

# The Log-Rank Conjecture for Linear Threshold Functions composed with $\wedge$

Nicholas Sieger, Aditya Krishnan and Anil Ada

August 8, 2016

## Abstract

In this note, we work towards showing 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 and argue that it is made of monotone parts. Based on the characterization of the polynomial, we suggest an algorithm in the AND decision tree setting to decide a boolean LTF which can be simulated by a communication protocol. Additionally, we show that each round of the communication protocol is at most polynomial in the rank of the function matrix.

## 1 Introduction

Let  $F : X \times Y \rightarrow \{0, 1\}$  be a function (hereafter referred to as a communication function) and  $M_F$  be its communication matrix, defined by  $(M_F)_{xy} = F(x, y)$ . The infamous Log-Rank Conjecture due to Lovász and Saks [3] claims that the deterministic communication complexity of  $F$ ,  $CC(F)$ , satisfies  $CC(F) \leq O(\text{polylog}(\text{rank}(M_F)))$ . Despite nearly 40 years of work, the best known result for general communication functions is that  $CC(F) \leq O(\sqrt{\text{rank}(M_F) \log(\text{rank}(M_F))})$ . Rather than tackle arbitrary communication functions, a number of authors focus on restricted classes of functions. One such class is functions of the form  $f(g(x_1, y_1), g(x_2, y_2), \dots, g(x_n, y_n))$  where  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  and  $g : \{0, 1\} \times \{0, 1\} \rightarrow \{0, 1\}$  is a “gadget” function, which we denote by  $f \circ g^{\oplus n}$ . By fixing  $g = \otimes$ , [7] and [5] began to verify the Log-Rank Conjecture in this restricted setting, and gave a close connection to XOR decision tree complexity. The connection between communication and decision tree complexity was also developed by [2] in a powerful query-to-communication lifting theorem. Furthermore, [2] used their lifting theorem to show an  $\Omega(\log^2)$  lower bound to  $CC(F)$ .

### 1.1 Our Work

We follow the direction proposed in [5] and consider functions of the form  $f \circ \wedge^{\otimes n}$ . In particular, we fix  $f$  to be an arbitrary *Linear Threshold Function* (LTF). Given our nicely restricted setting, we give a characterization of the multilinear polynomials which express an LTF, and propose an algorithm to build an AND decision tree from a communication protocol for  $f \circ \wedge^{\otimes n}$  in the hopes of proving

the Log-Rank Conjecture in this new setting. We compute the function by breaking the LTF into monotone and antimonotone counterparts. Then, by analysing the polynomial for the function, we show that the monomials of the polynomial consist of exactly the monotone and antimonotone functions that we compute in our proposed algorithm. Any AND decision tree algorithm for the  $\wedge$  gadget can be simulated by a communication protocol with atmost  $2DT_{\wedge}(f)$  bits communicated by Alice sending her required bit of the input followed by Bob sending the result of the querying the appropriate bit.

**Theorem 1.1.** *Let  $f : \{0,1\}^n \rightarrow \{0,1\}$  be a linear threshold function. Form a communication function  $F : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}$  by  $F(x_1, \dots, x_n, y_1, \dots, y_n) = f(x_1 \wedge y_1, x_2 \wedge y_2, \dots, x_n \wedge y_n)$ . Let  $M_F$  be the communication matrix of  $F$ . Then  $CC(F) \leq \text{polylog}(\text{rank}(M_F))$ .*

## 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  $\alpha_S$  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 [6] 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  $\text{sup}(f) := f^{-1}(\{1\})$ .

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  $CC(F) \leq 2DT_{\wedge}(f)$ . 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,  $DF_{\wedge}(f) \leq \log^c(\text{mon}(f))$  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,

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

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 corresponding to the set  $A \subseteq \bar{S}_y$  is 1 if  $|A|$  is even and  $-1$  otherwise. Notice that the  $\prod_{i \in S_y} x_i$  term exists in every monomial of the expansion of  $\mathbb{1}_y(x)$ . Hence we know a monomial corresponding to some  $S \subseteq [n]$  occurs in  $\mathbb{1}_y(x)$  if and only if  $S_y \subseteq S$ . Also notice that the coefficient of this monomial is 1 if  $|S| - |S_y|$  is even and  $-1$  otherwise since the parity of the coefficient corresponds to the number of indices picked from  $\bar{S}_y$ . Hence, summing over all  $\mathbb{1}_y(x)$  for all  $y \in \{0, 1\}^n$  gives us the result. □

### 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. We use the fact that the Log-Rank holds for monotone functions as an algorithm to compute them as a subroutine for our query algorithm. We then characterize the polynomials of LTFs showing that they're 'made up' of the monotone and antimonotone functions we used to compute it.

