1-1-1998

Average-Case Lower Bounds for Noisy Boolean Decision Trees

William Evans
Arizona University

Nicholas Pippenger
Harvey Mudd College

SIAM J. COMPUT.
Vol. 28, No. 2, pp. 433–446

AVERAGE-CASE LOWER BOUNDS FOR NOISY BOOLEAN DECISION TREES*

WILLIAM EVANS† AND NICHOLAS PIPPENGER‡

Abstract. We present a new method for deriving lower bounds to the expected number of queries made by noisy decision trees computing Boolean functions. The new method has the feature that expectations are taken with respect to a uniformly distributed random input, as well as with respect to the random noise, thus yielding stronger lower bounds. It also applies to many more functions than do previous results. The method yields a simple proof of the result (previously established by Reischuk and Schmeltz) that almost all Boolean functions of n arguments require $\Omega(n \log n)$ queries, and strengthens this bound from the worst-case over inputs to the average over inputs. The method also yields bounds for specific Boolean functions in terms of their spectra (their Fourier transforms). The simplest instance of this spectral bound yields the result (previously established by Feige, Peleg, Raghavan, and Upfal) that the parity function of n arguments requires $\Omega(n \log n)$ queries and again strengthens this bound from the worst-case over inputs to the average over inputs. In its full generality, the spectral bound applies to the "highly resilient" functions introduced by Chor, Friedman, Goldreich, Hastad, Rudich, and Smolensky, and it yields nonlinear lower bounds whenever the resiliency is asymptotic to the number of arguments.

Key words. fault-tolerance, reliability, noisy computation, error-correction

AMS subject classifications. 68M15, 68P10, 68R05

PII. S0097539796310102

1. Introduction.

We shall deal in this paper with dynamic decision trees for computing Boolean functions. A *dynamic decision tree* is a binary tree in which each internal node N is labelled with an argument index $\alpha(N) \in \{1, \ldots, n\}$, each child M of an internal node N is labelled with a Boolean value $\beta(M) \in \{0, 1\}$ that might be assumed by this argument (with siblings being labelled with distinct values), and each leaf L is labelled with a Boolean function value $\phi(L) \in \{0, 1\}$. Such a dynamic decision tree computes a Boolean function f of n Boolean arguments $x_1, \ldots, x_n$ in an obvious way: start at the root; when at an internal node N, query the argument $x_{\alpha(N)}$ and proceed to the child M of N such that $\beta(M) = x_{\alpha(N)}$; when at a leaf L, announce the function value $f(x_1, \ldots, x_n) = \phi(L)$. For such a dynamic decision tree, we may speak of the *worst-case cost* (the maximum over argument values of the depth of the leaf that announces the function value) or the *average-case cost* (the average with a uniform distribution over argument values of the depth of the leaf that announces the function value).

We shall be interested in the situation in which dynamic decision trees are noisy, that is, in which each internal node independently passes control to the incorrect child (that is, the child M of the internal node N such that $\beta(M) = \neg x_{\alpha(N)}$) with some fixed probability $0 < \varepsilon < 1/2$. We shall say that such a tree (ε, δ)-computes a

*Received by the editors October 2, 1996; accepted for publication (in revised form) April 10, 1997; published electronically July 7, 1998. A preliminary version of this paper appeared in *Proc. 28th Annual ACM Symposium on Theory of Computing*, ACM, New York, 1996, pp. 620–628.
http://www.siam.org/journals/sicomp/28-2/31010.html

†Department of Computer Science, The University of Arizona, Tucson, AZ 85721-0077 (will@cs.arizona.edu). The work of this author was supported by an NSERC Canada International Fellowship.

‡Department of Computer Science, The University of British Columbia, Vancouver, BC V6T 1Z4 Canada (nicholas@cs.ubc.ca). The work of this author was supported by an NSERC Operating Grant.

Boolean function f if, for all $x_1, \ldots, x_n \in \{0,1\}$, the probability that control reaches an incorrectly labelled leaf (that is, a leaf L labelled $\phi(L) = \neg f(x_1, \ldots, x_n)$) is at most $\delta < 1/2$. For such a noisy dynamic decision tree, we may again speak of the worst-case or average-case cost (where we may maximize or average over argument values but always average over noise).

An alternative to the error model we have adopted is to assume that errors occur with probability at most ε, rather than exactly ε. This alternative model gives stronger upper bounds. Our interest in this paper is in lower bounds, for which the model we have adopted gives stronger results.

To describe the history of our results, we shall need to refer to two additional computational models. The first of these is the *static decision tree*, which we may regard as a dynamic decision tree in which the argument queried by an internal node does not depend on the outcomes of previous queries (and thus depends only on the depth of the node in the tree), and in which all leaves appear at the same depth. The cost in this case is simply the common depth C of the leaves. It is not hard to see that we may ignore the tree structure, and simply focus on the number of queries C_i to each argument x_i. We then have $C_1 + \cdots + C_n = C$. Furthermore, we may ignore the sequence of answers to the queries to a given argument and focus on the number D_i of affirmative answers among answers to the C_i queries to x_i. We then have $0 \leq D_i \leq C_i$ for $1 \leq i \leq n$. While a noisy static decision tree might announce distinct function values for the same values of $D_1, \ldots, D_n$, it is not hard to see that these announcements can be replaced by a consistent announcement $\phi(D_1, \ldots, D_n)$, without increasing the probability of an incorrect announcement in any situation. Thus we may describe a static decision tree by specifying the numbers $C_1, \ldots, C_n$ and the labelling $\phi(D_1, \ldots, D_n)$ for $0 \leq D_1 \leq C_1, \ldots, 0 \leq D_n \leq C_n$.

