

On the Power of Parity Queries in Boolean Decision Trees

Raghav Kulkarni¹, Youming Qiao^{1(✉)}, and Xiaoming Sun²

¹ Centre for Quantum Technologies,
The National University of Singapore, Singapore, Singapore
{kulraghav86,jimmyqiao86}@gmail.com

² Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
sunxiaoming@ict.ac.cn

Abstract. In an influential paper, Kushilevitz and Mansour (1993) introduced a natural extension of Boolean decision trees called *parity decision tree* (PDT) where one may query the sum modulo 2, i.e., the *parity*, of an arbitrary subset of variables. Although originally introduced in the context of learning, parity decision trees have recently regained interest in the context of communication complexity (cf. Shi and Zhang 2010) and property testing (cf. Bhrushundi, Chakraborty, and Kulkarni 2013). In this paper, we investigate the power of parity queries. In particular, we show that the parity queries can be replaced by ordinary ones at the cost of the *total influence* aka *average sensitivity* per query. Our simulation is tight as demonstrated by the parity function.

At the heart of our result lies a qualitative extension of the result of O’Donnell, Saks, Schramme, and Servedio (2005) titled: *Every decision tree has an influential variable*. Recently Jain and Zhang (2011) obtained an alternate proof of the same. Our main contribution in this paper is a simple but surprising observation that the query elimination method of Jain and Zhang can indeed be adapted to eliminate, seemingly much more powerful, parity queries. Moreover, we extend our result to *linear* queries for Boolean valued functions over arbitrary finite fields.

1 Introduction

The decision tree model [8], perhaps due to its simplicity and fundamental nature has been extensively studied over decades, yet remains a fascinating source of some of the outstanding open questions. In the first part of this paper we focus on decision trees for Boolean functions, i.e., functions of the

Raghav Kulkarni—Research at the Centre for Quantum Technologies is funded by the Singapore Ministry of Education and the National Research Foundation.

Xiaoming Sun—Part of this work was done while the author was visiting the Centre for Quantum Technologies, National University of Singapore. He is supported in part by the National Natural Science Foundation of China Grant 61170062, 61222202, 61433014 and the China National Program for support of Top-notch Young Professionals.

form $f : \{0, 1\}^n \rightarrow \{0, 1\}$. In later section, we extend our results for decision trees over any finite field, i.e., for functions of the form $\mathbb{F}_q^n \rightarrow \{0, 1\}$. A deterministic decision tree D_f for f takes $x = (x_1, \dots, x_n)$ as an input and determines the value of $f(x_1, \dots, x_n)$ using queries of the form “is $x_i = 1$?”. Let $C(D_f, x)$ denote the cost of the computation, i.e., the number of queries made by D_f on input x . The *deterministic decision tree complexity* of f is defined as $D(f) = \min_{D_f} \max_x C(D_f, x)$.

Variants of decision tree model are fundamental for several reasons including their connection to other models such as communication complexity, their usability in analyzing more complicated models such as circuits, their mathematical elegance and richness, and finally the notoriety of some simple yet fascinating open questions about them such as the Evasiveness Conjecture [3, 14, 15, 19, 22] that have caught the imagination of generations of researchers over decades. In this paper we study a variant of decision trees called *parity decision tree* (PDT) and its extension over finite fields, which we call *linear decision tree* (LDT).

Motivation for Studying PDTs and LDTs

A parity decision tree may query “is $\sum_{i \in S} x_i \equiv 1 \pmod{2}$?” for an arbitrary subset $S \subseteq [n] = \{1, 2, \dots, n\}$. We call such queries *parity queries*. For a PDT P_f for f , let $C(P_f, x)$ denote the number of parity queries made by P_f on input x . The *parity decision tree complexity* of f is $D^\oplus(f) = \min_{P_f} \max_x C(P_f, x)$. Note that $D^\oplus(f) \leq D(f)$ as “is $x_i = 1$?” can be treated as a parity query.

The PDTs were introduced by Kushilevitz and Mansour [17] in the context of learning Boolean functions by estimating their Fourier coefficients. Several other models such as circuits and branching programs have been also been analysed in the past after augmenting their power by allowing counting operations.

