JMLR: Workshop and Conference Proceedings vol (2010) 15.1–15.10 25th Annual Conference on Learning Theory

# Learning Functions of Halfspaces Using Prefix Covers

**Parikshit Gopalan** PARIK@MICROSOFT.COM
*MSR Silicon Valley, Mountain View, California.*

**Adam R. Klivans** KLIVANS@CS.UTEXAS.EDU
*Department of Computer Science, UT Austin.*

**Raghu Meka** RAGHU@IAS.EDU
*IAS, Princeton.*

**Editor:** Shie Mannor, Nathan Srebro, Robert C. Williamson

## Abstract

We present a simple query-algorithm for learning arbitrary functions of $k$ halfspaces under any product distribution on the Boolean hypercube. Our algorithms learn any function of $k$ halfspaces to within accuracy $\varepsilon$ in time $O((nk/\varepsilon)^{k+1})$ under any product distribution on $\{0,1\}^n$ using read-once branching programs as a hypothesis.. This gives the first $\mathrm{poly}(n, 1/\varepsilon)$ algorithm for learning even the intersection of 2 halfspaces under the uniform distribution on $\{0,1\}^n$; previously known algorithms had an exponential dependence either on the accuracy parameter $\varepsilon$ or the dimension $n$.

To prove this result, we identify a new structural property of Boolean functions that yields learnability with queries: that of having a small prefix cover.

**Keywords:** Boolean functions, Halfspaces, Membership queries, Branching programs.

## 1. Introduction

The problem of learning functions of a few halfspaces is one of the most well-studied problems in computational learning. It includes as a special case the problem of learning intersections of halfspaces aka convex sets. There is a large body of work in computational learning dedicated to this problem (Baum, 1990; Blum and Kannan, 1997; Vempala, 1997, 2010b; Klivans et al., 2004; Kalai et al., 2008; Klivans et al., 2008, 2009; Harsha et al., 2010; Vempala, 2010a). The problem of PAC-learning under arbitrary distributions seems intractable, hence researchers have studied learnability various natural distributions such as the Gaussian distribution on $\mathbb{R}^n$, log-concave distributions on $\mathbb{R}^n$, and the uniform distribution on $\{0,1\}^n$.

Currently the best known algorithm for learning intersections of $k$ halfspaces in Gaussian space is due to Vempala (2010a), whose running time is bounded by $\mathrm{poly}(n, 1/\varepsilon, k)+n\min((k/\varepsilon)^k, k^{\log k/\varepsilon^4})$. This improves on a result by Klivans et al. (2008) which achieves a running time of $O(n^{\log k/\varepsilon^4})$ (even for agnostic learning) and earlier work by Vempala (1997, 2010b) which achieved a running time of $O((nk/\varepsilon)^k)$ (under any log-concave distribution). In summary, for constant $k$, we have algorithms with a running time of $\mathrm{poly}(n, 1/\varepsilon)$.

In the setting of the uniform distribution on Boolean hypercube, the known results are much weaker. The best previously known algorithm for learning intersection (or arbitrary functions) of $k$ halfspaces under the uniform distribution on the Boolean hypercube is due to Klivans et al. (2004). They showed that the low-degree algorithm of Linial, Mansour and Nisan (Linial et al., 1993) can learn arbitrary functions of $k$ halfspaces to accuracy $\varepsilon$ in time $O(n^{k^2/\varepsilon^2})$. Their bound is polynomial

© 2010 P. Gopalan, A.R. Klivans & R. Meka.

for any fixed $\varepsilon$ but becomes trivial for $\varepsilon = 1/\sqrt{n}$. Harsha et al. (2010) give an algorithm that can learn the intersection of $k$ $\varepsilon$-regular halfspaces in time $O(n^{\mathrm{poly}(\log k, 1/\varepsilon)})$ (we refer the reader to Harsha et al. (2010) for the definition of regularity). It is conjectured that a similar bound should hold for intersections of arbitrary halfspaces (Klivans et al., 2004; O'Donnell, 2012). If so, this would yield a similar running time for the intersections of $k$ arbitrary halfspaces.