Our final computational model is the *circuit with noisy gates*. We shall not describe this model in detail but merely remark that a lower bound to static decision tree cost yields a lower bound to the size (number of gates) of a circuit with noisy gates.

Work on reliable computation in the presence of noise was begun by von Neumann [14], who argued (although he did not give a rigorous proof) that a computation that can be performed by a noiseless network with L gates could be reliably performed by a noisy network with $O(L \log L)$ gates. Dobrushin and Ortyukov [4] provided a rigorous proof of this result, and [3] claimed the following matching lower bound: a noisy network that reliably computes a function f must have $\Omega(S \log S)$ gates, where S is the *sensitivity* of f (the maximum over inputs $x_1, \ldots, x_n$ of the number of indices i such that $f(x_1, \ldots, x_{i-1}, \neg x_i, x_{i+1}, \ldots, x_n) \neq f(x_1, \ldots, x_n)$). Since there are many functions (for example, the disjunction, conjunction, or parity of n arguments) that have sensitivity $S = n$ and can be computed by noiseless networks with $O(n)$ gates, this result shows that the logarithmic ratio of noisy to noiseless gates is necessary for certain functions.

There are, however, several errors in the proof of the lower bound of Dobrushin and Ortyukov [3]. These were pointed out by Pippenger, Stamoulis, and Tsitsiklis [16], who gave a proof of the weaker result that a noisy network that reliably computes the parity function of n arguments must have $\Omega(n \log n)$ gates. The full strength of the lower bound in terms of sensitivity was regained by Gál [9] (see also Gács and Gál [10]) and by Reischuk and Schmeltz [17]. An important consequence of this stronger result is that a noisy network that reliably computes the disjunction (or conjunction) of n arguments must have $\Omega(n \log n)$ gates. All of these lower bound

arguments apply to static decision trees as well as to circuits. For noisy static decision trees, lower bounds of $\Omega(n \log n)$ are best possible, since any Boolean function of n arguments can be computed by a noisy static decision tree with $O(n \log n)$ queries (with $2\log(n/\delta) \,/\, \log\bigl(1/4\varepsilon(1-\varepsilon)\bigr) = O(\log n)$ queries, it is possible to determine a single argument with error probability at most δ/n).

Noisy dynamic decision trees were considered by Feige et al. [6, 7], who showed that there are noisy dynamic decision trees that reliably compute the disjunction or conjunction of n arguments with $O(n)$ queries. Since we have seen that noisy static decision trees require $\Omega(n \log n)$ queries, this exhibits a clear separation between the two models. For noisy dynamic decision trees, Feige et al. [6, 7] showed that $\Omega(n \log n)$ queries are needed to compute the parity or majority of n arguments, and Reischuk and Schmeltz [17] showed that $\Omega(n \log n)$ queries are needed for almost all Boolean functions of n arguments. (This last result contrasts with results of Muller [13] and Pippenger [15] for circuits, to the effect that for almost all Boolean functions of n arguments, $\Omega(2^n/n)$ noiseless gates are necessary and $O(2^n/n)$ noisy gates are sufficient.) The lower bound proofs of both Feige et al. and of Reischuk and Schmeltz depend on locating particular sets of inputs that are difficult for a dynamic decision tree, and thus they yield lower bounds for the worst-case over inputs but not for the average-case over inputs (and clearly no proof that applied to disjunction or conjunction could give a nontrivial lower bound for the average over inputs).

The present paper gives a new method of establishing lower bounds for noisy dynamic decision trees. The gist of the method is to argue that for certain Boolean functions there cannot be even one leaf in the decision tree that has both a small depth and a small probability of error (conditional on control reaching the leaf). The Boolean functions to which the method applies are difficult to compute for all inputs rather than just for certain inputs. This implies that lower bounds established by the method apply to the average case over inputs rather than just the worst case. (It also implies of course that the method is powerless to deal with functions such as disjunction, conjunction, and majority that have inputs such as $x_1 = \cdots x_n = 1$, $x_1 = \cdots x_n = 0$, or both for which it is easy to reliably determine the function value.) These strengths and weaknesses of our new method are embodied in a new complexity measure for Boolean functions, which we call "noisy leaf complexity." In section 2 we shall define noisy leaf complexity and relate it to noisy dynamic decision tree complexity described above.

Our method considers the situation in which control has arrived at a leaf L. Arrival at L conditions the uniform prior distribution on the input x to a posterior distribution. Our method is based on the fact that, if the depth of L is small, this posterior distribution must be spread over a large range of possible input values. In section 3 we shall calculate this posterior distribution and derive quantitative versions of the assertion that it is spread over a large range.

Section 4 deals with random Boolean functions and establishes a lower bound of the form $\Omega(n \log n)$ for the noisy leaf complexity of "almost all" Boolean functions of n arguments. Specifically, we show that if L is a leaf of cost

$$C \le n \log_E(n/2) - n \log_E \log\bigl(2n^2/(1-2\delta)^2\bigr),$$

where $E = (1-\varepsilon)/\varepsilon$, then the probability is at most $2e^{-n^2}$ that L has error probability (conditional on arrival at L) at most δ for a random Boolean function of n arguments. This strengthens (from the worst-case over inputs to the average-case over inputs) the lower bound of Reischuk and Schmeltz [17].

