Polynomial Threshold Functions for Decision Lists

Vladimir Podolskii
Courant Institute of Mathematical Sciences, New York University, NY, USA
Steklov Mathematical Institute of Russian Academy of Sciences, Moscow, Russian Federation

Nikolay V. Proskurin
HSE University, Moscow, Russian Federation

Abstract

For $S \subseteq \{0,1\}^n$ a Boolean function $f\colon S \to \{-1,1\}$ is a polynomial threshold function (PTF) of degree d and weight W if there is a polynomial p with integer coefficients of degree d and with sum of absolute coefficients W such that $f(x) = \operatorname{sign} p(x)$ for all $x \in S$. We study a representation of decision lists as PTFs over Boolean cubes $\{0,1\}^n$ and over Hamming balls $\{0,1\}^n_{\leq k}$.

As our first result, we show that for all $d = O\left(\left(\frac{n}{\log n}\right)^{1/3}\right)$ any decision list over $\{0,1\}^n$ can be represented by a PTF of degree d and weight $2^{O(n/d^2)}$. This improves the result by Klivans and Servedio [22] by a $\log^2 d$ factor in the exponent of the weight. Our bound is tight for all $d = O\left(\left(\frac{n}{\log n}\right)^{1/3}\right)$ due to the matching lower bound by Beigel [3].

For decision lists over a Hamming ball $\{0,1\}^n_{\leq k}$ we show that the upper bound on weight above can be drastically improved to $n^{O(\sqrt{k})}$ for $d = \Theta(\sqrt{k})$. We also show that similar improvement is not possible for smaller degrees by proving the lower bound $W = 2^{\Omega(n/d^2)}$ for all $d = O(\sqrt{k})$.

2012 ACM Subject Classification Theory of computation → Models of computation

Keywords and phrases Threshold function, decision list, Hamming ball

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2022.52

Funding *Nikolay V. Proskurin*: This paper was prepared within the framework of the HSE University Basic Research Program.

1 Introduction

The main object of studies in this paper are polynomial threshold functions.

▶ **Definition 1.** *For $S \subseteq \{0,1\}^n$ a Boolean function $f\colon S \to \{-1,1\}$ is called a polynomial threshold function (PTF) of degree d if there is an integer polynomial p of degree d such that $f(x) = \operatorname{sign} p(x)$ for all $x \in S$. If p has integer coefficients, then the weight of a PTF is defined as a sum of absolute values of coefficients in p.*

Polynomial threshold functions have been studied intensively for decades. Much of this work was motivated by questions in computer science [28], and PTFs are now an important object of study in areas such as Boolean circuit complexity [7, 17, 2, 23], learning theory [20, 21, 16, 4, 15], and communication complexity [30].

In this paper, we study PTFs for the class of *decision lists*.

▶ **Definition 2.** *A decision list L on variables $x_1, \dots, x_n$ is a sequence of h pairs and a bit*

$$(\ell_1, b_1), (\ell_2, b_2), \dots, (\ell_h, b_h), b_{h+1},$$

where for all i ℓ_i is a literal (either a Boolean variable x_j or its negation) and $b_i \in \{-1,1\}$. On the input $x \in \{0,1\}^n$ the value of $L(x)$ is equal to b_i if i is the minimal index for which ℓ_i is true and if all ℓ_i are false, the value of $L(x)$ is equal to b_{h+1}.

Decision lists have been widely studied in computational complexity [1, 5, 12, 22]. Partially, their importance stems from computational learning theory where attribute efficient learning of decision lists is an important research direction. One of the approaches to learning decision lists is through their representation as PTFs, to which several known learning algorithms can be applied [18, 21, 22, 27]. This constitutes one of the reasons to study PTF representations of decision lists. Decision lists turned out to be important in other areas of theoretical computer science as well. They provide a good source of lower bounds in the studies of threshold functions, threshold circuits, oracle computations and in other fields [3, 8, 13, 34].

A lot of effort has been put into the studies of effective PTF representations of Boolean functions, and for decision lists in particular. Klivans and Servedio [22] have shown that for every $h < n$ any decision list can be computed by a PTF of degree $O(\sqrt{n}\log n)$ and weight $2^{O(\frac{n}{h}+\sqrt{h}\log^2 n)}$; in terms of degree d it results in $2^{O(\frac{n\log^2 d}{d^2}+d\log d)}$ weight bound. They used it to create the first online learning algorithm for decision lists that is subexponential in both running time and sample complexity; namely, their algorithm runs in time $n^{O(n^{1/3}\log^{1/3} n)}$ and mistake bound $2^{O(n^{1/3}\log^{4/3} n)}$. Servedio, Tan and Thaler proved in [29, Theorem 6] that for any $n^{1/4} \le d \le n$ any decision list $L\colon \{-1,1\}^n \to \{-1,1\}$[1] on n variables has a degree-d PTF of weight $2^{\tilde{O}_n((n/d)^{2/3})}$, where $\tilde{O}_n$ suppresses the polylog factor of n. This bound is weaker for small d, but gives an upper bound for $d \ge \Omega(\sqrt{n})$. As for the lower bounds, Beigel [3] provided a decision list that requires weight of $2^{\Omega(n/d^2)}$ to be computed by a degree-d PTF (we provide the details in Section 3). Servedio, Tan and Thaler [29] also proved a lower bound of $2^{\Omega(\sqrt{n/d})}$ for $d = o\left(\frac{n}{\log^2 n}\right)$, which is stronger than the bound of Beigel for $d = \Omega(n^{1/3})$.