The aforementioned results for the Boolean hypercube use polynomial approximations for functions of halfspaces. The exponential dependence on $\varepsilon$ is unavoidable via this approach. A different algorithm for learning function of $k$ halfspaces based on query learning of automata was presented in Klivans et al. (2004), its has a running time of $\mathrm{poly}(n^{2^k}, W^{2^k}, 1/\varepsilon)$ where $W$ is a bound on the weight of all halfspaces. It is known that there exist halfspaces for which $W$ needs to be exponential in $n$. Thus, while the dependence on $\varepsilon$ is polynomial, the dependence on $n$ could be exponential.

Thus, even for the intersection of two halfspaces on the hypercube under the uniform distribution with membership queries, there were no algorithms known with a running time of $\mathrm{poly}(n, 1/\varepsilon)$. For polynomially small error, there was no sub-exponential algorithm known.

### 1.1. Our Results

We present a query learning algorithm for learning functions of $k$ halfspaces with queries whose running time is polynomial in $n$ and $\varepsilon$ for constant $k$.

**Theorem 1 (Learning Functions of Halfspaces)** *The concept class of arbitrary Boolean functions of $k$ halfspaces can be PAC learned with membership queries under the uniform distribution $\{0,1\}^n$ to accuracy $\varepsilon$ in time $\tilde{O}((16nk/\varepsilon)^{k+1})$.*

We give a simple combinatorial algorithm which outputs a read-once branching program (ROBP) as its hypothesis. Analogous statements hold for any product distribution. Our work builds on classical results on learning branching programs using queries (Angluin, 1987; Beimel et al., 2000) and more recent work on approximating halfspaces using ROBPs (Meka and Zuckerman, 2010; Gopalan et al., 2011b); we next discuss the relation between our work and these results.

Learning branching programs is a classic area of study in computational learning theory. It is well-known that learning even width-five, polynomial-size branching programs is intractable under plausible cryptographic assumptions (Barrington, 1989; Kearns and Valiant, 1994). As a result, researchers have considered the possibility of learning natural subclasses of branching programs. One of the most common restrictions is to assume that the branching program is *read-once*; that is, no variable of the input is examined more than once on any traversal of the program. Additionally it is common to assume that the ordering of the variables is known to the learner apriori. The seminal work of Angluin (Angluin, 1987) on learning finite automata from membership and equivalence queries yields an efficient algorithm for learning read-once branching programs (ROBPs), as ROBPs can be viewed as special cases of polynomial-size finite automata.

Using Angluin's algorithm, Klivans et al. (2004) derive a query learning algorithm for halfspaces whose running time depends on the size of the smallest ROBP for that halfspace (they use the weight $W$ of the halfspace to bound this size). However this size can be exponential in $n$. The size can be reduced to polynomial in $n$ if one is willing to settle for approximations. This was shown in recent work on pseudorandom generators for halfspaces by Meka and Zuckerman (2010), who prove that every halfspace can be $\varepsilon$-approximated by an ROBP whose size is $\mathrm{poly}(n, 1/\varepsilon)$. Their

result also implies the existence of small ROBPs for approximating arbitrary functions of $k$ halfspaces. Thus, a learning algorithm for ROBPs that can tolerate small amounts of noise (to handle the approximation error) will directly yield better learning algorithms for functions of halfspaces.

Unfortunately, it is unclear how to analyze the performance of existing automata-based methods for learning in the presence of noise. At a very high level, these algorithms use membership queries to reconstruct the transitions among underlying states. If one of these queries returns a noisy label, the reconstructed automaton may have incorrect transitions, and its error with respect to the true concept may be large. Indeed, we do not know how to solve the problem of learning ROBPs with noise, and as we point out in Section 4, such a result will have several interesting consequences.

Instead, we identify a new structural property of Boolean functions that yields learnability with queries: that of having a small prefix cover. Given a function $f : \{0,1\}^n \to \{0,1\}$, every prefix $x \in \{0,1\}^i$ induces a function $f_x : \{0,1\}^{n-i} \to \{0,1\}$ given by $f_x(z) = f(x \circ z)$. We say that a function $f$ is $(\varepsilon, W)$-prefix coverable if for every length $i \in \{1, \ldots, n\}$, there is a set of at most $W$ special prefixes $x_1, \ldots, x_W$ so that for every other $x \in \{0,1\}^i$, $f_x$ is $\varepsilon$-close to $f_{x_j}$. We say that a function has small prefix covers if we can take $W = \mathrm{poly}(n, 1/\varepsilon)$.

