are smaller than $(1-\Omega(\epsilon^2))\widehat{M}(n)$ in magnitude. Hence, their algorithm works by selecting $\Theta(1/\epsilon^2)$ indices $i \in [n]$ uniformly at random, and for each i selected, determining (with high probability) whether $|\widehat{f}(i)| < (1-\Omega(\epsilon^2))\widehat{M}(n)$, in which case the algorithm rejects. The latter task is performed by taking a sample of size $\Theta\left(\frac{\log(1/\epsilon)}{\epsilon^4 \widehat{M}(n)}\right) = \Theta\left(\frac{\log(1/\epsilon)\sqrt{n}}{\epsilon^4}\right)$ so as to estimate $|\widehat{f}(i)|$. The linear dependence on $1/\widehat{M}(n) = \Theta(\sqrt{n})$ seems inherent in this approach since the degree-1 Fourier coefficients of f may indeed be very close to $\pm\widehat{M}(n)$ so that obtaining a good estimate of them requires a sample of size $\Omega(1/\widehat{M}(n))$.

How can we avoid the polynomial dependence on n? Roughly speaking[1], we show that if f is ϵ-far from every signed majority function, then one of the following holds: (1) We can find (statistical) evidence

[1] In particular, in all that follows we make statements that hold with high probability, without explicitly stating this.

to the fact that f is not a signed majority function without having to consider individual degree-1 Fourier coefficients but rather by considering the degree-1 Fourier coefficients "collectively" (by estimating the sum of certain powers of these coefficients); or (2) We can find a *very* large Fourier coefficient (i.e. of size $\mathrm{poly}(\epsilon/\log(n))$), which gives evidence that f is not a signed majority function; or (3) We can find a restriction f' of f such that f' is defined over approximately $n/2$ variables and f' is ϵ'-far (for ϵ' very close to ϵ) from every signed majority function (over the appropriate number of variables). In the first two cases we are done. In the third case we continue iteratively, working with f'.

In each iteration we either find evidence that the current function h (which is a restriction of f) is not a restriction of any signed majority function (recall that we are considering the case in which f is far from every signed majority function), or we obtain a restriction h' of h on approximately half the number of variables, where h' is still far from every signed majority function. This condition on h' is ensured by performing certain consistency checks relating the degree-1 Fourier coefficients of h to those of h'. The iterative process stops once the function we are dealing with has only $\mathrm{poly}(\log n, 1/\epsilon)$ input variables , at which point we can afford to run an "end-stage" algorithm with polynomial dependence on the number of input variables.

For the completeness argument (the case in which f is a signed majority function), it is clear that any restriction of f is a *signed threshold function* (see Definition 4). Our restrictions are constructed in such a way that if h is a signed threshold function with a small-magnitude threshold, then the restriction h' is also such a function. This ensures that the degree-1 Fourier coefficients of the restriction will behave appropriately, and that at each iteration the restriction h' will pass the consistency check with h. Finally, once we have only $\mathrm{poly}(\log n, 1/\epsilon)$ input variables, the restricted function is close to a signed majority function over those input variables (since it is a signed threshold function with a small-magnitude threshold), and the "end-stage" algorithm accepts.

2 Preliminaries

2.1 Basic definitions and the class of functions we are interested in

We consider functions whose domain is $\{+1,-1\}^n$ and whose range is $\{+1,-1\}$. We let $[n] = \{1,\ldots,n\}$.

Definition 1 (Distance between functions) *The* distance *between two functions* $f, g : \{+1,-1\}^n \to \{+1,-1\}$, *denoted* $\mathrm{dist}(f,g)$, *equals* $\Pr[f(x) \neq g(x)]$, *where the probability is taken over* x *that is selected uniformly from* $\{+1,-1\}^n$. *We say that* f *and* g *are* ϵ-far *from each other if* $\mathrm{dist}(f,g) > \epsilon$, *otherwise they are* ϵ-close. *For a family of functions* F_n *(from* $\{+1,-1\}^n$ *to* $\{+1,-1\}$*), we let* $\mathrm{dist}(f, F_n) \stackrel{\mathrm{def}}{=} \min_{g\in F_n}\{\mathrm{dist}(f,g)\}$, *and we say that* f *is* ϵ*-far from* F_n *if* $\mathrm{dist}(f,F_n) > \epsilon$ *(otherwise* f *is* ϵ*-close to* F_n*).*

Definition 2 (Property Testing) *For* $n \geq 1$ *let* F_n *denote a family of functions from* $\{+1,-1\}^n$ *to* $\{+1,-1\}$. *A* Property Testing *algorithm for (membership in) a family of functions* $\mathcal{F} = \bigcup_{n\geq 1} F_n$ *is given* n, *a distance parameter* $\epsilon > 0$, *and black-box query access to an unknown function* $f : \{+1,-1\}^n \to \{+1,-1\}$. *If* $f \in F_n$ *then the algorithm should* accept *with probability at least* $2/3$, *and if* f *is* ϵ*-far from* F_n, *then it should* reject *with probability at least* $2/3$.

A property testing algorithm is *non-adaptive* if it selects all its query strings (possibly using randomness) before making any queries; it is *adaptive* if for some $j > 1$ the j-th query made by the algorithm depends on the response received from the black-box oracle to previous queries.

The class that we shall be interested in testing is the class of all *signed majority functions*.

Definition 3 (Signed Majority Functions) *Fix $n \geq 1$. For each $\sigma \in \{+1,-1\}^n$, the* Signed Majority Function Maj_σ *is defined by* $\mathrm{Maj}_\sigma(x) = \mathrm{sign}\left(\sum_{i=1}^n \sigma_i x_i\right)$ *(where* $\mathrm{sign}(y) = +1$ *if* $y \geq 0$ *and* $\mathrm{sign}(y) = -1$ *otherwise). We denote the family of all 2^n signed majority functions over $\{+1,-1\}^n$ by SMAJ_n, and we write $\mathcal{SMAJ}$ to denote $\bigcup_{n \geq 1} \mathrm{SMAJ}_n$.*

For the sake of readability we will often suppress the dependence on n and simply write SMAJ to denote the set of all 2^n signed majority functions over $\{+1,-1\}^n$. In the analysis of our testing algorithm for SMAJ, we will have occasion to consider the following generalization of signed majority functions.

Definition 4 (Signed Threshold Functions) *Fix $n \geq 1$. For each integer $\theta \in [-n, n]$, and for each $\sigma \in \{+1,-1\}^n$, the* Signed Threshold Function $\mathrm{Thr}_\sigma^\theta$ *is defined by* $\mathrm{Thr}_\sigma^\theta(x) = \mathrm{sign}\left(\sum_{i=1}^n \sigma_i x_i - \theta\right)$. *We denote the family of n-variable signed threshold functions with threshold θ by by STHR_n^θ.*

As with signed majority functions, for readability we will often suppress the "n" and simply write STHR^θ for the class of all n-variable signed threshold functions with threshold θ.

Notation for substrings and restrictions. For $u \in \{+1,-1\}^m$ and $S \subseteq [m]$ we write $u_{|S}$ to denote the length-$|S|$ string consisting of u restricted to the coordinates in S. We shall use the notation $\{+1,-1\}^S$ for a set $S \subset [m]$ to denote length-$|S|$ substrings that are indexed by indices $i \in S$. For $h : \{+1,-1\}^m \rightarrow \{+1,-1\}$, $S \subseteq [m]$ and $y \in \{+1,-1\}^S$ we write $h_{S \leftarrow y}$ to denote the restriction of f obtained by fixing the coordinates in S according to y.

2.2 Fourier coefficients

See [29, 9] for excellent surveys on Fourier analysis over $\{+1,-1\}^n$; here we state only the very basics which we require.

Definition 5 (Fourier Coefficients) *For each subset $S \subseteq [n]$, the function $\chi_S : \{+1,-1\}^n \rightarrow \{+1,-1\}$ is defined by $\chi_S(x) \stackrel{\mathrm{def}}{=} \prod_{i \in S} x_i$. Given a function $f : \{+1,-1\}^n \rightarrow \{+1,-1\}$ we define its* Fourier coefficients *by* $\widehat{f}(S) \stackrel{\mathrm{def}}{=} \mathrm{E}[f(x)\chi_S(x)]$, *where the expectation is taken over a uniformly selected x, and we have that $f(x) = \sum_{S \subseteq [n]} \widehat{f}(S)\chi_S(x)$.*

We shall be interested especially in the *degree-1* Fourier coefficients $\widehat{f}(S)$ for $|S| = 1$; with a slight abuse of notation we use the notation $\widehat{f}(i)$ instead of $\widehat{f}(\{i\})$ for such coefficients. For $x \in \{+1,-1\}^n$, let $x^{+i} = (x_1, \ldots, x_{i-1}, +1, x_{i+1}, \ldots, x_n)$ and $x^{-i} = (x_1, \ldots, x_{i-1}, -1, x_{i+1}, \ldots, x_n)$. Observe that for each $i \in [n]$ we have that

$$\widehat{f}(i) \;=\; \mathrm{E}[f(x) \cdot x_i] \;=\; \Pr\left[f(x^{+i}) = +1 \;\&\; f(x^{-i}) = -1\right] - \Pr\left[f(x^{+i}) = -1 \;\&\; f(x^{-i}) = +1\right] \;. \quad (1)$$

In particular, if $f \in \mathrm{SMAJ}$ (that is, $f = \mathrm{Maj}_\sigma$ for some $\sigma \in \{+1,-1\}^n$), then for each $i \in [n]$ we have

$$\widehat{f}(i) = \sigma_i \cdot \Pr\left[f(x^{+i}) \neq f(x^{-i})\right] = \sigma_i \cdot \widehat{M}(n) \;, \quad (2)$$

where $\widehat{M}(n)$ denotes the value of the degree-1 Fourier coefficient $\widehat{\mathrm{Maj}}(i)$ of the n-variable majority function $\mathrm{sign}(x_1 + \cdots + x_n)$ (this value is the same for all $i \in [n]$). A well-known consequence of Stirling's formula (see [35]) is that $\widehat{M}(n) = \sqrt{\frac{2}{\pi n}} - \frac{c_n}{n^{3/2}}$, where $\frac{1}{9} \leq c_n \leq \frac{1}{8}$. We shall also use the notation $\widetilde{M}(n) \stackrel{\mathrm{def}}{=} n \cdot (\widehat{M}(n))^2$ so that

$$\widetilde{M}(n) = \frac{2}{\pi} - O(n^{-1}). \tag{3}$$

We sometimes refer to $\widetilde{M}(n)$ as "the level-1 Fourier weight of SMAJ."

The following lemma of Matulef et al. [26, Lemma 15] says that it is possible to efficiently estimate certain sums of products of degree-1 Fourier coefficients; we will use this in our testing algorithm.

Lemma 1 ([26]) *Let $p \geq 2$. Suppose we have black-box access to $f_1, \dots, f_p : \{+1, -1\}^n \to \{+1, -1\}$. Then for any $T \subseteq [n]$ we can estimate the sum of products of degree-1 Fourier coefficients*

$$\sum_{i \in T} \widehat{f_1}(i) \cdots \widehat{f_p}(i)$$

to within an additive η, with confidence $1 - \delta$, using $O(p \cdot \log(1/\delta)/\eta^4)$ total calls to the black-box oracles for $f_1, \dots, f_p$.

2.3 Results from probability

We briefly recall some useful bounds from probability theory. The first of these is the following standard Chernoff bound (see Theorem 1.1 of [12]):

Theorem 3 *Let $Y_1, \dots, Y_m$ be m independent random variables which take on values in $[0, 1]$, where $\mathrm{E}[Y_i] = p_i$, and $\sum_{i=1}^m p_i = P$. Then for any $\gamma > 0$, we have the additive bounds*

$$\Pr\left[\sum_{i=1}^n Y_i > P + \gamma m\right] \leq \exp(-2\gamma^2 m), \tag{4}$$

$$\Pr\left[\sum_{i=1}^n Y_i < P - \gamma m\right] \leq \exp(-2\gamma^2 m) \tag{5}$$

and the multiplicative bounds

$$\Pr\left[\sum_{i=1}^m Y_i > (1 + \gamma)P\right] < \exp(-\gamma^2 P/3) \tag{6}$$

and

$$\Pr\left[\sum_{i=1}^m Y_i < (1 - \gamma)P\right] < \exp(-\gamma^2 P/2). \tag{7}$$

Note that by an appropriate shifting and scaling it is possible to apply the above Chernoff bounds to sums of independent random variables which take on values in any interval $[a, b]$, and in particular we often apply the bounds to the interval $[-1, 1]$.

