Testing ±1-Weight Halfspaces

Kevin Matulef¹, Ryan O’Donnell², Ronitt Rubinfeld³, and Rocco A. Servedio⁴

¹ MIT
matulef@mit.edu

² Carnegie Mellon University
odonnell@cs.cmu.edu

³ Tel Aviv University and MIT
ronitt@theory.csail.mit.edu

⁴ Columbia University
rocco@cs.columbia.edu

Abstract. We consider the problem of testing whether a Boolean function $f : \{-1,1\}^n \rightarrow \{-1,1\}$ is a ±1-weight halfspace, i.e. a function of the form $f(x) = \text{sgn}(w_1 x_1 + w_2 x_2 + \cdots + w_n x_n)$ where the weights $w_i$ take values in $\{-1,1\}$. We show that the complexity of this problem is markedly different from the problem of testing whether $f$ is a general halfspace with arbitrary weights. While the latter can be done with a number of queries that is independent of $n$ [7], to distinguish whether $f$ is a ±1-weight halfspace versus $\epsilon$-far from all such halfspaces we prove that nonadaptive algorithms must make $\Omega(\log n)$ queries. We complement this lower bound with a sublinear upper bound showing that $O(\sqrt{n} \cdot \text{poly}(\frac{1}{\epsilon}))$ queries suffice.

1 Introduction

dfgfhfhdfg this is the best i can get it asdf to do it doesn’t seem to make much difference as far as i can tell

This is the best I can get it to do i I don’t know how to change $\frac{3}{4} \cdot \sum_{i=1}^{n}$ of

A fundamental class in machine learning and complexity is the class of halfspaces, or functions of the form $f(x) = (w_1 x_1 + w_2 x_2 + \cdots + w_n x_n - \theta)$. Halfspaces are a simple yet powerful class of functions, which for decades have played an important role in fields such as complexity theory, optimization, and machine learning (see e.g. [5, 12, 1, 9, 8, 11]).

Recently [7] brought attention to the problem of testing halfspaces. Given query access to a function $f : \{-1,1\}^n \rightarrow \{-1,1\}$, the goal of an $\epsilon$-testing algorithm is to output YES if $f$ is a halfspace and NO if it is $\epsilon$-far (with respect to the uniform distribution over inputs) from all halfspaces. Unlike a learning algorithm for halfspaces, a testing algorithm is not required to output an approximation to $f$ when it is close to a halfspace. Thus, the testing problem can be viewed as a relaxation of the proper learning problem (this is made formal in [4]). Correspondingly, [7] found that halfspaces can be tested more efficiently than they can be learned. In particular, while $\Omega(n/\epsilon)$ queries are required to learn halfspaces to accuracy $\epsilon$ (this follows from e.g. [6]), [7] show that $\epsilon$-testing halfspaces only requires $\text{poly}(1/\epsilon)$ queries, independent of the dimension $n$. 
In this work, we consider the problem of testing whether a function \( f \) belongs to a natural subclass of halfspaces, the class of \( \pm 1 \)-weight halfspaces. These are functions of the form \( f(x) = \text{sgn}(w_1 x_1 + w_2 x_2 + \cdots + w_n x_n) \) where the weights \( w_i \) all take values in \( \{-1, 1\} \). Included in this class is the majority function on \( n \) variables, and all \( 2^n \) “reorientations” of majority, where some variables \( x_i \) are replaced by \( -x_i \). Alternatively, this can be viewed as the subclass of halfspaces where all variables have the same amount of influence on the outcome of the function, but some variables get a “positive” vote while others get a “negative” vote.

For the problem of testing \( \pm 1 \)-weight halfspaces, we prove two main results:

1. **Lower Bound.** We show that any nonadaptive testing algorithm which distinguishes \( \pm 1 \)-weight halfspaces from functions that are \( \epsilon \)-far from \( \pm 1 \)-weight halfspaces must make at least \( \Omega(\log n) \) many queries. By a standard transformation (see e.g. \cite{3}), this also implies an \( \Omega(\log \log n) \) lower bound for adaptive algorithms. Taken together with \cite{7}, this shows that testing this natural subclass of halfspaces is more query-intensive than testing the general class of all halfspaces.

2. **Upper Bound.** We give a nonadaptive algorithm making \( O(\sqrt{n} \cdot \text{poly}(1/\epsilon)) \) many queries to \( f \), which outputs (i) YES with probability at least \( \frac{2}{3} \) if \( f \) is a \( \pm 1 \)-weight halfspace (ii) NO with probability at least \( \frac{2}{3} \) if \( f \) is \( \epsilon \)-far from any \( \pm 1 \)-weight halfspace.

We note that it follows from \cite{6} that learning the class of \( \pm 1 \)-weight halfspaces requires \( \Omega(n/\epsilon) \) queries. Thus, while some dependence on \( n \) is necessary for testing, our upper bound shows testing \( \pm 1 \)-weight halfspaces can still be done more efficiently than learning.

Although we prove our results specifically for the case of halfspaces with all weights \( \pm 1 \), we remark that similar results can be obtained using our methods for other similar subclasses of halfspaces such as \( \{-1, 0, 1\} \)-weight halfspaces (\( \pm 1 \)-weight halfspaces where some variables are irrelevant).

**Techniques.** As is standard in property testing, our lower bound is proved using Yao’s method. We define two distributions \( D_{YES} \) and \( D_{NO} \) over functions, where a draw from \( D_{YES} \) is a randomly chosen \( \pm 1 \)-weight halfspace and a draw from \( D_{NO} \) is a halfspace whose coefficients are drawn uniformly from \( \{+1, -1, +\sqrt{3}, -\sqrt{3}\} \). We show that a random draw from \( D_{NO} \) is with high probability \( \Omega(1) \)-far from every \( \pm 1 \)-weight halfspace, but that any set of \( o(\log n) \) query strings cannot distinguish between a draw from \( D_{YES} \) and a draw from \( D_{NO} \).