Having a small width ROBP easily implies the existance of small prefix covers. We show that the existance of a small prefix covers implies that the function can be approximated by an ROBP. This inclusion comes as a corollary of our main algorithm which learns concepts with small prefix covers using a small ROBP as hypothesis. It is worth noting that both the inclusions above are proper; thus having a small prefix cover is a stronger requirement than being approximated by an ROBP, but weaker then being computed exactly by an ROBP.

We prove the following result for learning functions with small prefix covers.

**Theorem 2 (Learning Prefix-Coverable Functions)** *There is a membership query algorithm which when given oracle access to a $(\varepsilon, W)$-prefix coverable function $f$, outputs a hypothesis which is a $W$-ROBP $M$ such that $d(M, f) \leq 4n\varepsilon$. The algorithm runs in time* $\mathrm{poly}(n, W, 1/\varepsilon)$.

Our algorithm is a natural modification of Angluin's algorithm for learning finite automata (Angluin, 1987). From this, we obtain our result for functions of halfspaces by showing that they have small prefix covers. The novel ingredient in this work is the notion of prefix covers, which seems to be the right relaxation of being computed exactly by a small-width branching program.

### 1.2. Other Related Work

Many researchers in computational learning theory have studied the problem of learning functions computable by read-once branching programs (for a discussion see Bshouty et al. (1998)). Positive results were known only for restricted classes of ROBPs, such as width-2 ROBPs (Ergün et al., 1995; Bshouty et al., 1998) (these algorithms use queries and succeed in the distribution-free model of learning) and do not apply in our setting.

Approximations of halfspaces by ROBPs have recently been used in work on pseudorandomness and derandomization. Meka and Zuckerman show that halfspaces can in fact be "sandwiched" between two small width ROBPs (Meka and Zuckerman, 2010). They use this to construct the first pseudorandom generators for halfspaces whose seed-length depends logarithmically on the error parameter. Recent work by Gopalan et al. (2011b) uses ROBPs to derive deterministic approximate counting algorithms for knapsack and some related problems. To achieve this, they show that the approximating ROBPs can be constructed algorithmically.

While our work is inspired by these works, the notion of prefix covers differs from the notions of approximation by ROBPs that is used by those papers.

## 2. ROBPs, Halfspaces and Prefix Covers

We start by defining ROBPs which are the hypotheses class for our learning algorithms.

**Definition 3 (Read-Once Branching Programs)** *A width $W$ read-once branching program $M$ ($W$-ROBP for short) is a layered multi-graph with a layer for each $0 \leq i \leq n$ and at most $W$ vertices each layer.*

- *Let $L(M,i)$ denote the vertices in layer $i$ of $M$. $L(M,0)$ consists of a single vertex $v_0$ and each vertex in $L(M,n)$ is labeled with $0$ (rejecting) or $1$ (accepting).*
- *For $0 \leq i \leq n$, a vertex $v \in L(M,i)$ has two outgoing edges labeled $\{0,1\}$ and ending at vertices in $L(M,i+1)$. The edges correspond to the possible values for the variable $x_i$.*
- *For a string $z$, $M(v,z)$ denotes the state reached by starting from $v$ and following edges labeled with $z$. For $z \in \{0,1\}^n$, let $M(z) = 1$ if $M(v_0,z)$ is an accepting state, and $M(z) = 0$ otherwise. Thus we view $M$ as both an ROBP and a Boolean function.*

Note that the branching programs in the above definition are necessarily read-once, hence we refer to them as read-once branching programs or ROBPs. Also, the ordering of the variables $x_1, \ldots, x_n$ is important: there are Boolean functions that have small ROBPs only under certain orderings of the variables. Henceforth, when we refer to a function having small ROBPs, we mean the ordering $x_1, \ldots, x_n$.

