

The Log-Rank Conjecture for Linear Threshold Functions composed with AND

Nicholas Sieger and Aditya Krishnan

October 8, 2016

Abstract

In this paper, we show that the Log-rank conjecture holds for linear threshold functions (LTFs) of the form $f \circ \wedge$. The result relies on conjecture holding for monotone functions of the same form. We show that any LTF can be recursively broken down into monotone functions. We further explore the multilinear polynomial of functions of the form $f \circ \wedge$ and give characterizations of its monomial complexity in terms of the support of the function.

1 Introduction

Woohoo

2 Notation and Definitions

We first set our notation for the relevant complexity measures. Consider an arbitrary boolean function $F : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}$ and define $CC(F)$ denote the *communication complexity* of F . For such an F , let M_F denote its communication matrix.

For each subset $S \subseteq [n]$ define the polynomial $\alpha_S : \{0,1\}^n \rightarrow \{0,1\}$ by $\alpha_S(x_1, \dots, x_n) := \prod_{i \in S} x_i$ with $\alpha_\emptyset(x) = 1$. An *AND decision tree* is then a binary decision tree where each node makes an α_S query for some $S \subseteq [n]$. Such an AND decision tree T decides a boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$ if T agrees with f on all inputs $x \in \{0,1\}^n$. The *AND decision tree complexity* $DT_\wedge(f)$ is the minimum depth of an AND decision tree which decides f .

Given a boolean function f , we will frequently write f as a multilinear polynomial. It is shown in [2] that any boolean function f can uniquely written as $f(x) = \sum_{S \subseteq [n]} \hat{a}[S] \alpha_S(x)$ where $\hat{a}[S]$ is the coefficient of $\alpha_S(x)$. We denote by $\text{mon}(f)$ be number of nonzero $\hat{a}[S]$ for a function f . The *support* of a function f is defined as $\text{supp}(f) := f^{-1}(\{1\})$.

We will only consider a subclass of all boolean functions $f : \{0,1\}^n \rightarrow \{0,1\}$ which we now define. Let

$$f(x_1, \dots, x_n) := \begin{cases} 1 & w_1 w_1 + w_2 x_2 + \dots + w_n x_n \geq w_0 \\ 0 & \text{otherwise} \end{cases}$$

be a *linear threshold function* (LTF). We will assume without loss of generality that $w_0 \geq 0$. We form our communication function $F(x_1, \dots, x_n, y_1, \dots, y_n) := f(x_1 \wedge y_1, x_2 \wedge y_2, \dots, x_n \wedge y_n)$.

Finally, there are several relevant results to state. It is shown in [1] that $\text{rank}(M_F) = \text{mon}(f)$ (Note this result holds for a general f , not just LTFs). In addition, it is easy to see that $\text{CC}(F) \leq 2\text{DT}_\wedge(f)$. Thus, showing the following claim implies the Log-Rank Conjecture for our class of communication functions F .

Proposition 2.1. *If f is an LTF, $\text{DF}_\wedge(f) \leq \log^c(\text{mon}(f))$ for some $c \in \mathbb{R}^+$.*

3 Properties of Monomials

We characterize the coefficients of the polynomial expansion of a function based on its support. Although seemingly unmotivated, this characterization will help us in analysing the monomials of LTFs with respect to our proposed query algorithm.

Given a set $S \subseteq [n]$ let us define the set $E_S^f = \{x \in \text{sup}(f) \mid S_x \subseteq S \text{ and } |S_x| \text{ is even}\}$ and $O_S^f = \{x \in \text{sup}(f) \mid S_x \subseteq S \text{ and } |S_x| \text{ is odd}\}$ where x is the characteristic vector of the set S_v . Given these notions, we can explicitly calculate the coefficients $\hat{a}[S]$ of the polynomial computing f in terms of E_S^f and O_S^f .

Definition 3.1. Let us define $\mathbb{1}_x$ to be the identity multilinear polynomial that returns 1 only for input x . Then the expansion for $\mathbb{1}_x(y)$ is

$$\mathbb{1}_x(y) = \prod_{i \in S_x} y_i \prod_{j \in \bar{S}_x} (1 - y_j)$$

We now propose the lemma that characterizes the coefficients.