Section 5 establishes a lower bound of the form $\Omega(n \log n)$ for the noisy leaf complexity of the parity function of n arguments. Specifically, we show that if a leaf with cost C has conditional error probability at most δ for the parity function of n arguments, then

$$C \geq n \log_E n - n \log_E \log\bigl(1/(1-2\delta)\bigr),$$

where $E = (1-\varepsilon)/\varepsilon$. This strengthens (from the worst-case over inputs to the average-case over inputs) the lower bound of Feige et al. [6, 7] for the parity function. The proof of our lower bound uses the Fourier transform of the parity function, which has a particularly simple form. Other examples of the use of the Fourier transform to derive lower bounds to the computational complexity of Boolean functions are given by Brandman, Orlitsky, and Hennessy [1] (noiseless decision trees) and by Linial, Mansour, and Nisan [12] (bounded-depth circuits). It would be possible to rephrase this proof so as not to refer to the Fourier transform. Indeed, Fourier analysis on finite groups such as the Boolean n-cube is tantamount to linear algebra in finite-dimensional vector spaces. Fourier analysis lends this linear algebra a certain suggestive terminology, however, that provides a vivid intuition to guide the manipulations. This intuition was valuable in discovering the more general results of section 6.

A general class of Boolean functions to which our method applies is the class of "highly resilient" functions. If a Boolean function is significantly "biased" (that is, if it assumes the values 0 and 1 with significantly unequal probabilities under the uniform input distribution), then even a leaf at depth 0 can announce the function value with a probability of output error significantly less than $1/2$. This suggests we focus our attention on "unbiased" functions, which assume the values 0 and 1 each with probability $1/2$. Extending this reasoning, we see that if a Boolean function can be significantly biased by substituting constants for a small number of arguments, then a leaf with small depth can achieve a probability of output error significantly less than $1/2$. This suggests we focus our attention on functions that are unbiased and which remain unbiased even when constants are substituted for some number t of arguments. Such functions are called "t-resilient" by Chor et al. [2]. Though defined combinatorially, the highly resilient functions have natural characterizations in terms of their "spectra," either in the sense of their Fourier transforms or in the sense of the eigenvalues of the adjacency matrix of the Boolean hypercube. These characterizations are discussed by Friedman [8].

Section 6 establishes a lower bound for the noisy leaf complexity of t-resilient Boolean functions. Specifically, we show that if f is t-resilient and a leaf with cost C has conditional error probability at most δ for f, then

$$C \geq (t+1) \log_E \frac{t+1}{\frac{n}{2} H\left(\frac{t+1}{n}\right) + \log \frac{1}{1-2\delta}},$$

where $E = (1-\varepsilon)/\varepsilon$, and $H(\eta) = -\eta \log \eta - (1-\eta)\log(1-\eta)$ for $0 < \eta < 1$, extended by continuity to $H(0) = H(1) = 0$. The most resilient function of n arguments is the parity function, which is $(n-1)$-resilient. Thus we recover the lower bound of section 5 in this special case. There are, however, many highly resilient functions that are not parity functions. For these functions, our method yields a nonlinear lower bound whenever $t \sim n$, that is, whenever the resiliency is asymptotic to the number of arguments.

2. Noisy leaf complexity. Let f be a Boolean function of n arguments $x_1, \ldots, x_n$. Let T be a decision tree and let L be a leaf of T. By the *cost* of L we shall mean the number of queries along the path from the root of T to L. Suppose now that the input x is chosen at random with the uniform distribution (with each possible input having probability 2^{-n}). Suppose further that the tree T is applied to the input x with query error probability $\varepsilon > 0$ at each internal node. We shall say that L is (ε, δ)-*good* for f if the probability $\Pr(\phi(L) = \neg f(x) \mid L)$ of output error at L, conditional on control reaching L, is at most $\delta < 1/2$. It is clear that whether or not a leaf L is (ε, δ)-good for f depends only on the numbers $C_1, \ldots, C_n$ of queries to the arguments $x_1, \ldots, x_n$, and on the numbers $D_1, \ldots, D_n$ of affirmative responses to these queries, and not on the rest of T. By the (ε, δ)-*leaf complexity* of a Boolean function f, we shall mean the smallest possible cost of a leaf that is (ε, δ)-good for f.

PROPOSITION 2.1. *Suppose that the noisy dynamic decision tree T (ε, δ)-computes the Boolean function f with expected cost C averaged over both inputs and noise. Let δ' be such that $\delta < \delta' < 1/2$. Then f has (ε, δ')-leaf complexity at most $C' = C/(1 - \delta/\delta')$.*

Proof. Let the input x be chosen with the uniform distribution. For each leaf L in T, let $p_L = \Pr(L)$ denote the probability that control reaches L, let $\delta_L = \Pr(\phi(L) = \neg f(x) \mid L)$ denote the probability of error conditional on control reaching L, and let C_L denote the cost of L. Let A denote the set of leaves L such that $\delta_L > \delta'$. If A is nonempty we have

$$\delta' \sum_{L \in A} p_L < \sum_{L \in A} p_L \delta_L \leq \sum_{L} p_L \delta_L = \delta,$$

and if A is empty we have

$$\delta' \sum_{L \in A} p_L = 0 < \delta,$$

so in any case we have

$$\sum_{L \in A} p_L < \delta/\delta'.$$

Let B denote the set of leaves L such that $C_L > C'$. If B is nonempty we have

$$C' \sum_{L \in B} p_L < \sum_{L \in B} p_L C_L \leq \sum_{L} p_L C_L = C,$$

and if B is empty we have

$$C' \sum_{L \in B} p_L = 0 < C,$$

so in any case we have

$$\sum_{L \in B} p_L < C/C'.$$

These inequalities yield

$$\sum_{L \notin A \cup B} p_L > 1 - \delta/\delta' - C/C' = 0.$$

Thus with positive probability control arrives at a leaf L such that $\delta_L \leq \delta'$ and $C_L \leq C'$, which shows that the (ε, δ')-leaf complexity of f is at most C'. □