**Definition 4 (Halfspaces)** *A halfspace $h : \{0,1\}^n \to \{0,1\}$ is a Boolean function defined by $f(x) = 1$ if $\sum_i a_i x_i \leq b$ and $0$ otherwise, where $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$.*

Given a halfspace $h$, let $W(h) = \min(\sum_{i=1}^n |a_i| + |b|)$ denote the minimum weight over all possible ways of representing the halfspace. There is a natural $W(h)$-ROBP for any halfspace $h$ which keeps track of partial sums. But it is easy to construct halfspaces $h$ where $W(h)$ is exponential in $n$, and where any branching program computing $h$ must have $S = \exp(n)$; the Greater Than function is one example. One could relax the requirement and ask for an ROBP that approximates the halfspace $h$. Meka and Zuckerman show that every halfspace can be well approximated by a $\mathrm{poly}(n)$-width ROBP (Meka and Zuckerman, 2010).

Let $f : \{0,1\}^n \to \{0,1\}$ and let $\mu_n$ denote the uniform distribution over $\{0,1\}^n$. For two functions $f, g : \{0,1\}^n \to \{0,1\}$, we define $d(f,g) = \mathsf{Pr}_{x \leftarrow \mu_n}[f(x) \neq g(x)]$.

**Theorem 5 (Appropximating halfspaces by ROBPs)** *(Meka and Zuckerman, 2010) For any halfspace $h : \{0,1\}^n \to \{0,1\}$ and any $\varepsilon > 0$, there is an $O(\frac{n}{\varepsilon}$-ROBP $M$ such that $d(h,M) \leq \varepsilon$.*

A similar approximation result holds for arbitrary functions of $k$ halfspaces. Thus if one could design an agnostic learning algorithm for learning ROBPs (which can even tolerate small amounts of noise), one could then use it for learning functions of $k$ halfspaces. However, none of the known algorithms for ROBP learning appear to be robust to (even small amounts of) adversarial noise.

### 2.1. Prefix-Coverable Functions

We introduce the notion of Prefix-Covers for a class of functions. This class includes all functions that are computable by ROBPs but not all functions that can be approximated by them. Crucially for us, it also includes arbitrary functions of $k$ halfspaces. Our learning results hold for all functions that have small Prefix Covers.

Let $f : \{0,1\}^n \to \{0,1\}$ and let $\mu_i$ denote the uniform distribution over $\{0,1\}^i$. For each prefix $x \in \{0,1\}^i$, we define the function $f_x : \{0,1\}^{n-i} \to \{0,1\}$ by $f_x(z) = f(x \circ z)$ where $\circ$ denotes concatenation. Thus given two prefixes $x, y \in \{0,1\}^i$, $d(f_x, f_y) = \mathsf{Pr}_{z \leftarrow \mu_{n-i}}[f(x \circ z) \neq f(y \circ z)]$.

**Definition 6 (Prefix Coverable Functions)** *A function $f : \{0,1\}^n \to \{0,1\}$ is $(\varepsilon, W)$-prefix coverable if there exist sets $S^i \subseteq \{0,1\}^i$, $i \in \{1, \ldots, n\}$ with $|S^i| \leq W$ such that for every $y \in \{0,1\}^i$ there exists an $x \in S^i$ such that $d(f_x, f_y) \leq \varepsilon$. The sets $S^1, \ldots, S^n$ are called $(\varepsilon, W)$-prefix covers for $f$.*

It is easy to see that functions computed by ROBPs have small prefix covers with $\varepsilon = 0$, in fact these conditions are equivalent.

**Lemma 7** *A function $f$ is computable by a $W$-ROBP iff $f$ is $(0, W)$-prefix coverable.*

**Proof** If $f$ is computable by a $W$-ROBP $M$, we can get prefix covers for prefixes of length $i$ by picking one representative string for each state in $L(M, i)$. The other direction is via Angluin's algorithm, or equivalently by using $\varepsilon = 0$ in Corollary 8 below. ■

If we allow $\varepsilon > 0$ in our prefix cover, we get a richer class of functions than small width ROBPs. To see this, observe that while not all halfspaces have small ROBPs, every halfspace is $(\varepsilon, 2/\varepsilon)$-prefix coverable by Lemma 10 for any $\varepsilon > 0$.

In the other direction, one could ask whether being $(\varepsilon, W)$-prefix coverable implies that the function can be approximated by a small ROBP. We prove that this is indeed the case. The proof is via our main learning algorithm (Theorem 13).