Above we have discussed PTF representations of decision lists over the Boolean cube $\{0,1\}^n$. However, representations over its subsets are also relevant. The case of the Hamming ball $\{0,1\}^n_{\le k}$ consisting of all vectors with at most k 1s received some attention. Long and Servedio [25] gave bounds for the weights of PTFs for degree $d = 1$. Their main motivation to study this setting comes from learning theory: in scenarios involving learning categorical data the common representation for examples is the one-hot encoded vector, which might have an extremely large amount of features, but only a small fraction of them can be active at the same time. The Winnow algorithm used in [22] can be applied to learn functions not only on the Boolean cube but also on its subsets, including $\{0,1\}^n_{\le k}$, so it makes sense to study not only linear representations but also polynomial ones. Boolean functions on Hamming balls are also important outside of learning theory. For example, approximation of such functions by polynomials of low weight arises in the problem of indistinguishability [6, 19], and recently the approximation on Hamming Balls in general received a lot of attention as a stepping stone for the approximation of some important Boolean functions [9, 31], including constant-depth circuits (AC^0) [10, 11, 32, 33].

Our results

First, we address the PTF representations of decision lists over $\{0,1\}^n$. In our first result, we show that for every $h < n$ any decision list can be computed by a PTF of degree $O(\sqrt{n})$ and weight $2^{O(\frac{n}{h}+\sqrt{h}\log n)}$. This improves the result of [22] by a logarithmic factor both in degree and exponent of weight. As a corollary, we slightly improve upon the upper bound for the attribute efficient learning of decision lists: we prove that there is an algorithm that learns decision lists in time $n^{O(n^{1/3})}$ and mistake bound $2^{O(n^{1/3}\log n)}$.

[1] In this result, input Boolean variables range over $\{-1,1\}$.

In terms of degree vs. weight tradeoff, our results imply that for all $d = O\left(\left(\frac{n}{\log n}\right)^{1/3}\right)$ any decision list over $\{0,1\}^n$ can be computed by a PTF of degree d and weight $2^{O(n/d^2)}$. The analogous result that follows the construction in [22] gave the weight upper bound of $2^{O((n\log^2 d)/d^2)}$. Our bound is tight for all $d = O\left(\left(\frac{n}{\log n}\right)^{1/3}\right)$ due to the matching upper bound by Beigel [3] mentioned above.

Clearly, our upper bound for the Boolean cube works also for decision lists over the Hamming ball $\{0,1\}^n_{\leq k}$ as they are just restrictions to a smaller domain. However, the tightness of this bound for the case of Hamming balls has to be investigated. We actually show that the upper bound can be improved to a polynomial in n for d roughly equal to $\sqrt{k}$. More precisely, we show that for $d = \Theta(\sqrt{k})$ any decision list over $\{0,1\}^n_{\leq k}$ can be represented with weight $n^{O(\sqrt{k})}$. However, we show that such an improvement is impossible for smaller degrees. In order to do so, we extend Beigel's lower bound to this setting, showing that there is a decision list that requires weight $2^{\Omega(n/d^2)}$ when computed by a PTF of degree $d = O(\sqrt{k})$.

Our techniques

To prove our upper bounds we adopt the same strategy as in [22]. We first represent a decision list as a sum of sublists, then we pointwise approximate each of the sublists. The improvement comes from the better approximation technique originated by Sherstov [31], which we adapt for decision lists.

For the lower bound, we extend the proof of Beigel [3] to the setting of low-weight inputs. We use the same proof strategy of inductively constructing a sequence of inputs on which the value of the polynomial in PTF grows exponentially. The new ingredient in the proof is to keep the number of 1's in the input low by reusing them from the previous blocks.

Organization

In Section 2 we provide the necessary definitions and theorems. In Section 3 we give our PTF construction for decision lists on $\{0,1\}^n$, and in Section 4 we prove the upper and the lower bounds for representing decision lists as a PTF on $\{0,1\}^n_{\leq k}$.

2 Preliminaries

We use the following notation for the *arithmetization* of a literal ℓ:

$$\tilde{\ell} = \begin{cases} x, & \ell \text{ is an unnegated variable } x, \\ 1-x, & \ell \text{ is a negated variable } \overline{x}, \end{cases}$$

We also use the notations $[n] = \{1,2,\dots,n\}$ and $|||p|||$ for the sum of absolute values of coefficients in p (therefore, the weight of a PTF p is equals to $|||p|||$ if p has integer coefficients).

For technical reasons, we need a modified definition of a decision list, where the last b_i is always equal to 0 instead of ± 1. We call those decision lists *modified*.

As mentioned in the introduction, one of the technical steps involves the pointwise approximation of Boolean functions.

▶ **Definition 3.** *Given a Boolean function $f\colon X \to \{0,1\}$, we say that a polynomial p is ε-approximates the function f iff $\max_{x\in X} |f(x) - p(x)| \leq \varepsilon$. The ε-approximate degree $\deg_\varepsilon(f)$ is the minimum degree such that there exists a polynomial of degree d that ε-approximates for the function f.*

Polynomial threshold functions may be viewed as a generalization of ε-approximators: if p ε-approximates the function f on $X \subseteq \{0,1\}^n$, then $p(x) - \frac{1}{2}$ is a PTF that computes f on X.

One of the commonly used families of polynomials for the Boolean function approximation is the Chebyshev polynomials of the first kind.

▶ **Definition 4.** *The Chebyshev polynomials of the first kind are defined by the recurrence relation* $T_d(x) = 2xT_{d-1}(x) - T_{d-2}(x)$ *with* $T_0(x) = 1$ *and* $T_1(x) = x$.

The solution of the recurrence above gives the following equation.

$$T_d(x) = \frac{1}{2}\left(\left(x - \sqrt{x^2-1}\right)^d + \left(x + \sqrt{x^2-1}\right)^d\right). \tag{1}$$

It is obvious from the definition that $\deg(T_d) = d$ and T_d has integer coefficients. We also need the following claims about the Chebyshev polynomials.