In spite of being combinatorially rich and beautiful model, the PDT somehow remained dormant until recently where it was brought back into light in an entirely different context, namely the *communication complexity* of XOR functions [23, 31]. Shi and Zhang [31] and Montanaro and Osborne [23] have observed that the deterministic communication complexity $CC(f^\oplus)$ of computing $f(x \oplus y)$, when x and y are distributed between the two parties, is upper bounded by $D^\oplus(f)$. The importance for communication complexity comes from the conjecture [23, 31] that for some positive constant c , every Boolean function f satisfies $D^\oplus(f) = O((\log \|\widehat{f}\|_0)^c)$; where $\|\widehat{f}\|_0$ is the *sparsity* (number of non-zero Fourier coefficients) of f . Settling this conjecture in affirmative would confirm the famous Log-rank Conjecture [24] in the important special case of XOR functions. Recently Tsang et al. [36] confirm it for functions with constant degree over \mathbb{F}_2 and Kulkarni and Santha [18] confirm it for AC^0 functions.

Very recently, Bhrushundi, Chakraborty, and Kulkarni [4] connected parity decision trees to property testing of linear and quadratic functions. Their approach for instance can potentially be used to solve a long-standing open question of closing the gap for k -linearity by analysing the randomized PDT complexity of the function E_k that evaluates to 1 iff the number of 1s in the input is exactly k . Recently PDTs were analysed further in several papers including [18, 32, 34, 36] and many more to come.

Similar to PDTs, the LDTs are closely related to the Fourier spectrum of functions over \mathbb{Z}_p . In recent paper by Shpilka, Tal, and Volk [32] the authors derive various structural results of the Fourier spectrum by analysing LDTs. Given the evidence of abundance of connections to other models and mathematics, and given the rich combinatorial structure of PDTs and LDTs, we believe that they deserve a systematic and independent study at this point. Our paper is a step in this direction.

Motivation for Studying Influence Lower Bounds

Proving lower bounds on the influence of Boolean functions has had a long history in Theoretical Computer Science. It is nicely summarized in the paper [29], we restate a part from that for illustration. Influence lower bounds have been crucial part of several fundamental results such as threshold phenomenon, lower bound on randomized query complexity of graph properties, quantum and classical equivalence etc. Ben-Or and Linial [6], in their 1985 paper on collective coin flipping, observe that the maximum influence $\text{Inf}_{\max}(f) \geq 1/n$ for any balanced function and conjectured $\Theta(\log n/n)$ bound. The seminal paper by Kahn, Kalai, Linial [16] confirmed the conjecture via an application of the *Hypercontractive Inequality*. This result was subsequently generalized by Talagrand [35] in order to show *sharp threshold behaviour* for monotone functions.

In their celebrated paper *Every decision tree has an influential variable*, O'Donnell, Saks, Schramme, and Servedio [29] showed a crucial inequality lower bounding the maximum influence: $\text{Inf}_{\max}(f) \geq \text{Var}(f)/\Delta(f)$, where $\Delta(f)$ denotes the minimum possible average depth of a decision tree for f . This inequality found application in the lower bounds on randomized query complexity of monotone graph properties. Homin Lee [20] found a simple inductive proof of the OSSS result. Recently Jain and Zhang [13] found another simple and conceptually different proof via the method of query elimination, which we use here.

Aaronson and Ambainis [1] study a conjecture lower bounding the maximum influence of real valued polynomials in terms of their degree. This conjecture, if true, would imply polynomial equivalence between bounded-error quantum and classical query complexity. These previous results seems to indicate the importance of lower bounds on influence in terms of several complexity measures. In this paper, we present such new lower bounds in terms of PDT and LDT complexity.

Our Results

Let $D_\epsilon(f)$ and $D_\epsilon^\oplus(f)$ denote the minimum depth of a DT and a PDT (resp.) computing f correctly on at least $1 - \epsilon$ fraction of the inputs.

Theorem 1. *For any Boolean function f and any $\epsilon \geq 0$:*

$$\text{Inf}_{\max}(f) \geq \frac{\text{Var}(f) - \epsilon}{D_\epsilon^\oplus(f)}.$$

Corollary 1. *For any Boolean function f and any $\epsilon > 0$:*

$$D_\epsilon(f) \leq \frac{1}{\epsilon^2} \cdot D^\oplus(f) \cdot \text{Inf}(f).$$

Corollary 2. *If f is computable by a polynomial size constant depth circuit, i.e., $f \in AC^0$, then:¹*

$$D_\epsilon(f) = \tilde{O}_\epsilon(D^\oplus(f)).$$

To prove Theorem 1 we use an adaptation of the query elimination method of Jain and Zhang. Our main observation is that assuming the uniform distribution on the inputs, one can eliminate seemingly powerful parity queries at the expense of $\text{Inf}_{\max}(f)$ error per elimination. Corollary 1 is obtained by analysing the ‘query the most influential variable’ strategy using our new bound. We extend Theorem 1 for LDTs over arbitrary fields (see Sect. 4). The Corollary 1 can also be extended with similar techniques; we omit its simple proof.