**Corollary 8** *If $f : \{0,1\}^n \to \{0,1\}$ is $(\varepsilon, W)$-prefix coverable, there exists a $W$-ROBP $M$ such that $d(M, f) \leq 4n\varepsilon$.*

This statement does not have a converse. Approximation by a $W$-ROBP does not guarantee small prefix covers.

**Lemma 9** *There exists a function $f$ such that $d(f, g) \leq 1/n$ for some function $g$ which is computable by a width 2 ROBP, but such that $f$ is not $(0.1, W)$-prefix coverable for $W = \mathrm{poly}(n)$.*

**Proof** Let $f$ be the Parity function on $n$ bits if $\vee_{i=1}^{\log n} x_i$ and a random function otherwise. It is clear that $f$ is $1/n$ far from the Parity function which is computable by a width 2 ROBP. However any easy counting argument shows that random functions do not have small prefix covers, hence $f$ does not have small prefix covers either. ■

### 2.2. Prefix Covers for functions of Halfspaces

We now show that functions of halfspaces have small prefix covers. We first consider the case of one halfspace. Our proof of this claim is similar to the argument used in Meka and Zuckerman (2010) to show that one can approximate halfspaces by ROBPs, we explain the intuition below:

Consider a halfspace

$$f \equiv 1\{\sum_{i \leq n} a_i x_i \leq b\}.$$

If we set $x_1, \ldots, x_i$, we get a halfspace of the form

$$f \equiv 1\{\sum_{i < j \leq n} a_j x_j \leq b'\}.$$

Thus different prefixes result in halfspaces which differ only in the constant term. Thus by picking prefixes that correspond to well-spaced values for $b'$, we can cover all possible prefixes.

**Lemma 10 (Prefix Covers for halfspaces)** *For every $\varepsilon > 0$, every halfspace is $(\varepsilon, 2/\varepsilon)$-prefix coverable.*

**Proof** Fix $\varepsilon > 0$ and a halfspace

$$f \equiv 1\{\sum_{i \leq n} a_i x_i \leq b\}.$$

For each $i \leq n$, we will show that there exist a cover $S^i$, $|S^i| \leq 2/\varepsilon$ for prefixes of length $i$.

If we fix a prefix $x \in \{0,1\}^i$, and let $v(x) = \sum_{j \leq i} a_j x_j$ then we get the function

$$f_x(z) \equiv 1\{\sum_{i < j \leq n} a_j z_j \leq b - v(x)\}.$$

It follows that if $v(x) \geq v(x')$, then $f_x^{-1}(1) \subseteq f_{x'}^{-1}(1)$. In other words, as $v(x)$ gets larger, the set of suffixes $z$ that are accepted become smaller.

Let $p(x) = \Pr_{z \in \mu^{n-i}}[f_x(z) = 1]$. Then $p(x)$ is a decreasing function of $v(x)$. Now arrange all prefixes in $\{0,1\}^i$ in decreasing order with respect to $p(x)$, call this ordering $x[1], \ldots, x[2^i]$. Each time $p(x)$ drops by $\varepsilon$, we pick a new prefix and add it to $S^i$. Formally, we start by adding $x_1 = x[1]$ to $S_i$. Assuming we have added $x_1, \ldots, x_j$, we next add a string $x_{j+1}$ which maximizes $p(x)$ over all strings $x \in \{0,1\}^i$ such that $p(x) < p(x_j) - \varepsilon$. Assume the resulting set is $S^i = \{x_1, \ldots, x_k\}$.

It is clear that $k \leq \lceil 1/\varepsilon \rceil \leq 2/\varepsilon$ strings, since every time we add a string to $S$, $p(x)$ drops by at least $\varepsilon$ and it lies in the range $[0,1]$. We claim that $S^i = \{x_1, \ldots, x_k\}$ forms a cover for prefixes of length $i$. Indeed, for any $y \in \{0,1\}^i$, assuming $y \notin S^i$, there exists $x_j \in S^i$ such that $p(x_j) \geq p(y) \geq p(x_j) - \varepsilon$. Further, it follows that $f_y^{-1}(1) \subseteq f_{x_j}^{-1}(1)$ and hence $d(f_{x_j}, f_y) \leq \varepsilon$. This completes the proof. ■

It is easy to deduce a similar claim for functions of halfspaces using the following lemma which shows that the property of having small prefix covers is preserved under composition.