Lemma 3.2. *For any $S \subseteq [n]$ and function $f : \{0, 1\}^n \rightarrow \{0, 1\}$, the coefficients $\hat{a}[S]$ are given by*

$$\hat{a}[S] = -1^{|S|} (|E_S^f| - |O_S^f|)$$

Proof. Every function can be written in terms of the identity polynomial for each input. That is,

$$f(y) = \sum_{x \in \{0,1\}^n} f(y) \mathbb{1}_x(y) = \sum_{x \mid f(x)=1} \mathbb{1}_x(y)$$

We analyse the polynomial expansion for each $\mathbb{1}_y(x)$. The expansion is given by the product $\prod_{j \in \bar{S}_y} (1 - x_j)$. The monomials formed by expanding this product correspond to selecting either $-x_j$ or 1 from each term in the product. Hence we know that every monomial, denoted by a subset $A \subseteq \bar{S}_y$, must occur in the expansion. The coefficient of the monomial, $A \subseteq \bar{S}_y$, is 1 if $|A|$ is even and -1 otherwise. Hence from the polynomial expansion of $\mathbb{1}_y$ we know a monomial corresponding to some $S \subseteq [n]$ occurs in $\mathbb{1}_y x$ if $S_y \subseteq S$. Its coefficient is 1 if $|S| - |S_y|$ is even and -1 otherwise. The result follows by casing on whether $|S|$ is even or odd. \square

4 Breaking Down LTFs

In this section we show how an LTF can be broken down into monotone and antimonotone functions. We provide an AND Decision Tree algorithm to compute an LTF based on this decomposition and

then conclude that there exists a communication protocol that computes the LTF, formed by bitwise \wedge of the inputs to Alice and Bob, that is polylogarithmic in the monomial complexity of the LTF. Since $\text{rank}(M_{f_\wedge}) = \text{mon}(f_\wedge)$ from [1], we have that the Log-rank holds for LTFs of the mentioned form.

Throughout this section we will assume that f is a linear threshold function (LTF) of the form $f(x) = w_1x_1 + \dots + w_nx_n > w_0$. We assume without loss of generality that $w_0 \geq 0$.

For an input x , we will denote $W_f(x)$ to denote the set $\{w_i \mid x_i = 1\}$ for the LTF f . Naturally, we define $W_f^+(x)$ to be subset of $W_f(x)$ that has positive weights and correspondingly we define $W_f^-(x)$. Similarly, we define x^+ and x^- to be the substrings that correspond to the weights.

Definition 4.1. An input x is a *pivot* if $f(x) = 1$ and for any $x_i = 1$, flipping x_i to 0, x^{-i} , causes the function value to flip, $f(x^{-i}) = 0$.

Definition 4.2. An input x is an *antipivot* if $f(x) = 0$ and for any $x_i = 1, w_i < 0$, we have that $f(x^{-i}) = 1$.

Fact 4.3. There does not exist an input x' and a pivot x such that $S_{x'} \subset S_x$ and $f(x') = 1$. Similarly, this extends to bits corresponding to negative weights in antipivots.

The concept of function restrictions easily extends to LTFs. We shall consider the restrictions that restrict some of the input bits to 1. Let f_x be defined as the function that is restricted to the 0 bits of x . Clearly, the restricted function f_x has weights $\{w_1, \dots, w_n\} \setminus W_f(x)$ and has the condition that the linear combination of weights must be more than $w_0 - \sum_{w_i \in W_f(x)} w_i$. Notice that now that this expression is negative. We will define P_f, A_f to the set of inputs x for f that are pivots and antipivots respectively.

We are now ready to propose the AND Decision Tree (ADT) algorithm that computes $f(x)$.

```

Input:  $x \in \{0, 1^n\}$ 
1 while  $f$  is not a constant function do
2   if  $\exists p \in P_f$  such that  $p \subset x$  then
3     if  $\exists a \in A_{f_p}$  then
4        $f \leftarrow f_{p+a}$ 
5     else
6       return 1
7   else
8     return 0
9 return value of  $f$ 

```

Algorithm 1: LTF Decision Tree Query Algorithm

We now analyse the communication complexity of the protocol that simulates this AND Decision Tree query algorithm based on the monomial complexity. By inducting on n , the number of input variables, we shall inductively assume that $\text{CC}(f(x \wedge y)) \leq \log^c(\text{mon}(f))$. It is then left to show that in lines 2, 3 the pivot and antipivots can be found in polylogarithmic number of queries in the number of monomials. We shall define 2 such functions. Define the function $f'(x) = 1 \iff \exists p \in P_f \mid p \subseteq x$ and for some fixed pivot $p \in P_f$, we define $f^p(x) = 1 \iff \exists a \in A_{f_p} \mid a \subseteq x$.