Theorem 2. *Let q be a prime power. For any $f : \mathbb{F}_q^n \rightarrow \{0, 1\}$ and any $\epsilon \geq 0$:*

$$\text{Inf}_{\max}(f) \geq \frac{1}{q-1} \cdot \frac{\text{Var}(f) - \epsilon}{D_\epsilon^{\oplus q}(f)}.$$

Further we explore the power of PDTs for monotone functions and show:

Theorem 3. *For any monotone Boolean function f and any $\epsilon > 0$:*

$$D_\epsilon(f) \leq \frac{3}{\epsilon^2} \cdot D^\oplus(f)^{3/2}.$$

To prove Theorem 3 we show an upper bound on L_1 norm of Fourier spectrum in terms of PDT depth, which in turn gives an upper bound on sum of linear Fourier coefficients restricted to monotone functions. We adapt the proof of the same for ordinary decision trees by O’Donnell and Servedio. Our main observation is that under the uniform distribution on inputs their proof can be extended for PDTs as well. Our result naturally raises the following question:

Question 1. Is it true that for every monotone Boolean function f and for every $\epsilon > 0$ we have:

$$D_\epsilon(f) = \tilde{O}_\epsilon(D^\oplus(f))?$$

It is also interesting to see if our results can be strengthened to D_ϵ^\oplus rather than just D^\oplus as zero-error and bounded error complexities may behave differently.

We believe that our observations, although might appear simple, are indeed surprising. They seem to make a crucial qualitative point, that under the uniform distribution, the method of lower bounding the ordinary (randomized) decision tree complexity by $\text{Var}(f)/\text{Inf}_{\max}(f)$ works equally well for seemingly much more powerful PDTs and LDTs as well. For non-balanced functions the uniform distribution does not seem to be an optimal choice for maximizing $\text{Var}(f)/\text{Inf}_{\max}(f)$ but for balanced functions it does. As an application, finally we exhibit a gap between randomized PDT complexity and approximate L_1 , both of which are relevant for communication complexity of XOR functions.

¹ The O_ϵ notation hides a multiplicative constant depending on ϵ and the \tilde{O}_ϵ notation hides a further poly-logarithmic multiplicative factor.

Organization. Section 2 contains preliminaries. Section 3 contains the proof of Theorem 1. Section 4 contains the proof of Theorem 2. Unfortunately, we had to move the other proofs to appendix and hence omit it from this version due to space constraint.

2 Preliminaries

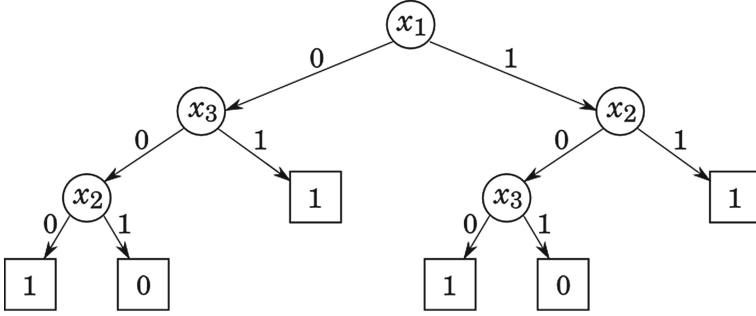


Fig. 1. A boolean decision tree

Randomized Decision Trees

A bounded error randomized decision tree R_f is a probability distribution over all deterministic decision trees such that for every input, the expected error of the algorithm is bounded by some fixed constant less than $1/2$ (say $1/3$). The cost $C(R_f, x)$ is the highest possible number of queries made by R_f on x , and the *bounded error randomized decision tree complexity* of f is $R(f) = \min_{R_f} \max_x C(R_f, x)$. Similarly one can define bounded error randomized PDT complexity of f , denoted by $R^\oplus(f)$. Using Yao’s min-max principle one may obtain: $D_{1/3}(f) \leq R(f)$ and $D_{1/3}^\oplus(f) \leq R^\oplus(f)$. (Fig. 1)

Variance and Influence

Let μ_p denote the p -biased distribution on the Boolean cube, i.e., each coordinate is independently chosen to be 1 with probability p . The *variance* of a Boolean function is $\text{Var}(f, p) := 4 \cdot \Pr_{x \leftarrow \mu_p}(f(x) = 0) \Pr_{x \leftarrow \mu_p}(f(x) = 1)$. The *influence* of the i^{th} variable under μ_p is $\text{Inf}_i(f, p) := \Pr_{x \leftarrow \mu_p}(f(x) \neq f(x \oplus e_i))$. Let $\text{Inf}_{\max}(f) := \max_i \text{Inf}_i(f)$. The *total influence* aka *average sensitivity* of f is $\text{Inf}(f, p) := \sum_i \text{Inf}_i(f, p)$. In this paper we focus on $p = 1/2$ case.