**Lemma 11** *Let $f^1, \ldots, f^k : \{0,1\}^n \to \{0,1\}$ be $(\varepsilon, W)$-prefix coverable and $g : \{0,1\}^k \to \{0,1\}$. Then $h : \{0,1\}^n \to \{0,1\}$ defined by $h(x) = g(f^1(x), \ldots, f^k(x))$ is $(2k\varepsilon, W^k)$-prefix coverable.*

**Proof** For $j \leq k$, let $S_j^1, \ldots, S_j^n$ be $(\varepsilon, W)$-prefix covers for $f^j$. Fix $i \leq n$ and form a set of prefixes $T^i \subseteq \{0,1\}^i$ as follows: for every $(x_1, \ldots, x_k) \in S_1^i \times \ldots \times S_k^i$ let

$$U(x_1, \ldots, x_k) = \{z \in \{0,1\}^i : d(f_{x_j}^j, f_z^j) \leq \varepsilon, \forall j \leq k\}.$$

If $U(x_1, \ldots, x_k)$ is not empty, add a single element of $U$ to $T^i$.

By construction, $|T^i| \leq W^k$. Further, for every $y \in \{0,1\}^i$, there exists $(x_1, \ldots, x_k)$ such that $y \in U(x_1, \ldots, x_k)$. Let $u$ be the element of $U(x_1, \ldots, x_k)$ added to $T$. Observe that for two inputs $x, y$ if $h(x) \neq h(y)$, there must be some $i$ such that $f^i(x) \neq f^i(y)$. Hence by a union bound,

$$d(h_y, h_u) \leq \sum_j d(f_y^j, f_u^j) \leq 2k\varepsilon.$$

■

By combining these two Lemmas, we get small prefix covers for functions of $k$ halfspaces.

**Corollary 12** *Every function of $k$ halfspaces is $(\varepsilon, (4k/\varepsilon)^k)$-prefix coverable.*

## 3. Learning Prefix-Coverable Functions via ROBPs

The algorithm learns a prefix coverable $f$, given query access to $f$, by constructing a ROBP $M$ that approximates $f$. The ROBP $M$ has $n$ layers numbered 0 through $n$. The set of vertices in layer $i$ is denoted by $L(M, i)$. Each vertex $x \in L(M, i)$ corresponds to a string $x \in \{0,1\}^i$. $L(M, 0)$ consist of a single start state, identified with the null string $\varphi$. By abuse of notation, we will think of $M$ both as a branching program and a Boolean function.

**Main Algorithm**. Input $n, \varepsilon, W$.

Let $L(M, 0)$ contain the null string, while $L(M, i)$ are empty sets for $i \in \{1, \ldots, n\}$.
For $i = 1, \ldots, n$:
  For each $x \in L(M, i-1)$ and $b \in \{0,1\}$,
    Check if there is $y \in L(M, i)$ such that $d(f_{x \circ b}, f_y) \leq 3\varepsilon$.
      If so, add an edge labeled $b$ from $x$ to $y$.
      If not, add $x \circ b$ to $L(M, i)$, add an edge labeled $b$ from $x$ to $x \circ b$.
  If $|L(M, i)| > W$, then output FAIL and halt.

In line 4 of our algorithm, to check if there is a vertex $y$ that is $\varepsilon$-close to $x \circ b$, we pick $R$ random suffixes $z \in \{0,1\}^{n-i}$ and check if $f(x \circ b \circ z) = f(y \circ z)$. By the Chernoff bound, if $R = O(\log(nW^2/\delta)/\varepsilon)$, then the probability that our estimate of $d(f_{x \circ b}, f_y)$ is off by more than an additive $\varepsilon$ is at most $\delta/2nW^2$. Since each layer has at most $W$ vertices in total, we estimate at most $2nW^2$ such quantities. Hence the probability that the error is more than $\varepsilon$ in any of our estimates is at most $\delta$.