Our upper bound is achieved by an algorithm which uniformly selects a small set of variables and checks, for each selected variable \( x_i \), that the magnitude of the corresponding singleton Fourier coefficient \( |\hat{f}(i)| \) is close to the right value. We show that any function that passes this test with high probability must have its degree-1 Fourier coefficients very similar to those of some \( \pm 1 \)-weight halfspace, and that any function whose degree-1 Fourier coefficients have this property must be close to a \( \pm 1 \)-weight halfspace. At a high level this approach is similar to some of what is done in \cite{7}, but in the setting of the current paper this approach incurs a dependence on \( n \) because of the level of accuracy that is required to adequately estimate the Fourier coefficients.
2 Notation and Preliminaries

Throughout this paper, unless otherwise noted \( f \) will denote a Boolean function of the form \( f : \{-1, 1\}^n \to \{-1, 1\} \). We say that two Boolean functions \( f \) and \( g \) are \( \epsilon \)-far if \( \Pr_x[f(x) \neq g(x)] > \epsilon \), where \( x \) is drawn from the uniform distribution on \( \{-1, 1\}^n \).

We say that a function \( f \) is unate if it is monotone increasing or monotone decreasing as a function of variable \( x_i \) for each \( i \).

Fourier analysis. We will make use of standard Fourier analysis of Boolean functions. The set of functions from the Boolean cube \( \{-1, 1\}^n \) to \( \mathbb{R} \) forms a \( 2^n \)-dimensional inner product space with inner product given by \( \langle f, g \rangle = \mathbb{E}_x[f(x)g(x)] \). The set of functions \( \{\chi_S\}_{S \subseteq [n]} \) defined by \( \chi_S(x) = \prod_{i \in S} x_i \) forms a complete orthonormal basis for this space. Given a function \( f : \{-1, 1\}^n \to \mathbb{R} \) we define its Fourier coefficients by \( \hat{f}(S) = \mathbb{E}_x[f(x)x_S] \), and we have that \( f(x) = \sum_S \hat{f}(S)x_S \). We will be particularly interested in \( f \)’s degree-1 coefficients, i.e., \( \hat{f}(S) \) for \( |S| = 1 \); for brevity we will write these as \( \hat{f}(i) \) rather than \( \hat{f}(|i|) \). Finally, we have Plancherel’s identity \( \langle f, g \rangle = \sum_{S \subseteq \{1, \ldots, n\}} \hat{f}(S) \hat{g}(S) \), which has as a special case Parseval’s identity, \( \mathbb{E}_x[f(x)^2] = \sum_{S \subseteq \{1, \ldots, n\}} \hat{f}(S)^2 \). It follows that for every \( f : \{-1, 1\}^n \to \{-1, 1\} \) we have \( \sum_{S \subseteq \{1, \ldots, n\}} \hat{f}(S)^2 = 1 \).

Probability bounds. To prove our lower bound we will require the Berry-Esseen theorem, a version of the Central Limit Theorem with error bounds (see e.g. [2]):

**Theorem 1.** Let \( \ell(x) = c_1x_1 + \cdots + c_nx_n \) be a linear form over the random \( \pm 1 \) bits \( x_i \). Assume \( |c_i| \leq \tau \) for all \( i \) and write \( \sigma = \sqrt{\sum_c c_i^2} \). Write \( F \) for the c.d.f. of \( \ell(x)/\sigma \); i.e., \( F(t) = \Pr[\ell(x)/\sigma \leq t] \). Then for all \( t \in \mathbb{R} \),

\[
|F(t) - \Phi(t)| \leq O(\tau/\sigma) \cdot \frac{1}{1 + |t|^{3/2}},
\]

where \( \Phi \) denotes the c.d.f. of \( X \), a standard Gaussian random variable. In particular, if \( A \subseteq \mathbb{R} \) is any interval then \( |\Pr[\ell(x)/\sigma \in A] - \Pr[X \in A]| \leq O(\tau/\sigma) \).

A special case of this theorem, with a sharper constant, is also useful (the following can be found in [10]):

**Theorem 2.** Let \( \ell(x) \) and \( \tau \) be as defined in Theorem 1. Then for any \( \lambda \geq \tau \) and any \( \theta \in \mathbb{R} \) it holds that \( \Pr[|\ell(x) - \theta| \leq \lambda] \leq 6\lambda/\sigma \).

3 A \( \Omega(\log n) \) Lower Bound for Testing \( \pm 1 \)-Weight Halfspaces

In this section we prove the following theorem:

**Theorem 3.** There is a fixed constant \( \epsilon > 0 \) such that any nonadaptive \( \epsilon \)-testing algorithm \( A \) for the class of all \( \pm 1 \)-weight halfspaces must make at least \( (1/26) \log n \) many queries.
To prove Theorem 3, we define two distributions $D_{YES}$ and $D_{NO}$ over functions. The “yes” distribution $D_{YES}$ is uniform over all $2^n \pm 1$-weight halfspaces, i.e., a function $f$ drawn from $D_{YES}$ is $f(x) = \text{sgn}(r_1 x_1 + \cdots + r_n x_n)$ where each $r_i$ is independently and uniformly chosen to be $\pm 1$. The “no” distribution $D_{NO}$ is similarly a distribution over halfspaces of the form $f(x) = \text{sgn}(s_1 x_1 + \cdots + s_n x_n)$, but each $s_i$ is independently chosen to be $\pm \sqrt{1/2}$ or $\pm \sqrt{3/2}$ each with probability $1/4$.