Fourier Spectrum, Polynomial Degree, and Sparsity

Let $f_\pm : \{-1, 1\}^n \rightarrow \{-1, 1\}$ be represented by the following polynomial with real coefficients: $f_\pm(z_1, \dots, z_n) = \sum_{S \subseteq [n]} \hat{f}(S) \prod_{i \in S} z_i$. The above polynomial is unique and it is called the Fourier expansion of f . The $\hat{f}(S)$ are called the Fourier coefficients of f . The *polynomial degree* of f is $\text{deg}(f) := \max\{|S| \mid \hat{f}(S) \neq 0\}$. The *sparsity* of a Boolean function f is $\|\hat{f}\|_0 := |\{S \mid \hat{f}(S) \neq 0\}|$. We know that $\text{deg}(f) \leq D(f)$, $\log \|\hat{f}\|_0 \leq D_\oplus(f)$ and $\log \|\hat{f}\|_0 \leq \text{deg}(f)$.

Representing Decision Trees

We represent a decision tree T as $T = (x_i, T_0, T_1)$ where x_i denotes the first variable queried by T , i.e., x_i is the variable at the root of T : if $x_i = 0$ then T_0 is consulted; if $x_i = 1$ then T_1 is consulted. A leaf labeled 1 is represented as $(1, \emptyset, \emptyset)$ and the one labeled 0 is represented as $(0, \emptyset, \emptyset)$. We represent a parity decision tree as $T = (x_S, T_0, T_1)$; if $\sum_{i \in S} x_i = 0 \pmod{2}$ then consult T_0 , else consult T_1 . A leaf labeled 1 is represented as $(1, \emptyset, \emptyset)$ and the one labeled 0 is represented as $(0, \emptyset, \emptyset)$.

The Query Elimination Lemma (Jain and Zhang)

Jain and Zhang prove the following simple yet powerful lemma:

Lemma 1 (Query Elimination Lemma). *If $T = (x_i, T_0, T_1)$ is an ordinary decision tree that computes f correctly on at least $1 - \delta$ fraction of the inputs then either T_0 or T_1 computes f correctly on at least $1 - \delta - \text{Inf}_i(f)$ fraction of the inputs.*

In this paper we observe that the above lemma can be adapted for parity decision trees. This observation is a crucial part of our results.

Overview of the Query Elimination Method

The *query elimination method* of Jain and Zhang works as follows: Suppose we have a decision tree of depth $D_\epsilon(f)$ that computes f correctly on at least $1 - \epsilon$ fraction of the inputs. We repeatedly apply the Query Elimination Lemma to obtain a decision tree that computes f correctly on at least $1 - \epsilon - D_\epsilon(f) \cdot \text{Inf}_{\max}(f)$ fraction of the inputs without making any single query. Of course, such (zero-query) decision tree must make error on at least $\text{Var}(f)$ fraction of the inputs. Hence: the error of the zero-query decision tree that we obtained ($\epsilon + D_\epsilon(f) \cdot \text{Inf}_{\max}(f)$) can be lower bounded by $\text{Var}(f)$. In other words:

$$D_\epsilon(f) \geq \frac{\text{Var}(f) - \epsilon}{\text{Inf}_{\max}(f)}.$$

3 Every PDT Has an Influential Variable

In this section we present the proof of Theorem 1. We start with eliminating queries in PDTs.

Eliminating Ordinary Queries in PDTs

First we note that Jain and Zhang's proof of the Query Elimination Lemma generalizes when T_i are parity decision trees instead of ordinary ones. In other words, if the first query in a parity decision tree is an ordinary query then one can remove it at the expense of $\text{Inf}_i(f)$ increase in the error. We formulate this below.