**Theorem 13 (Correctness of the Main Algorithm)** *For $\varepsilon, \delta > 0$, given oracle access to a $(\varepsilon, W)$-prefix coverable function $f$, the above algorithm runs in time $O(nW \log(nW/\delta)/\varepsilon)$ and constructs a $W$-ROBP $M$ such that $d(M, f) \leq 4n\varepsilon$ with probability at least $1 - \delta$.*

By a simple Chernoff bound, by setting $L = O(\log(nW^2/\delta)/\varepsilon)$ to be sufficiently large we can assume that all the estimates made by the algorithm in line (4) are within $\varepsilon$, which happens with probability $1 - \delta$. The proof of Theorem 13 follows from two lemmas.

**Lemma 14** *The algorithm never outputs FAIL.*

**Proof** Let $S^1, \ldots, S^n$ be $(\varepsilon, W)$-prefix covers for $f$. For each $x \in S^i$, consider the ball

$$B(x) = \{y \in \{0,1\}^i : d(f_y, f_x) \leq \varepsilon\}.$$

By definition, they cover all of $\{0,1\}^i$. We claim that $L(M,i)$ cannot have two distinct vertices $y, y' \in \{0,1\}^i$ in layer $i$ that belong to the same ball $B(x)$. For, if $y, y'$ lie in the same ball, $d(f_y, f_{y'}) \leq 2\varepsilon$. Since the sampling error is at most $\varepsilon$, our estimate for $d(f_y, f_{y'})$ would be at most $3\varepsilon$, thus we would not add both of them to $L(M,i)$. Hence $|L(M,i)| \leq |S^i| \leq W$. ■

**Lemma 15** *We have* $\mathsf{Pr}_\mu[M(x) \neq f(x)] \leq 4n\varepsilon$.

**Proof** By induction on $n - i$, we will show that for every $x \in L(M,i)$ , $d(M_x, f_x) \leq 4(n-i)\varepsilon$. This implies that when $i = 0$, $d(M, f) \leq 4n\varepsilon$ as desired.

For $i = n$ there is nothing to prove. Suppose the statement is true for all vertices in $L(M, i+1)$. Consider a vertex $x \in L(M,i)$. Let $y_0, y_1 \in L(M, i+1)$ be it's neighbors in $M$. Then, by our assumption on sampling errors, for $b \in \{0,1\}$, $d(f_{x \circ b}, f_{y_b}) \leq 4\varepsilon$. By the induction hypothesis, we know that $d(f_{y_b}, M_{y_b}) \leq 4(n-i-1)\varepsilon$. Putting these together, we get

$$\begin{aligned} d(f_x, M_x) &= \frac{1}{2} \sum_{b \in \{0,1\}} d(f_{x \circ b}, M_{y_b}) \\ &\leq \frac{1}{2} \sum_{b \in \{0,1\}} (d(f_{x \circ b}, f_{y_b}) + d(f_{y_b}, M_{y_b})) \quad \text{(by triangle inequality)} \\ &\leq 4\varepsilon + 4(n-i-1)\varepsilon = 4\varepsilon(n-i). \end{aligned}$$

■

Theorem 13 now follows as the probability of sampling error is at most $\delta$.

To derive Theorem 1, for learning functions of $k$ halfspaces to accuracy $\varepsilon_0$, we apply Theorem 13 with parameters

$$\varepsilon = \frac{\varepsilon_0}{4n}, W = \left(\frac{4k}{\varepsilon}\right)^k = \left(\frac{16nk}{\varepsilon_0}\right)^k.$$

where the setting of $W$ comes from Corollary 12.

We can get similar results for learning under any explicitly given small-space source in the sense of Kamp et al. (2006). In particular, we can learn functions of halfspaces under $p$-biased and symmetric distributions. We refer the reader to Gopalan et al. (2011a) for details.

## 4. Conclusions

Our algorithm for learning functions of halfspaces exploits the connection between halfspaces and ROBPs. It would be interesting if this connection could lead to improved algorithms for agnostically learning halfspaces. In particular, the problem of agnostically learning $\mathrm{poly}(n)$-ROBPs with membership queries is open. A (query) algorithm for this problem will give a (query) algorithm for agnostically learning halfspaces.