We will also use the Berry-Esséen theorem, which is a version of the Central Limit Theorem with explicit error bounds:

Theorem 4 *(Berry-Esséen) Let $X_1, \ldots, X_n$ be a sequence of independent random variables satisfying $\mathrm{E}[X_i] = 0$ for all i, $\sqrt{\sum_i \mathrm{E}[X_i^2]} = \sigma$, and $\sum_i \mathrm{E}[|X_i|^3] = \rho_3$. Let $S = (X_1 + \cdots + X_n)/\sigma$ and let F denote the cumulative distribution function (cdf) of S. Then*

$$\sup_x |F(x) - \Phi(x)| \leq C\rho_3/\sigma^3, \tag{8}$$

where Φ is the cdf of a standard Gaussian random variable (with mean zero and variance one), and C is a universal constant. It is known [34] that one can take $C = .7915$.

3 Useful Technical Claims

We note that in our technical claims about functions that are ϵ-far from SMAJ we will always be dealing with the case that $\epsilon \geq 1/n^{c_0}$ where $c_0 > 0$ is a small (unspecified) absolute constant. This is because for $\epsilon < 1/n^{c_0}$ it is possible to non-adaptively test the class SMAJ using $\mathrm{poly}(1/\epsilon)$ queries and $\mathrm{poly}(1/\epsilon)$ time since $\mathrm{poly}(1/\epsilon) = \mathrm{poly}(n, 1/\epsilon)$ in this case.[2]

The following technical lemma states that if f is far from every signed majority function and f does not have "too little" Fourier weight at level 1, then the level-1 Fourier coefficients of f must be far from those of every signed majority function. The proof of the lemma is similar to the proof of [26, Theorem 34].

Lemma 2 *Let $\epsilon \geq 1/n^{c_0}$. If $f : \{+1,-1\}^n \to \{+1,-1\}$ is ϵ-far from SMAJ and $\sum_{i=1}^n (\widehat{f}(i))^2 \geq \widetilde{M}(n) - \gamma$, then for every $\sigma \in \{+1,-1\}^n$ we have that*

$$\sum_{i=1}^n \left(\widehat{f}(i) - \sigma_i \cdot \widehat{M}(n)\right)^2 > \frac{7\epsilon^2}{32} - \gamma \ . \tag{9}$$

Proof: We first observe that the left-hand-side of Equation (9) is minimized when $\sigma_i = \mathrm{sign}(\widehat{f}(i))$, and hence it suffices to prove the inequality for this setting of σ, which we denote by σ^f. For this setting we have:

$$\begin{aligned} \sum_{i=1}^n \left(\widehat{f}(i) - \sigma_i^f \cdot \widehat{M}(n)\right)^2 &= \sum_{i=1}^n \left(\widehat{f}(i)\right)^2 + \widetilde{M}(n) - 2\sum_{i=1}^n |\widehat{f}(i)| \cdot \widehat{M}(n) \\ &\geq 2\left(\widetilde{M}(n) - \widehat{M}(n) \cdot \sum_{i=1}^n |\widehat{f}(i)|\right) - \gamma \ , \end{aligned} \tag{10}$$

where we have used the premise of the lemma concerning $\sum_{i=1}^n \left(\widehat{f}(i)\right)^2$.

Let $w_i^f \stackrel{\mathrm{def}}{=} \sigma_i^f \cdot \widehat{M}(n)$ and consider the linear function $\ell^f(x) \stackrel{\mathrm{def}}{=} \sum_i w_i^f \cdot x_i$. We next relate the right-hand-side of Equation (10) to this linear function. First, we observe that

$$\mathrm{E}\left[\left|\ell^f(x)\right|\right] = \mathrm{E}\left[\left|\sum_{i=1}^n w_i^f \cdot x_i\right|\right] = \widehat{M}(n) \cdot \mathrm{E}\left[\left|\sum_{i=1}^n x_i\right|\right] \leq \widetilde{M}(n) + O(n^{-1/2}) \ . \tag{11}$$

[2] This can be done either by running the $O(\sqrt{n} \cdot \mathrm{poly}(1/\epsilon))$-queries algorithm of [27] (which runs in $\mathrm{poly}(n, 1/\epsilon)$ time), or by simply running a proper learning algorithm with time and query complexity $\mathrm{poly}(n, 1/\epsilon)$ for the class SMAJ and applying the well-known result that the query complexity of testing a class of functions is essentially upper bounded by the query complexity of proper learning the class [Prop. 3.1.1][19]. To properly learn a function $f \in$ SMAJ, i.e., $f = \mathrm{Maj}_\sigma$ for $\sigma \in \{+1,-1\}^n$, it suffices to find σ_i for each $i \in [n]$ and this can easily be done by performing $\mathrm{poly}(n)$ queries.

The final equality above is by symmetry of $\{+1,-1\}^n$, and the inequality can be obtained either by using Proposition 32 of [26] (which gives a fairly simple proof using the Berry-Esséen theorem) or by using Theorem 2 of [25] (which has a more involved proof providing a sharper constant in the low-order term, which we will not need here). On the other hand, we also have that

$$\mathrm{E}\left[f(x)\cdot \ell^f(x)\right] \;=\; \mathrm{E}\left[f(x)\cdot \sum_{i=1}^{n} w_i^f \cdot x_i\right] \;=\; \sum_{i=1}^{n} w_i^f \cdot \widehat{f}(i) \;=\; \widehat{M}(n)\cdot \sum_{i=1}^{n} |\widehat{f}(i)|\;, \tag{12}$$

where in the second equality we applied the definition of $\widehat{f}(i)$, and in the third equality we applied the definition of w_i^f. By combining Equation (10) with Equations (11) and (12) we get that

$$\sum_{i=1}^{n}\left(\widehat{f}(i) - \sigma_i^f \cdot \widehat{M}(n)\right)^2 \;\geq\; 2\left(\mathrm{E}[|\ell^f(x)|] - \mathrm{E}[f(x)\cdot \ell^f(x)]\right) - \gamma - O(n^{-1/2})\;. \tag{13}$$

Since the range of f is $\{+1,-1\}$, we have that

$$\begin{aligned}\mathrm{E}[|\ell^f(x)|] - \mathrm{E}[f(x)\cdot \ell^f(x)] \;&=\; 2^{-n}\cdot \sum_{x: f(x)\neq \mathrm{sign}(\ell^f(x))} 2|\ell^f(x)| \\ &\geq\; \frac{\epsilon}{8}\cdot\left(\Pr[f(x)\neq \mathrm{sign}(\ell^f(x))] - \Pr[2|\ell^f(x)|<\epsilon/8]\right)\;. \end{aligned}\tag{14}$$

Since f is ϵ-far from SMAJ, it is in particular ϵ-far from $\mathrm{sign}(\ell^f)$, so we have $\Pr[f(x)\neq \mathrm{sign}(\ell^f(x))] > \epsilon$. On the other hand, we have

$$\Pr\left[2|\ell^f(x)|<\epsilon/8\right] \;=\; \Pr\left[\left|\sum_{i=1}^{n} x_i\right| < \frac{\epsilon}{16}\cdot\frac{1}{\widehat{M}(n)}\right] \;\leq\; \frac{\epsilon}{16} + O(n^{-1/2}), \tag{15}$$

where the equality is by the definition of $\ell^f(x)$ and the symmetry of $\{+1,-1\}^n$, and the first inequality follows from the Berry-Esséen Theorem. We thus get that the right-hand-side of Equation (14) is lower bounded by $\frac{\epsilon}{8}\left(\frac{15\epsilon}{16} - O(n^{-1/2})\right)$, and the lemma follows by combining this inequality with Equation (13) and recalling the assumed lower bound on ϵ. ■

We next prove a simple claim concerning binomial coefficients.

Claim 3 *For any two integers m and j such that $m>0$ and $0\leq j\leq \sqrt{m}$, when m is even we have*

$$\left(1-\frac{2j^2}{m}\right)\binom{m}{m/2} \;\leq\; \binom{m}{m/2+j} \;\leq\; \binom{m}{m/2}\;, \tag{16}$$

and when m is odd we have

$$\left(1-\frac{2j(j+1)}{m}\right)\binom{m}{\lceil m/2\rceil} \;\leq\; \binom{m}{\lceil m/2\rceil+j} \;\leq\; \binom{m}{\lceil m/2\rceil}\;. \tag{17}$$

Proof: We prove the claim for even m. The proof for odd m is very similar. For any $k\in[0, m/2-1]$,

$$\begin{aligned}\binom{m}{m/2+k+1} \;&=\; \frac{m/2-k}{m/2+k+1}\cdot\binom{m}{m/2+k} \\ &=\; \frac{m/2+k+1-(2k+1)}{m/2+k+1}\cdot\binom{m}{m/2+k} \\ &\geq\; \left(1-\frac{2(2k+1)}{m}\right)\cdot\binom{m}{m/2+k}\;. \end{aligned}\tag{18}$$

Hence

$$\binom{m}{m/2+j} \;\geq\; \prod_{k=0}^{j-1}\left(1-\frac{2(2k+1)}{m}\right)\cdot\binom{m}{m/2} \;\geq\; \left(1-\frac{2}{m}\sum_{k=0}^{j-1}(2k+1)\right)\cdot\binom{m}{m/2}\,, \tag{19}$$

and we get that

$$\left(1-\frac{2j^2}{m}\right)\cdot\binom{m}{m/2} \;\leq\; \binom{m}{m/2+j} \;\leq\; \binom{m}{m/2}\,, \tag{20}$$

as claimed. ■

As a corollary of Claim 3 we get:

Lemma 4 *Let $X_1,\ldots,X_m$ be i.i.d. $\{+1,-1\}$-valued random variables with $\Pr[X_i=+1]=1/2$ for all i. For any α,β such that $-1<\alpha<\beta<1$ and where $\alpha\sqrt{m}$ and $\beta\sqrt{m}$ are even integers when m is even and odd integers when m is odd, we have that:*

$$\frac{\beta-\alpha}{6} \leq \Pr\left[\alpha\sqrt{m} \leq \sum_{i=1}^{m} X_i \leq \beta\sqrt{m}\right] \leq \frac{\beta-\alpha}{2} + 1/\sqrt{m}\,. \tag{21}$$

Proof: Let $\overline{X} \overset{\text{def}}{=} \sum_{i=1}^{m} X_i$. Consider the case that m is even (the case that m is odd is very similar). Observe that for any even k we have that $\Pr[\overline{X}=k]=2^{-m}\binom{m}{m/2+k/2}$. Therefore,

$$\Pr\left[\alpha\sqrt{m} \leq \overline{X} \leq \beta\sqrt{m}\right] = 2^{-m}\sum_{j=\alpha\sqrt{m}/2}^{\beta\sqrt{m}/2}\binom{m}{m/2+j}\,. \tag{22}$$

The upper bound on $\Pr\left[\alpha\sqrt{m} \leq \overline{X} \leq \beta\sqrt{m}\right]$ now follows simply because $\binom{m}{m/2+j} \leq \binom{m}{m/2}$ and since $2^{-m}\binom{m}{m/2} = \sqrt{\frac{2}{\pi m}} - O(m^{-3/2})$. For the lower bound we apply (the lower bound in) Claim 3 to get that

$$\Pr\left[\alpha\sqrt{m} \leq \overline{X} \leq \beta\sqrt{m}\right] \;\geq\; 2^{-m}\binom{m}{m/2}\cdot\sum_{j=\alpha\sqrt{m}/2}^{\beta\sqrt{m}/2}\left(1-\frac{2j^2}{m}\right)\,, \tag{23}$$

and we use $2^{-m}\binom{m}{m/2} = \sqrt{\frac{2}{\pi m}} - O(m^{-3/2})$ once again. ■

In the following lemma $\beta>0$ should be thought of as "large" and $\kappa>0$ as "small", and the values for $\alpha>0$ which make the claim non-trivial are essentially those which satisfy $\alpha^2>\kappa/\beta$. Intuitively, the lemma says that if a set of small non-negative weights have average value β, then summing a random subset is extremely likely to yield a value which is very close to β times the number of weights in the subset.