3. The posterior distribution. Suppose that we choose an input x at random with a uniform distribution: $\Pr(x) = 2^{-n}$. Then suppose that we apply a noisy dynamic decision tree T with query error probability $\varepsilon > 0$ and arrive at a leaf L. We shall calculate the posterior probability distribution on x, given arrival at L: $\Pr(x \mid L)$.

Suppose that along the path from the root of T to L the input x_i is queried C_i times, with D_i affirmative responses (and thus $C_i - D_i$ negative responses). The event of arrival at L is the conjunction of n events $L_1, \ldots, L_n$, where L_i specifies a particular sequence of responses of the C_i queries to x_i. The prior distribution of x_i is $\Pr_i(x_i) = 1/2$. The conditional probability $\Pr_i(L_i \mid x_i)$ of L_i given x_i is

$$\Pr_i(L_i \mid 0) = \varepsilon^{D_i}(1-\varepsilon)^{C_i - D_i},$$
$$\Pr_i(L_i \mid 1) = \varepsilon^{C_i - D_i}(1-\varepsilon)^{D_i},$$

and thus

$$\Pr(L_i) = \frac{\varepsilon^{D_i}(1-\varepsilon)^{C_i - D_i} + \varepsilon^{C_i - D_i}(1-\varepsilon)^{D_i}}{2}.$$

Thus the posterior distribution $\Pr_i(x_i \mid L_i)$ of x_i, conditioned on L_i, is

$$\Pr_i(0 \mid L_i) = \frac{\varepsilon^{D_i}(1-\varepsilon)^{C_i - D_i}}{\varepsilon^{D_i}(1-\varepsilon)^{C_i - D_i} + \varepsilon^{C_i - D_i}(1-\varepsilon)^{D_i}}, \tag{3.1}$$

$$\Pr_i(1 \mid L_i) = \frac{\varepsilon^{C_i - D_i}(1-\varepsilon)^{D_i}}{\varepsilon^{D_i}(1-\varepsilon)^{C_i - D_i} + \varepsilon^{C_i - D_i}(1-\varepsilon)^{D_i}}. \tag{3.2}$$

Finally, since the x_i and the responses to the queries given the x_i are all independent, we have

$$\Pr(x \mid L) = \prod_{1 \le i \le n} \Pr_i(x_i \mid L_i). \tag{3.3}$$

Formulas (3.1), (3.2), and (3.3) give the desired posterior distribution of x.

It will be convenient to have bounds for $\Pr_i(x_i \mid L_i)$ that are independent of D_i. If we divide the numerator and denominator of (3.1) by the numerator, we obtain

$$\Pr_i(0 \mid L_i) = \frac{1}{1 + E^{2D_i - C_i}},$$

where $E = (1-\varepsilon)/\varepsilon$ (and $E > 1$, since $\varepsilon < 1/2$). The right-hand side is maximized when $D_i = 0$, so we have

$$\Pr_i(0 \mid L_i) \le \frac{1}{1 + E^{-C_i}}.$$

Similar reasoning from (3.2) yields an expression that is maximized when $D_i = C_i$, resulting in the same bound for $\Pr_i(1 \mid L_i)$. Thus if we set $P_i = \max\{\Pr_i(0 \mid L_i), \Pr_i(1 \mid L_i)\}$, we have

$$\begin{aligned} P_i &\le \frac{1}{1 + E^{-C_i}} \\ &= \frac{E^{C_i}}{E^{C_i} + 1} \\ &= 1 - \frac{1}{E^{C_i} + 1} \\ &\le 1 - \frac{1}{2E^{C_i}}. \end{aligned} \tag{3.4}$$

This is the desired bound.

4. Random Boolean functions. Throughout this section, f will denote a random Boolean function of n arguments, for which $f(x)$ is equally likely to be 0 or 1, independently for each value of x. Our strategy will be to consider a leaf L of small depth and bound the probability that L is (ε,δ)-good for f. Our main result is the following.

THEOREM 4.1. *Let L be a leaf of cost*

$$C \le n\log_E(n/2) - n\log_E\log\bigl(2n^2/(1-2\delta)^2\bigr),$$

where $E = (1-\varepsilon)/\varepsilon$. Then L is (ε,δ)-good for a random Boolean function of n arguments with probability at most $2e^{-n^2}$.

This result easily yields a lower bound for the noisy leaf complexity of almost all Boolean functions.