The best known algorithm for agnostically learning a halfspace under the uniform distribution on $\{0,1\}^n$ due to Kalai et al. (2008) runs in time $O(n^{1/\varepsilon^2})$. Their paper also shows that a $\mathrm{poly}(n, 1/\varepsilon)$ algorithm for agnostically learning halfspaces from random examples alone will result in a $\mathrm{poly}(n)$ time algorithm for the notorious noisy parity problem. This seems to suggest that such an algorithm is perhaps unlikely, but it leaves open the possibility of a $\mathrm{poly}(n, 1/\varepsilon)$ algorithm that uses membership queries. ROBP-based algorithms seem to be the only tool we currently have to exploit membership queries in the context of halfspace learning.

## 5. Acknowledgments

Adam Klivans is supported by an NSF CAREER Award and NSF CCF 0728536.

## References

D. Angluin. Learning Regular Sets from Queries and Counterexamples. *Information and Computation*, 75(2):87–106, 1987.

D. Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in $\mathrm{nc}^1$. *Journal of Computer and System Sciences*, 38(1):150–164, 1989.

E. Baum. On learning a union of halfspaces. *Journal of Complexity*, 6(1):67–101, 1990.

Amos Beimel, Francesco Bergadano, Nader H. Bshouty, Eyal Kushilevitz, and Stefano Varricchio. Learning functions represented as multiplicity automata. *J. ACM*, 47(3):506–530, 2000.

A. Blum and R. Kannan. Learning an intersection of a constant number of halfspaces over a uniform distribution. *J. Comput. Syst. Sci. (JCSS)*, 54(2):371–380, 1997.

N. Bshouty, C. Tamon, and D. Wilson. On learning width two branching programs. *Information Processing Letters*, 65:217–222, 1998.

F. Ergün, R. Kumar, and R. Rubinfeld. On learning bounded-width branching programs. In *COLT*, pages 361–368, 1995.

P. Gopalan, A. Klivans, and R. Meka. Polynomial-time approximation schemes for knapsack and related counting problems using branching programs. In *Electronic Colloquium on Computational Complexity (ECCC)*, 2011a.

P. Gopalan, A. Klivans, R. Meka, D. Stefankovic, S. Vempala, and E. Vigoda. An FPTAS for knapsack and related counting problems. In *FOCS*, pages 817–826, 2011b.

P. Harsha, A. Klivans, and R. Meka. An invariance principle for polytopes. In *STOC*, pages 543–552, 2010.

A. Kalai, A. Klivans, Y. Mansour, and R. Servedio. Agnostically learning halfspaces. *SIAM Journal on Computing*, 37(6):1777–1805, 2008.

J. Kamp, A. Rao, S. P. Vadhan, and D. Zuckerman. Deterministic extractors for small-space sources. In *STOC*, pages 691–700, 2006.

M. Kearns and L. Valiant. Cryptographic limitations on learning Boolean formulae and finite automata. *Journal of the ACM*, 41(1):67–95, 1994.

A. Klivans, R. O'Donnell, and R. Servedio. Learning intersections and thresholds of halfspaces. *Journal of Computer & System Sciences*, 68(4):808–840, 2004.

A. Klivans, P. M. Long, and A. Tang. Baum's algorithm learns intersections of halfspaces with respect to log-concave distributions. In *APPROX-RANDOM*, pages 588–600, 2009.

A. R. Klivans, R. O'Donnell, and R. Servedio. Learning geometric concepts via Gaussian surface area. In *FOCS*, pages 541–550, 2008.

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

R. Meka and D. Zuckerman. Pseudorandom generators for polynomial threshold functions. In *STOC*, pages 427–436, 2010.

R. O'Donnell. Noise sensitivity of intersections of halfspaces, Open problem collection (Simons Symposium). http://analysisofbooelanfunctions.org, 2012.

S. Vempala. A random sampling based algorithm for learning the intersection of halfspaces. In *FOCS*, pages 508–513, 1997.

S. Vempala. Learning convex concepts from gaussian distributions with PCA. In *FOCS*, pages 124–130, 2010a.

S. Vempala. A random-sampling-based algorithm for learning intersections of halfspaces. *J. ACM*, 57(6):32, 2010b.