Lemma 5 *Let $0 \leq w_1,\ldots,w_m \in \mathbb{R}$ be such that $w_i \leq \kappa m$ for all $i=1,\ldots,m$ and $\sum_{i=1}^{m} w_i = \beta m$. Let $X_1,\ldots,X_m$ be i.i.d. Bernoulli random variables with $\Pr[X_i=1]=1/2$ for all i, and let $\overline{X} \overset{\text{def}}{=} \sum_{i=1}^{m} X_i$. For any $\alpha\in[0,1]$, with probability at least $1-\exp(-\Omega(\alpha^2\beta/\kappa))$ we have that*

$$\left|\sum_{i=1}^{m} w_i X_i - \beta\overline{X}\right| \leq \alpha\beta\overline{X}. \tag{24}$$

Proof: We define m random variables, $Y_1, \ldots, Y_m$, where $Y_i = (w_i X_i)/(\kappa m)$. By the premise of the lemma, $Y_i \in [0,1]$, and $\mathrm{E}\left[\sum_{i=1}^m Y_i\right] = (\beta/2\kappa)$. By applying a multiplicative Chernoff bound (see Theorem 3),

$$\begin{aligned} \Pr\left[\left|\sum_{i=1}^m Y_i - (\beta/2\kappa)\right| \le \gamma(\beta/2\kappa)\right] &\ge 1 - 2\exp(-(1/3)\gamma^2(\beta/2\kappa)) \\ &= 1 - \exp\left(-\Omega(\gamma^2\beta/\kappa)\right) . \end{aligned} \tag{25}$$

By the definition of Y_i, this implies that

$$\Pr\left[\left|\sum_{i=1}^m w_i X_i - (\beta/2)m\right| \le \gamma(\beta/2)m\right] \ge 1 - \exp\left(-\Omega(\gamma^2\beta/\kappa)\right) . \tag{26}$$

By applying another multiplicative Chernoff bound, this time to $\overline{X} = \sum_{i=1}^m X_i$, we have that

$$\Pr\left[\left|\overline{X} - m/2\right| \le \gamma' m/2\right] \ge 1 - 2\exp(-(1/3)(\gamma')^2 m/2) , \tag{27}$$

which is of course equivalent to

$$\Pr\left[\left|\beta\overline{X} - \beta m/2\right| \le \beta\gamma' m/2\right] \ge 1 - 2\exp(-(1/3)(\gamma')^2 m/2) . \tag{28}$$

We thus have that with probability $1 - \exp\left(-\Omega(\gamma^2\beta/\kappa)\right) - \exp\left(-\Omega((\gamma')^2 m)\right)$,

$$\left|\sum_{i=1}^m w_i X_i - \beta\overline{X}\right| \le (\gamma+\gamma')\beta(m/2) \le \frac{\gamma+\gamma'}{1-\gamma'}\beta\overline{X} . \tag{29}$$

The lemma follows by setting $\gamma = \gamma' = \alpha/3$ and observing that $m \ge \beta/\kappa$ (since $\sum_{i=1}^m w_i = \beta m$ and $w_i \le \kappa m$) so that $\alpha^2 m \ge \alpha^2\beta/\kappa$). ■

We end this section by recording some properties of signed threshold functions (with a small-magnitude threshold) that will be useful later. Since two threshold functions $\mathrm{Thr}_\sigma^\theta$ and $\mathrm{Thr}_\sigma^{\theta'}$ can be equivalent even though $\theta \ne \theta'$, when we write $h = \mathrm{Thr}_\sigma^\theta$, we consider the minimal $|\theta|$ for which h has threshold θ.

Lemma 6 *Let $h : \{+1,-1\}^m \to \{+1,-1\}$, $h = \mathrm{Thr}_\sigma^\theta$ where $|\theta| \le \alpha\sqrt{m}$, $\alpha < 1$, and $\sigma \in \{+1,-1\}^m$.*

1. *For every $i \in [m]$ we have that $(1-\alpha^2/2)\cdot\widehat{M}(m) \le |\widehat{h}(i)| \le \widehat{M}(m)$.*
2. *$|\mathrm{E}[h]| \le \alpha + 1/\sqrt{m}$.*

On the other hand, suppose that $h' : \{+1,-1\}^m \to \{+1,-1\}$, $h' = \mathrm{Thr}_{\sigma'}^{\theta'}$ is a signed threshold function where $|E[h']| \le \gamma$. Then:

3. *$|\theta'| \le 3\gamma\sqrt{m}$.*

Proof: For the first item, assume m is even. The odd case is proved similarly. As noted in the text preceding the lemma, we consider the minimal $|\theta|$ defining h, so that θ is even. Since $h = \mathrm{sign}\left(\sum_{i=1}\sigma_i x_i - \theta\right)$, we have that

$$\widehat{h}(i) = \frac{\binom{m}{m/2+\theta/2}(m/2+\theta/2)}{m2^{m-1}} . \tag{30}$$

Using $\binom{m}{m/2+k} = \binom{m}{m/2-k}$ (since θ might be negative), the first item now follows by applying Claim 3 and using $\widehat{M}(m) = \frac{\binom{m}{m/2}}{2^m}$.

The second and third items follow from Lemma 4 by observing that

$$|\mathrm{E}[h]| \in \left[\Pr\left[\left|\sum_{i=1}^{m}\sigma_i x_i\right| < \theta\right] - 1/\sqrt{m}, \Pr\left[\left|\sum_{i=1}^{m}\sigma_i x_i\right| < \theta\right] + 1/\sqrt{m}\right] \tag{31}$$

and the lemma follows. ■

4 A $\mathrm{poly}(\log n, 1/\epsilon)$-Query Adaptive Testing Algorithm for SMAJ

4.1 A high-level overview and intuition for our approach

As explained in the introduction, our testing algorithm (Algorithm 4) works in an iterative manner. At the start of the j^{th} iteration it holds a function $f^{(j-1)}$ that is a restriction of the tested function f, and is defined over a set of variables $T^{(j-1)} \subseteq [n]$ (where $f^{(0)} = f$ and $T^{(0)} = [n]$ for the first iteration). In the course of the j^{th} iteration, the algorithm selects a restriction $f^{(j)}$ of $f^{(j-1)}$ where the size of the subset of variables over which $f^{(j)}$ is defined, that is, $T^{(j)}$, is a constant fraction of the size of $T^{(j-1)}$. The iterative process ends once $|T^{(j)}|$ goes below a certain threshold that is $\mathrm{poly}(\log n, 1/\epsilon)$, at which point we can afford running a procedure whose query complexity is linear (or even polynomial) in the number of variables.

In each iteration, the restriction is selected by calling a procedure (Algorithm 1) that ensures (with probability at least $1-\delta$) that if $f^{(j-1)}$ is a signed threshold function with a small threshold (so that its bias $|\mathrm{E}[f^{(j-1)}]|$ is small) then the same is true of $f^{(j)}$. If $f^{(0)} = f$ is a signed majority function (i.e., a signed threshold function with 0 threshold), then by calling Algorithm 1 each time with $\delta = O(1/\log n)$ we have that with high probability all restrictions $f^{(j)}$ are signed threshold functions with a small threshold. This is essentially what ensures the completeness of the algorithm, since in each iteration the checks performed on the new restriction $f^{(j)}$ should pass (with high probability) when $f^{(j)}$ is a signed threshold function with a small threshold, and the same is true of the final procedure. The main focus of what follows is to explain what the algorithm does in each iteration (as well as initially and after the last iteration) so as to ensure the algorithm's soundness, where we point out in the appropriate places how this is done without compromising the completeness of the algorithm.

Consider the vector $v^f \stackrel{\mathrm{def}}{=} (\widehat{f}(1), \ldots, \widehat{f}(n))$ of degree-1 Fourier coefficients of f. First recall that by Lemma 2, if f is ϵ-far from SMAJ then either $\|v^f\|_2^2 = \sum_{i=1}^n (\widehat{f}(i))^2$ is significantly smaller (i.e., by $\Omega(\epsilon^2)$) than $\widetilde{M}(n) = n(\widehat{M}(n))^2 \approx 2/\pi$, or $\|v^f - v^{\mathrm{Maj}_\sigma}\|_2^2 = \sum_{i=1}^n \left(\widehat{f}(i) - \sigma_i \widehat{M}(n)\right)^2$ is relatively large (i.e., is $\Omega(\epsilon^2)$) for every $\sigma = \{+1,-1\}^n$. By Lemma 1 (setting $p = 2$ and $f_1 = f_2 = f$), the first case can be detected by performing only $\mathrm{poly}(1/\epsilon)$ queries to f.

However, the first case does not necessarily hold, and $\|v^f\|_2^2$ might actually be very close to "what it should be" (i.e., $\widetilde{M}(n) \approx 2/\pi$). We thus turn to the second case where $\|v^f - v^{\mathrm{Maj}_\sigma}\|_2^2$ is relatively large (for every $\sigma = \{+1,-1\}^n$). Suppose first that there exist few (possibly just one) degree-1 Fourier coefficients $\widehat{f}(i)$ that are relatively large, that is, of the order of $\mathrm{poly}(\epsilon)$ (or even $\mathrm{poly}(\epsilon/\log n)$). In such a case we call a procedure (Algorithm 2) that detects (with high probability) the existence of such a large coefficient (note that if f is a signed majority function, then no such large Fourier coefficient exists).

We are left to deal with the case in which $\|v^f\|_2^2$ is close to $\widetilde{M}(n)$ and $\widehat{f}(i)$ is not very large for every $i \in [n]$. As explained in the foregoing discussion, this implies that $\|v^f - v^{\mathrm{Maj}_\sigma}\|_2^2$ is relatively large for every $\sigma \in \{+1,-1\}^n$ (but no individual $\widehat{f}(i)$ is very large). In this case we show that if we select each i independently with probability $1/2$, then with high probability, the restriction of v^f to the subset of selected indices $T = T^{(1)}$, which we denote by $v^f_{|T}$, is relatively far from every $|T|$-dimensional all-$(\pm\widehat{M}(|T|))$ vector (recall that v^{Maj_σ} is an n-dimensional all-$(\pm\widehat{M}(n))$ vector). The algorithm performs such a selection of T and then finds an assignment y to the subset of remaining variables, $R = R^{(1)}$ (by calling the aforementioned Algorithm 1) thus obtaining a restricted function $f^{(1)} = f_{R\leftarrow y}$.

As noted previously, such an assignment y is selected so as to ensure (with high probability) that when $f \in \mathrm{SMAJ}$, then $f^{(1)}$ is a signed threshold function with a small threshold (though this is immaterial to the case that f is ϵ-far from SMAJ). The algorithm now estimates $\|v^{f^{(1)}}\|_2^2$ as well as appropriately scaled versions of $\|v^f_{|T}\|_2^2$ and $\langle v^{f^{(1)}}, v^f_{|T}\rangle$, and rejects if any of them deviates from $2/\pi$ by more than a small amount. In the analysis we show that this ensures (with high probability) that $\|v^{f^{(1)}} - v^{\mathrm{Maj}_\sigma}\|_2^2$ is relatively large for every $\sigma \in \{+1,-1\}^{|T|}$. The algorithm then calls a procedure for detecting the existence of any large degree-1 Fourier coefficient of $f^{(1)}$, and if no such coefficient is detected, then the algorithm continues to the next iteration with $f^{(1)}$.

Since the number of relevant variables decreases by a factor of approximately $1/2$ in each iteration, after $O(\log n)$ iterations (in which the algorithm does not reject) this number goes below a sufficiently small threshold $s = \mathrm{poly}(\log n, 1/\epsilon)$. At this point, the final function obtained, h, is tested for having any degree-1 Fourier coefficient that deviates by a sufficiently large multiplicative factor $(1 \pm \zeta)$ from $\widehat{M}(k)$, where $k \leq s$ is the number of variables over which h is defined. We show that if f is ϵ-far from SMAJ, then (with high probability over all the iterations), h must have at least one such coefficient. On the other hand, if $f \in \mathrm{SMAJ}$ then (with high probability over all the iterations), all degree-1 Fourier coefficients of h are sufficiently close to $\widehat{M}(k)$. The algorithm can distinguish between the two cases by performing a number of queries to h that is polynomial in k and $1/\epsilon$.