To show that this approach yields a lower bound we must prove two things. First, we must show that a function drawn from $D_{NO}$ is with high probability far from any $\pm 1$-weight halfspace. This is formalized in the following lemma:

**Lemma 1.** Let $f$ be a random function drawn from $D_{NO}$. With probability at least $1 - o(1)$ we have that $f$ is $\epsilon$-far from any $\pm 1$-weight halfspace, where $\epsilon > 0$ is some fixed constant independent of $n$.

Next, we must show that no algorithm making $o(\log n)$ queries can distinguish $D_{YES}$ and $D_{NO}$. This is formalized in the following lemma:

**Lemma 2.** Fix any set $x^1, \ldots, x^q$ of $q$ query strings from $\{-1,1\}^n$. Let $\tilde{D}_{YES}$ be the distribution over $\{-1,1\}^q$ obtained by drawing a random $f$ from $D_{YES}$ and evaluating it on $x^1, \ldots, x^q$. Let $\tilde{D}_{NO}$ be the distribution over $\{-1,1\}^q$ obtained by drawing a random $f$ from $D_{NO}$ and evaluating it on $x^1, \ldots, x^q$. If $q = (1/26) \log n$ then $\|D_{YES} - \tilde{D}_{NO}\|_1 = o(1)$.

We prove Lemmas 1 and 2 in subsections 3.1 and 3.2 respectively. A standard argument using Yao’s method (see e.g. Section 8 of [3]) implies that the lemmas taken together prove Theorem 3.

### 3.1 Proof of Lemma 1.

Let $f$ be drawn from $D_{NO}$, and let $s_1, \ldots, s_n$ denote the coefficients thus obtained. Let $T_1$ denote $\{i : |s_i| = \sqrt{1/2}\}$ and $T_2$ denote $\{i : |s_i| = \sqrt{3/2}\}$. We may assume that both $|T_1|$ and $|T_2|$ lie in the range $[n/2 - \sqrt{n \log n}, n/2 + \sqrt{n \log n}]$ since the probability that this fails to hold is $1 - o(1)$. It will be slightly more convenient for us to view $f$ as $\text{sgn}(\sqrt{2}(s_1 x_1 + \cdots + s_n x_n))$, that is, such that all coefficients are of magnitude $1$ or $\sqrt{3}$.

It is easy to see that the closest $\pm 1$-weight halfspace to $f$ must have the same sign pattern in its coefficients that $f$ does. Thus we may assume without loss of generality that $f$’s coefficients are all $+1$ or $+\sqrt{3}$, and it suffices to show that $f$ is far from the majority function $\text{Maj}(x) = \text{sgn}(x_1 + \cdots + x_n)$.

Let $Z$ be the set consisting of those $z \in \{-1,1\}^{|T_2|}$ (i.e. assignments to the variables in $T_1$) which satisfy $S_{T_1} = \sum_{i \in T_2} z_i \in [\sqrt{n/2}, 2\sqrt{n/2}]$. Since we are assuming that $|T_2| \approx n/2$, using Theorem 1, we have that $|Z|/2^{|T_1|} = C_1 \pm o(1)$ for constant $C_1 = \Phi(2) - \Phi(1) > 0$.

Now fix any $z \in Z$, so $\sum_{i \in T_1} z_i$ is some value $V_z \cdot \sqrt{n/2}$ where $V_z \in [1, 2]$. There are $2^{n - |T_1|}$ extensions of $z$ to a full input $z' \in \{-1,1\}^n$. Let $C_{\text{Maj}}(z)$ be the fraction of those extensions which have $\text{Maj}(z') = -1$; in other words, $C_{\text{Maj}}(z)$ is the fraction of
strings in \(\{-1,1\}^T_2\) which have \(\sum_{i\in T_2} z_i < -V_z \sqrt{n/2}\). By Theorem 1, this fraction is \(\Phi(-V_z) \pm o(1)\). Let \(C_f(z)\) be the fraction of the \(2^n - |T_1|\) extensions of \(z\) which have \(f(x') = -1\). Since the variables in \(T_2\) all have coefficient \(\sqrt{3}\), \(C_f(z)\) is the fraction of strings in \(\{-1,1\}^T_2\) which have \(\sum_{i\in T_2} z_i < -(V_z/\sqrt{3})\sqrt{n/2}\), which by Theorem 1 is \(\Phi(-V_z/\sqrt{3}) \pm o(1)\).

There is some absolute constant \(c > 0\) such that for all \(z \in \mathbb{Z}, |C_f(z) - C_{\text{Maj}}(z)| \geq c\). Thus, for a constant fraction of all possible assignments to the variables in \(T_1\), the functions \(\text{Maj}\) and \(f\) disagree on a constant fraction of all possible extensions of the assignment to all variables in \(T_1 \cup T_2\). Consequently, we have that \(\text{Maj}\) and \(f\) disagree on a constant fraction of all assignments, and the lemma is proved.