▷ Claim 5 ([31, Proposition 2.6]). For every $\delta > 0$, $T_d(1+\delta) \geq 1 + d^2\delta$.

▷ Claim 6. For every $\delta \in [0;2]$, $T_d(1+\delta) \leq (1+\sqrt{\delta})^{2d}$.

Proof. Note that $\sqrt{(1+\delta)^2-1} = \sqrt{\delta(\delta+2)} \leq 2\sqrt{\delta}$ for $0 \leq \delta \leq 2$. By substituting $x = 1+\delta$ into (1), we get

$$T_d(1+\delta) = \frac{1}{2}\left(\left(1+\delta-\sqrt{(1+\delta)^2-1}\right)^d + \left(1+\delta+\sqrt{(1+\delta)^2-1}\right)^d\right) \leq$$

$$\leq \left(1+\delta+\sqrt{(1+\delta)^2-1}\right)^d \leq (1+\delta+2\sqrt{\delta})^d = (1+\sqrt{\delta})^{2d}.$$

◁

▷ Claim 7 ([6, Section 2.2, Property 5]). $|||T_d||| \leq 2^{2d}$.

Klivans and Servedio used the Expanded-Winnow algorithm [22, Theorem 2] to derive the bounds on learning the decision lists class in the attribute-efficient learning model. It essentially runs the Winnow algorithm [24, Algorithm 4] on a set of all possible monomials of degree up to d. We note that the Winnow algorithm can learn Boolean functions on any domain $X \subseteq \{0,1\}^n$, and therefore we can apply the same bounds from the Expanded-Winnow to learn the decision lists on $\{0,1\}^n_{\leq k}$. Thus, the next theorem implicitly follows from the Expanded-Winnow algorithm.

▶ **Theorem 8.** *Let C be a class of Boolean functions over $S \subseteq \{0,1\}^n$ with the property that each $f \in C$ has a PTF of degree at most d and weight at most W. Then there is an online learning algorithm for C which runs in n^d time per example and has mistake bound $O(W \cdot d^2 \cdot \log n)$.*

3 Decision lists on the Boolean cube

In this section, we prove the upper bound for representing decision lists as a PTF over the Boolean cube. Our proof can be viewed as an improvement of the proof of the following theorem by Klivans and Servedio.

▶ **Theorem 9** ([22, Theorem 7])**.** *Let L be a decision list of length n on $\{0,1\}^n$. Then for any $h < n$, L is computed by a PTF of degree $O(\sqrt{h}\log h)$ and weight $2^{O(n/h+\sqrt{h}\log^2 h)}$.*

We start with the outline of their construction and then proceed to prove the missing pieces to tighten up the upper bound. This construction consists of two parts. The "outer" part is the decomposition of the original decision list into the sum of modified sublists, i.e. we represent the $L(x)$ as the $L(x) = \sum_i^{n/h} a_i L_i(x)$, where each sublist is responsible for the independent block of size roughly h. The "inner" part is the pointwise approximation of the sublist. To achieve a close enough approximation, Klivans and Servedio represent each of the sublists as a sum of conjunctions and approximate each of the conjunctions separately. The degree-weight traidoff is adjusted by the parameter h.

The problem with this approach is that in order to approximate the sublist, one has to very closely approximate the inner conjunctions, i.e. with a precision of at least $O(\frac{1}{h})$. But it was proved in [14] that such an approximation requires the degree of $\Theta(\sqrt{h \log h})$, and while we can improve on the logarithmic factor, we can not get rid of it completely. However, we can notice that the set of conjunctions we are dealing with is not entirely random; in fact, every two successive conjunctions share most of their literals. That means we can adjust our approximator so it would increase it's precision as more terms in conjunction are set to zero. Luckily for us, it was proved in [31] that such an adjustment can be done with the same degree as in the regular approximator. In the next theorem, we adapt it for our needs and also prove the weight bound for it.

▶ **Theorem 10.** *For every $d = \Theta(1)$ and every $\varepsilon = \Theta(1)$ there exists an univariate polynomial $P_{n,d,\varepsilon}(x)$ such that $\deg(P_{n,d,\varepsilon}) = O(\sqrt{n})$, $|||P_{n,d,\varepsilon}||| = 2^{O(\sqrt{n})}$ and*

$$\forall t \in [0;1] \quad |P_{n,d,\varepsilon}(t) - 1| \le \varepsilon,$$

$$\forall t \in (2;n] \quad |P_{n,d,\varepsilon}(t)| \le \frac{\varepsilon}{t^d}.$$

Moreover, there exists a constant $C = n^{O(\sqrt{n})}$ such that $C \cdot P_{n,d,\varepsilon}(x)$ has integer coefficients.

Proof. We can get such a polynomial from [31, Theorem 3.7] (in fact, our statement is much more limited than the original one) with a proven degree bound but with no weight estimate. So it suffices for us to prove that the claimed weight bound holds as well.

The polynomial $P_{n,d,\varepsilon}(x)$ from [31, Theorem 3.7] is given as $P_{n,d,\varepsilon}(x) = p_1(x)^d p_2(p_1(x)) p_3(x)$, where

$$p_1(t) = \frac{T_{d_1+1}\left(1 - 2\frac{t-1}{n-1}\right)}{T_{d_1+1}\left(\frac{n+1}{n-1}\right)}$$

$$p_2(t) = \sum_{i=0}^{D} \binom{i+d-1}{i} (1-t)^i$$

$$p_3(t) = B_{d_3}\left(\frac{1}{e^7} T_{\sqrt{n}}\left(1 + \frac{2-t}{n}\right)\right), \quad B_{d_3}(t) = \sum_{i=\lceil 2.5e^{-7} d_3 \rceil}^{d_3} \binom{d_3}{i} t^i (1-t)^i$$

with $d_1 = \lfloor \sqrt{2(n-1)} \rfloor$, $D = O\left(d + \log \frac{1}{\varepsilon}\right)$ and $d_3 = O\left(\log \frac{1}{\varepsilon}\right)$.