4.2 Some useful subroutines for our testing algorithm

In this subsection we describe several subroutines that are used by our testing algorithm.

4.2.1 Finding small-bias restrictions.

Definition 6 (Biased restrictions) *Let $h : \{+1,-1\}^m \to \{+1,-1\}$. We say that a restriction $h_{S\leftarrow y}$ (where $S \subseteq [m]$, $y \in \{+1,-1\}^S$) is a* γ-bias restriction *of h if $|\mathrm{E}[h_{S\leftarrow y}]| \leq \gamma$.*

As described in the intuitive overview in Section 4.1, our approach involves constructing small-bias restrictions to signed threshold functions that have a small-magnitude threshold. This is done using Algorithm 1. It is given query access to a function $h : \{+1,-1\}^{T\cup R}$ and disjoint sets $T, R \subseteq [n]$ of input variables to h. It either fails, or else outputs an assignment $y \in \{+1,-1\}^R$ to the variables in R. The key property of the algorithm is that if h is a signed threshold function which itself has small bias, then with high probability it outputs an assignment $y \in \{+1,-1\}^R$ such that $h_{R\leftarrow y}$ is similarly a small-bias restriction of h. The algorithm works simply by trying random assignments for y and estimating the bias of $h_{R\leftarrow y}$ for each try (giving up if there are too many unsuccessful tries).

Algorithm 1: Procedure for finding small-bias restrictions

```
Input: query access to a function $h : \{+1,-1\}^{T \cup R} \to \{+1,-1\}$ over a set of variables $T \cup R$ and
       parameters $\delta$ and $\gamma$
  1. Repeat the following at most $s = \Theta(\log(1/\delta)/\gamma)$ times:
     (a) Select $y \in \{+1,-1\}^R$ uniformly at random and let $h' : \{+1,-1\}^T \to \{+1,-1\}$ be
         $h' := h_{R \leftarrow y}$.
     (b) Estimate $\mathrm{E}[h']$ to within an additive error of $\gamma$ with confidence $1 - \delta/(4s)$ (by performing
         $\Theta(\log(s/\delta)/(\gamma)^2)$ queries to $h'$), and denote the estimate by $\mu$.
     (c) If $|\mu| \le 2\gamma$ then return $y$.
  2. Return failure (this step is reached if $|\mu| > 2\gamma$ in all $s$ repetitions above).
```

Lemma 7 *Let $h : \{+1,-1\}^{T \cup R}$ be a signed threshold function satisfying $|\mathrm{E}[h]| \le 3\gamma$ where $T, R \subset [n]$ are disjoint sets that satisfy $\frac{1}{3}|T \cup R| \le |T|, |R| \le \frac{2}{3}|T \cup R|$. Suppose that $|T \cup R| \ge C/\gamma^2$ (for a sufficiently large absolute constant C). Then*

1. *Algorithm 1 makes $O\left(\frac{\log(1/\delta)\log(1/(\delta\gamma))}{\gamma^3}\right)$ queries to h.*
2. *With probability at least $1-\delta$, Algorithm 1 outputs an assignment y for which $h_{R \leftarrow y}$ is a (3γ)-bias restriction of h.*

Proof: Part (1) is immediate from inspection of the algorithm.

We now turn to part (2). First observe that given the sample size used in Step 1b of the algorithm, by an additive Chernoff bound (see Theorem 3) for each fixed choice of y selected in Step 1a, with probability at least $1 - \delta/(2s)$ (for an appropriate constant in the sample size $\Theta(\cdot)$ notation) we have $|\mathrm{E}[h_{R \leftarrow y}] - \mu| < \gamma$. By taking a union bound over the (at most s) choices of y, we have that with probability at least $1 - \delta/2$ all estimates μ are within γ of their expected value. Assume from this point on that this is indeed the case. This implies that the algorithm will not return an assignment y for which $|\mathrm{E}[h_{R \leftarrow y}]| > 3\gamma$. It remains to show that with probability at least $1 - \delta/2$, for some estimate μ we have that $|\mu| \le 2\gamma$ so that the corresponding assignment y is returned. To this end we show that the probability over a single choice of y that $|\mathrm{E}[h_{R \leftarrow y}]| \le \gamma$ is $\Omega(\gamma)$, implying that the probability that no y is selected in $s = \Theta(\log(1/\delta)/\gamma)$ iterations of Step 1 is at most $\delta/2$.

By the premise of the lemma, h is a $|T \cup R|$-variable signed threshold function $h = \mathrm{Thr}_{\sigma^h}^{\theta^h}$, with $|\mathrm{E}[h]| \le 3\gamma$, where $|T \cup R| \ge C/\gamma^2$ for some large constant C. By Part 3 of Lemma 6 we have that the threshold θ^h satisfies $|\theta^h| \le 9\gamma\sqrt{|T \cup R|}$. By the definition of h we have that $h(x) = \mathrm{sign}\left(\sum_{i \in T} \sigma_i^h x_i + \sum_{i \in R} \sigma_i^h x_i - \theta^h\right)$. For any assignment y to R, let $\tau^h(y) = \sum_{i \in R} \sigma_i^h y_i$. Thus, for $h' = h_{R \leftarrow y}$ we have that h' is a signed threshold function (over the variables in T) with threshold $\theta^{h'} = \theta^h - \tau^h(y)$. It remains to lower bound (by $\Omega(\gamma)$) the probability (over the choice of y) that $|\theta^{h'}| \le \gamma\sqrt{|T|}$.

The constraint $|\theta^{h'}| \le \gamma\sqrt{|T|}$, that is, $-\gamma\sqrt{|T|} \le \theta^{h'} \le \gamma\sqrt{|T|}$, is equivalent to $\theta^h - \gamma\sqrt{|T|} \le \tau^h(y) \le \theta^h + \gamma\sqrt{|T|}$. Since $|\theta^h| \le 3\gamma\sqrt{|T \cup R|}$, $\gamma \ge \sqrt{C}/\sqrt{|T \cup R|}$, and $|R|, |T| \ge |T \cup R|/3$, we can apply Lemma 4, and get that $\Pr[|\theta^{h'}| \le \gamma\sqrt{|T|}] = \Omega(\gamma)$. ■

4.2.2 Detecting the existence of large degree-1 Fourier coefficients.

Recall that by Lemma 6, if h is a signed threshold over m variables with a small-magnitude threshold, then $|\widehat{h}(i)| \leq \widehat{M}(m) < \sqrt{\frac{2}{\pi m}}$ for every $i \in [m]$. Hence, if given query access to a function h over m variables we can detect that $|\widehat{h}(i)|$ is significantly larger than $\sqrt{\frac{2}{\pi m}}$ for some $i \in [m]$, then we have evidence that h is not such a signed threshold function. In this subsection we present a simple procedure (see Algorithm 2) that (roughly) performs such a detection. The precise claim regarding Algorithm 2 is stated in Lemma 8. We note that the procedure is called by the testing algorithm (Algorithm 4) on restrictions h of the tested function f, i.e., $h = f_{R \leftarrow y}$ for some $R \subset [n]$ and $y \in \{+1,-1\}^T$. Thus, each such h is defined over $\{+1,-1\}^T$ for some $T = [n] \setminus R$. By reordering the variables, we may view h as being over $\{+1,-1\}^m$.

```
Algorithm 2: Procedure for detecting large Fourier coefficients
  Input: query access to a function $h : \{+1,-1\}^m \to \{+1,-1\}$ and parameters $\delta > 0$ and
         $\gamma > C\sqrt{(\log m)/m}$ for some large constant $C$
    1. Let $Z_1, \ldots, Z_t$ be an arbitrary partition of $[m]$ into $t = \Theta(\log(1/\gamma)/\gamma^2)$ disjoint subsets, each of size
       $\lfloor m/t \rfloor$ or $\lceil m/t \rceil$ (which is $\Theta((\gamma^2/\log(1/\gamma))m)$).
    2. For $j = 1$ to $t$ do:
       (a) Initialize $d_j := 0$ and repeat the following $s = \Theta(\log(t/\delta)/\gamma^2)$ times:
           i. Select an assignment $y$ to the variables in $[m] \setminus Z_j$ uniformly at random, and two
              assignments, $z$ and $z'$ to $Z_j$ uniformly at random.
           ii. If $h_{[m]\setminus Z_j \leftarrow y}(z) \neq h_{[m]\setminus Z_j \leftarrow y}(z')$ then $d_j := d_j + 1$.
       (b) If $d_j \geq (\gamma/4)s$, then return fail (and exit).
    3. Return pass (this step is reached if $d_j < (\gamma/4)s$ for all $j$).
```

Lemma 8 *Let $h : \{+1,-1\}^m \to \{+1,-1\}$ and $\gamma \geq C\sqrt{(\log m)/m}$ for some large enough constant C.*

1. *Algorithm 2 makes $O\left(\frac{\log(1/(\delta\gamma))\log(1/\gamma)}{\gamma^4}\right)$ queries to h.*
2. *If there exists $i \in [m]$ such that $|\widehat{h}(i)| \geq \gamma$ then with probability at least $1-\delta$ Algorithm 2 returns* fail, *and if h is a signed threshold function then with probability at least $1-\delta$ Algorithm 2 returns* pass.

In order to prove Lemma 8 we recall the well-studied notion of the *variation* of a set of variables (see e.g., [14]).

Definition 7 (Variation) *For a function $h : \{+1,-1\}^m \to \{+1,-1\}$ and a set of variables $Z \subset [m]$ the* variation *of the set Z with respect to h, denoted $\mathrm{Vr}^h(Z)$, is*

$$\Pr_{y,z,z'}\left[h_{[m]\setminus Z \leftarrow y}(z) \neq h_{[m]\setminus Z \leftarrow y}(z')\right]$$

where y is selected uniformly at random in $\{+1,-1\}^{[m]\setminus Z}$ and z, z' are selected uniformly at random in $\{+1,-1\}^Z$. When $Z = \{i\}$, we use the shorthand notation $\mathrm{Vr}^h(i)$ for $\mathrm{Vr}^h(\{i\})$.

By the definition of the degree-1 Fourier coefficients (see Equation (1)), we have that $\mathrm{Vr}^h(i) \geq |\widehat{h}(i)|/2$, where the factor of $1/2$ is due to the event that $z = z'$ for the assignment to variable i. We shall use the fact that the variation is monotone, so that in particular, for every nonempty subset Z of variables, and every variable i in Z, we have that $\mathrm{Vr}^h(Z) \geq \mathrm{Vr}^h(i)$.

Proof of Lemma 8: Part (1) is immediate from inspection of the algorithm.

In order to prove part (2), first observe that what the algorithm does is simply estimate the variation of each subset Z_j. Given the setting of the sample size s, by an additive Chernoff bound and a union bound over all t subsets we have that with probability at least $1 - \delta$ all $j \in [t]$ satisfy $|d_j/s - \mathrm{Vr}^h(Z_j)| \leq \gamma/8$. Assume from this point on that this is in fact the case. It remains to verify that if $|\widehat{h}(i)| \geq \gamma$ for some $i \in [m]$ then $\mathrm{Vr}^h(Z_j) \geq \gamma/2$ for some subset Z_j, while if h is a signed threshold function, then $\mathrm{Vr}^h(Z_j) \leq \gamma/8$ for every subset Z_j.