COROLLARY 4.2. *For all sufficiently large n (depending on $E = (1-\varepsilon)/\varepsilon > 1$ and $\delta < 1/2$), the fraction of all Boolean functions of n arguments having (ε,δ)-leaf complexity at most $(n/2)\log_E(n/2)$ is at most $2e^{-n^2/2}$.*

Proof. For all sufficiently large n, we have

$$C = (n/2)\log_E(n/2) \le n\log_E(n/2) - n\log_E\log\bigl(2n^2/(1-2\delta)^2\bigr),$$

so we may apply Theorem 4.1 to any leaf of cost at most C. But such a leaf is determined by specifying (1) which of the n arguments is queried at each of the C queries and (2) the response (affirmative or negative) to each query. Thus there are at most $(2n)^C$ leaves, and thus the probability that some leaf is (ε,δ)-good for f is at most $2e^{-n^2}(2n)^{(n/2)\log_E(n/2)}$. For sufficiently large n, this bound is at most $2e^{-n^2/2}$. □

It will be convenient to work not only with the Boolean function f but also with the rescaled real-valued function $F(x) = 1 - 2f(x)$, which is equally likely to be $+1$ or -1, independently for each value of x. Similarly, it will be convenient to work not only with the probability of error δ_L associated with a leaf L but also with the *correlation* $\xi_L = 1 - 2\delta_L$ between the rescaled label $\Phi(L) = 1 - 2\phi(L)$ of L and the rescaled function $F(x)$. If $\delta_L \le \delta < 1/2$, then $\xi_L \ge 1-2\delta > 0$. Thus if L is (ε,δ)-good for f we have

$$1 - 2\delta \le \xi_L = \mathrm{Ex}_x\bigl(\Phi(L)F(x)\bigr) = \Phi(L)\sum_x \Pr(x \mid L)F(x).$$

Since $\Phi(L) = \pm 1$, this implies

$$1 - 2\delta \le \left|\sum_x \Pr(x \mid L)F(x)\right|. \tag{4.1}$$

The terms $\Pr(x \mid L)F(x)$ are independent random variables that assume the values $\pm\Pr(x \mid L)$ each with probability 1/2. Thus, to estimate the probability that (4.1) holds, it will suffice to use an estimate for the probability of large deviations for sums of independent, but not necessarily identically distributed, random variables. The following result of Hoeffding [11, Theorem 2] suits our purpose.

PROPOSITION 4.3 (see [11]). *If A_x are independent random variables with mean 0 and range $|A_x| \le \Delta_x$, then*

$$\Pr\left(\sum_x A_x \ge T\right) \le \exp(-T^2/2S),$$

where

$$S = \sum_x \Delta_x^2.$$

Since the random variables $\Pr(x \mid L)F(x)$ are distributed symmetrically about 0, the probability that (4.1) holds is just twice the probability that

$$1 - 2\delta \le \sum_x \Pr(x \mid L)F(x) \tag{4.2}$$

holds. We can bound this using Proposition 4.3 by taking $A_x = \Pr(x \mid L)F(x)$, so that $\Delta_x = \Pr(x \mid L)$, and $T = 1 - 2\delta$. Thus we seek an estimate for

$$S = \sum_x \Pr(x \mid L)^2.$$

We observe that by virtue of (3.3) we have

$$S = \sum_x \Pr(x \mid L)^2 = \prod_{1 \le i \le n} \left(\Pr_i(0 \mid L_i)^2 + \Pr_i(1 \mid L_i)^2\right).$$

Since

$$u^2 + (1-u)^2 = 1 - 2u(1-u) \le \max\{u, 1-u\},$$

we have

$$S \le \prod_{1 \le i \le n} P_i,$$

with $P_i = \max\{\Pr_i(0 \mid L_i), \Pr_i(1 \mid L_i)\}$ as defined in section 3. Using (3.4) we have

$$S \le \prod_{1 \le i \le n} \left(1 - \frac{1}{2E^{C_i}}\right).$$

Since $1 - u \le \exp(-u)$, we have

$$\begin{aligned} S &\le \prod_{1 \le i \le n} \exp\left(-\frac{1}{2}E^{-C_i}\right) \\ &= \exp\left(-\frac{1}{2}\sum_{1 \le i \le n} E^{-C_i}\right). \end{aligned}$$

Since $E^u = \exp_E u$ is a convex function of u, we have

$$\begin{aligned} S &\le \exp\left(-\frac{n}{2}\exp_E\left(-\frac{1}{n}\sum_{1 \le i \le n} C_i\right)\right) \\ &= \exp\left(-\exp_E\left(\log_E \frac{n}{2} - \frac{C}{n}\right)\right). \end{aligned}$$

Thus if

$$C \le n\log_E(n/2) - n\log_E\log\left(2n^2/(1-2\delta)^2\right),$$

we have $S \le (1-2\delta)^2/2n^2$. Proposition 4.3 then implies that (4.2) holds with probability at most e^{-n^2}, so (4.1) holds with probability at most $2e^{-n^2}$. This completes the proof of Theorem 4.1.

5. The parity function. In this section we shall derive a lower bound for the (ε, δ)-leaf complexity of the parity function:

$$f(x_1, \ldots, x_n) = x_1 + \cdots + x_n \pmod 2.$$

Our result is the following.

THEOREM 5.1. *If the leaf L with cost C is (ε, δ)-good for the parity function f of n arguments, then*

$$C \geq n \log_E n - n \log_E \log\bigl(1/(1-2\delta)\bigr),$$

where $E = (1-\varepsilon)/\varepsilon$.

The proof of this theorem depends on the notion of the Fourier transform of a Boolean function. This notion has already been applied to the computational complexity of Boolean functions by circuits (see Linial, Mansour, and Nisan [12]) and noiseless dynamic decision trees (see Brandman, Orlitsky, and Hennessy [1]), but the present paper appears to mark its debut for the complexity of noisy computation.