To start with,

$$T_{d_1+1}\left(\frac{n+1}{n-1}\right) \le \left(1 + \sqrt{\frac{2}{n-1}}\right)^{2\sqrt{2(n-1)}} = \left(1 + \frac{2}{\sqrt{2(n-1)}}\right)^{2\sqrt{2(n-1)}} \le e^4,$$

where the first inequality is Claim 5 and the last one is $(1+\frac{a}{x})^x < e^a$ for $a > 0$ and $x > 1$. On the other hand, by Claim 6

$$T_{d_1+1}\left(\frac{n+1}{n-1}\right) \geq 1 + 2(n-1)\cdot\frac{2}{n-1} = 5,$$

so the scaling factor in $p_1(t)$ is bounded by constants, and by Claim 7 $|||p_1||| = 2^{O(\sqrt{n})}$. By multiplying p_1 by $(n-1)^{d_1+1}$, we clear the denominators from $1 - 2\frac{t-1}{n-1}$, and thus it would have integer coefficients. The same holds for $T_{\sqrt{n}}\left(1+\frac{2-t}{n}\right)$ and $n^{\sqrt{n}}$. As for the $T_{d_1+1}\left(\frac{n+1}{n-1}\right)$ in p_1, we know that it is a rational number $\frac{a}{b}$ with $b \leq (n-1)^{d_1+1}$, which is bounded by constants. Therefore, $a = O((n-1)^{d_1+1})$, and to clear all the denominators from $P_{n,d,\varepsilon}(x)$ it suffices to multiply it by $C = n^{O(\sqrt{n})}$.

Next, both $p_2(t)$ and $B_{d_3}(t)$ have integer and independent of n coefficients, as well as independent of n degrees, so their degrees and weights are constant. Finally, the product of a constant number of terms of degree $O(\sqrt{n})$ and weight $n^{O(\sqrt{n})}$ has the same degree and weight bounds, and the overall weight bound holds. The bound for the constant C holds for the same reasons. ◀

With the new approximator, we are ready to proceed to the main proof of this section.

▶ **Theorem 11.** *Let L be a modified decision list of length h on $\{0,1\}^n$. Then for every $\varepsilon = \Theta(1)$ L can be ε-approximated by a polynomial of degree $O(\sqrt{h})$ and weight $n^{O(\sqrt{h})}$. Moreover, $p(0^n) = 0$.*

Proof. Consider the decision list $L = (\ell_1, b_1), \ldots (\ell_h, b_h), 0$. It is straightforward to check that L can be expressed as $L(x) = \sum_{i=1}^{h} b_i \ell_i \bigwedge_{j=1}^{h-1} \overline{\ell_j}$, where the term corresponding to b_i is non-zero iff ℓ_i is the first condition satisfied in L.

For every i put $T_i(x) = \bigwedge_{j=1}^{h-1} \overline{\ell_j(x)}$ and $A_i(x) = 3(i-1) - 3\tilde{\ell_1} - \ldots - 3\tilde{\ell}_{i-1}$. Notice that $A_i(x) = 0 \Leftrightarrow T_i(x) = 1$ and $A_i(x) \geq 3$ otherwise. Moreover, for every $j < i$ such that $\ell_j = 1$ the value of $A_i(x)$ increases by 3. By corollary, for every $i < j$ we have $\ell_i = 1 \Rightarrow A_i(x) + 3 \leq A_j(x)$ because A_j consists of every zero literal of T_i and $\overline{\ell_i} = 0$.

Now consider $p_i(x) = P_{3h,2,\varepsilon/2}(A_i(x))$ where $P_{3h,2,\varepsilon/2}$ is from Theorem 10. We will prove that $p(x) = \sum_{i=1}^{h} b_i \tilde{\ell_i} p_i(x)$ is the desired polynomial. To start with, if $\ell_i = 0$ then the corresponding term is also equals to zero; in particular, $p(0^n) = 0$. Next, let $j_1 < \ldots < j_m$ be the set of all indexes such that $\ell_{j_i} = 1$. For j_1 we have $A_{j_1}(x) = 0$, and so $\operatorname{sign}(b_{j_1}\tilde{\ell_{j_1}} p_{j_1}(x)) = b_{j_1}$ and $|b_{j_1}\tilde{\ell_{j_1}} p_{j_1}(x)| \geq 1 - \varepsilon/2$. Finally, for the remaining indexes we have $3 \leq A_{j_2}(x) < A_{j_3}(x) < \ldots < A_{j_m}(x) \leq 3h$, and so

$$\left|\sum_{i=2}^{m} b_{j_i}\ell_{j_i} p_{j_i}(x)\right| \leq \sum_{i=2}^{m} |p_{j_i}(x)| \leq \frac{\varepsilon}{2}\sum_{i=3}^{\infty}\frac{1}{i^2} = \frac{\varepsilon}{2}\left(\sum_{i=1}^{\infty}\frac{1}{i^2} - \frac{5}{4}\right) \leq \frac{\varepsilon}{2}\left(\frac{\pi^2}{6} - \frac{5}{4}\right) \leq \frac{\varepsilon}{2}.$$