Lemma 2. *If $T = (x_{\{i\}}, T_0, T_1)$ is a parity decision tree that computes f correctly on at least $1 - \delta$ fraction of the inputs then either T_0 with every occurrence of x_i hard-wired to 0 or T_1 with every occurrence of x_i hard-wired to 1 computes f correctly on at least $1 - \delta - \text{Inf}_i(f)$ fraction of the inputs.*

Eliminating Parity Queries in PDTs

Let T be a parity decision tree that computes f correctly on at least $1 - \delta$ fraction of the inputs. Our idea is to convert the parity queries to an ordinary one and then eliminate the queries at the root of the tree. Let

$$Lf(x) := f(Lx).$$

We apply the linear transformation L on the input space \mathbb{F}_2^n and work with Lf instead of f .

Observation 4. $\text{Var}(f) = \text{Var}(Lf)$ and $D_{\oplus}(f) = D_{\oplus}(Lf)$.

Rotating the PDT T : Without loss of generality, let us assume that the first parity query in T is the parity of the first k bits, i.e., $x_1 \oplus \dots \oplus x_k$ (for some k). Let $g(x_1, \dots, x_n) := f(x_1 \oplus \dots \oplus x_k, x_2, \dots, x_n)$. Note that $g = Lf$ where L is the following invertible linear transformation on the vector space $\mathbb{F}_2^n : L(x_1, \dots, x_n) := (x_1 \oplus \dots \oplus x_k, x_2, \dots, x_n)$. Also note that: $f(x_1, \dots, x_n) = g(x_1 \oplus \dots \oplus x_k, x_2, \dots, x_n)$. Thus by querying $x_1 \oplus \dots \oplus x_k$, we know the value of the ‘first input bit’ of g . Moreover the influence of the first variable remains unchanged.

Observation 5. $\text{Inf}_1(g) = \text{Inf}_1(f)$.

Note however that the influences of the variables x_2, \dots, x_k might have changed!

A PDT $T = (x_{[k]}, T_0, T_1)$ for f can be easily modified to a PDT LT for $Lf = g$. We call the transformation from T to LT as the *rotation* of T and it is defined as follows:

$$L(x_S, T_0, T_1) := (L(x_S), L(T_0), L(T_1)),$$

$$\text{(base case) } L(0, \emptyset, \emptyset) = (0, \emptyset, \emptyset),$$

$$\text{(base case) } L(1, \emptyset, \emptyset) = (1, \emptyset, \emptyset).$$

Next we observe that the error is preserved by a rotation.

Observation 6. *If T computes f correctly on $1 - \delta$ fraction of the inputs then LT computes $g = Lf$ correctly on $1 - \delta$ fraction of the inputs.*

Moreover: the tree LT has a nice property that the query at the root is not an arbitrary parity query but in fact an ordinary query, i.e., a variable x_1 . Hence we can use Lemma 2 to remove the first query at the expense of $\text{Inf}_1(g) = \text{Inf}_1(f)$ increase in the error. Thus we conclude that:

Proposition 1. *If T computes f with error δ then either LT_0 or LT_1 computes LF correctly on at least $1 - \delta - \text{Inf}_{\max}(f)$ fraction of inputs.*

Rotating the PDT LT_i back to T_i :

Observation 7. For the particular L above, $L^{-1} = L$.

Suppose that LT_i computes Lf correctly on at least $1 - \delta - \text{Inf}_{\max}(f)$ fraction of the inputs.

Thus we can rewrite Observation 6 as follows:

Observation 8. If LT computes Lf correctly on $1 - \delta$ fraction of the inputs then $L(LT)$ computes $f = L(Lf)$ correctly on $1 - \delta$ fraction of the inputs.

Proof of Theorem 1. Since $L(LT_i) = T_i$ and since LT_i computes Lf correctly on at least $1 - \delta - \text{Inf}_{\max}(f)$ fraction of the inputs, T_i computes f with the same error. Notice that T_i makes one less parity query than T . So we have eliminated one parity query with an increase in error at most $\text{Inf}_{\max}(f)$. Now we can repeat this process starting from a parity tree T of depth $D_\epsilon^\oplus(f)$ that makes error on at most ϵ fraction of the inputs to obtain a zero-query parity decision tree that makes at most $\epsilon + D_\epsilon^\oplus(f) \cdot \text{Inf}_{\max}(f)$ error. The error of any zero-query parity decision tree must be at least $\text{Var}(f)$. This completes the proof of Theorem 1. \square

Remark 1. OR and AND functions on n variables can be computed with error probability at most $1/n$ on every input, using $O(\log n)$ parity queries chosen uniformly at random. Thus our Theorem 1 can be extended (up to a multiplicative poly-logarithmic factor) to the decision trees that use AND, OR, and PARITY queries. More generally, one can extend it to so called 1+ queries (see [10]) involving parities of (say polynomially many) arbitrary subsets.

