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

Nicholas Sieger and Aditya Krishnan

August 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 montone functions of the same form. We show that any LTF can be recursively broken down into montone functions. We further explore the multilinear polynomial of functions of the form $f \circ \wedge$ and give characterizations of its the monomial complexity in terms of the support of the function. We further suggest an algorithm in the AND decision tree setting to decide a boolean LTF.

1 Introduction

The *Log-Rank Conjecture* of Lovász and Saks [3] claims that the communication complexity of any boolean function $f : X \times Y \to \{0,1\}$ is a polynomial of the rank of the natural communication matrix. Despite decades of work, the best known bound, due to Lovett [5], is $\sqrt{\mathsf{r}\log(\mathsf{r})}$. Given the challenges of showing general upper bounds, many authors focused on restricted classes of functions, especially functions of the form $f(g(x_1, y_1), g(x_2, y_2), \ldots, f(x_n, y_n))$ where $f : \{0,1\}^n \to \{0,1\}$ and $g : \{0,1\} \times \{0,1\} \to \{0,1\}$ is a "gadget" function. This direction proved quite fruitful in [2],[8],and [6].

In this note, we consider communication functions of the form $f(x_1 \wedge y_1, x_2 \wedge y_2, \ldots, x_n \wedge y_n)$ where $f : \{0,1\}^n \to \{0,1\}$ is a linear threshold function. Moreover, our methods follow the ideas of [6] in that we relate the Communication complexity to AND decision tree complexity. Using these techniques, we show the following:

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

2 Notation and Definitions

We first set our notation for the relevant complexity measures. Consider an arbitrary boolean function $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ and define $\mathsf{CC}(F)$ 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 $\alpha_S : \{0,1\}^n \to \{0,1\}$ 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 $f : \{0,1\}^n \to \{0,1\}$ if T agrees with f on all inputs $x \in \{0,1\}^n$. The AND *decision tree complexity* $\mathsf{DT}_\wedge(f)$ 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 [7] that any boolean function f can uniquely written as $f(x) = \sum_{S \subseteq [n]} \hat{\alpha}[S] \alpha_S(x)$ where $\hat{\alpha}[S]$ is the coefficient of $\alpha_S(x)$. We denote by $\mathsf{mon}(f)$ be number of nonzero $\hat{\alpha}[S]$ for a function f. The *support* of a function f is defined as $\mathsf{sup}(f) := f^{-1}(\{1\})$.

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

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

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 $\mathsf{rank}(M_F) = \mathsf{mon}(f)$ (Note this result holds for a general f, not just LTFs). In addition, it is easy to see that $\mathsf{CC}(F) \leq 2\mathsf{DT}_\wedge(f)$. 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, $\mathsf{DF}_\wedge(f) \leq \log^c(\mathsf{mon}(f))$ 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 $E_S^f = \{x \in \mathsf{sup}(f) \mid S_x \subseteq S \text{ and } |S_x| \text{ is even}\}$ and $O_S^f = \{x \in \mathsf{sup}(f) \mid S_x \subseteq S \text{ and } |S_x| \text{ is odd}\}$ 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 $f : \{0,1\}^n \to \{0,1\}$, 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 indentity polynomial for each input. That is,

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

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 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 $\text{rank}(M_{f_\wedge}) = mon(f_\wedge)$ 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 $f(x) = w_1x_1 + ...w_nx_n > w_0$. 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 $\{w_1, ..., w_n\} \backslash W_f(x)$ 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 $CC(f(x \wedge y)) \leq \log^c(mon(f))$. 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 montone function. □

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 $mon(f'), mon(\neg f^p) \leq mon(f)$.

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

Lemma 4.5. *If $g : \{0,1\}^n \to \{0,1\}$ 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 $\sup(g)$. 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
$-1((1-1)^{|A|} - \binom{|A|}{0}) = (\sum_{k=0}^{|A|} -1^{k+1}\binom{|A|}{k}) + \binom{|A|}{0}$. □

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 $s \in \sup(f)$ of this form (a union of pivots), s has the property that any $s' \in \sup(f)$, $s' \subseteq s$ also occurs in $\sup(f')$ 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, $(p \cup s) \in \sup(f)$. 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 $CC(f \circ \wedge) \leq \log^c(mon(f))$ for $c \in O(1)$.

For $\vec{x} \in \mathbb{F}_2^n$ we define $\vec{x}_{-i} := \vec{x} + 1_i$.

Definition 4.7. Consider $f : \mathbb{F}_2^n \to \mathbb{F}_2$. We say that $\vec{x} \in \mathbb{F}_2^n$ is a pivot of f if $f(\vec{x}) = 1$ and for any i if $\vec{x}_i = 1$ then $f(\vec{x}_{-i}) = 0$.

Definition 4.8. Consider $f : \mathbb{F}_2^n \to \mathbb{F}_2$. We say that $\vec{x} \in \mathbb{F}_2^n$ is a antipivot of f if $f(\vec{x}) = 1$ and for any i if $\vec{x}_i = 0$ then $f(\vec{x}_{-i}) = 0$.