Let $F : \mathbf{B}^n \to \mathbf{R}$ be a real-valued function of n Boolean arguments. By the *Fourier transform* of F we shall mean the function $\hat{F} : \mathbf{B}^n \to \mathbf{R}$ defined by

$$\hat{F}(y) = \frac{1}{\sqrt{2^n}} \sum_x (-1)^{x \cdot y} F(x),$$

where $x \cdot y = \sum_{1 \leq j \leq n} x_j y_j$ denotes the inner product of x and y. (The factor $(-1)^{x \cdot y}$ is the specialization of the usual Fourier kernel $e^{2\pi i x \cdot y/m}$ to $m = 2$.) The normalization factor $1/\sqrt{2^n}$ has been chosen to make the transform an involution: we have

$$\begin{aligned}
\hat{\hat{F}}(z) &= \frac{1}{\sqrt{2^n}} \sum_y (-1)^{y \cdot z} \hat{F}(y) \\
&= \frac{1}{\sqrt{2^n}} \sum_y (-1)^{y \cdot z} \frac{1}{\sqrt{2^n}} \sum_x (-1)^{x \cdot y} F(x) \\
&= \frac{1}{2^n} \sum_x F(x) \sum_y (-1)^{x \cdot y + y \cdot z} \\
&= F(z),
\end{aligned}$$

since

$$\sum_y (-1)^{x \cdot y + y \cdot z} = \begin{cases} 2^n & \text{if } x = z, \\ 0 & \text{otherwise.} \end{cases}$$

(The general Fourier transform is not an involution, but rather has period four, and the effect of applying the transform twice is to reverse the function by negating its argument. But in $\mathbf{B}$, regarded as an additive group of order two, every element is its own negative, so each function is its own reversal.)

The key result we shall need is the Parseval identity

$$\sum_y \hat{F}(y)\, \hat{G}(y) = \sum_y F(y)\, G(y),$$

which says that the Fourier transform is an isometry of the Hilbert space $\mathbf{R}^{\mathbf{B}^n}$. This follows from a calculation similar to the one above:

$$\begin{aligned}\sum_y \hat{F}(y)\,\hat{G}(y) &= \sum_y \frac{1}{\sqrt{2^n}} \sum_x (-1)^{x\cdot y} F(x) \frac{1}{\sqrt{2^n}} \sum_z (-1)^{z\cdot y} G(z) \\ &= \frac{1}{2^n} \sum_x \sum_z F(x)\,G(z) \sum_y (-1)^{x\cdot y + z\cdot y} \\ &= \sum_x F(x)\,G(x).\end{aligned}$$

For the proof of Theorem 5.1, we take $F(x) = 1 - 2f(x)$ to be the rescaled parity function. As in the preceding section, we have

$$1 - 2\delta \le \left| \sum_x \Pr(x \mid L) F(x) \right|.$$

Setting $G(x) = \Pr(x \mid L)$ and applying the Parseval identity, we have

$$1 - 2\delta \le \left| \sum_y \hat{G}(y) \hat{F}(y) \right|. \tag{5.1}$$

For F the rescaled parity function, a simple calculation yields $\hat{F}$:

$$\hat{F}(y) = \begin{cases} \sqrt{2^n} & \text{if } y = (1, \ldots, 1), \\ 0 & \text{otherwise.} \end{cases}$$

Substituting this formula into (5.1) yields

$$1 - 2\delta \le \sqrt{2^n}\ \ |\hat{G}(1, \ldots, 1)|.$$

From the definitions of G and $\hat{G}$, this reduces to

$$1 - 2\delta \le \left| \sum_x (-1)^{|x|} \Pr(x \mid L) \right|, \tag{5.2}$$

where $|y| = \sum_{1 \le i \le n} y_i$ denotes the number of i such that $y_i = 1$.

To estimate the right-hand side of (5.2), we observe that

$$\begin{aligned}\sum_x (-1)^{|x|} \Pr(x \mid L) &= \prod_{1 \le i \le n} \big(\Pr_i(0 \mid L_i) - \Pr_i(1 \mid L_i) \big) \\ &= \prod_{1 \le i \le n} \big(1 - 2\Pr_i(1 \mid L_i) \big).\end{aligned}$$

Since

$$|1 - 2u| = 2 \max\{u, 1 - u\} - 1,$$

we have

$$\begin{aligned}1 - 2\delta &\le \left| \sum_x (-1)^{|x|} \Pr(x \mid L) \right| \\ &= \prod_{1 \le i \le n} (2P_i - 1),\end{aligned}$$

with $P_i = \max\{\Pr_i(0 \mid L_i), \Pr_i(1 \mid L_i)\}$ as defined in section 3. Using (3.4) we have

$$1 - 2\delta \leq \prod_{1 \leq i \leq n} \left(1 - \frac{1}{E^{C_i}}\right).$$

Since $1 - u \leq \exp -u$, we have

$$\begin{aligned} 1 - 2\delta &\leq \prod_{1 \leq i \leq n} \exp\left(-E^{-C_i}\right) \\ &= \exp\left(-\sum_{1 \leq i \leq n} E^{-C_i}\right). \end{aligned}$$

Since $E^u = \exp_E u$ is a convex function of u, we have

$$\begin{aligned} 1 - 2\delta &\leq \exp\left(-n \exp_E\left(-\frac{1}{n} \sum_{1 \leq i \leq n} C_i\right)\right) \\ &= \exp\left(-\exp_E\left(\log_E n - \frac{C}{n}\right)\right). \end{aligned}$$

Thus we obtain

$$C \geq n \log_E n - n \log_E \log\bigl(1/(1 - 2\delta)\bigr).$$

This completes the proof of Theorem 5.1.