Therefore, for every $x \in \{0,1\}^n$ we have $|L(x) - p(x)| \leq \varepsilon$ and $p(0^n) = 0$. The bounds on degree and weight follows from the Theorem 10 and the following observation: if $p = a_d x^d + \ldots + a_1 x_1 + a_0$ then

$$|||p(q(x))||| = |||a_d q(x)^d + \ldots + a_1 q(x) + a_0||| \leq |||a_d q(x)^d + \ldots + a_1 q(x)^d + a_0||| =$$

$$= |||q(x)^d(a_d + \ldots + a_1 + a_0)||| \leq |||q|||^d \cdot |||p|||$$

By composing A_i of weight $O(n)$ and degree 1 with $P_{3h,2,\varepsilon/2}$ of weight $2^{O(\sqrt{h})}$ and degree $\sqrt{h}$, we get a polynomial of degree $\sqrt{h}$ and weight at most $n^{O(\sqrt{h})}$. ◀

▶ **Corollary 12.** *Let L be a modified decision list of length h on $\{0,1\}^n$. Then L is computed by a PTF p of degree $O(\sqrt{h})$ and weight $h^{O(\sqrt{h})}$. Moreover, the $p(0^n) = 0$.*

Proof. Multiply the polynomial from the previous theorem by a constant $C = n^{O(\sqrt{n})}$. By Theorem 10, every p_i from the proof of Theorem 11 now has integer coefficients. The resulting polynomial does not pointwise approximate the decision list anymore, but for every $x \in \{0,1\}^n$ we have $|CL(x) - p(x)| \leq C/\varepsilon$ and $p(0^n) = 0$. Therefore, for any sufficiently large constant ε, say $\frac{1}{100}$, Cp is the desired PTF for L. ◀

Using Corollary 12 as the inner approximator, we can use the outer construction by Klivans and Servedio to achieve the final PTF for the decision lists.

▶ **Theorem 13.** *Let L be a decision list of length n on $\{0,1\}^n$. Then for any $h < n$, L is computed by a PTF of degree $O(\sqrt{h})$ and weight $2^{O(n/h+\sqrt{h}\log h)}$.*

The proof goes exactly the same as in [22, Theorem 7], except the inner approximator has a constant precision, which does not affect the overall correctness. For the sake of completeness, we provide the proof in Appendix A.

▶ **Corollary 14.** *Let L be a decision list of length n on $\{0,1\}^n$. Then for any $d < \sqrt{n}$, L is computed by a PTF of degree d and weight $2^{O(n/d^2+d\log d)}$.*

In [3], Beigel proved the lower bound for the weight of PTF for the specific decision list called ODD-MAX-BIT.

▶ **Definition 15.** *The ODD-MAX-BIT$_n$ function on input $x \in \{0,1\}^n$ is equal to $(-1)^i$ where i is the position of the rightmost 1 in x. If $x = 0^n$ then ODD-MAX-BIT$_n(x) = 1$.*

▶ **Theorem 16** ([3])**.** *Let p be a degree d PTF with integer coefficients which computes ODD-MAX-BIT$_n$ on $\{0,1\}^n$. Then $|||p||| = 2^{\Omega(n/d^2)}$.*

Note that Corollary 14 gives a PTF of weight $2^{O(n/d^2)}$ for any decision list and $d = O\left(\left(\frac{n}{\log n}\right)^{1/3}\right)$. Thus, in this range our bound is asymptotically optimal.

Using Theorem 13, we can also provide a slightly more efficient online learning algorithm for decision lists.

▶ **Corollary 17.** *Let L be a decision list of length n on $\{0,1\}^n$. Then L is computed by a PTF of degree $O(n^{1/3})$ and weight $2^{O(n^{1/3}\log n)}$.*

Proof. Apply Theorem 13 with $h = n^{1/3}$. ◀

▶ **Corollary 18.** *There is an algorithm that learns decision lists on $\{0,1\}^n$ in time $n^{O(n^{1/3})}$ and mistake bound $2^{O(n^{1/3}\log n)}$.*

Proof. Follow immediately from the previous corollary and Theorem 8. ◀

4 Decision lists on Hamming balls

In this section, we shift our focus from $\{0,1\}^n$ to $\{0,1\}^n_{\leq k}$. We show that both upper and lower bounds can be generalized on the new domain, and the degree-weight dependency is much more significant than it is on the Boolean cube. To be more precise, we show that if the degree parameter is low, then the Hamming ball scenario is not much different from the Boolean cube one, and we still need an exponential in terms of n weight for a PTF. However, after a certain threshold it can be drastically improved.

4.1 Upper bound

We first start with the upper bound; to achieve it, we can straightforwardly modify the proof of Theorem 11 for the $\{0,1\}^n_{\leq k}$ domain.

▶ **Theorem 19.** *Let L be a modified decision list of length h on $\{0,1\}^n_{\leq k}$. Then for every $\varepsilon = \Theta(1)$ L can be ε-approximated by a polynomial of degree $O(\sqrt{k})$ and weight $n^{O(\sqrt{k})}$. Moreover, $p(0^n) = 0$.*

Proof. Recall the following definitions from the proof of Theorem 11

$$L(x) = \sum_{i=1}^{h} b_i \ell_i \bigwedge_{j=1}^{h-1} \overline{\ell_j} = \sum_{i=1}^{h} b_i \ell_i T_i(x) \qquad A_i(x) = 3(i-1) - \tilde{\ell_1} - \ldots - \tilde{\ell}_{i-1}$$

We need to reexamine $A_i(x)$. If for every $x \in \{0,1\}^n_{\leq k}$ we have $A_i(x) > 0$, then $T_i(x) \equiv 0$ on $\{0,1\}^n_{\leq k}$ and we can safely remove the corresponding term from the sum without affecting it's value on any input. Otherwise, let $x' \in \{0,1\}^n_{\leq k}$ be such an input that $A_i(x') = 0$. Flipping the value of one bit in x' can affect at most one literal in T_i, so if $|x \oplus x'| = 1$ then $A_i(x) \leq 3 + A_i(x')$.