Definition 4.9. Let $f : \{0,1\} \to \{0,1\}$ be a boolean function. We say that f is *convex on the hypercube* if for $\vec{x}, \vec{y} \in \mathsf{supp}(f)$ there is a shortest path P between the vertices x, y on the hypercube and every vertex of P is also in $\mathsf{supp}(f)$

Lemma 4.10. *LTFs are convex on the hypercube.*

Proof. Let $f(\vec{z}) = \langle \vec{w}, \vec{z} \rangle \geq w_0$ be a linear threshold function. and let $\vec{x}, \vec{y}$ be arbitrary vertices of the hypercube H^n. Without loss of generality, we can take $\vec{x} = \vec{0}$. To see this, rotate the hypercube and relabel the vertices $\vec{z} \in H^n$ by $\vec{z} \to \vec{z} + \vec{x}$ (where addition is taken over $\mathbb{F}_2^n$). Then consider $f'(\vec{z}) = \langle \vec{w}, \vec{z} + \vec{x} \rangle \geq w_0$, which is again an LTF. FILL IN MORE DETAILS

Take the subcube $C_{\vec{y}} \subseteq H^n$ formed by all $\vec{z} \leq \vec{y}$ (under the hamming order). Clearly, $C_{\vec{y}}$ contains all shortest paths from $\vec{0}$ to $\vec{y}$. Let $i_1, i_2, \ldots, i_k$ be the nonzero indices of $\vec{y}$ ordered such that $w_{i_1} \geq w_{i_2} \geq \cdots \geq w_{i_k}$. Consider the shortest

$\square$

Lemma 4.11. *Let f be an LTF and $\vec{a}$ an antipivot of f.If $\vec{x} \geq \vec{a}$ in the Hamming order, then $f(\vec{x}) = 0$.*

Proof. Assume not, and let $\vec{x} \in \mathsf{supp}(f)$ satisfy $\vec{a} \leq \vec{x}$ for some antipivot $\vec{a}$. By 4.9 some shortest path between $\vec{x}$ and $\vec{a}$ must be part of the support. Thus, $\vec{a}_{-i} \in \mathsf{supp}(f)$ for some i, contradicting the definition of antipivot 4.8. $\square$

Lemma 4.12. *Let f be an LTF and $\vec{p}$ a pivot of f.If $\vec{x} \leq \vec{p}$ in the Hamming order, then $f(\vec{x}) = 0$.*

Proof. $\square$

Definition 4.13. Let f be an LTF. Define

$$f^+(\vec{x}) := \begin{cases} 1 & \vec{x} \geq \vec{p} \text{ for some pivot } \vec{p} \text{ of } f \\ 0 & \text{otherwise} \end{cases}$$

$$f^-(\vec{x}) := \begin{cases} 1 & \vec{x} > \vec{a} \text{ for some antipivot } \vec{a} \text{ of } f \\ 0 & \text{otherwise} \end{cases}$$

Proposition 4.14. *f_+, f_- are monotone.*

Proof. This follows from the transitivity of the hamming order. □

Proposition 4.15. $f(\vec{x}) = f^+(\vec{x}) * (1 + f_-(\vec{x}))$

Proof. By 4.12 and 4.11, □

Corollary 4.16. *Let F, F_+, F_- be the communication functions formed by f, f_+, f_- respectively, and M_F, M_+, M_- their corresponding communication matrices. Then*

$$\mathsf{rank}(M_F) = \mathsf{rank}(M_+) * (1 + \mathsf{rank}(M_-)) = \Theta(\mathsf{rank}(M_+)\mathsf{rank}(M_-))$$

.

Proof. By 4.15 and the uniqueness of multilinear representations, $\mathsf{mon}(f) = \mathsf{mon}(f_+)(1 + \mathsf{mon}(f_-))$. By [1],

$$\mathsf{rank}(M_F) = \mathsf{mon}(f) = \mathsf{mon}(f_+)(1+\mathsf{mon}(f_-) = \mathsf{rank}(M_+)(1+\mathsf{rank}(M_-)) = \Theta(\mathsf{rank}(M_+)\mathsf{rank}(M_-))$$

□

We can now combine all these parts to prove our main claim.

proof of 1.1. By 4.14, there are protocols P_+, P_- for F_+, F_- such that P_+ communicates $O(\mathsf{polylog}(\mathsf{rank}(M_+)))$ bits and P_- communicates $O(\mathsf{polylog}(\mathsf{rank}(M_-)))$ bits. By 4.15, we can form the following protocol. First run P_+ on the input $\vec{x}$. If $\vec{x} \le \vec{p}$ for some pivot $\vec{p}$, we return 0. Otherwise, run P_-. If $\vec{x} \ge \vec{a}$ for some antipivot $\vec{a}$, we can return 0. Otherwise, return 1. Thus,

$$\begin{aligned} CC(F) &\le CC(F_+) + CC(F_-) \\ &\le O(\mathsf{polylog}(\mathsf{rank}(M_+))) + \mathsf{polylog}(\mathsf{rank}(M_-))) \end{aligned}$$

By 4.16 and submodularity

$$\le O\left(\mathsf{polylog}(\mathsf{rank}(M_+)\mathsf{rank}(M_-))\right)$$

□

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. (Illc):16, 1999. ISSN 10930159. doi: 10.1109/CCC.2001.933879. URL http://arxiv.org/abs/cs/9910010. 2, 4, 9, 5

[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. Communication is bounded by root of rank. *Journal of the ACM*, 63(1):1:1–1:9, 2016. ISSN 00045411. doi: 10.1145/2591796.2591799. URL http://10.0.4.121/2724704{%}255Cnhttp://search.ebscohost.com/login.aspx?direct=true{&}db=buh{&}AN=113046637{&}site=ehost-live. 1

[6] S. Lovett. The fourier structure of low degree polynomials. *Electronic Colloquim on Computational Complexity*, 25(25):1–17, feb 2016. URL http://arxiv.org/abs/1603.00002. 1

[7] R. O'Donnell. *Analysis of Boolean Functions*, volume 1. 2007. ISBN 9781107038325. doi: 10.1017/CBO9781139814782. 2

[8] 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