### 3.2 Proof of Lemma 2.

For \(i = 1, \ldots, n\) let \(Y^i \in \{-1,1\}^q\) denote the vector of \((x^i_1, \ldots, x^i_q)\), that is, the vector containing the values of the \(i^{th}\) bits of each of the queries. Alternatively, if we view the \(n\)-bit strings \(x^1, \ldots, x^n\) as the rows of a \(q \times n\) matrix, the strings \(Y^1, \ldots, Y^n\) are the columns. If \(f(x) = \text{sgn}(a_1 x_1 + \cdots + a_n x_n)\) is a halfspace, we write \(\text{sgn}(\sum_{i=1}^n a_i Y^i)\) to denote \((f(x^1), \ldots, f(x^n))\), the vector of outputs of \(f\) on \(x^1, \ldots, x^n\); note that the value \(\text{sgn}(\sum_{i=1}^n a_i Y^i)\) is an element of \(\{-1,1\}^q\).

Since the statistical distance between two distributions \(D_1, D_2\) on a domain \(D\) of size \(N\) is bounded by \(N \cdot \max_{x \in D} |D_1(x) - D_2(x)|\), we have that the statistical distance \(\|\tilde{D}_{YES} - \tilde{D}_{NO}\|_1\) is at most \(2^q \cdot \max_{Q \in \{-1,1\}^q} |\text{Pr}_r[\text{sgn}(\sum_{i=1}^n r_i Y^i) = Q] - \text{Pr}_s[\text{sgn}(\sum_{i=1}^n s_i Y^i) = Q]|\). So let us fix an arbitrary \(Q \in \{-1,1\}^q\); it suffices for us to bound

\[
\left| \text{Pr}_r[\text{sgn}(\sum_{i=1}^n r_i Y^i) = Q] - \text{Pr}_s[\text{sgn}(\sum_{i=1}^n s_i Y^i) = Q] \right|. \tag{1}
\]

Let \(\text{In}Q\) denote the indicator random variable for the quadrant \(Q\), i.e. given \(x \in \mathbb{R}^q\) the value of \(\text{In}Q(x)\) is 1 if \(x\) lies in the quadrant corresponding to \(Q\) and is 0 otherwise. We have

\[
(1) = \left| \mathbf{E}_r[\text{In}Q(\sum_{i=1}^n r_i Y^i)] - \mathbf{E}_s[\text{In}Q(\sum_{i=1}^n s_i Y^i)] \right| \tag{2}
\]

We then note that since the \(Y^i\) vectors are of length \(q\), there are at most \(2^q\) possibilities in \(\{-1,1\}^q\) for their values which we denote by \(\tilde{Y}^1, \ldots, \tilde{Y}^{2^q}\). We lump together those vectors which are the same: for \(i = 1, \ldots, 2^q\) let \(c_i\) denote the number of times that \(\tilde{Y}^i\) occurs in \(Y^1, \ldots, Y^n\). We then have that \(\sum_{i=1}^n r_i Y^i = \sum_{i=1}^{2^q} a_i \tilde{Y}^i\) where each \(a_i\) is an independent random variable which is a sum of \(c_i\) independent \(\pm 1\) random variables (the \(r_j\)'s for those \(j\) that have \(Y^j = \tilde{Y}^i\)). Similarly, we have \(\sum_{i=1}^n s_i Y^i = \sum_{i=1}^{2^q} b_i \tilde{Y}^i\) where each \(b_i\) is an independent random variable which is a sum of \(c_i\) independent variables distributed as the \(s_j\)'s (these are the \(s_j\)'s for those \(j\) that have \(Y^j = \tilde{Y}^i\)). We thus can re-express (2) as

\[
\left| \mathbf{E}_a[\text{In}Q(\sum_{i=1}^{2^q} a_i \tilde{Y}^i)] - \mathbf{E}_b[\text{In}Q(\sum_{i=1}^{2^q} b_i \tilde{Y}^i)] \right|. \tag{3}
\]
Let us define a sequence of random variables that hybridize between $\sum_{i=1}^{2^q} a_i \tilde{Y}^i$ and $\sum_{i=1}^{2^q} b_i \tilde{Y}^i$. For $1 \leq \ell \leq 2^q + 1$ define

$$Z_\ell := \sum_{i<\ell} b_i \tilde{Y}^i + \sum_{i \geq \ell} a_i \tilde{Y}^i,$$

so $Z_1 = \sum_{i=1}^{2^q} a_i \tilde{Y}^i$ and $Z_{2^q+1} = \sum_{i=1}^{2^q} b_i \tilde{Y}^i$. As is typical in hybrid arguments, by telescoping (3), we have that (3) equals

$$\left| \mathbb{E}_{a,b} \left[ \sum_{\ell=1}^{2^q} \text{InQ}(Z_\ell) - \text{InQ}(Z_{\ell+1}) \right] \right| = \left| \sum_{\ell=1}^{2^q} \mathbb{E}_{a,b} \left[ \text{InQ}(Z_\ell) - \text{InQ}(Z_{\ell+1}) \right] \right|$$

where $W_\ell := \sum_{i<\ell} b_i \tilde{Y}^i + \sum_{i>\ell} a_i \tilde{Y}^i$. The RHS of (5) is at most

$$2^q \cdot \max_{\ell=1,\ldots,2^q} \left| \mathbb{E}_{a,b} \left[ \text{InQ}(W_\ell + a_{\ell} \tilde{Y}^{\ell}) - \text{InQ}(W_\ell + b_{\ell} \tilde{Y}^{\ell}) \right] \right|.$$

So let us fix an arbitrary $\ell$; we will bound

$$\left| \mathbb{E}_{a,b} \left[ \text{InQ}(W_\ell + a_{\ell} \tilde{Y}^{\ell}) - \text{InQ}(W_\ell + b_{\ell} \tilde{Y}^{\ell}) \right] \right| \leq B$$

(we will specify $B$ later), and this gives that $\| \tilde{D}_{YES} - \tilde{D}_{NO} \|_1 \leq 4^q B$ by the arguments above. Before continuing further, it is useful to note that $W_\ell$, $a_{\ell}$, and $b_{\ell}$ are all independent from each other.

