Exploiting Product Distributions to Identify Relevant Variables of Correlation Immune Functions

Lisa Hellerstein
Bernard Rosell
Eric Bach
Soumya Ray
David Page

Technical Report #1627

January 2008

Exploiting Product Distributions to Identify Relevant Variables of Correlation Immune Functions

Lisa Hellerstein HSTEIN@CIS.POLY.EDU
Department of Computer and Information Science
Polytechnic University
5 Metrotech Center
Brooklyn, NY 11201, USA

Bernard Rosell BROSELL@ATT.COM
AT&T Laboratories
200 South Laurel Ave.
Middletown, NJ 07748, USA

Eric Bach BACH@CS.WISC.EDU
Department of Computer Sciences
University of Wisconsin, Madison
Madison, WI 53706, USA

Soumya Ray SRAY@EECS.OREGONSTATE.EDU
School of Electrical Engineering and Computer Science
1148 Kelley Engineering Center
Oregon State University
Corvallis, OR 97331, USA

David Page PAGE@BIOSTAT.WISC.EDU
Department of Biostatistics and Medical Informatics
University of Wisconsin, Madison
Madison, WI 53706, USA

Abstract

A Boolean function f is *correlation immune* if each input variable is independent of the output, under the uniform distribution on inputs. (For example, the parity function is correlation immune.) We consider the problem of identifying relevant variables of a correlation immune function, in the presence of irrelevant variables. We address this problem in two different contexts. First, we analyze *Skewing*, a heuristic method that was developed to improve the ability of greedy decision tree algorithms to identify relevant variables of correlation immune Boolean functions, given examples drawn from the uniform distribution (Page and Ray, 2003). We present theoretical results revealing both the capabilities and limitations of skewing. Second, we explore the problem of identifying relevant variables in the *Product Distribution Choice* (PDC) learning model, a model in which the learner can choose product distributions and obtain examples from them. We give two new algorithms for finding relevant variables of correlation immune functions in the PDC model.

1. Introduction

A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is *correlation immune* if for every input variable x_i, the values of x_i and $f(x_1, \ldots, x_n)$ are independent, with respect to the uniform distribution on $\{0,1\}^n$ (cf. Roy, 2002). Examples of correlation immune functions include parity of $k \geq 2$ variables, the constant functions $f \equiv 1$ and $f \equiv 0$, and the function $f(x) = 1$ iff all bits of x are equal.

If a function f is not correlation immune, then given access to examples of f drawn from the uniform distribution, one can easily identify (at least one) relevant variable of f by finding an input variable that is correlated with the output of f. This approach clearly fails if f is correlation immune.

We consider the problem of identifying relevant variables of a correlation immune function, in the presence of irrelevant variables. We address this problem in two different contexts. First, we present a theoretical analysis of *Skewing*, a heuristic method that was developed to improve the ability of greedy decision tree learning algorithms to identify relevant variables of correlation immune functions, given examples drawn from the uniform distribution (Page and Ray, 2003; Ray and Page, 2004). We present theoretical results that reveal both the strengths and limitations of skewing. Second, we present algorithms for identifying relevant variables in the *Product Distribution Choice* (PDC) model of learning. The PDC model, which we introduce below, is a variant of the standard PAC learning model (Valiant, 1984) in which the learner can specify product distributions and sample from them.

Greedy decision tree learning algorithms perform poorly on correlation immune functions because they rely on measures such as Information Gain (Quinlan, 1997) and Gini gain (Breiman et al., 1984) to choose which variables to place in the nodes of the decision tree. The correlation immune functions are precisely those in which every attribute has zero gain under all standard gain measures, when the gain is computed on the complete dataset (i.e. the truth table) for the function. Thus when examples of a correlation immune function are drawn uniformly at random from the complete dataset, the learning algorithms have no basis for distinguishing between relevant and irrelevant attributes.

Experiments have shown skewing to be successful in learning many correlation immune functions (Page and Ray, 2003). One of the original motivations behind skewing was the observation that obtaining examples from non-uniform product distributions can be helpful in learning particular correlation immune functions such as parity. Skewing works by reweighting the given training set to simulate receiving examples from a subclass of product distributions called *skewed* distributions.