Notice that because we are only interested in $x \in \{0,1\}^n_{\leq k}$, for any x we have $|x \oplus x'| \leq |x \vee x'| \leq 2k$ and $A_i(x) \leq 3 \cdot 2k + A_i(x') = 6k$. Thus, we can approximate $A_i(x)$ with $p_i(x) = P_{6k,2,2\varepsilon}(A_i(x))$ instead of $P_{3n,2,\varepsilon/2}$ (where both polynomials are from Theorem 10), and this results in the desired bounds on degree and weight. ◀

▶ **Corollary 20.** *Let L be a decision list of length n on $\{0,1\}^n_{\leq k}$. Then L is computed by a PTF p of degree $O(\sqrt{k})$ and weight $n^{O(\sqrt{k})}$.*

Proof. The proof is absolutely the same as in Theorem 13. ◀

We note that the same ideas can be applied to [22, Theorem 6], but it would result in a bound of $\deg(p) = O(\sqrt{k}\log n)$ and $|||p||| = 2^{O(\sqrt{k}\log^2 n)}$, which is much worse if $k \ll n$.

As a side result, we also get an online-learning algorithm for decision lists on $\{0,1\}^n_{\leq k}$.

▶ **Corollary 21.** *There is an algorithm that learns decision lists on $\{0,1\}^n_{\leq k}$ in time $n^{O(\sqrt{k})}$ and with mistake bound $n^{O(\sqrt{k})}$.*

4.2 Lower bound

While the previous proof is straightforward, it requires $d = \Theta(\sqrt{k})$. It turns out that it is not a coincidence: if we want to lower the degree even further, we would need to drastically increase the weight parameter. In fact, Corollary 14 is tight in the case of Hamming Balls as well as in the case of the Boolean Cube.

▶ **Theorem 22.** *Let p be a degree d PTF with integer coefficients which computes ODD-MAX-BIT$_n$ on $\{0,1\}^n_{\leq k}$ and $d = O(\sqrt{k})$. Then $|||p||| = 2^{\Omega(n/d^2)}$.*

The new proof is heavily based on the proof of Theorem 16; the main difference is the way we construct a polynomial that achieves a contradiction on its approximate degree.

Proof. The proof goes as follows. We first partition $[n]$ into blocks of even size about $t = O(d^2)$. Note that we got $r = \Omega(n/d^2)$ blocks in total. Then we prove that for every block i we can find an input y_i such that it may have 1's only in the first i blocks and $|p(y_i)| \geq 2^i$. If we succeed, then we get an input y_r such that $|p(y_r)| \geq 2^r = 2^{\Omega(n/d^2)}$ and, as a corollary, $|||p||| = 2^{\Omega(n/d^2)}$. We will prove this claim by induction.

We adjust the hidden constant in $d = O(\sqrt{k})$ so we can assume $k \geq t$. With that in mind, let $y_0 = 1^t 0^{n-t}$, i.e. the first t bits are 1s and the remaining bits are 0s. Because p has integer coefficients, we have $|p(y_0)| \geq 1$ and we are done with the base case. Now suppose without loss of generality that we have $p(y_i) \geq 2^i$ (the case of negative $p(y_i)$ is completely analogous), we will prove that we can fill the $(i+1)$'s block in y_i and get y_{i+1} such that $p(y_{i+1}) \leq -2^{i+1}$.

Let $P(z)$: $\{0,1\}^{t/2} \to \mathbb{R}$ be constructed by the following constraints on the input of p.

1. Every block up to i's is filled as in y_i.
2. In the $(i+1)$'s block every odd bit becomes a new variable, and every other bit is set to 0.
3. Let $x_1, \ldots, x_t$ be the positive bits in the current input and let $z = (z_1, \ldots, z_{t/2})$ be the new set of variables. We change the value of every x_i to $1 - z_i$.
4. All the remaining bits are also set to 0.

We first prove that for any z the achieved input is indeed in $\{0,1\}^n_{\leq k}$. Note that because we started with t 1s in y_0, every time we set z_i to 1 we change the corresponding x_i to 0. Thus, we never increase the number of 1s in the input, and because $k \geq t$ it is in $\{0,1\}^n_{\leq k}$.

Now suppose by contradiction that the desired y_{i+1} does not exist, i.e. for every x achieved by filling the $(i+1)$'s block of y_i, we have $|p(x)| \leq 2^{i+1}$. By definition, $P(0^{t/2}) = p(y_i) > 2^i$ but for every $z \neq 0^{t/2}$ we have $-2^{i+1} < p(z) < 0$ where the first inequality follows from $|P(x)| \leq 2^{i+1}$ and the second one follows from the definition of the ODD-MAX-BIT$_n$.

We claim that the polynomial $\tilde{P}(z) = -\frac{1}{3 \cdot 2^i} P(z)$ $\frac{1}{3}$-approximates the $OR_{t/2}$ function. Indeed, $\tilde{P}(0) = \frac{1}{3}$ and for every $z \in \{0,1\}^{t/2} \setminus \{0^{t/2}\}$ we have $\frac{2}{3} \leq \tilde{P}(z) \leq \frac{4}{3}$. It is well known (see [26]) that $\deg_{1/3}(OR_n) = \Omega(\sqrt{n})$. For a suitable choice of the hidden constant in t we get that $\deg(\tilde{P}) > \sqrt{t} = d$, but at the same time $\deg(\tilde{P}) = \deg(P) \leq \deg(p) = d$, which contradicts the non-existence of the desired y_{i+1}. ◀

5 Discussion

Despite having optimal bounds on decision lists for almost all $d = O(n^{1/3})$, the situation is less clear for other values of d. As we mentioned in the introduction, Servedio, Tan and Thaler [29, Theorem 6] proved that for any $n^{1/4} \leq d \leq n$ any decision list L: $\{-1,1\}^n \to \{-1,1\}$ on n variables has a degree-d PTF of weight $2^{\tilde{O}_n((n/d)^{2/3})}$. This gives an upper bound for $d = \Omega(n^{1/2})$, but it uses $\{-1,1\}^n$ domain rather than $\{0,1\}^n$. They also proved a lower bound of $2^{\Omega(\sqrt{n/d})}$ for both domains and $d = o\left(\frac{n}{\log^2 n}\right)$, which is stronger than Theorem 16 for $d = \Omega(n^{1/3})$. It uses the same approach, but a different technique for bounding the degree on a block, which suggests that a matching upper bound (if there is one) should have different construction than the one achieved by Corollary 12. We also note that because of the block approach this lower bound can be also adapted for Hamming Balls.