4 Every Linear Decision Tree Has an Influential Variable

Let q be a prime power and \mathbb{F}_q be the finite field with q elements. In this section we consider computing functions from \mathbb{F}_q^n to $\{0, 1\}$ with the model called linear decision trees, denoted by \oplus_q -DT. It is a computation tree, with each internal node v labeled by a linear form $\ell : \mathbb{F}_q^n \rightarrow \mathbb{F}_q$. v has q children, whose edges connecting to v are labeled by elements from \mathbb{F}_q . The branching at node v is based on the evaluation of ℓ on the input vector. It is clear that when $q = 2$, this model becomes the parity decision tree model for computing boolean functions. We use $D_\epsilon^{\oplus q}(f)$ to denote the smallest \oplus_q -DT for computing $f : \mathbb{F}_q^n \rightarrow \{0, 1\}$ with error ϵ .

We will focus on the setting of uniform distribution over \mathbb{F}_q^n . For $f : \mathbb{F}_q^n \rightarrow \{0, 1\}$, its variance is defined the same as $\text{Var}(f) = 4 \cdot \Pr(f(x) = 0) \Pr(f(x) = 1)$. If x and y in \mathbb{F}_q^n differ only at the k th position, $k \in [n]$, we denote this by $x \sim_k y$. The influence of the k^{th} variable is $\text{Inf}_k(f) := \Pr_{x \sim_k y}(f(x) \neq f(y))$. Our main result is the following analogue of Theorem 1.

Theorem 2, restated. For any function $f : \mathbb{F}_q^n \rightarrow \{0, 1\}$ and any $\epsilon \geq 0$:

$$\text{Inf}_{\max}(f) \geq \frac{1}{q-1} \cdot \frac{\text{Var}(f) - \epsilon}{D_\epsilon^{\oplus q}(f)}.$$

We now prove Theorem 2. We shall adapt the proof of the query elimination lemma to \oplus_q -DT as follows.

Suppose T is a \oplus_q -DT for $f : \mathbb{F}_q^n \rightarrow \{0, 1\}$. Let $\ell : \mathbb{F}_q^n \rightarrow \{0, 1\}$ be the first query made by T , and $\ell(x_1, \dots, x_n) = \alpha_1 x_1 + \alpha_2 x_2 + \dots + \alpha_n x_n$. As ℓ is not trivial, there exists some $k \in [n]$ s.t. $\alpha_k \neq 0$. Fix such a $k \in [n]$. For $i \in \mathbb{F}_q$, let T_i be the \oplus_q -DT to be executed when $\ell(x) = i$.

For every T_i , $i \in \mathbb{F}_q$, construct a new \oplus_q -DT T'_i , by replacing every occurrence of x_k in T_i with

$$\frac{1}{\alpha_k} (i - (\alpha_1 x_1 + \dots + \alpha_{k-1} x_{k-1} + \alpha_{k+1} x_{k+1} + \dots + \alpha_n x_n)).$$

It is clear that T'_i and T_i are related as follows. Let $a = (a_1, \dots, a_n) \in \mathbb{F}_q^n$. Then $T'_i(a_1, \dots, a_n) = T_i(a_1, \dots, a_{k-1}, b_k, a_{k+1}, \dots, a_n)$, where $b_k \in \mathbb{F}_q$ s.t.

$$\ell(a_1, \dots, a_{k-1}, b_k, a_{k+1}, \dots, a_n) = i.$$

For $a = (a_1, \dots, a_n) \in \mathbb{F}_q^n$, we use $a|_k^{\ell, i}$ to denote $(a_1, \dots, a_{k-1}, b_k, a_{k+1}, \dots, a_n) \in \mathbb{F}_q^n$ satisfying the above. Then we have $T'_i(a) = T_i(a|_k^{\ell, i})$.

As T computes f with error ϵ , there exists some $j \in \mathbb{F}_q$, s.t. when restricting to $\{a \in \mathbb{F}_q^n \mid \ell(a) = j\}$, T_j computes f with error $\leq \epsilon$. Fix such T_j , and consider T'_j . We claim that T'_j computes f with error no more than $\epsilon + (q-1) \text{Inf}_k(f)$.