**Bounding (6).** Let $N := (n/2^q)^{1/3}$. Without loss of generality, we may assume that the $c_i$’s are in monotone increasing order, that is $c_1 \leq c_2 \leq \ldots \leq c_{2^q}$. We consider two cases depending on the value of $c_\ell$. If $c_\ell > N$ then we say that $c_\ell$ is *big*, and otherwise we say that $c_\ell$ is *small*. Note that each $c_i$ is a nonnegative integer and $c_1 + \cdots + c_{2^q} = n$, so at least one $c_i$ must be big; in fact, we know that the largest value $c_{2^q}$ is at least $n/2^q$.

If $c_\ell$ is big, we argue that $a_{\ell}$ and $b_{\ell}$ are distributed quite similarly, and thus for any possible outcome of $W_\ell$ the LHS of (6) must be small. If $c_\ell$ is small, we consider some $k \neq \ell$ for which $c_k$ is very big (we just saw that $k = 2^q$ is such a $k$) and show that for any possible outcome of $a_{\ell}$, $b_{\ell}$ and all the other contributors to $W_\ell$, the contribution to $W_\ell$ from this $c_k$ makes the LHS of (6) small (intuitively, the contribution of $c_k$ is so large that it “swamps” the small difference that results from considering $a_{\ell}$ versus $b_{\ell}$).

**Case 1: Bounding (6) when $c_\ell$ is big, i.e. $c_\ell > N$.** Fix any possible outcome for $W_\ell$ in (6). Note that the vector $\tilde{Y}^{\ell}$ has all its coordinates ±1 and thus it is “skew” to each of the axis-aligned hyperplanes defining quadrant $Q$. Since $Q$ is convex, there is some interval $A$ (possibly half-infinite) of the real line such that for all $t \in R$ we have $\text{InQ}(W_\ell + t \tilde{Y}^{\ell}) = 1$ if and only if $t \in A$. It follows that

$$| \Pr_{a_{\ell}}[\text{InQ}(W_\ell + a_{\ell} \tilde{Y}^{\ell}) = 1] - \Pr_{b_{\ell}}[\text{InQ}(W_\ell + b_{\ell} \tilde{Y}^{\ell}) = 1] | = | \Pr[a_{\ell} \in A] - \Pr[b_{\ell} \in A] | .$$

(7)
Now observe that as in Theorem 1, \(a_\ell\) and \(b_\ell\) are each sums of \(c_\ell\) many independent zero-mean random variables (the \(r_j\)'s and \(s_j\)'s respectively) with the same total variance \(\sigma = \sqrt{\ell}\) and with each \(|r_j|, |s_j| \leq O(1)\). Applying Theorem 1 to both \(a_\ell\) and \(b_\ell\), we get that the RHS of (7) is at most \(O(1/\sqrt{\ell}) = O(1/\sqrt{N})\). Averaging the LHS of (7) over the distribution of values for \(W_\ell\), it follows that if \(c_\ell\) is big then the LHS of (6) is at most \(O(1/\sqrt{N})\).

**Case 2: Bounding (6) when \(c_\ell\) is small, i.e. \(c_\ell \leq N\).** We first note that every possible outcome for \(a_\ell, b_\ell\) results in \(|a_\ell - b_\ell| \leq O(N)\). Let \(k = 2^\ell\) and recall that \(c_k \geq n/2^\ell\). Fix any possible outcome for \(a_\ell, b_\ell\) and for all other \(a_j, b_j\) such that \(j \neq k\) (so the only “unfixed” randomness at this point is the choice of \(a_k\) and \(b_k\)). Let \(W'_\ell\) denote the contribution to \(W_\ell\) from these \(2^\ell - 2\) fixed \(a_j, b_j\) values, so \(W_\ell = W'_\ell + a_k\bar{Y}^k\) (since \(k > \ell\)). (Note that under this supposition there is actually no dependence on \(b_k\) now; the only randomness left is the choice of \(a_k\).)

We have

\[
\begin{align*}
|\Pr_{a_k}[\ln Q(W_\ell + a_\ell \bar{Y}^\ell) = 1] - \Pr_{a_k}[\ln Q(W_\ell + b_\ell \bar{Y}^\ell) = 1]| \\
= |\Pr_{a_k}[\ln Q(W'_\ell + a_\ell \bar{Y}^\ell + a_k\bar{Y}^k) = 1] - \Pr_{a_k}[\ln Q(W'_\ell + b_\ell \bar{Y}^\ell + a_k \bar{Y}^k) = 1]| \quad (8)
\end{align*}
\]

The RHS of (8) is at most

\[
\Pr_{a_k}[\text{the vector } W'_\ell + a_\ell \bar{Y}^\ell + a_k \bar{Y}^k \text{ has any coordinate of magnitude at most } |a_\ell - b_\ell|]. \quad (9)
\]

(If each coordinate of \(W'_\ell + a_\ell \bar{Y}^\ell + a_k \bar{Y}^k\) has magnitude greater than \(|a_\ell - b_\ell|\), then each corresponding coordinate of \(W'_\ell + b_\ell \bar{Y}^\ell + a_k \bar{Y}^k\) must have the same sign, and so such an outcome affects each of the probabilities in (8) in the same way – either both points are in quadrant \(Q\) or both are not.) Since each coordinate of \(\bar{Y}^k\) is of magnitude 1, by a union bound the probability (9) is at most \(q\) times

\[
\max_{\text{all intervals } A \text{ of width } 2|a_\ell - b_\ell|} \Pr_{a_k}[a_k \in A]. \quad (10)
\]