6. Resilient Boolean functions. A Boolean function f of n arguments is *unbiased* if

$$\sum_x F(x) = 0,$$

where $F(x) = 1 - 2f(x)$ is the rescaled real-valued function as in the preceding section, and the sum is over all 2^n values of x. Thus a function is unbiased if it assumes the values 0 and 1 for equal numbers of inputs.

A Boolean function f is *t-resilient* if every function obtained from f by substituting constants for at most t arguments is an unbiased function of the remaining arguments. Thus a function is 0-resilient if and only if it is unbiased. Our main result in this section is the following.

THEOREM 6.1. *If f is t-resilient and the leaf L with cost C is (ε, δ)-good for f, then*

$$C \geq (t+1) \log_E \frac{t+1}{\frac{n}{2} H\left(\frac{t+1}{n}\right) + \log \frac{1}{1-2\delta}},$$

where $E = (1 - \varepsilon)/\varepsilon$, and $H(\eta) = -\eta \log \eta - (1 - \eta) \log(1 - \eta)$ for $0 < \eta < 1$, extended by continuity to $H(0) = H(1) = 0$.

The projection functions, of the form $f(x_1, \ldots, x_n) = x_i$, are 0-resilient but not 1-resilient. The parity functions, of the form $f(x_1, \ldots, x_n) = x_1 + \cdots + x_n + c \pmod 2$, are $(n-1)$-resilient, which is the maximum possible for a function of n arguments.

Theorem 5.1 applies to many other functions however. If g and h are t-resilient functions of k arguments, then

$$f(x_1,\dots,x_{k+1}) = \begin{cases} g(x_1,\dots,x_k) & \text{if } x_{k+1}=0,\\ h(x_1,\dots,x_k) & \text{if } x_{k+1}=1, \end{cases}$$

defines a t-resilient function of $k+1$ arguments. Since there are two distinct t-resilient parity functions of $t+1$ arguments, and this scheme allows us to square the number of functions by adding one argument, we conclude that there are at least $2^{2^{n-t-1}}$ t-resilient functions of n arguments.

Our proof of Theorem 5.1 will exploit a characterization of resilient functions in terms of their Fourier transforms. Friedman [8] has observed that this characterization is implicit in the work of Chor et al. [2], although the terminology of Fourier transforms is not used there.

PROPOSITION 6.2 (see [2]). *Let $\hat{F}$ be the Fourier transform of $F(x) = 1 - 2f(x)$ for some Boolean function f of n arguments. Then for $t \geq 0$, f is t-resilient if and only if $\hat{F}(y) = 0$ for all y such that $|y| \leq t$.*

In particular, a function f is unbiased if and only if $\hat{F}(0,\dots,0) = 0$, and the parity functions are the only functions for which $\hat{F}(y) = 0$ for all y except $y = (1,\dots,1)$.

We shall also need the following standard estimate for sums of binomial coefficients.

LEMMA 6.3. *If $l \geq n/2$, then*

$$\sum_{k\geq l} \binom{n}{k} \leq \exp\bigl(nH\bigl(l/n\bigr)\bigr).$$

Proof. For $\xi \geq 1$ we have

$$\sum_{k\geq l} \binom{n}{k} \leq \xi^{-l} \sum_{k} \binom{n}{k} \xi^k = \xi^{-l}(1+\xi)^n.$$

Taking $\xi = l/(n-l)$, so that $\xi \geq 1$ follows from $l \geq n/2$, we obtain

$$\sum_{k\geq l} \binom{n}{k} \leq \frac{n^n}{l^l (n-l)^{n-l}} = \exp\bigl(nH(l/n)\bigr),$$

as claimed. □

As in the preceding, section we have

$$1 - 2\delta \leq \left| \sum_y \hat{G}(y)\hat{F}(y) \right|, \tag{6.1}$$

where $\hat{G}$ is the Fourier transform of $G(x) = \Pr(x \mid L)$. Since f is t-resilient, we have $\hat{F}(y) = 0$ for $|y| \leq t$, and thus we have

$$1 - 2\delta \leq \sum_{\substack{y\\ |y|\geq t+1}} |\hat{G}(y)||\hat{F}(y)|.$$

Using Cauchy's inequality we obtain

$$(1-2\delta)^2 \leq \left(\sum_{\substack{y\\ |y|\geq t+1}} \hat{G}(y)^2 \right) \left(\sum_{\substack{y\\ |y|\geq t+1}} \hat{F}(y)^2 \right). \tag{6.2}$$

Since $F(x) = \pm 1$, Parseval's identity yields

$$\sum_{\substack{y \\ |y| \geq t+1}} \hat{F}(y)^2 \leq \sum_y \hat{F}(y)^2 = \sum_x F(x)^2 = 2^n.$$

Thus from (6.2) we obtain

$$(1-2\delta)^2 \leq \left(\sum_{\substack{y \\ |y| \geq t+1}} \hat{G}(y)^2 \right) 2^n. \tag{6.3}$$