To see this, for $i \in \mathbb{F}_q$, $i \neq j$, define

$$A|_k^{\ell, j}(f, i) = \Pr_{a \in \mathbb{F}_q^n, \ell(a)=i} (f(a) \neq f(a|_k^{\ell, j})).$$

It is obvious that T'_j computes f with error $\leq \epsilon + 1/q \cdot (\sum_{i \in \mathbb{F}_q, i \neq j} A|_k^{\ell, j}(f, i))$. Now we verify that $1/q \cdot (\sum_{i \in \mathbb{F}_q, i \neq j} A|_k^{\ell, j}(f, i)) \leq (q-1) \text{Inf}_k(f)$. Fix $a = (a_1, \dots, a_n)$ from $\{a \in \mathbb{F}_q^n \mid \ell(a) = j\}$. Then the contribution of $(a_1, \dots, a_{k-1}, a_{k+1}, \dots, a_n)$ in $1/q \cdot (\sum_{i \in \mathbb{F}_q, i \neq j} A|_k^{\ell, j}(f, i))$ is $\frac{1}{q} \cdot \frac{1}{q^n - 1} \cdot s$, where $s \in \{0, \dots, q-1\}$ is the number of field elements b s.t. $f(a_1, \dots, a_{k-1}, b, a_{k+1}, \dots, a_n) \neq f(a_1, \dots, a_n)$. On the other hand, its contribution in $(q-1) \cdot \text{Inf}_k(f)$ is $(q-1) \cdot \frac{1}{q^n - 1} \cdot \frac{s \binom{q-s}{2}}{\binom{q}{2}}$. Finally note that $\frac{s}{(q-1)q} \leq \frac{s \binom{q-s}{2}}{\binom{q}{2}}$ for $q \geq 2$ and $s \in \{0, \dots, q-1\}$.

As eliminating the first query introduces an extra error of at most $(q-1) \text{Inf}_{\max}(f)$, similar to the argument in proving Theorem 1, we have $\epsilon + (q-1) D^{\oplus_q}(f) \cdot \text{Inf}_{\max}(f) \geq \text{Var}(f)$, therefore proving that

$$\text{Inf}_{\max}(f) \geq \frac{1}{q-1} \cdot \frac{\text{Var}(f) - \epsilon}{D^{\oplus_q}(f)}.$$

Acknowledgements. We thank Rahul Jain, Supartha Poddar, Miklos Santha, and Avishay Tal for several helpful discussions. We also thank Ben vee Volk for pointing out that the super-linear separation in [27] works for PDTs as well.