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  $f$ . We then 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 of  $x$  that correspond to the positive and negative weights respectively.

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

**Definition 4.2.** An input  $x$  is an antipivot if  $f(x) = 0$  and for any  $x_i = 1$  and  $w_i < 0$ , 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 for any LTF that has non-zero support. We will define the sets  $P_f, A_f$  to be 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  $\exists a \in A_{f_p}$  such that  $a \subset x$  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. Let us first show that in lines 2, 3 the pivot and antipivots can be found in polylogarithmic number of queries in the number of monomials of the original function  $f$ . To do so, we shall define two 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 monotone

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

Recall that a building set for a monotone function correspond to the inputs in the support with minimum Hamming weight that are not subsets of each other.

The functions  $f', \neg f^p$  can clearly be used directly to compute lines 2, 3 respectively in the algorithm. As per [1, 4], 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*

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

where a string represents a monomial corresponding to its characteristic set 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. It is easy to see that inputs not in the support will be evaluate to 0 since the function is monotone and the polynomial contains monomials only in the support. Since every monomial  $y = \bigcup_{z \in A'} z$  such that  $A' \subseteq A$  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  $s' \in \text{sup}(f)$  with  $s' \subseteq s$  also occurs in  $\text{sup}(f')$  and vice versa. Thus, as per Lemma 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  $f$ .  $\square$

## 5 Conclusion

It is left to show in the analysis of our algorithm that the monomial complexity of our function drops sufficiently fast enough. Although unable to complete this analysis we conjecture that this query algorithm when simulated by a communication protocol is  $\text{poly log}(\text{rank } M_f)$ . Notice that a subroutine used by our algorithm is a communication protocol that computes a monotone function. Hence, it is not clear that this subroutine can be computed with few queries in the  $\text{DT}_\wedge$  setting. Thus, an interesting problem would be to show that the Log-Rank conjecture holds for the AND Decision Tree complexity also. A stepping-stone in showing this would be to give an explicit algorithm for monotone functions for which  $\text{DT}_\wedge(f) \leq \text{poly log}(\text{mon } f)$ .

## 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, 9
- [2] M. Goos, T. Pitassi, and T. Watson. Deterministic communication vs. partition number. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, volume 2015-Decem, pages 1077–1088. IEEE, oct 2015. ISBN 978-1-4673-8191-8. doi: 10.1109/FOCS.2015.70. URL <http://ieeexplore.ieee.org/document/7354444/>. 1
- [3] L. Lovasz and M. Saks. Lattices, mobius functions and communications complexity. In *[Proceedings 1988] 29th Annual Symposium on Foundations of Computer Science*, pages 81–90. IEEE, 1988. ISBN 0-8186-0877-3. doi: 10.1109/SFCS.1988.21924. URL <http://ieeexplore.ieee.org/document/21924/>. 1
- [4] L. Lovasz and M. Saks. Communication complexity and combinatorial lattice theory. *Journal of Computer and System Sciences*, 47(2):322–349, 1993. ISSN 10902724. doi: 10.1016/0022-0000(93)90035-U. 9
- [5] S. Lovett. The fourier structure of low degree polynomials. *Electronic Colloquium on Computational Complexity*, 25(25):1–17, feb 2016. URL <http://arxiv.org/abs/1603.00002>. 1, 1.1
- [6] R. O’Donnell. *Analysis of Boolean Functions*, volume 1. 2007. ISBN 9781107038325. doi: 10.1017/CBO9781139814782. 2
- [7] H. Y. Tsang, C. H. Wong, N. Xie, and S. Zhang. Fourier sparsity, spectral norm, and the log-rank conjecture. pages 0–24, 2013. ISSN 02725428. doi: 10.1109/FOCS.2013.76. URL <http://arxiv.org/abs/1304.1245>. 1