However, simulating alternative distributions is not the same as sampling directly from them. The *Product Distribution Choice* (PDC) model allows such direct sampling. This model can be seen as a variant of the PAC model, and has similarities with other learning models studied previously (see Section 4). In the PDC model, the learner has access to an oracle from which it can request examples. Before requesting an example, the learner specifies a product distribution. The oracle then supplies an example drawn from that distribution. In our study of the PDC model, we focus on a fundamental learning task: the problem of identifying relevant variables in the presence of irrelevant ones.

Note that by setting the parameters of the product distribution to be equal to 0 and 1, one can simulate membership queries in the PDC model. However, we are particularly interested in exploring learning in the PDC model when the parameters of the chosen product distributions are bounded away from 0 and 1.

Our interest in the PDC model is motivated not just by our study of skewing, but by a more general question: In learning, how much does it help to have access to data from different distributions? In practice, it may be possible to obtain data from different distributions by collecting it from different sources or populations. Alternatively, one may be able to alter environmental conditions to change the distribution from which data is obtained. In such settings, it can be expensive to sample from too many distributions, and it may be impossible to sample from certain "extreme" distributions. Thus in the PDC model, we are concerned not just with time and sample complexity, but also in the number and type of product distributions specified.

2. Summary of results

We begin by proving that in an idealized setting, skewing will succeed. More particularly, we show that when the learning algorithm has access to the complete truth table of a target Boolean function, skewing will succeed in finding a relevant variable of that function. (More particularly, under any random choice of skewing parameters, a single round of the skewing procedure will find a relevant variable with probability 1.)

We also prove a result in the idealized setting for a variant of skewing called *sequential skewing* (Ray and Page, 2004). Experiments indicate that sequential skewing scales better to higher dimensional problems than standard skewing. We show here, however, that even when the entire truth table is available as the training set, sequential skewing is ineffective for a subset of the correlation immune functions known as the *2-correlation immune* functions. A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is 2-correlation immune if, for every pair of distinct input variables x_i and x_j, the variables x_i, x_j, and $f(x_1, \ldots, x_n)$ are mutually independent. Thus, any practical advantage sequential skewing has over standard skewing comes at the cost of not working on this subset of functions.

We present two new algorithms in the PDC model for identifying a relevant variable of an n-variable Boolean function with r relevant variables. The first algorithm uses only r distinct p-biased distributions (i.e. distributions in which each input variable is set to 1 with some fixed probability p). It runs in time polynomial in n and the sample size $O((r+1)^{2r} \ln \frac{2nr}{\delta})$. The second algorithm uses $O(e^{4r} \ln \frac{1}{\delta})$ p-biased distributions, and runs in time polynomial in n and the sample size, $O(e^{28r} \ln^2 \frac{n}{\delta})$. For $r = O(\log n)$, only the second algorithm runs in time polynomial in n, but the first uses $O(\log n)$ distributions, whereas the second uses a number of distributions that depends polynomially on n.

Both algorithms are non-adaptive: they request all examples before processing them. Since the algorithms use only p-biased distributions, and each such distribution is a skewed distribution, they can be viewed as skewing algorithms for a setting in which it is possible to sample directly from skewed distributions, rather than just to simulate those distributions.

The second of the two algorithms is based on a new sample complexity result that we prove using martingales.

We also briefly describe two PDC algorithms that are implicit in the literature. One follows directly from the techniques of Bshouty and Feldman (2002). When $r = O(\log n)$ it runs in time polynomial in n and uses a number of distributions that is linear in n. The exact sample complexity of this algorithm is somewhat different than the sample complexity of our second new algorithm, and the distributions used are not p-biased. The other algorithm from the literature is a simple membership query algorithm.

Finally, we analyze skewing in the context for which it was originally designed: learning from a random sample drawn from the uniform distribution. We present a negative result on learning parity functions with skewing in this context, based on techniques from statistical query learning. One implication of the result is that skewing requires a sample of size at least $n^{\Omega(\log n)}$ to find (with constant probability of failure) a relevant variable of an n-variable Boolean function computing the parity of $\log n$ of its variables. (Technically, we prove the result for a variant of skewing called skewing with independent samples. We give evidence that the lower bound should also apply to standard skewing.)