References

1. Aaronson, S., Ambainis, A.: The need for structure in quantum speedups. In: ICS 2011, pp. 338–352 (2011)
2. Ada, A., Fawzi, O., Hatami, H.: Spectral norm of symmetric functions. In: Gupta, A., Jansen, K., Rolim, J., Servedio, R. (eds.) APPROX 2012 and RANDOM 2012. LNCS, vol. 7408, pp. 338–349. Springer, Heidelberg (2012)
3. Babai, L., Banerjee, A., Kulkarni, R., Naik, V.: Evasiveness and the distribution of prime numbers. In: STACS 2010, pp. 71–82 (2010)
4. Bhruhundi, A., Chakraborty, S., Kulkarni, R.: Property testing bounds for linear and quadratic functions via parity decision trees. In: Hirsch, E.A., Kuznetsov, S.O., Pin, J.É., Vereshchagin, N.K. (eds.) CSR 2014. LNCS, vol. 8476, pp. 97–110. Springer, Heidelberg (2014). Electronic colloquium on Computational Complexity (ECCC)
5. Benjamini, I., Kalai, G., Schramm, O.: Noise sensitivity of boolean functions and its application to percolation. *Inst. Hautes Etudes Sci. Publ. Math.* **90**, 5–43 (1999)
6. Ben-Or, M., Linial, N.: Collective coin flipping. In: Proceedings of the 26th FOCS, pp. 408–416 (1985)
7. Bollobas, B.: *Combinatorics: Set Systems, Hypergraphs, Families Of Vectors And Combinatorial Probability*. Cambridge University Press, New York (1986)
8. Buhrman, H., de Wolf, R.: Complexity measures and decision tree complexity: a survey. *Theor. Comput. Sci.* **288**(1), 21–43 (2002)
9. Efron, B., Stein, C.: The jackknife estimate of variance. *Ann. Stat.* **9**, 586–596 (1981)
10. Gopalan, P., O’Donnell, R., Servedio, R.A., Shpilka, A., Wimmer, K.: Testing fourier dimensionality and sparsity. In: Albers, S., Marchetti-Spaccamela, A., Matias, Y., Nikolettseas, S., Thomas, W. (eds.) ICALP 2009, Part I. LNCS, vol. 5555, pp. 500–512. Springer, Heidelberg (2009)
11. Hayes, T.P., Kutin, S., van Melkebeek, D.: The quantum black-box complexity of majority. *algorithmica* **34**(4), 480–501 (2002)
12. Hatami, P., Kulkarni, R., Pankratov, D.: Variations on the sensitivity conjecture. *Theor. Comput. Grad. Surv.* **2**, 1–27 (2011)
13. Jain, R., Zhang, S.: The influence lower bound via query elimination. *Theor. Comput.* **7**(1), 147–153 (2011)
14. Kulkarni, R.: Evasiveness through a circuit lens. In: ITCS 2013 pp. 139–144 (2013)
15. Kulkarni, R.: Gems in decision tree complexity revisited. *SIGACT News* **44**(3), 42–55 (2013)
16. Kahn, J., Kalai, G., Linial, N.: The influence of variables on boolean functions (extended abstract). In: FOCS 1988, pp. 68–80 (1988)
17. Kushilevitz, E., Mansour, Y.: Learning decision trees using the fourier spectrum. *SIAM J. Comput.* **22**(6), 1331–1348 (1993)
18. Kulkarni, R., Santha, M.: Query complexity of matroids. In: Spirakis, P.G., Serna, M. (eds.) CIAC 2013. LNCS, vol. 7878, pp. 300–311. Springer, Heidelberg (2013)
19. Kahn, J., Saks, M.E., Sturtevant, D.: A topological approach to evasiveness. *Combinatorica* **4**(4), 297–306 (1984)
20. Lee, H.K.: Decision trees and influence: an inductive proof of the OSSS inequality. *Theor. Comput.* **6**(1), 81–84 (2010)
21. Linial, N., Mansour, Y., Nisan, N.: Constant depth circuits, fourier transform, and learnability. *J. ACM* **40**(3), 607–620 (1993)

22. Lovasz, L., Young, N. E.: Lecture Notes on Evasiveness of Graph Properties arXiv:cs/020503 (2002)
23. Montanaro, A., Osborne, T.: On the communication complexity of XOR functions. CoRR abs/0909.3392 (2009)
24. Mehlhorn, K., Schmidt, E.: Las Vegas is better than determinism in VLSI and distributed computing. In: Proceedings of the 14th STOC, pp. 330–337. ACM Press, New York (1982)
25. Nisan, N.: CREW PRAMs and decision trees. In: Proceedings of the 21st STOC, pp. 327–335. ACM Press, New York (1989)
26. Nisan, N., Szegedy, M.: On the degree of boolean functions as real polynomials. *Comput. Complex.* **4**, 301–313 (1994)
27. Nisan, N., Wigderson, A.: On rank vs. communication complexity. *Combinatorica* **15**(4), 557–565 (1995)
28. O’Donnell, R., Servedio, R.A.: Learning monotone decision trees in polynomial time. *SIAM J. Comput.* **37**(3), 827–844 (2007)
29. O’Donnell, R., Saks, M.E., Schramm, O., Servedio, R.A.: Every decision tree has an influential variable. In: FOCS, pp. 31–39 (2005)
30. Sherstov, A.A.: Making polynomials robust to noise. In: STOC 2012, pp. 747–758 (2012)
31. Shi, Y., Zhang, Z.: Communication Complexities of XOR functions CoRR abs/0808.1762 (2008)
32. Shpilka, A., Tal, A., Volk, B.L.: On the Structure of Boolean Functions with Small Spectral Norm: [arXiv:1304.0371](https://arxiv.org/abs/1304.0371)
33. Saks, M.E., Wigderson, A.: Probabilistic boolean decision trees and the complexity of evaluating game trees. In: FOCS, pp. 29–38 (1986)
34. Zhang, Z., Shi, Y.: On the parity complexity measures of boolean functions. *Theor. Comput. Sci.* **411**(26–28), 2612–2618 (2010)
35. Talagrand, M.: On russo’s approximate 0-1 law. *Ann. Probab.* **22**(3), 1576–1587 (1994)
36. Tsang, H.Y., Wong, C.H., Xie, N., Zhang, S.: Fourier sparsity, spectral norm, and the Log-rank conjecture. CoRR abs/1304.1245 (2013) FOCS (2014)