The first statement follows directly from the monotonicity of the variation by considering the subset Z_j that contains i. To establish the second statement, we write $h(x) = \mathrm{sign}\left(\sum_{i \in Z_j} \sigma_i x_i + \sum_{i \in [m] \setminus Z_j} \sigma_i x_i - \theta\right)$. For an assignment y to $[m] \setminus Z_j$ let $\tau^h(y) = \sum_{i \in [m] \setminus Z_j} \sigma_i y_i$, and similarly define $\tau^h(z)$ for an assignment z to Z_j. If $h_{[m] \setminus Z \leftarrow y}(z) \neq h_{[m] \setminus Z \leftarrow y}(z')$ then this means that $\tau^h(y) + \tau^h(z) - \theta$ has a different sign from $\tau^h(y) + \tau^h(z') - \theta$. Since the probability (taken over the random choice of y, z and z') of such an event decreases with $|\theta|$ we may consider the case $\theta = 0$. Let $m' = m - |Z_j| = (1 - \tilde{\Theta}(\gamma^2))m$. For any choice of $\beta \in (-1, 1)$

$$\begin{aligned} &\Pr\left[\mathrm{sign}(\tau^h(y) + \tau^h(z)) \neq \mathrm{sign}(\tau^h(y) + \tau^h(z'))\right] \\ &\leq \Pr\left[|\tau^h(y)| \leq \beta\sqrt{m'}\right] + \Pr\left[|\tau^h(z)| > \beta\sqrt{m'}\right] + \Pr\left[|\tau^h(z')| > \beta\sqrt{m'}\right] . \end{aligned} \tag{32}$$

By Lemma 4, the first term on the right-hand-side of Equation (32) is upper bounded by $O(\beta)$. Taking β to be a sufficiently small constant multiple of γ, this term is at most $\gamma/16$. Observing that $\sqrt{m'} \geq \sqrt{|Z_j| \log(1/\gamma)}/(\gamma/c)$ for some sufficiently large constant $c > 1$, by an additive Chernoff bound, the second and third terms are upper bounded by $\gamma/32$ each, and the lemma follows. ■

4.2.3 Dealing with functions on few variables.

As described in Section 4.1, our algorithm works by successively fixing more and more variables, until eventually we are working with a restriction f' of the original function f which has only $k = \mathrm{poly}(\log n, 1/\epsilon)$ input variables left unrestricted. At this point the algorithm checks whether all the degree-1 Fourier coefficients of f' are sufficiently close in magnitude to $\widehat{M}(k)$ (as is the case when f' is a small-bias signed threshold function). The procedure for performing this task is Algorithm 3, and the following lemma states the properties of this algorithm. (As in the case of Algorithm 2, Algorithm 3 is called on a restriction $h : \{+1, -1\}^T$ of the tested function f, but for simplicity we let $T = [k]$.)

Lemma 9 *Given query access to $h : \{+1, -1\}^k \rightarrow \{+1, -1\}$ and $\zeta \in (0, 1)$,*

1. *Algorithm 3 makes $O\left(\frac{k \log k}{\zeta^2}\right)$ queries to h.*

2. *If* $|\widehat{h}(i)| \in \left[(1-\zeta)\widehat{M}(k), (1+\zeta)\widehat{M}(k)\right]$ *for every* $i \in [k]$*, then Algorithm 3 accepts with probability at least* $9/10$*, and if* $|\widehat{h}(i)| \notin \left[(1-2\zeta)\widehat{M}(i), (1+2\zeta)\widehat{M}(i)\right]$ *for some* $i \in [k]$*, then Algorithm 3 rejects with probability at least* $9/10$.

Proof: The bound on the query complexity (the first part of the lemma) follows immediately by inspection of the algorithm.

For the second part, consider any specific degree-1 Fourier coefficient $\widehat{h}(i)$ of h and recall that $\widehat{h}(i) = \mathrm{E}[h(x)x_i]$. By an additive Chernoff bound, given that the sample size s satisfies $s = \Theta(k \log k/\zeta^2))$, the probability that $|\tilde{h}_i - \widehat{h}(i)| > \zeta/(4\sqrt{k})$ is at most $1/(10k)$. By taking a union bound over all $i \in [k]$, with probability at least $9/10$ we have that $|\tilde{h}_i - \widehat{h}(i)| \leq \zeta/(4\sqrt{k})$ for every $i \in [k]$. The lemma now follows since $1/(4\sqrt{k}) < \widehat{M}(k)/2$. ■

Algorithm 3: Procedure for detecting deviation of Fourier coefficients from $\pm\widehat{M}(k)$

```
**Input**: query access to a function $h : \{+1,-1\}^k \to \{+1,-1\}$ and parameter $\zeta \in (0,1)$.
    1. Draw $s = \Theta(k\log(k)/\zeta^2)$ strings $x^1, \ldots, x^s$ independently and uniformly from $\{+1,-1\}^k$ and query $h(x^j)$ for each $j \in [s]$.
    2. For each $i \in [k]$ let $\tilde{h}_i := \frac{\sum_{j=1}^{s} h(x^j)x_i^j}{s}$.
    3. If for any $i \in [k]$ it holds that $|\tilde{h}_i| \notin [(1-3\zeta/2)\cdot\widehat{M}(k), (1+3\zeta/2\cdot\widehat{M}(k)]$ then return reject, otherwise return accept.
```

4.3 The Algorithm

In this subsection we present our algorithm for testing signed majority functions – Algorithm 4. Lemmas 10 and 11 establish completeness and soundness of the testing algorithm, respectively.

Lemma 10 (Completeness) *If* $f \in \mathrm{SMAJ}$*, then Algorithm 4 accepts with probability at least* $2/3$.

Proof: By the premise of the lemma $f = \mathrm{Maj}_{\sigma^f}$ for some $\sigma^f \in \{+1,-1\}^n$. We consider a collection of failure events (whose total probability is at most $1/3$) and show that if none of the failure events occurs then the algorithm accepts.

We first note that the algorithm may reject in Step 5b if the random subset $T^{(j)}$ of $T^{(j-1)}$ ever has size smaller than $\frac{1}{3}|T^{(j)}|$ or larger than $\frac{2}{3}|T^{(j)}|$. Since Step 5b is only performed when $|T^{(j-1)}| \geq s$, at each iteration of Step 5b the probability of such an "unbalanced split" is at most $1/n^{\omega(1)}$ by a standard additive Chernoff bound. Given that the splits are all roughly balanced (between $\frac{1}{3}$ and $\frac{2}{3}$), the "while" loop of Step 5 is carried out at most $O(\log n)$ times, and thus the probability that the algorithm ever outputs reject and exits in Step 5b is at most $1/n^{\omega(1)}$.

We next observe that the algorithm computes various estimates in Steps 2 and 5d, on a total of $O(\log n)$ occasions. The probability that any of these estimates deviates by more than the allowed error from the correct value is $O(\delta \cdot \log(n))$, which is at most $1/20$. The algorithm also performs $O(\log n)$ calls to

Algorithm 4: Testing Algorithm for Signed Majority

```
1. Let $\delta := \Theta(1/\log n)$, $s := \Theta((\log n)^4(\log\log n)(\log(\log n/\epsilon))/\epsilon^4)$, $\gamma_1 := \Theta(\epsilon^2/(\log n)^2)$,
   $\gamma_2 := \Theta(\epsilon^2/((\log n)^2\sqrt{\log\log n}))$, $\gamma_3 := \Theta(\epsilon/\log n)$.

2. Obtain an estimate of $\sum_{i=1}^{n}(\widehat{f}(i))^2$ to within additive error $\gamma_1$ and with confidence $1-\delta$ by applying
   Lemma 1. If this estimate deviates from $\widetilde{M}(n)$ by more than $\pm\gamma_1$, then output reject (and exit).

3. Call Algorithm 2 with query access to $f$ and parameters $\delta$ and $\gamma = \gamma_2$. If it returns pass then
   continue, otherwise output reject (and exit).

4. Initialize $j := 0$, $f^0 := f$, $T^0 := [n]$, $t^{(0)} = n$.

5. While $t^j \geq s$ do:

   (a) Increment $j := j + 1$.
   (b) Select a subset $T^{(j)} \subseteq T^{(j-1)}$ by independently putting each $i \in T^{(j-1)}$ into $T^{(j)}$ with
       probability $1/2$. Set $t^{(j)} := |T^{(j)}|$ and $R^{(j)} := T^{(j-1)} \setminus T^{(j)}$. If $t^{(j)} \notin [\frac{1}{3}t^{(j-1)}, \frac{2}{3}t^{(j-1)}]$ then
       output reject and exit.
   (c) Call Algorithm 1 with query access to $f^{(j-1)}$, the pair of sets of variables $T^{(j)}$ and $R^{(j)}$ and the
       parameters $\delta$ and $\gamma = \gamma_3$. If it returns failure then output reject (and exit). Otherwise, let $y^{(j)}$
       be the assignment it returns to $R^{(j)}$ and set $f^{(j)} := f^{(j-1)}_{R^{(j)}\leftarrow y^{(j)}}$.
   (d) Let (i) $\phi^{(j)}_{j-1}$ be $\left(\widehat{M}(t^{(j)})/\widehat{M}(t^{(j-1)})\right)^2$ times an estimate of $\sum_{i\in T^{(j)}}\left(\widehat{f^{(j-1)}}(i)\right)^2$;
           (ii) $\phi^{(j)}_{j}$ be an estimate of $\sum_{i\in T^{(j)}}\left(\widehat{f^{(j)}}(i)\right)^2$; and
           (iii) $\phi^{(j)}_{j-1,j}$ be $\widehat{M}(t^{(j)})/\widehat{M}(t^{(j-1)})$ times an estimate of $\sum_{i\in T^{(j)}}\widehat{f^{(j-1)}}(i)\cdot\widehat{f^{(j)}}(i)$.
       All estimates are obtained via Lemma 1 with additive error $\gamma_1$ and confidence $1-\delta$. If any one
       of $\phi^{(j)}_{j-1}$, $\phi^{(j)}_{j}$, or $\phi^{(j)}_{j-1,j}$ deviates from $2/\pi$ by more than $2\gamma_1$, then output reject (and exit).
   (e) Call Algorithm 2 with query access to $f^{(j)}$ and parameters $\delta$ and $\gamma = \gamma_2$. If it returns pass then
       continue, otherwise output reject (and exit).

6. Let $\ell = j$ be the index of the last iteration of the "while" loop (Step 5). Run Algorithm 3 with query
   access to $f^{(\ell)}$ and $\zeta = \epsilon/8$, and output what it outputs.
```

Algorithm 2. Each call is with a restriction of f (which is a signed threshold function because $f \in$ SMAJ) over at least $s/3$ variables, and parameters δ and γ_2. By Lemma 8 (note that the relation between γ_2 and the number of variables is as required), the probability that any call rejects is $O(\delta \cdot \log(n))$, which is at most $1/20$. For the rest of this proof we may assume that all estimates are indeed as required and that all calls to Algorithm 2 return pass. (In particular the algorithm does not reject in Step 2 or Step 3 and reaches the while loop in Step 5.)

Since $f \in$ SMAJ, by the definition of $f^{(j)}$ it holds that $f^{(j)} \in \mathrm{STHR}^{\theta^{(j)}}$ for all j. For $j = 0$ we have that $\theta^{(0)} = 0$, so $|\mathrm{E}[f^{(j)}]| \leq 3\gamma_3$ as required by Lemma 7. Indeed, Lemma 7 gives that if $|\mathrm{E}[f^{(j-1)}]| \leq 3\gamma_3$ then $f^{(j)}$ (defined in Step 5c) also satisfies $|\mathrm{E}[f^{(j)}]| \leq 3\gamma_3$ except with failure probability at most $\delta =$

$\Theta(1/\log n)$. Since Step 5c is executed $O(\log n)$ times, the probability that the algorithm ever outputs reject in Step 5c is at most $1/10$.

At this point we have established that we may assume that at each iteration of Step 5d, both $f^{(j-1)}$ and $f^{(j)}$ are signed threshold functions, with thresholds $\theta^{(j-1)}$ and $\theta^{(j)}$ respectively, which satisfy $|\mathrm{E}[f^{(j-1)}]|, |\mathrm{E}[f^{(j)}]| \le 3\gamma_3$. Part (3) of Lemma 6 implies that $|\theta^{(j-1)}| = O(\gamma_3\sqrt{t^{(j-1)}})$ and $|\theta^{(j)}| = O(\gamma_3\sqrt{t^{(j)}})$, so that by Part (1) of Lemma 6,

$$(1 - O((\gamma_3)^2))\widehat{M}(t^{(j-1)}) \le \widehat{f^{(j-1)}}(i) \le \widehat{M}(t^{(j-1)}) \tag{33}$$

and

$$(1 - O((\gamma_3)^2))\widehat{M}(t^{(j)}) \le \widehat{f^{(j)}}(i) \le \widehat{M}(t^{(j)})\ . \tag{34}$$

By Equation (34),

$$\sum_{i\in T^{(j)}} \left(\widehat{f^{(j)}}(i)\right)^2 = (1 - O((\gamma_3)^2)) \cdot \widetilde{M}(t^{(j)}) \tag{35}$$

Given that $\phi_j^{(j)}$ is accurate as required, recalling that $\gamma_1 = \Omega((\gamma_3)^2)$ and that $\widetilde{M}(m) = 2/\pi - O(m^{-1})$, we see that the algorithm does not reject in Step 5d because of a large deviation of $\phi_j^{(j)}$ from $2/\pi$. Similarly, by Equation (33),

$$\left(\frac{\widehat{M}(t^{(j)})}{\widehat{M}(t^{(j-1)})}\right)^2 \cdot \sum_{i\in T^{(j)}} \left(\widehat{f^{(j-1)}}(i)\right)^2 \ = \ (1 - O((\gamma_3)^2))\widetilde{M}(t^{(j)}) \tag{36}$$

and by Equations (33) and (34),

$$\frac{\widehat{M}(t^{(j)})}{\widehat{M}(t^{(j-1)})} \cdot \sum_{i\in T^{(j)}} \widehat{f^{(j-1)}}(i) \cdot \widehat{f^{(j)}}(i) \ = \ (1 - O((\gamma_3)^2))\widetilde{M}(t^{(j)})\ . \tag{37}$$

Given that $\phi_{j-1}^{(j)}$ and $\phi_{j-1,j}^{(j)}$ are accurate as required (as well as $\gamma_1 = \Omega((\gamma_3)^2)$ and $\widetilde{M}(m) = 2/\pi - O(m^{-1})$), the algorithm does not reject in Step 5d because of large deviations of these estimates as well. Recall that we have already bounded the probability that the algorithm rejects in Step 5e due to a failed call to Algorithm 2.