Now using the fact that \(|a_\ell - b_\ell| = O(N)\), the fact that \(a_k\) is a sum of \(c_k \geq n/2^\ell\) independent \(\pm 1\)-valued variables, and Theorem 2, we have that (10) is at most \(O(Nq\sqrt{2^\ell})/\sqrt{n}\). So we have that (8) is at most \(O(Nq\sqrt{2^\ell})/\sqrt{n}\). Averaging (8) over a suitable distribution of values for \(a_1, b_1, \ldots, a_{k-1}, b_{k-1}, a_{k+1}, b_{k+1}, \ldots, a_{2^\ell}, b_{2^\ell}\), gives that the LHS of (6) is at most \(O(Nq\sqrt{2^\ell})/\sqrt{n}\).

So we have seen that whether \(c_\ell\) is big or small, the value of (6) is upper bounded by

\[
\max\{O(1/\sqrt{N}), O(Nq\sqrt{2^\ell})/\sqrt{n}\}.
\]

Recalling that \(N = (n/2^\ell)^{1/3}\), this equals \(O(q^{2^\ell/n^{1/6}})\), and thus \(\|\tilde{D}_{YES} - \tilde{D}_{NO}\|_1 \leq O(q^{2^{13q/6}/n^{1/6}})\). Recalling that \(q = (1/26) \log n\), this equals \(O((\log n)/n^{1/12}) = o(1)\), and Lemma 2 is proved.
4 A Sublinear Algorithm for Testing ±1-Weight Halfspaces

In this section we present the ±1-Weight Halfspace-Test algorithm, and prove the following theorem:

**Theorem 4.** For any \(36/n < \varepsilon < 1/2\) and any function \(f : \{-1, 1\}^n \rightarrow \{-1, 1\}\),

- if \(f\) is a ±1-weight halfspace, then ±1-Weight Halfspace-Test \((f, \varepsilon)\) passes with probability \(\geq 2/3\),
- if \(f\) is \(\varepsilon\)-far from any ±1-weight halfspace, then ±1-Weight Halfspace-Test \((f, \varepsilon)\) rejects with probability \(\geq 2/3\).

The query complexity of ±1-Weight Halfspace-Test \((f, \varepsilon)\) is \(O(\sqrt{n/\varepsilon \log 1/\varepsilon})\). The algorithm is nonadaptive and has two-sided error.

The main tool underlying our algorithm is the following theorem, which says that if most of \(f\)’s degree-1 Fourier coefficients are almost as large as those of the majority function, then \(f\) must be close to the majority function. Here we adopt the shorthand \(\text{Maj}_n\) to denote the majority function on \(n\) variables, and \(\hat{\text{M}}_n\) to denote the value of the degree-1 Fourier coefficients of \(\text{Maj}_n\).

**Theorem 5.** Let \(f : \{-1, 1\}^n \rightarrow \{-1, 1\}\) be any Boolean function and let \(\varepsilon > 36/n\). Suppose that there is a subset of \(m \geq (1 - \varepsilon)n\) variables \(i\) each of which satisfies \(\hat{f}(i) \geq (1 - \varepsilon)\hat{\text{M}}_n\). Then \(\Pr[f(x) \neq \text{Maj}_n(x)] \leq 32\sqrt{\varepsilon}\).

In the following subsections we prove Theorem 5 and then present our testing algorithm.

4.1 Proof of Theorem 5.

Recall the following well-known lemma, whose proof serves as a warmup for Theorem 5:

**Lemma 3.** Every \(f : \{-1, 1\}^n \rightarrow \{-1, 1\}\) satisfies \(\sum_{i=1}^n |\hat{f}(i)| \leq n\hat{\text{M}}_n\).

**Proof.** Let \(G(x) = \text{sgn}(\hat{f}(1))x_1 + \cdots + \text{sgn}(\hat{f}(n))x_n\) and let \(g(x) = \text{sgn}(G(x))\). We have

\[
\sum_{i=1}^n |\hat{f}(i)| = \mathbb{E}[|fG|] \leq \mathbb{E}[|G||] = \mathbb{E}[|G(x)g(x)|] = \sum_{i=1}^n \hat{\text{M}}_n,
\]

where the first equality is Plancherel (using the fact that \(G\) is linear), the inequality is because \(f\) is a ±1-valued function, the second equality is by definition of \(g\) and the third equality is Plancherel again, observing that each \(\hat{g}(i)\) has magnitude \(\hat{\text{M}}_n\) and sign \(\text{sgn}(\hat{f}(i))\).

**Proof of Theorem 5.** For notational convenience, we assume that the variables whose Fourier coefficients are “almost right” are \(x_1, x_2, \ldots, x_m\). Now define \(G(x) = x_1 + \ldots +
$x_2 + \cdots + x_n$, so that $\text{Maj}_n = \text{sgn}(G)$. We are interested in the difference between the following two quantities:

$$E[|G(x)|] = E[G(x)\text{Maj}_n(x)] = \sum_S \hat{G}(S)\hat{\text{Maj}}_n(S) = \sum_{i=1}^n \hat{\text{Maj}}_n(i) = n\hat{M}_n,$$

$$E[G(x)f(x)] = \sum_S \hat{G}(S)\hat{f}(S) = \sum_{i=1}^m \hat{f}(i) = \sum_{i=m+1}^n \hat{f}(i).$$

The bottom quantity is broken into two summations. We can lower bound the first summation by $(1 - \epsilon)^2 n\hat{M}_n \geq (1 - 2\epsilon)n\hat{M}_n$. This is because the first summation contains at least $(1 - \epsilon)n$ terms, each of which is at least $(1 - \epsilon)\hat{M}_n$. Given this, Lemma 3 implies that the second summation is at least $-2\epsilon n\hat{M}_n$. Thus we have

$$E[G(x)f(x)] \geq (1 - 4\epsilon)n\hat{M}_n$$

and hence

$$E[|G| - Gf] \leq 4\epsilon n\hat{M}_n \leq 4\epsilon \sqrt{n} \tag{11}$$

where we used the fact (easily verified from Parseval’s equality) that $M_n \leq \frac{1}{\sqrt{n}}$.