Correlation immunity is defined in terms of the uniform distribution. We discuss a natural extension of correlation immunity to non-uniform product distributions. We give a simple example of a function that is correlation immune with respect to a non-uniform product distribution. Thus while functions like parity are difficult for greedy learners when examples come from the uniform distribution, other functions can be difficult when examples come from another product distribution.

Our analysis of skewing in the idealized setting, and our two new algorithms in the PDC model, are both based on a lemma that we prove concerning a property of Boolean functions. Specifically, we show that every non-constant Boolean function f on $\{0,1\}^n$ has a variable x_i such that induced functions $f_{x_i \leftarrow 0}$ and $f_{x_i \leftarrow 1}$ on $\{0,1\}^{n-1}$ (produced by hardwiring x_i to 0 and 1) do not have the same number of positive examples of Hamming weight k, for some k.

The paper is organized as follows. We first give some background on skewing in Section 3. In Section 4, we discuss related work. Section 5 contains basic definitions and lemmas, including characterizations of correlation immune functions, and simple lemmas on quantities such as Gini gain and the magnitude of the first-order Fourier coefficients. It also contains a simple example of a function that is correlation immune with respect to a non-uniform product distribution. Section 6 discusses sample complexity bounds used later in the paper, and proves an upper bound on the estimation of Gini gain, based on martingales.

In Section 7, we prove our structural result on Boolean functions.

We begin our analysis of skewing in Section 8 with results in the idealized setting in which the entire truth table is given as the training set.

Section 9 contains our two new algorithms for the PDC model. It also the discussion of the two PDC algorithms implicit in the literature. Finally, Section 10 contains our sample complexity lower bounds on learning parity functions.

3. Background on Skewing

As a motivating example, suppose we have a Boolean function $f(x_1, \ldots, x_n)$ whose value is the parity of r of its variables. Those r variables are relevant, and the rest are irrelevant.

Function f is correlation immune. With respect to the uniform distribution on the domain of f, both relevant and irrelevant variables have zero gain. Equivalently, the first-order Fourier coefficients are all zero (cf. Section 5.3). But, with respect to other product distributions on the examples, relevant variables have non-zero gain, while irrelevant variables still have zero gain (see Page and Ray, 2003; Arpe and Reischuk, 2006). This suggests that learning correlation immune functions might be easier if examples could be obtained from non-uniform product distributions.

In many machine learning applications, however, we have little or no control over the distribution from which we obtain training data. The approach taken by skewing is to reweight the training data, to simulate receiving examples from another distribution. More particularly, the skewing algorithm works by choosing a "preferred setting" (either 0 or 1) for every variable x_i in the examples, and a weighting factor p where $\frac{1}{2} < p < 1$. These choices define a product distribution over examples $x \in \{0,1\}^n$ in which each variable x_i has its preferred setting with probability p, and the negation of that setting with probability $1 - p$.

To simulate receiving examples from this product distribution, the skewing algorithm begins by initializing the weight of every example in the training set to 1. Then, for each x_i, and each example, it multiplies the weight of the example by p if the value of x_i in the example matches its preferred setting, and by $1-p$ otherwise. This process is called "skewing" the distribution. The algorithm computes the gain of each variable after the reweighting. The algorithm repeats this procedure a number of times, with different preferred settings chosen each time. Finally, it uses all the calculated gains to determine which variable to output. The exact method used varies in different skewing implementations. In the paper that introduced skewing, the variable chosen was the one whose calculated gains exceeded a certain threshold the maximum number of times (Page and Ray, 2003).

In the context of decision tree learning, skewing is applied at every node of the decision tree, in place of standard gain calculations. After running skewing on the training set at that node, the variable chosen by the skewing procedure is used as the split variable at that node.

In investigating skewing, we are particularly interested in cases in which the number of relevant variables is much less than the total number of variables. Optimally, we would like sample complexity and running time to depend polynomially on n, 2^r (and $\log \frac{1}{\delta}$), so that we have a polynomial-time algorithm when $r = O(\log n)$.

4. Related Work

Throughout this paper, we focus on the problem of finding a relevant variable of a target Boolean function, given a labeled sample drawn from the uniform distribution. Given a procedure that finds a single relevant variable x_i of a Boolean function f (for any f with at most r relevant variables), it is easy to extend this procedure to find all relevant variables of the target by recursively applying it to the induced functions obtained by hardwiring x_i to 1 and 0 respectively.