As for the approximation on $\{0,1\}^n_{\leq k}$, an obvious open question is to generalize Theorem 22 for $k = o(d^2)$. Another open question is to find optimal degree-weight tradeoffs for pointwise approximation of Boolean functions on both $\{0,1\}^n$ and $\{0,1\}^n_{\leq k}$, as suggested in [19]. We suspect that by including ε into analysis of Theorem 10 it can be generalized for any $\varepsilon = o(1)$ as well to achieve a similar to Theorem 11 result. After that, the similar ideas used in [6] and [19] can be applied to get degree-weight traidoff results for a class of decision lists.

References

1 Martin Anthony, Yves Crama, and Peter L. Hammer. Decision lists and related classes of boolean functions. In Yves Crama and Peter L. Hammer, editors, *Boolean Models and Methods in Mathematics, Computer Science, and Engineering*, pages 577–596. Cambridge University Press, 2010. doi:10.1017/cbo9780511780448.017.

2 Richard Beigel. The polynomial method in circuit complexity. In *Proceedings of the Eighth Annual Structure in Complexity Theory Conference*, pages 82–95. IEEE Computer Society Press, 1993.

3 Richard Beigel. Perceptrons, PP, and the polynomial hierarchy. *Comput. Complex.*, 4:339–349, 1994. doi:10.1007/BF01263422.

4 Arnab Bhattacharyya, Suprovat Ghoshal, and Rishi Saket. Hardness of learning noisy halfspaces using polynomial thresholds. In Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet, editors, *Conference On Learning Theory, COLT 2018, Stockholm, Sweden, 6-9 July 2018*, volume 75 of *Proceedings of Machine Learning Research*, pages 876–917. PMLR, 2018. URL: http://proceedings.mlr.press/v75/bhattacharyya18a.html.

5 Avrim Blum. Rank-r decision trees are a subclass of r-decision lists. *Inf. Process. Lett.*, 42(4):183–185, 1992. doi:10.1016/0020-0190(92)90237-P.

6 Andrej Bogdanov and Christopher Williamson. Approximate bounded indistinguishability. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPIcs*, pages 53:1–53:11. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.ICALP.2017.53.

7 Jehoshua Bruck and Roman Smolensky. Polynomial threshold functions, AC^0 functions, and spectral norms. *SIAM Journal on Computing*, 21(1):33–42, 1992.

8 Harry Buhrman, Nikolai K. Vereshchagin, and Ronald de Wolf. On computation and communication with small bias. In *22nd Annual IEEE Conference on Computational Complexity (CCC 2007), 13-16 June 2007, San Diego, California, USA*, pages 24–32. IEEE Computer Society, 2007. doi:10.1109/CCC.2007.18.

9 Mark Bun, Robin Kothari, and Justin Thaler. The polynomial method strikes back: Tight quantum query bounds via dual polynomials. *Theory Comput.*, 16:1–71, 2020. doi:10.4086/toc.2020.v016a010.

10 Mark Bun and Justin Thaler. A nearly optimal lower bound on the approximate degree of AC^0. *SIAM J. Comput.*, 49(4), 2020. doi:10.1137/17M1161737.

11 Mark Bun and Justin Thaler. The large-error approximate degree of AC^0. *Theory Comput.*, 17:1–46, 2021. URL: https://theoryofcomputing.org/articles/v017a007/.

12 Arkadev Chattopadhyay, Meena Mahajan, Nikhil S. Mande, and Nitin Saurabh. Lower bounds for linear decision lists. *Chic. J. Theor. Comput. Sci.*, 2020, 2020. URL: http://cjtcs.cs.uchicago.edu/articles/2020/1/contents.html.

13 Arkadev Chattopadhyay and Nikhil S. Mande. A short list of equalities induces large sign rank. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 47–58. IEEE Computer Society, 2018. doi:10.1109/FOCS.2018.00014.

14 Ronald de Wolf. A note on quantum algorithms and the minimal degree of ϵ-error polynomials for symmetric functions. *Quantum Inf. Comput.*, 8(10):943–950, 2008. doi:10.26421/QIC8.10-4.

15 Ilias Diakonikolas, Daniel M. Kane, and Alistair Stewart. Learning geometric concepts with nasty noise. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 1061–1073. ACM, 2018. doi:10.1145/3188745.3188754.

16 Ilias Diakonikolas, Ryan O'Donnell, Rocco A. Servedio, and Yi Wu. Hardness results for agnostically learning low-degree polynomial threshold functions. In Dana Randall, editor, *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 1590–1606. SIAM, 2011. doi:10.1137/1.9781611973082.123.

17 Mikael Goldmann, Johan Håstad, and Alexander A. Razborov. Majority gates vs. general weighted threshold gates. *Computational Complexity*, 2(4):277–300, 1992.

18 Lisa Hellerstein and Rocco A. Servedio. On PAC learning algorithms for rich boolean function classes. *Theor. Comput. Sci.*, 384(1):66–76, 2007. doi:10.1016/j.tcs.2007.05.018.