Let $p$ denote the fraction (easily verified from Parseval’s equality) that $M_n \leq \frac{1}{\sqrt{n}}$.

Let $k$ be such that $\sqrt{\epsilon} = (4k + 2)/\sqrt{n}$, so in particular $k \geq 1$. It is well known (by Stirling’s approximation) that each “layer” $\{x \in \{-1, 1\}^n : x_1 + \cdots + x_n = \ell\}$ of the Boolean cube contains at most a $\frac{1}{\sqrt{n}}$ fraction of $\{-1, 1\}^n$, and consequently at most a $\frac{2k + 1}{\sqrt{n}}$ fraction of points have $|G(x)| \leq 2k$. It follows that at least a $p/2$ fraction of points satisfy both $|G(x)| > 2k$ and $f(x) \neq \text{Maj}_n(x)$. Since $|G(x)| - G(x)f(x)$ is at least $4k$ on each such point and $|G(x)| - G(x)f(x)$ is never negative, this implies that the LHS of (11) is at least

$$\frac{p}{2} \cdot 4k \geq (16\sqrt{\epsilon}) \cdot (4k) \geq (16\sqrt{\epsilon})/(2k + 1) = (16\sqrt{\epsilon}) \cdot 8\epsilon \sqrt{n},$$

but this contradicts (11). This proves the theorem. \qed

### 4.2 A Tester for ±1-Weight Halfspaces.

Intuitively, our algorithm works by choosing a handful of random indices $i \in [n]$, estimating the corresponding $|\hat{f}(i)|$ values (while checking unateness in these variables), and checking that each estimate is almost as large as $\hat{M}_n$. The correctness of the algorithm is based on the fact that if $f$ is unate and most $|\hat{f}(i)|$ are large, then some reorientation of $f$ (that is, a replacement of some $x_i$ by $-x_i$) will make most $\hat{f}(i)$ large. A simple application of Theorem 5 then implies that the reorientation is close to $\text{Maj}_n$, and therefore that $f$ is close to a ±1-weight halfspace.

We start with some preliminary lemmas which will assist us in estimating $|\hat{f}(i)|$ for functions that we expect to be unate.
Lemma 4.

\[ \hat{f}(i) = \Pr_x[f(x^-) < f(x^+)] - \Pr_x[f(x^-) > f(x^+)] \]

where \( x^- \) and \( x^+ \) denote the bit-string \( x \) with the \( i \)th bit set to \(-1\) or \(1\) respectively.

We refer to the first probability above as the positive influence of variable \( i \) and the second probability as the negative influence of \( i \). Each variable in a monotone function has only positive influence. Each variable in a unate function has only positive influence or negative influence, but not both.

Proof. (of Lemma 4) First note that \( \hat{f}(i) = E_x[f(x)x_i] \), then

\[ E_x[f(x)x_i] = \Pr_x[f(x) = 1, x_i = 1] + \Pr_x[f(x) = -1, x_i = -1] \]
\[ - \Pr_x[f(x) = -1, x_i = 1] - \Pr_x[f(x) = 1, x_i = -1]. \]

Now group all \( x \)'s into pairs \( (x^-, x^+) \) that differ in the \( i \)th bit. If the value of \( f \) is the same on both elements of a pair, then the total contribution of that pair to the expectation is zero. On the other hand, if \( f(x^-) < f(x^+) \), then \( x^- \) and \( x^+ \) each add \( \frac{1}{\delta} \) to the expectation, and if \( f(x^-) > f(x^+) \), then \( x^- \) and \( x^+ \) each subtract \( \frac{1}{\delta} \). This yields the desired result.

Lemma 5. Let \( f \) be any Boolean function, \( i \in [n] \), and let \( |\hat{f}(i)| = p \). By drawing \( m = \frac{\log \frac{\delta}{\epsilon}}{p^2} \) uniform random strings \( x \in \{-1, 1\}^n \), and querying \( f \) on the values \( f(x^+) \) and \( f(x^-) \), with probability \( 1 - \delta \) we either obtain an estimate of \( |\hat{f}(i)| \) accurate to within a multiplicative factor of \( 1 \pm \epsilon \), or discover that \( f \) is not unate.

The idea of the proof is that if neither the positive influence nor the negative influence is small, random sampling will discover that \( f \) is not unate. Otherwise, \( |\hat{f}(i)| \) is well approximated by either the positive or negative influence, and a standard multiplicative form of the Chernoff bound shows that \( m \) samples suffice.

Proof. (of Lemma 5) Suppose first that both the positive influence and negative influence are at least \( \frac{\epsilon}{10} \). Then the probability that we do not observe any pair with positive influence is \( \leq (1 - \frac{\epsilon}{10})^m \leq e^{-\epsilon pm/2} = e^{-\epsilon p(3/2) \log(2/\delta)} < \frac{\delta}{2} \), and similarly for the negative influence. Therefore, the probability that we observe at least some positive influence and some negative influence (and therefore discover that \( f \) is not unate) is at least \( 1 - 2\delta = 1 - \delta \).