It is a major open problem whether there is a polynomial-time algorithm for finding relevant variables of a Boolean function of $\log n$ relevant variables (out of n total variables) using examples from the uniform distribution (cf. Blum, 2003). Mossel et al. (2003) gave an algorithm for learning arbitrary functions on r relevant variables, using examples drawn from the uniform distribution, in time polynomial in n^{cr} and $\ln(1/\delta)$, for some $c < 1$. This improves on the naïve algorithm which requires time polynomial in n^r. The heart of the algorithm is a procedure to find a relevant variable. The algorithm of Mossel et al. uses both Gaussian elimination and estimates of Fourier coefficients, and is based on structural properties of Boolean functions. It is not noise-tolerant.

Mossel et al. also briefly considered the question of finding a relevant variable, given examples drawn from a single product distribution $[p_1, \ldots, p_n]$. [1] They stated a result that is similar to our Theorem 8.1, namely that if a product distribution is chosen at random, then with probability 1, the Fourier coefficient (for that distribution) associated with any relevant variable will be non-zero. The important difference between that result and Theorem 8.1 is that our theorem applies not to all random product distributions, but just to random skewed distributions. Since random product distributions have different properties than random skewed distributions, the proof given by Mossel et al. does not suffice to prove Theorem 8.1.

The problem of learning parity functions has been extensively studied in various learning models. It is a well-known open question whether it is possible to PAC-learn parity functions in polynomial time, using examples drawn from the uniform distribution, in the presence of random classification noise. This problem is at least as difficult as other open problems in learning; in fact, a polynomial time algorithm for this problem would imply a polynomial-time algorithm for the problem mentioned above, learning functions of $\log n$ relevant variables using examples from the uniform distribution (Feldman et al., 2006).

At the other extreme from correlation-immune functions are functions for which all first order Fourier coefficients are non-zero (i.e. all relevant variables have non-zero gain). This is true of monotone functions and symmetric functions (see Mossel et al., 2003). Arpe and Reischuk, extending previous results, gave a Fourier-based characterization of the class of functions that can be learned using a standard greedy covering algorithm (Arpe and Reischuk, 2006; Akutsu et al., 2003; Fukagawa and Akutsu, 2005). This class is a superset of the set of functions for which all relevant variables have non-zero degree-1 Fourier coefficients.

The PDC model investigated in this paper has some similarity to the extended statistical query model of Bshouty and Feldman (2002). In that model, the learner can specify a product distribution in which each variable is set to 1 with probability $\rho, 1/2$ or $1-\rho$, for some constant $1/2 > \rho > 0$. The learner can then ask a *statistical query* which will be answered with respect to the specified distribution. In the PDC model the user can specify an arbitrary product distribution, and can ask for random examples with respect to that distribution. One could simulate the extended statistical query model in the PDC model by using random examples (drawn with respect to the specified distribution) to answer the statistical queries.

As noted in the introduction, it is possible to simulate membership queries in the PDC model by setting the parameters of the chosen product distribution to 0 and 1. The problem of efficiently learning Boolean functions with few relevant variables, using membership

[1] They also claimed that this result implies an algorithm for learning functions with r relevant variables in time polynomial in 2^r, n, and $\ln(1/\delta)$, given examples drawn from almost any product distribution. However, the justification for their claim was faulty, since it does not take into account the magnitude of the non-zero Fourier coefficient.

queries alone, has been addressed in a number of papers (Blum et al., 1995; Bshouty and Hellerstein, 1998; Damaschke, 2000). The goal in these papers is to have *attribute-efficient* algorithms that use a number of queries that is polynomial in r, the number of relevant variables, but only logarithmic in n, the total number of variables. Guijarro et al. (1999) investigated the problem of identifying relevant variables in the PAC model with membership queries.

We use Fourier-based techniques in proving some of our results. There is an extensive literature on using Fourier methods in learning, including some of the papers mentioned above. Some of the most important results are described in the excellent survey of Mansour (1994).