Finally, by Lemma 9 (using Equation (34) for $j = \ell$ and the fact that Algorithm 3 is called with $\zeta = \epsilon/8 = \omega((\gamma_3)^2)$), the algorithm rejects in Step 6 with probability at most $1/10$. Summing all probabilities of failure we see that the overall probability that the algorithm rejects is at most $1/n^{\omega(1)} + 1/10 + 1/10 + 1/10 < 1/3$, and the lemma is proved. ■

Lemma 11 (Soundness) *If* $\mathrm{dist}(f, \mathrm{SMAJ}) > \epsilon$*, then Algorithm 4 rejects with probability at least* $2/3$.

Proof: Fix $f : \{+1,-1\}^n \to \{+1,-1\}$ to be any function that is ϵ-far from every n-variable signed majority. As in the proof of Lemma 10, the probability that any estimate computed by the algorithm has an additive error greater than indicated is $O(\delta \log n)$, so we may assume from this point on (incurring failure probability at most $1/20$) that all estimates are indeed within the desired additive error. In particular, this means that $\sum_{i=1}^n (\widehat{f}(i))^2 \ge \widetilde{M}(n) - 2\gamma_1$. By Lemma 2, for every $\sigma \in \{+1,-1\}^n$ and for $\epsilon_0 = \frac{7\epsilon^2}{32} - 2\gamma_1$, we have

$$\sum_{i=1}^n \left(\widehat{f}(i) - \sigma_i \cdot \widehat{M}(n)\right)^2 > \epsilon_0\ . \tag{38}$$

By Lemma 8, for each call to Algorithm 2 on some $f^{(j)}$ (i.e., in Step 3 for $j = 0$ or in Step 5e for $j > 1$), if $|\widehat{f^{(j)}}(i)| \geq \gamma_2$ for some $i \in T^{(j)}$, then Algorithm 2 returns fail with probability at least $1 - \delta$. Thus we may assume from this point on (incurring failure probability $O(\delta \log n) \leq 1/20$) that for each j, if Algorithm 4 reached iteration j, then $|\widehat{f^{(j-1)}}(i)| < \gamma_2$ for every $i \in T^{(j-1)}$.

We next define the following vectors:

$$a^{(j)} \overset{\text{def}}{=} \left(\widehat{f^{(j)}}(i)\right)_{i \in T^{(j)}} \quad \text{for } j \geq 0, \tag{39}$$

$$b^{(j)} = \left(\frac{\widehat{M}(t^{(j)})}{\widehat{M}(t^{(j-1)})} \cdot \widehat{f^{(j-1)}}(i)\right)_{i \in T^{(j)}} \quad \text{for } j \geq 1, \tag{40}$$

and for each $\sigma \in \{+1, -1\}^{T^{(j)}}$,

$$c_\sigma^{(j)} = \left(\sigma_i \cdot \widehat{M}(t^{(j)})\right)_{i \in T^{(j)}} \quad \text{for } j \geq 0. \tag{41}$$

Recalling that $f^{(0)} = f$, $T^{(0)} = [n]$ and $t^{(0)} = n$, Equation (38) implies that

$$\|a^{(0)} - c_\sigma^{(0)}\|_2^2 > \epsilon_0 \tag{42}$$

for each $\sigma \in \{+1, -1\}^n$. Observe that $\phi_{j-1}^{(j)}$, $\phi_j^{(j)}$, and $\phi_{j-1,j}^{(j)}$ are, respectively, estimates of $\|b^{(j)}\|_2^2$, $\|a^{(j)}\|_2^2$ and $\langle a^{(j)}, b^{(j)} \rangle$. From the foregoing discussion at the start of the proof we may assume that these estimates are all within additive error at most γ_1 of the true values; moreover, we may assume that each estimate is within $\pm 2\gamma_1$ from $2/\pi$, or else the algorithm would reject. Consequently we have that all j satisfy

$$\|a^{(j)} - b^{(j)}\|_2^2 = \|b^{(j)}\|_2^2 + \|a^{(j)}\|_2^2 - 2\langle a^{(j)}, b^{(j)} \rangle \leq 8\gamma_1 . \tag{43}$$

Similarly, by the assumption concerning the correctness of the executions of Algorithm 2, if we have reached iteration j, then $|a^{(j-1)}(i)| < \gamma_2$ for every $i \in T^{(j-1)}$, which implies that $(a_i^{(j-1)})^2 < (\gamma_2)^2$ for each $i \in T^{(j)}$.

We have reached the main step of the proof, which is establishing the following claim.

Claim 12 *There is an absolute constant C such that the following holds: for each fixed $j \geq 1$, given that*

$$\text{all } \sigma \in \{+1, -1\}^{T^{(j-1)}} \text{ satisfy}$$
$$\|a^{(j-1)} - c_\sigma^{(j-1)}\|_2 \geq \sqrt{\epsilon_0} \cdot \left(1 - C\sqrt{\gamma_1/\epsilon_0}\right)^{j-1} \tag{44}$$

and that

$$(a_i^{(j-1)})^2 \leq (\gamma_2)^2 \text{ for each } i \in T^{(j)} \quad , \tag{45}$$

with probability at least $1 - \delta$ over the random choice of $T^{(j)} \subseteq T^{(j-1)}$, all $\sigma \in \{+1, -1\}^{T^{(j)}}$ satisfy

$$\|a^{(j)} - c_\sigma^{(j)}\|_2 \geq \sqrt{\epsilon_0} \cdot \left(1 - C\sqrt{\gamma_1/\epsilon_0}\right)^{j} . \tag{46}$$

Since there are $O(\log n)$ iterations of the "while" loop (Step 5), Claim 12 implies (given that all estimates are as required as described above) that conditioned on the algorithm not rejecting in Step 2 or Step 3 or in any iteration of the "while" loop, with probability at least $1 - O(\delta \log n) \geq \frac{9}{10}$ (over all choices of $T^{(j)}$) the following holds: once the algorithm exits the "while" loop, the final vector $a^{(\ell)}$ (corresponding to $f^{(\ell)}$) satisfies (recall that $\epsilon_0 = 7\epsilon^2/32 - 2\gamma_1$)

$$\|a^{(\ell)} - c_\sigma^{(\ell)}\|_2^2 \geq \epsilon_0 \cdot (1 - C\sqrt{\gamma_1/\epsilon_0})^{O(\log n)} \geq \epsilon^2/8 \tag{47}$$

for all $\sigma \in \{+1,-1\}^{T^{(\ell)}}$. But if this is the case then we claim that Step 6 will reject with probability at least $9/10$ by Lemma 9 because for at least one $i \in [k]$ it holds that $|\widehat{f^{(\ell)}}(i)| \notin \Big[(1-\epsilon/4)\widehat{M}(k), (1-\epsilon/4)\widehat{M}(k)\Big]$ (recall that Algorithm 3 is called with $\zeta = \epsilon/8$). To verify this, assume in contradiction that $|\widehat{f^{(\ell)}}(i)| \in \Big[(1-\epsilon/4)\widehat{M}(k), (1-\epsilon/4)\widehat{M}(k)\Big]$ for every $i \in [k]$. But then, setting $\sigma_i = \mathrm{sign}(\widehat{f^{(\ell)}}(i))$ we get

$$\|a^{(\ell)} - c_\sigma^{(\ell)}\|_2^2 \leq k \cdot (\epsilon^2/16)(\widehat{M}(k))^2 \ , \tag{48}$$

which (using $\widehat{M}(k) < 1/\sqrt{k}$) is less than $\epsilon^2/8$. Consequently the overall probability that the algorithm outputs reject is at least $2/3$.

Thus, to complete the proof of Lemma 11, it remains to prove Claim 12.

Proof of Claim 12: Recall that $b^{(j)}$ is obtained from $a^{(j-1)}$ by selecting each coordinate of $a^{(j-1)}$ independently with probability $1/2$, and multiplying it by $\widehat{M}(t^{(j)})/\widehat{M}(t^{(j-1)})$. Given a vector $\sigma \in \{+1,-1\}^{T^{(j-1)}}$, we shall write the i^{th} coordinate of $a^{(j-1)}$ (that is, $\widehat{f^{(j-1)}}(i)$) as $(1 + \rho_\sigma(i)) \cdot \sigma_i \widehat{M}(t^{(j-1)})$. Using this notation we have

$$\begin{aligned} \|a^{(j-1)} - c_\sigma^{(j-1)}\|_2^2 &= \sum_{i \in T^{(j-1)}} \Big((1+\rho_\sigma(i)) \cdot \sigma_i \widehat{M}(t^{(j-1)}) - \sigma_i \widehat{M}(t^{(j-1)})\Big)^2 \\ &= (\widehat{M}(t^{(j-1)}))^2 \sum_{i \in T^{(j-1)}} (\rho_\sigma(i))^2 \ . \end{aligned} \tag{49}$$

Recalling that Equation (44) (the first premise of the claim) gives $\|a^{(j-1)} - c_\sigma^{(j-1)}\|_2 \geq \sqrt{\epsilon_0} \cdot (1 - C\sqrt{\gamma_1/\epsilon_0})^{j-1}$, by Equation (49) we get (recalling that $(\widehat{M}(m))^2 \leq \frac{2}{\pi m}$) that for each $\sigma \in \{+1,-1\}^{T^{(j-1)}}$,

$$\sum_{i \in T^{(j-1)}} (\rho_\sigma(i))^2 = \beta_\sigma t^{(j-1)} \tag{50}$$

for $\beta_\sigma \geq \frac{\pi}{2} \cdot \epsilon_0 \cdot (1 - C\sqrt{\gamma_1/\epsilon_0})^{2(j-1)}$.

Recalling that $c_\sigma^{(j)} = \Big(\sigma_i \cdot \widehat{M}(t^{(j)})\Big)_{i \in T^{(j)}}$ for $\sigma \in \{+1,-1\}^{T^{(j)}}$, we have that for all $\sigma \in \{+1,-1\}^{T^{(j)}}$,

$$\begin{aligned} \|b^{(j)} - c_\sigma^{(j)}\|_2^2 &= \sum_{i \in T^{(j)}} \Big((\widehat{M}(t^{(j)})/\widehat{M}(t^{(j-1)})) \cdot (1+\rho_\sigma(i)) \cdot \sigma_i \widehat{M}(t^{(j-1)}) - \sigma_i \widehat{M}(t^{(j)})\Big)^2 \\ &= (\widehat{M}(t^{(j)}))^2 \sum_{i \in T^{(j)}} (\rho_\sigma(i))^2. \end{aligned} \tag{51}$$

We also know from Equation (45) (the second premise of the claim) that $(a_i^{(j-1)})^2 \leq (\gamma_2)^2$ for each i; since $a_i^{(j-1)} = (1+\rho_\sigma(i)) \cdot \sigma_i \widehat{M}(t^{(j-1)})$, recalling that $\widehat{M}(m)^2 \geq \frac{2}{\pi m} - c/m^{3/2}$ for some constant c and $t^{(j-1)} = \Omega(s)$, we get that for each $\sigma \in \{+1,-1\}^{T^{(j-1)}}$ we have $(\rho_\sigma(i))^2 \leq 2(\gamma_2)^2 t^{(j-1)}$.