We have

$$\begin{aligned} \sum_{\substack{y \\ |y| \geq t+1}} \hat{G}(y)^2 &\leq \left(\max_{\substack{y \\ |y| \geq t+1}} \hat{G}(y)^2 \right) \left(\sum_{\substack{y \\ |y| \geq t+1}} 1 \right) \\ &\leq \left(\max_{\substack{y \\ |y| \geq t+1}} \hat{G}(y)^2 \right) \left(\sum_{k \geq t+1} \binom{n}{k} \right) \\ &\leq \left(\max_{\substack{y \\ |y| \geq t+1}} \hat{G}(y)^2 \right) \exp\left(nH\left(\frac{t+1}{n} \right) \right). \end{aligned}$$

Thus from (6.3) we obtain

$$(1-2\delta)^2 \leq \left(\max_{\substack{y \\ |y| \geq t+1}} \hat{G}(y)^2 \right) \exp\left(nH\left(\frac{t+1}{n} \right) \right) 2^n.$$

We have

$$\hat{G}(y)^2 = \frac{1}{2^n} \left(\sum_x (-1)^{x \cdot y} \Pr(x \mid L) \right)^2.$$

The sum on the right-hand side can be estimated in the same way as the sum in (5.2): if $|y| = k \geq t+1$, the sum factors into a product of k factors, and the final result is

$$\hat{G}(y)^2 \leq \frac{1}{2^n} \exp\left(-2 \exp_E\left(\log_E k - \frac{C}{k} \right) \right),$$

so that

$$\max_{\substack{y \\ |y| \geq t+1}} \hat{G}(y)^2 \leq \frac{1}{2^n} \exp\left(-2 \exp_E\left(\log_E(t+1) - \frac{C}{t+1} \right) \right).$$

Thus from (6.3) we obtain

$$(1-2\delta)^2 \leq \exp\left(-2 \exp_E\left(\log_E(t+1) - \frac{C}{t+1} \right) \right) \exp\left(nH\left(\frac{t+1}{n} \right) \right).$$

This yields

$$C \geq (t+1) \log_E \frac{t+1}{\frac{n}{2} H\left(\frac{t+1}{n} \right) + \log \frac{1}{1-2\delta}},$$

which completes the proof of Theorem 6.1.

REFERENCES

[1] Y. Brandman, A. Orlitsky, and J. Hennessy, *A spectral lower bound technique for the size of decision trees and two-level AND/OR circuits*, IEEE Trans. Comput., 39 (1990), pp. 282–287.

[2] B. Chor, O. Goldreich, J. Hastad, J. Friedman, S. Rudich, and R. Smolensky, *The bit extraction problem or t-resilient functions*, in Proc. 26th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1985, pp. 396–407.

[3] R. L. Dobrushin and S. I. Ortyukov, *Lower bound for the redundancy of self-correcting arrangements of unreliable functional elements*, Problems Inform. Transmission, 13 (1977), pp. 59–65.

[4] R. L. Dobrushin and S. I. Ortyukov, *Upper bound for the redundancy of self-correcting arrangements of unreliable functional elements*, Problems Inform. Transmission, 13 (1977), pp. 203–218.

[5] W. Evans and N. Pippenger, *Lower bounds for noisy Boolean decision trees*, in Proc. 28th Annual ACM Symposium on Theory of Computing, ACM, New York, 1996, pp. 620–628.

[6] U. Feige, D. Peleg, P. Raghavan, and E. Upfal, *Computing with unreliable information*, in Proc. 22nd Annual ACM Symposium on Theory of Computing, ACM, New York, 1990, pp. 128–137.

[7] U. Feige, P. Raghavan, D. Peleg, and E. Upfal, *Computing with noisy information*, SIAM J. Comput., 23 (1994), pp. 1001–1018.

[8] J. Friedman, *On the bit extraction problem*, in Proc. 33rd Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1992, pp. 314–319.

[9] A. Gál, *Lower bounds for the complexity of reliable Boolean circuits with noisy gates*, in Proc. 32nd Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1991, pp. 594–601.

[10] P. Gács and A. Gál, *Lower bounds for the complexity of reliable Boolean circuits with noisy gates*, IEEE Trans. Inform. Theory, 40 (1994), pp. 579–583.

[11] W. Hoeffding, *Probability inequalities for sums of bounded random variables*, J. Amer. Statist. Assoc., 58 (1963), pp. 13–30.

[12] N. Linial, Y. Mansour, and N. Nisan, *Constant depth circuits, Fourier transform, and learnability*, J. Assoc. Comput. Mach., 40 (1993), pp. 607–620.

[13] D. E. Muller, *Complexity in electronic switching circuits*, Institute of Radio Engineers Trans. Elec. Comput., 5 (1956), pp. 15–19.

[14] J. von Neumann, *Probabilistic logics and the synthesis of reliable organisms from unreliable components*, in Automata Studies, C. E. Shannon and J. McCarthy, eds., Princeton University Press, Princeton, NJ, 1956, pp. 43–98.

[15] N. Pippenger, *On networks of noisy gates*, in Proc. 26th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1985, pp. 30–36.

[16] N. Pippenger, G. D. Stamoulis, and J. N. Tsitsiklis, *On a lower bound for the redundancy of reliable networks with noisy gates*, IEEE Trans. Inform. Theory, 37 (1991), pp. 639–643.

[17] R. Reischuk and B. Schmeltz, *Reliable computation with noisy circuits and decision trees—A general $n \log n$ lower bound*, in Proc. 32nd Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA, 1991, pp. 602–611.