19 Xuangui Huang and Emanuele Viola. Approximate degree, weight, and indistinguishability. *ACM Trans. Comput. Theory*, 14(1):3:1–3:26, 2022. doi:10.1145/3492338.

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

21 Adam R. Klivans and Rocco A. Servedio. Learning DNF in time $2^{\widetilde{O}(n^{1/3})}$. *Journal of Computer and System Sciences*, 68(2):303–318, 2004.

22 Adam R. Klivans and Rocco A. Servedio. Toward attribute efficient learning of decision lists and parities. *J. Mach. Learn. Res.*, 7:587–602, 2006. URL: http://jmlr.org/papers/v7/klivans06a.html.

23 Matthias Krause and Pavel Pudlák. Computing boolean functions by polynomials and threshold circuits. *Computational Complexity*, 7(4):346–370, 1998.

24 Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Mach. Learn.*, 2(4):285–318, 1987. doi:10.1007/BF00116827.

25 Philip M. Long and Rocco A. Servedio. On the weight of halfspaces over hamming balls. *SIAM J. Discret. Math.*, 28(3):1035–1061, 2014. doi:10.1137/120868402.

26 Noam Nisan and Mario Szegedy. On the degree of boolean functions as real polynomials. *Comput. Complex.*, 4:301–313, 1994. doi:10.1007/BF01263419.

27 Ryan O'Donnell and Rocco A. Servedio. New degree bounds for polynomial threshold functions. *Comb.*, 30(3):327–358, 2010. doi:10.1007/s00493-010-2173-3.

28 Micheal E. Saks. Slicing the hypercube. In Keith Walker, editor, *Surveys in Combinatorics*, number 187 in London Mathematical Society Lecture Note Series, pages 211–256. Cambridge University Press, 1993.

29 Rocco A. Servedio, Li-Yang Tan, and Justin Thaler. Attribute-efficient learning and weight-degree tradeoffs for polynomial threshold functions. In Shie Mannor, Nathan Srebro, and Robert C. Williamson, editors, *COLT 2012 - The 25th Annual Conference on Learning Theory, June 25-27, 2012, Edinburgh, Scotland*, volume 23 of *JMLR Proceedings*, pages 14.1–14.19. JMLR.org, 2012. URL: http://proceedings.mlr.press/v23/servedio12/servedio12.pdf.

30 Alexander A. Sherstov. Communication lower bounds using dual polynomials. *Bulletin of the EATCS*, 95:59–93, 2008.

31 Alexander A. Sherstov. Algorithmic polynomials. *SIAM J. Comput.*, 49(6):1173–1231, 2020. doi:10.1137/19M1278831.

32 Alexander A. Sherstov. The approximate degree of DNF and CNF formulas. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 1194–1207. ACM, 2022. doi:10.1145/3519935.3520000.

33 Alexander A. Sherstov and Pei Wu. Near-optimal lower bounds on the threshold degree and sign-rank of AC^0. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 401–412. ACM, 2019. doi:10.1145/3313276.3316408.

34 Kei Uchizawa and Eiji Takimoto. Lower bounds for linear decision trees with bounded weights. In Giuseppe F. Italiano, Tiziana Margaria-Steffen, Jaroslav Pokorný, Jean-Jacques Quisquater, and Roger Wattenhofer, editors, *SOFSEM 2015: Theory and Practice of Computer Science - 41st International Conference on Current Trends in Theory and Practice of Computer Science, Pec pod Sněžkou, Czech Republic, January 24-29, 2015. Proceedings*, volume 8939 of *Lecture Notes in Computer Science*, pages 412–422. Springer, 2015. doi:10.1007/978-3-662-46078-8_34.

A Omitted proofs

Proof of Theorem 13. We can represent L as follows:

$$L(x) = \operatorname{sign} C\left(\sum_{i=1}^{n/h} 3^{n/h-i+1} f_i(x) + b_{n+1}\right).$$

with C from Theorem 10, where each f_i is a sublist responsible for the bits on the interval $x_{(i-1)h+1}, \ldots, x_{ih}$, which returns zero if no literals on this interval is satisfied. It is straightforward to check the correctness of the above decomposition: if f_i contains the first satisfied literal, then it determines the sign of the sum because $3^{n/h-i+1} > \sum_{j<i} 3^{n/h-j+1}$ and $f_j(x) = 0$ for every $j < i$.

Every sublist f_i is a modified decision list, so we can apply Corollary 12 and replace every $Cf_i(x)$ with $p_i(x)$, resulting in

$$H(x) = \sum_{i=1}^{n/h} 3^{n/h-i+1} p_i(x) + Cb_{n+1}.$$

Our goal is to show that $H(x)$ is the desired PTF. First of all, if $x = 0^n$ then every $p_i(x) = 0$ and $\operatorname{sign} H(x) = b_{n+1}$. Otherwise, let $r = (i-1)h + c$ be the first bit such that ℓ_r is satisfied. We have several cases:

1. If $j < i$ then $3^{n/h-i+1} p_j(x) = 0$;
2. $3^{n/h-i+1} p_i(x)$ differs from $3^{n/h-i+1} Cb_r$ by at most $C3^{n/h-i+1}\frac{1}{100}$;
3. The magnitude of each of the remaining values is at most $C3^{n/h-j+1}\left(1 + \frac{1}{100}\right)$.

Combining these bounds, the value of $H(x)$ differs from $3^{n/h-i+1} Cb_r$ by at most

$$C\left(1 + \frac{3^{n/h-i+1}}{100} + \left(1 + \frac{1}{100}\right)\sum_{j=1}^{n/h-i} 3^j\right).$$

The value of the sum is less than $\frac{3^{n/h-i+1}}{2}$, so the overall value is less than $C3^{n/h-i+1}$ and $\operatorname{sign} H(x) = b_r$. The bounds on degree and weight follows from Corollary 12. ◀