Now let us define a "distinguished" vector of signs $\sigma^{(j-1)} \in \{+1,-1\}^k$ by $\sigma_i^{(j-1)} = \mathrm{sign}(a_i^{(j-1)})$, so $\sigma^{(j-1)}$ is the sign vector whose coordinates match the signs of the degree-1 Fourier coefficients of $f^{(j-1)}$. We apply Lemma 5 where the $(\rho_{\sigma^{(j-1)}}(i))^2$'s play the role of the w_i's, $2(\gamma_2)^2$ plays the role of "κ", $\beta_{\sigma^{(j-1)}}$ plays the role of "β", and $t^{(j-1)}$ plays the role of m, taking α to be $\gamma_1/\beta_{\sigma^{(j-1)}}$. Lemma 5 gives us that

$$\sum_{i \in T^{(j)}} (\rho_{\sigma^{(j-1)}}(i))^2 \geq (\beta_{\sigma^{(j-1)}} - \gamma_1) t^{(j)} \tag{52}$$

with probability at least $1 - \exp(-\Omega(\gamma_1^2/(\beta_{\sigma^{(j-1)}} \cdot (\gamma_2)^2)))$ (where the probability is taken over the random choice of $T^{(j)}$ from $T^{(j-1)}$). We now observe that Equations (49) and (50) imply that $\beta_{\sigma^{(j-1)}}$ is certainly upper bounded by $O(1)$ (recall that $a^{(j-1)}$ and $c^{(j-1)}$ are vectors of norm at most 1), so the bound $\gamma_1 = \Omega(\sqrt{\log(1/\delta)} \cdot \gamma_2)$ implies that the above probability is at least $1-\delta$ as desired.

We now observe that for every $\sigma \in \{+1,-1\}^{T^{(j-1)}}$ and every $i \in T^{(j-1)}$, we have

$$|\rho_\sigma(i)| \geq |\rho_{\sigma^{(j-1)}}(i)|. \tag{53}$$

(This follows easily from the facts that $\sigma_i^{(j-1)} = \mathrm{sign}(a_i^{(j-1)})$ and $a_i^{(j-1)} = (1+\rho_\sigma(i)) \cdot \sigma_i \cdot \widehat{M}(t^{(j-1)})$.) Consequently, Equation (52) yields that with probability at least $1-\delta$ over the random choice of $T^{(j)}$, *every* $\sigma \in \{+1,-1\}^{T^{(j-1)}}$ satisfies

$$\sum_{i \in T^{(j)}} (\rho_\sigma(i))^2 \geq (\beta_{\sigma^{(j-1)}} - \gamma_1) t^{(j)}. \tag{54}$$

Using Equation (51) this gives

$$\begin{aligned} \|b^{(j)} - c_\sigma^{(j)}\|_2^2 &\geq (\widehat{M}(t^{(j)}))^2 \cdot (\beta_{\sigma^{(j-1)}} - \gamma_1) \cdot t^{(j)} \\ &= \widetilde{M}(t^{(j)}) \cdot (\beta_{\sigma^{(j-1)}} - \gamma_1) \\ &\geq \widetilde{M}(t^{(j-1)}) \cdot (1 - O(1/t^{(j-1)})) \cdot (\beta_{\sigma^{(j-1)}} - \gamma_1) \\ &\geq \beta_{\sigma^{(j-1)}} \cdot \widetilde{M}(t^{(j-1)})(1 - O(1)/t^{(j-1)}) - \gamma_1 \end{aligned} \tag{55}$$

where the first inequalities follow from the definition and bound on $\widetilde{M}(\cdot)$ given in Equation (3). Combining Equations (49) and (50) we may re-express $\|a^{(j-1)} - c_\sigma^{(j-1)}\|_2^2$ as $\beta_{\sigma^{(j-1)}} \widetilde{M}(t^{(j-1)})$, and thus the final inequality above yields

$$\|b^{(j)} - c_\sigma^{(j)}\|_2^2 \;\geq\; \|a^{(j-1)} - c_\sigma^{(j-1)}\|_2^2 \cdot \left(1 - O(1)/t^{(j-1)}\right) - \gamma_1 \,. \tag{56}$$

Combining with Equation (44) (and using the elementary identity $\sqrt{x-y} \geq \sqrt{x} - \sqrt{y}$ for $0 < y < x$), we get

$$\|b^{(j)} - c_\sigma^{(j)}\|_2 \;\geq\; \sqrt{\epsilon_0} \cdot (1 - C\sqrt{\gamma_1/\epsilon_0})^{j-1} \cdot \sqrt{1 - O(1)/t^{(j-1)}} - \sqrt{\gamma_1} \,. \tag{57}$$

Now the fact that $\|a^{(j)} - b^{(j)}\|_2 \leq \sqrt{8\gamma_1}$ (which follows from Equation (43) implies that

$$\begin{aligned} \|a^{(j)} - c_\sigma^{(j)}\|_2 &\geq \sqrt{\epsilon_0} \cdot (1 - C\sqrt{\gamma_1/\epsilon_0})^{j-1} \cdot \sqrt{1 - O(1)/t^{(j-1)}} - \sqrt{\gamma_1} - \sqrt{8\gamma_1} \\ &\geq \sqrt{\epsilon_0} \cdot (1 - C\sqrt{\gamma_1/\epsilon_0})^{j} \end{aligned} \tag{58}$$

for a suitable absolute constant C, and Claim 12 follows. ■ **(Claim 12 and Lemma 11)**

Theorem 5 *Algorithm 4 is a testing algorithm for Signed Majority functions. Its query complexity is* $\mathrm{poly}(\log n, 1/\epsilon)$.

Proof: The correctness of the algorithm follows by Lemmas 10 and 11. Inspection of the algorithm and the parameter settings shows that the query complexity is the query complexity is

$$O\left(\log n\cdot\left(\overbrace{\frac{\log\frac{1}{\delta}}{(\gamma_1)^4}}^{\text{Step 5d}}+\overbrace{\frac{\log(\frac{1}{\delta\gamma_2})\log\frac{1}{\gamma_2}}{(\gamma_2)^4}}^{\text{Step 5e}}+\overbrace{\frac{\log(\frac{1}{\delta})\log(\frac{1}{\delta\gamma_3})}{(\gamma_3)^3}}^{\text{Step 5c}}\right)+\overbrace{\frac{s\log s}{(\epsilon/8)^4}}^{\text{Step 6}}\right)=O\left(\frac{(\log n)^9(\log(\log n/\epsilon))^4}{\epsilon^8}\right)$$

where the dominant contribution comes from the ($O(\log n)$ many) executions of Step 5e. ■

5 A $\mathrm{poly}(n)$ lower bound for non-adaptive testing of SMAJ

In this section we prove the following lower bound which yields Theorem 2 of Section 1:

Theorem 6 *There is a universal constant* $\epsilon_0 > 0$ *such that any non-adaptive algorithm for testing* $\mathcal{SMAJ}$ *with distance parameer* ϵ_0 *must make at least* $\Omega(n^{1/12})$ *queries.*

Theorem 6 gives an exponential improvement over the lower bound of [27], which showed that any non-adaptive algorithm for ϵ_0-testing $\mathcal{SMAJ}$ must make $\Omega(\log n)$ queries. The theorem is proved via an improved analysis of the construction of [27], using ideas from [5] and [21]. In Section 5.1 we first recall the construction and then in Section 5.2 we prove Theorem 6.

5.1 The construction

The construction is based on two distributions $\mathcal{D}_{\mathrm{yes}}$ and $\mathcal{D}_{\mathrm{no}}$ over functions from $\{+1,-1\}^n$ to $\{+1,-1\}$. As defined in [27], the distribution $\mathcal{D}_{\mathrm{yes}}$ is uniform over all 2^n signed majority functions, so a function f_{yes} drawn from $\mathcal{D}_{\mathrm{yes}}$ is $f_{\mathrm{yes}}(x) = \mathrm{sign}(\sigma_1 x_1 + \cdots \sigma_n x_n)$ where each σ_i is independently and uniformly drawn from $\{+1,-1\}$. The distribution $\mathcal{D}_{\mathrm{no}}$ is similarly a distribution over halfspaces of the form $f_{\mathrm{no}}(x) = \mathrm{sign}(\nu_1 x_1 + \cdots \nu_n x_n)$, but each ν_i is independently chosen to be $\pm\sqrt{1/2}$ or $\pm\sqrt{3/2}$ each with probability $1/4$.

Lemma 4 of [27] gives the following:

Lemma 13 (Lemma 4 of [27]) *Let* f_{no} *be a random function drawn from* $\mathcal{D}_{\mathrm{no}}$. *With probability at least* $1-o(1)$ *we have that* f_{no} *is* ϵ_0*-far from* SMAJ, *where* $\epsilon_0 > 0$ *is some fixed constant independent of* n.

In the next subsection we will prove the following lemma:

Lemma 14 *Let* $\mathcal{T}$ *be any deterministic non-adaptive q-query algorithm for testing whether a black-box* $f : \{+1,-1\}^n \to \{+1,-1\}$ *is a signed majority. Then*

$$\left|\Pr_{f_{\mathrm{yes}}\sim\mathcal{D}_{\mathrm{yes}}}[\mathcal{T}\textit{ accepts } f_{\mathrm{yes}}] - \Pr_{f_{\mathrm{no}}\sim\mathcal{D}_{\mathrm{no}}}[\mathcal{T}\textit{ accepts } f_{\mathrm{no}}]\right| \;=\; O(q^{3/2}/n^{1/8})\ . \qquad (59)$$

A standard argument using Yao's minmax method [36] yields Theorem 6 from Lemmas 13 and 14.

5.2 Proof of Lemma 14

Fix $\mathcal{T}$ to be a deterministic q-query non-adaptive tester. We may view its q queries as a $q \times n$ query matrix $Q \in \{+1,-1\}^{q\times n}$. Following the terminology of [5], we define a "Response Vector" random variable $R_{\text{yes}} \in \{+1,-1\}^q$ which is obtained by drawing a random $\sigma \in \{+1,-1\}^n$, evaluting $\text{sign}(\sigma \cdot x)$ on the q different choices of x corresponding to the q rows of Q, and writing down the results as the q entries of R_{yes}. Similarly we define a "Response Vector" random variable $R_{\text{no}} \in \{+1,-1\}^q$ which is obtained by drawing a random coefficient vector ν uniformly from $\{\pm\sqrt{1/2}, \pm\sqrt{3/2}\}$, evaluating $\text{sign}(\nu \cdot x)$ on the q rows of Q, and writing down the results as the q entries of R_{no}.

By the definition of total variation distance, the left-hand side of Equation (59) is upper bounded by $d_{\text{TV}}(R_{\text{yes}}, R_{\text{no}})$, the total variation distance between the random variables R_{yes} and R_{no}. Let S denote the column vector $Q\sigma$ and let T denote the column vector $Q\nu$ where σ, ν are uniform over $\{+1,-1\}^q$ and $\{\pm\sqrt{1/2}, \pm\sqrt{3/2}\}^q$ respectively as described above. The Response Vector R_{yes} is determined by the orthant of $\mathbb{R}^q$ in which S lies, and the Response Vector R_{no} is determined by the orthant of $\mathbb{R}^q$ in which T lies. So as in [5], to prove Lemma 14 it suffices for us to prove the following:

Lemma 15 *For S, T as defined above and $\mathcal{O}$ any union of orthants in $\mathbb{R}^q$, we have*

$$|\Pr[S \in \mathcal{O}] - \Pr[T \in \mathcal{O}] = O(q^{3/2}/n^{1/8}). \tag{60}$$

We proceed to prove Lemma 15. We have that

$$S = X_1 + \cdots + X_n \tag{61}$$

where each X_i is a vector-valued random variable which is independently Q^i (the i-th column of Q) with probability $1/2$ and $-Q^i$ with probability $1/2$. Similarly we have that

$$T = Y_1 + \cdots + Y_n \tag{62}$$

where each Y_i is a vector-valued random variable which is equiprobably independently $\pm\sqrt{\frac{1}{2}}Q^i, \pm\sqrt{\frac{3}{2}}Q^i$.