Now consider the case when either the positive influence or the negative influence is less than \( \frac{\epsilon}{10} \). Without loss of generality, assume that the negative influence is less than \( \frac{\epsilon}{10} \). Then the positive influence is a good estimate of \( |\hat{f}(i)| \). In particular, the probability that the estimate of the positive influence is not within \( (1 \pm \frac{\epsilon}{2})p \) of the true value (and therefore the estimate of \( |\hat{f}(i)| \) is not within \( (1 \pm \epsilon)p \)), is at most \( 2e^{-mp^2/3} = 2e^{-\log \frac{2}{\delta}} = \delta \) by the multiplicative Chernoff bound. So in this case, the probability that the estimate we receive is accurate to within a multiplicative factor of \( 1 \pm \epsilon \) is at least \( 1 - \delta \). This concludes the proof.

Now we are ready to present the algorithm and prove its correctness.
\[ \pm 1\text{-Weight Halfspace-Test} \text{ (inputs are } \epsilon > 0 \text{ and black-box access to } f : \{-1,1\}^n \rightarrow \{-1,1\}) \]

1. Let \( \epsilon' = \left(\frac{\epsilon}{32}\right)^2 \).
2. Choose \( k = \frac{1}{2\epsilon^2} \ln 6 = O\left(\frac{1}{\epsilon^2}\right) \) many random indices \( i \in \{1, \ldots, n\} \).
3. For each \( i \), estimate \( |\hat{f}(i)| \). Do this as in Lemma 5 by drawing \( m = \frac{24\log 12k}{\hat{M}_n \epsilon^2} = O\left(\frac{\sqrt{\epsilon^2 \log \frac{1}{\epsilon'}}}{\epsilon}ight) \) random \( x \)'s and querying \( f(x^+) \) and \( f(x^-) \). If a violation of unateness is found, reject.
4. Pass if and only if each estimate is larger than \( (1 - \epsilon')\hat{M}_n \).

**Proof. (of Theorem 4)** To prove that the test is correct, we need to show two things: first that it passes functions which are \( \pm 1 \)-weight halfspaces, and second that anything it passes with high probability must be \( \epsilon \)-close to a \( \pm 1 \)-weight halfspace. To prove the first, note that if \( f \) is a \( \pm 1 \)-weight halfspace, the only possibility for rejection is if any of the estimates of \( |\hat{f}(i)| \) is less than \( (1 - \epsilon')\hat{M}_n \). But applying lemma 5 (with \( p = \hat{M}_n \), \( \epsilon = \frac{\epsilon'}{32}, \delta = \frac{1}{64k} \)), the probability that a particular estimate is wrong is less than \( \frac{1}{64k} \), and therefore the probability that any estimate is wrong is less than \( \frac{1}{6} \). Thus the probability of success is at least \( \frac{5}{6} \).

The more difficult part is showing that any function which passes the test whp must be close to a \( \pm 1 \)-weight halfspace. To do this, note that if \( f \) passes the test whp then it must be the case that for all but a \( \epsilon' \) fraction of variables, \( |\hat{f}(i)| \) is less than \( (1 - \epsilon')\hat{M}_n \). If this is not the case, then Step 2 will choose a “bad” variable – one for which \( |\hat{f}(i)| \leq (1 - \epsilon')\hat{M}_n \) – with probability at least \( \frac{5}{6} \). Now we would like to show that for any bad variable \( i \), the estimate of \( |\hat{f}(i)| \) is likely to be less than \( (1 - \epsilon')\hat{M}_n \). Without loss of generality, assume that \( |\hat{f}(i)| = (1 - \epsilon')\hat{M}_n \) (if \( |\hat{f}(i)| \) is less than that, then variable \( i \) will be even less likely to pass step 3). Then note that it suffices to estimate \( |\hat{f}(i)| \) to within a multiplicative factor of \( 1 + \frac{\epsilon'}{2} \) (since \( (1 + \frac{\epsilon'}{2} - (1 - \epsilon')\hat{M}_n < (1 - \epsilon')\hat{M}_n) \)).

Again using Lemma 5 (this time with \( p = (1 - \epsilon')\hat{M}_n, \epsilon = \frac{\epsilon'}{2}, \delta = \frac{1}{64k} \)), we see that \( \frac{12k}{\hat{M}_n^2(1 - \epsilon')} \log 12k < \frac{24\log 12k}{\hat{M}_n^2} \log 12k \) samples suffice to achieve this. The total probability of failure (the probability that we fail to choose a bad variable, or that we mis-estimate one when we do) is thus less than \( \frac{1}{6} + \frac{1}{64k} < \frac{1}{3} \).

The query complexity of the algorithm is \( O(km) = O\left(\sqrt{n} \frac{1}{\epsilon^2} \log \frac{1}{\epsilon'}\right) = O\left(\sqrt{n} \cdot \frac{1}{\epsilon^2} \log \frac{1}{\epsilon'}\right) \).

## 5 Conclusion

We have proven a lower bound showing that the complexity of testing \( \pm 1 \)-weight halfspaces is at least \( \Omega(\log n) \) and an upper bound showing that it is at most \( O\left(\sqrt{n} \cdot \text{poly}\left(\frac{1}{\epsilon}\right)\right) \). An open question is to close the gap between these bounds and determine the exact dependence on \( n \). One goal is to use some type of binary search to get a poly log \( (n) \)-query adaptive testing algorithm; another is to improve our lower bound to \( n^{\Omega(1)} \) for nonadaptive algorithms.
References