Proposition 4.4. $f'(x)$ and $f^p(x)$ are montone

Proof. Since there cannot be a pair of pivots (antipivots) where one is a subset of another, Fact 4.3, the set $P_f(A_{f_p})$ forms a building set for the monotone function. \square

The functions $f', \neg f^p$ can clearly be used directly to compute lines 2, 3 respectively in the algorithm. As per [1?], the log-rank conjecture holds for monotone functions and thus there exists communication protocols that compute these functions in polylogarithmic number of communicated bits in the number of their monomials. It is left to show $\text{mon}(f'), \text{mon}(\neg f^p) \leq \text{mon}(f)$.

We shall show a property about monotone functions and their monomials before we reason about the monomial complexity of $f', \neg f^p$.

Lemma 4.5. *If $g : \{0, 1\}^n \rightarrow \{0, 1\}$ is a monotone function and let $M \subset \{0, 1\}^n$ be the building set for g , then we have that*

$$g(x) = \sum_{k=1}^{|M|} \sum_{A \subseteq M, |A|=k} -1^{k-1} \bigcup_{x \in A} x$$

where a string represents a monomial and the union of strings is the union of their characteristic sets.

Proof. Consider any input x in $\text{sup}(g)$. Let $A \subseteq M$ such that $\forall a \in A, a \subseteq x$. We can assume that x is simply a union of strings from A since the suggested polynomial contains monomials only of that form. Since any monomial of the form $y \not\subseteq x$ will evaluate to 0 and every monomial $y = \bigcup_{z \in A'} z, A' \subseteq A$ will evaluate to 1, we will have that the polynomial reduces to $\sum_{i=1}^{|A|} -1^{i-1} \binom{|A|}{i}$. We can see that this sum evaluates to 1 by looking at the binomial expansion of $-1((1-1)^{|A|} - \binom{|A|}{0}) = (\sum_{k=0}^{|A|} -1^{k+1} \binom{|A|}{k}) + \binom{|A|}{0}$. \square

Proposition 4.6. *The monomials of the function $f'(x), \neg f^p(x)$ are contained in those of $f(x)$; taking into account the restriction of f in f^p .*

Proof. Let us first consider f' . Notice that any monomial in f' , as per Lemma 4.5, is a union of pivots. For all $s \in \text{sup}(f)$ of this form (a union of pivots), s has the property that any $s' \in \text{sup}(f), s' \subseteq s$ also occurs in $\text{sup}(f')$ and vice versa. Thus, as per 3.2, the quantity $(-1)^{|s|}(|E_s| - |O_s|)$ is the same in f and f' .

Now, let us consider $\neg f^p$ where $p \in P_f$. The support of $\neg f^p$ is characterized by all s such that $\nexists a \in A_{f_p} \mid a \subseteq s$. By the definition of an antipivot, $(p \cup s) \in \text{sup}(f)$. Thus, by Lemma 3.2, any monomial a , which is a union of antipivots from A_{f_p} , must occur as $p \cup a$ in f . \square

Combining Proposition 4.6 with Algorithm 1 we have inductively shown that $\text{CC}(f \circ \wedge) \leq \log^c(\text{mon}(f))$ for $c \in O(1)$.

5 Conclusion

Using subroutines for monotone functions in an iteration for which the log-rank conjecture holds, we have inductively shown that the log-rank conjecture holds for all LTFs of the form $f \circ \wedge$. Although this result heavily uses a previous result, we conjecture that this algorithm is the optimum AND Decision Tree algorithm. Our result uses [1] to show the log-rank conjecture for the communication complexity. An interesting problem would be to show it holds for the AND Decision Tree complexity also.

References

- [1] H. Buhrman and R. de Wolf. Communication Complexity Lower Bounds by Polynomials. (Ilc): 16, 1999. ISSN 10930159. doi: 10.1109/CCC.2001.933879. URL <http://arxiv.org/abs/cs/9910010>. 2, 4, 9, 5
- [2] R. O'Donnell. *Analysis of Boolean Functions*, volume 1. 2007. ISBN 9781107038325. doi: 10.1017/CBO9781139814782. 2