Similar to [5], we will prove Lemma 15 using multidimensional invariance principle tools. The following lemma is a combination of Lemmas 4.5 and 4.6 in [5], which in turn are respectively essentially Theorem 4.1 in [28] (see [21]), and a result established in [21].

Lemma 16 ([28, 21]) *Let $\mathcal{O}$ be any union of orthants in $\mathbb{R}^q$. Let $S = S_1 + \cdots + S_n$ where the S_i's are independent $\mathbb{R}^q$-valued random variables and let $T = T_1 + \cdots + T_n$ similarly.*

Assume that for each $i \in [n]$, S_i and T_i have matching means and covariance matrices, i.e. $\text{E}[S_i] = \text{E}[T_i]$ *and* $\text{Cov}[S_i] = \text{Cov}[T_i]$. *Given a multi-index $J = (j_1, \ldots, j_q) \in \mathbb{N}^q$ and a vector $U \in \mathbb{R}^q$, let U^J denote $U_1^{j_1} \cdot \cdots \cdot U_q^{j_q}$, and let $|J|$ denote $j_1 + \cdots + j_q$.*

Let

$$W_r := \{x \in \mathbb{R}^q : |x_i| \le r/2 \textit{ for some } i \in [q]\}. \tag{63}$$

Then we have

$$|\Pr[S \in \mathcal{O}] - \Pr[T \in \mathcal{O}]| \ \le \ \Pr[S \in W_r] + O(1/r^3) \sum_{i=1}^{n} \sum_{|J|=3} \left(\text{E}[|(S_i)^J|] + \text{E}[|(T_i)^J|]\right) . \tag{64}$$

A suitable application of Lemma 16 will give us Equation (60) as desired. To show that the conditions for Lemma 16 are satisfied, we begin with the following straightforward lemma, which is analogous to Lemma 4.7 of [5]:

Lemma 17 *For all $j \in [n]$ we have* $\mathrm{E}[S_j] = \mathrm{E}[T_j] = 0$ *and* $\mathrm{Cov}[S_j] = \mathrm{Cov}[T_j]$.

Proof: It suffices to prove the desired equalities for $j = 1$. We have

$$\mathrm{E}[S_1] = \mathrm{E}[(X_1 + \cdots + X_n)_1] = \sum_{i=1}^{n} \mathrm{E}[(X_i)_1] = 0 \tag{65}$$

since each $(X_i)_1$ is independently equally likely to be $(Q^i)_1$ or $-(Q^i)_1$. Similarly, we have

$$\mathrm{E}[T_1] = \mathrm{E}[(Y_1 + \cdots + Y_n)_1] = \sum_{i=1}^{n} \mathrm{E}[(Y_i)_1] = 0 \tag{66}$$

since each $(Y_i)_1$ is independently equally likely to be $\pm\sqrt{1/2}(Q^i)_1$ or $\pm\sqrt{3/2}(Q^i)_1$.

For the covariances, fix $i, i' \in [q]$. Let us write q to denote the i-coordinate of vector Q^1 and q' to denote the i'-coordinate of Q^1. Similarly, let us write x for the i-coordinate of vector X_1 and x' for the i'-coordinate of X_1.

Observing that $\mu_x := \mathrm{E}[x] = 0$ and $\mu_{x'} := \mathrm{E}[x'] = 0$, we have that

$$\mathrm{Cov}[S_1]_{i,i'} \;=\; \mathrm{E}[(x-\mu_x)(x'-\mu_{x'})] \;=\; \mathrm{E}[xx'] \;=\; \frac{1}{2}(qq') + \frac{1}{2}(-q)(-q') = qq' \; . \tag{67}$$

Similarly let us write y for the i-coordinate of Y_1 and y' for the i'-coordinate of Y_1. Observing that $\mu_y := \mathrm{E}[y] = 0$ and $\mu_{y'} := \mathrm{E}[y'] = 0$, we similarly have that

$$\begin{aligned} \mathrm{Cov}[T_1]_{i,i'} &= \mathrm{E}[(y-\mu_y)(y'-\mu_{y'})] = \mathrm{E}[yy'] \\ &= \frac{1}{4} \cdot qq' \cdot \left(\left(\sqrt{\frac{1}{2}}\right)^2 + \left(-\sqrt{\frac{1}{2}}\right)^2 + \left(\sqrt{\frac{3}{2}}\right)^2 + \left(-\sqrt{\frac{3}{2}}\right)^2 \right) \\ &= qq' \, , \end{aligned} \tag{68}$$

and the lemma follows. ■

As in Lemma 16 let $W_r := \{x \in \mathbb{R}^q : |x_i| \le r \text{ for some } i \in [q]\}$ be the region around the orthant boundaries. Recalling that the binomial distribution $B(n, 1/2)$ puts probability mass at most $O(1/\sqrt{n})$ on each outcome, it is easy to see that for $r \ge 1$ we have

$$\Pr[S \in W_r] = O(qr/\sqrt{n}), \tag{69}$$

where the factor of q comes from a union bound over the q coordinates.

Finally, to apply Lemma 16 it remains only to observe that for any $|J| \le 3$ the quantities $\mathrm{E}[|X_i^J|]$ and $\mathrm{E}[|Y_i^J|]$ can be bounded uniformly by an absolute constant for all i. Putting the pieces together Lemma 16 gives that

$$|\Pr[S \in \mathcal{O}] - \Pr[T \in \mathcal{O}]| \le O(qr/\sqrt{n}) + O(nq^3/r^3). \tag{70}$$

Taking $r = n^{3/8}q^{1/2}$, we get that the bound is $O(q^{3/2}/n^{1/8})$ as desired. This concludes the proof of Lemma 14.

Acknowledgement. We would like to thank Ryan O'Donnell and Gilad Tsur for discussions in the initial stages of this work.

References

[1] N. Ailon and B. Chazelle. Information theory in property testing and monotonicity testing in higher dimension. *Information and Computation*, 204:1704–1717, 2006.

[2] N. Alon, T. Kaufman, M. Krivelevich, S. Litsyn, and D. Ron. Testing Reed-Muller Codes. *IEEE Transactions on Information Theory*, 51(11):4032–4039, 2005.

[3] Eric Blais. Improved bounds for testing juntas. In *Proc. 12th International Workshop on Randomization and Computation (RANDOM)*, pages 317–330, 2008.

[4] Eric Blais. Testing juntas nearly optimally. In *Proc. 41st Annual ACM Symposium on Theory of Computing (STOC)*, pages 151–158, 2009.

[5] Eric Blais and Ryan O'Donnell. Lower bounds for testing function isomorphism. In *Proc. 25th Annual IEEE Conference on Computational Complexity (CCC)*, pages 235–246, 2010.

[6] M. Blum, M. Luby, and R. Rubinfeld. Self-testing/correcting with applications to numerical problems. *J. Comp. Sys. Sci.*, 47:549–595, 1993.

[7] J. Briët, S. Chakraborty, D. García-Soriano, and A. Matsliah. Monotonicity testing and shortest-path routing on the cube. In *Proc. 14th International Workshop on Randomization and Computation (RANDOM)*, pages 462–475, 2010.

[8] J. Brody, K. Matulef, and C. Wu. Lower bounds for testing computability by small width OBDDs. In *Proc. 8th Annual Conference on Theory and Applications of Models of Computation (TAMC)*, pages 320–331, 2011.

[9] Ronald de Wolf. *A Brief Introduction to Fourier Analysis on the Boolean Cube*. Number 1 in Graduate Surveys. Theory of Computing Library, 2008.

[10] I. Diakonikolas, H. Lee, K. Matulef, K. Onak, R. Rubinfeld, R. Servedio, and A. Wan. Testing for concise representations. In *Proc. 48th Ann. Symposium on Computer Science (FOCS)*, pages 549–558, 2007.

[11] Y. Dodis, O. Goldreich, E. Lehman, S. Raskhodnikova, D. Ron, and A. Samorodnitsky. Improved testing algorithms for monotonocity. In *Proc. Randomization, Approximation, and Combinatorial Algorithms and Techniques, Third International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM)*, pages 97–108, 1999.

[12] D. Dubhashi and A. Panconesi. *Concentration of measure for the analysis of randomized algorithms*. Cambridge University Press, Cambridge, 2009.

[13] E. Fischer. The art of uninformed decisions: A primer to property testing. *Bulletin of the European Association for Theoretical Computer Science*, 75:97–126, 2001.

[14] E. Fischer, G. Kindler, D. Ron, S. Safra, and A. Samorodnitsky. Testing juntas. *J. Computer & System Sciences*, 68(4):753–787, 2004.

[15] E. Fischer, E. Lehman, I. Newman, S. Raskhodnikova, R. Rubinfeld, and A. Samorodnitsky. Monotonicity testing over general poset domains. In *Proc. 34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 474–483, 2002.

[16] O. Goldreich. On testing computability by small width OBDDs. In *Proc. 14th International Workshop on Randomization and Computation (RANDOM)*, pages 574–587, 2010.

[17] O. Goldreich, editor. *Property Testing: Current Research and Surveys*. Springer, 2010. LNCS 6390.

[18] O. Goldreich, S. Goldwasser, E. Lehman, D. Ron, and A. Samordinsky. Testing monotonicity. *Combinatorica*, 20(3):301–337, 2000.

[19] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, 45:653–750, 1998.

[20] P. Gopalan, R. O'Donnell, R. Servedio, A. Shpilka, and K. Wimmer. Testing Fourier dimensionality and sparsity. In *Proc. 36th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 500–512, 2009.

[21] P. Gopalan, R. O'Donnell, Y. Wu, and D. Zuckerman. Fooling functions of halfspaces under product distributions. In *Proc. 25th Annual IEEE Conference on Computational Complexity (CCC)*, pages 223–234, 2010.

[22] S. Halevy and E. Kushilevitz. Testing monotonicity over graph products. *Random Structures and Algorithms*, 33(1):44–67, 2008.

[23] C. S. Jutla, A. C. Patthak, A. Rudra, and D. Zuckerman. Testing low-degree polynomials over prime fields. In *Proc. 45th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 423–432. IEEE Computer Society Press, 2004.

[24] T. Kaufman and D. Ron. Testing polynomials over general fields. *SIAM J. on Comput.*, 35(3):779–802, 2006.

[25] H. König, C. Schütt, and N. Tomczak-Jaegermann. Projection constants of symmetric spaces and variants of khintchine's inequality. *J. Reine Agnew. Math.*, 511:1–42, 1999.

[26] K. Matulef, R. O'Donnell, R. Rubinfeld, and R. Servedio. Testing halfspaces. *SIAM J. on Comput.*, 39(5):2004–2047, 2010.

[27] K. Matulef, R. O'Donnell, R. Rubinfeld, and R. A. Servedio. Testing ± 1-weight halfspaces. In *Proc. 13th International Workshop on Randomization and Computation (RANDOM)*, pages 646–657, 2009.

[28] E. Mossel. Gaussian bounds for noise correlation of functions and tight analysis of long codes. *Proc. 49th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 156–165, 2008.

[29] R. O'Donnell. Some topics in analysis of boolean functions. In *Proc. 40th Annual ACM Symposium on Theory of Computing (STOC)*, pages 569–578, 2008.

[30] M. Parnas, D. Ron, and A. Samorodnitsky. Testing Basic Boolean Formulae. *SIAM J. Disc. Math.*, 16:20–46, 2002.

[31] D. Ron. Property Testing: A Learning Theory Perspective. *Foundations and Trends in Machine Learning*, 1(3):307–402, 2008.

[32] D. Ron. Algorithmic and analysis techniques in property testing. *Foundations and Trends in Theoretical Computer Science*, 5:73–205, 2010.

[33] D. Ron and G. Tsur. Testing computability by width-two OBDDs. *Theoretical Computer Science*, 420:64–79, 2012.

[34] I. S. Shiganov. Refinement of the upper bound of the constant in the central limit theorem. *Journal of Soviet Mathematics*, pages 2545–2550, 1986.

[35] Wikipedia contributors. Central binomial coefficient. Wikipedia, The Free Encyclopedia, accessed June 8, 2012. http://en.wikipedia.org/wiki/Central_binomial_coefficient.

[36] A. Yao. Probabilistic computations: Towards a unified measure of complexity. In *Proc. Seventeenth Annual Symposium on Foundations of Computer Science (STOC)*, pages 222–227, 1977.