Correlation immune functions and k-correlation immune functions have applications to cryptography. They have been widely studied in that field (see Roy, 2002, for a survey), beginning with the work of Siegenthaler (1984). Correlation immune functions have also been studied in other fields under different guises. The truth table of a k-correlation immune function corresponds to a certain orthogonal array (Camion et al., 1991). Orthogonal arrays are used in experimental design. The positive examples of a k-correlation immune function form a k-wise independent set. Such sets are used in derandomization (see e.g. Alon, 1996).

It is natural to ask how many n-variable Boolean functions are correlation immune, since these actually *need* skewing. The question has been addressed in a number of different papers, as described by Roy (2002). Counts of correlation immune functions up to $n = 6$, separated by Hamming weight, were computed by Palmer et al. (1992). For larger n one can use the analytic approximation $2^{2^n} \cdot P_n$, where

$$P_n = \frac{1}{2}\left(\frac{8}{\pi}\right)^{n/2} 2^{-n^2/2}\left(1 - \frac{n^2}{4 \cdot 2^n}\right). \tag{1}$$

Since there are 2^{2^n} Boolean functions in toto, P_n approximates the probability that a random Boolean function is correlation immune. Its main term was found by Denisov (1992), and the rest is the beginning of an asymptotic series investigated by Bach (2007). Even for small n, the above approximation is fairly accurate. For example, there are 503483766022188 6-variable correlation immune functions, and the above formula gives 4.99×10^{14}.

Skewing was developed as an applied method for learning correlation-immune Boolean functions. Skewing has also been applied to non-Boolean functions, and to Bayes nets (Lantz et al., 2007; Ray and Page, 2005).

The main results in Sections 7 and 8 of this paper appeared in preliminary form in Rosell et al. (2005).

5. Preliminaries

We begin with basic definitions and fundamental lemmas.

5.1 Notation and terminology

We consider two-class learning problems, where the features, or variables, are Boolean. A *target function* is a Boolean function $f(x_1, \ldots, x_n)$. An *example* is an element of $\{0,1\}^n$. Example $a \in \{0,1\}^n$ is a *positive example* of Boolean function $f(x_1, \ldots, x_n)$ if $f(a) = 1$, and a *negative example* of f if $f(a) = 0$. A *labeled example* is an element $(a, b) \in \{0,1\}^n \times \{0,1\}$; it is a labeled example of f if $f(a) = b$.

Let $f(x_1, \ldots, x_n)$ be a Boolean function. The function f is a mapping from $\{0,1\}^n$ to $\{0,1\}$. An *assignment* $a = (a_1, \ldots, a_n)$ to the variables $x_1, \ldots, x_n$ is an element of $\{0,1\}^n$. The assignment obtained from a by negating the ith bit of a is denoted by $a_{\neg x_i}$. Given a Boolean function $f(x_1, \ldots, x_n)$, variable x_i is a *relevant variable* of f if there exists $a \in \{0,1\}^n$ such that $f(a) \neq f(a_{\neg x_i})$.

For $\sigma \in \{0,1\}^n$, let $\sigma^i = (\sigma_1, \ldots, \sigma_{i-1}, \sigma_{i+1}, \ldots, \sigma_n)$, that is, σ^i denotes σ with its ith bit removed.

A *truth table* for a function f over a set of variables is a list of all assignments over the variables, together with the mapping of f for each assignment. For $i \in [1 \ldots n]$ and $b \in \{0,1\}$, $f_{x_i \leftarrow b}$ denotes the function on $n-1$ variables produced by "hardwiring" the ith variable of f to b. More formally, $f_{x_i \leftarrow b} : \{0,1\}^{n-1} \rightarrow \{0,1\}$ such that for all $a \in \{0,1\}^{n-1}$, $f_{x_i \leftarrow b}(a) = f(a_1, a_2, \ldots, a_{i-1}, b, a_i, \ldots, a_{n-1})$.

The integers between 1 and n are denoted by $[1 \ldots n]$. For real a and b, (a, b) denotes the open interval from a to b.

For probability distribution D, we use $\Pr_D$ and E_D to denote the probability and expectation with respect to distribution D. For any probability distribution D over a finite set X, and any $A \subseteq X$, we define $\Pr_D(A)$ to be equal to $\sum_{a \in A} \Pr_D(a)$. We omit the subscript D when it is clear from context.

Given a probability distribution D on $\{0,1\}^n$, and a Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$, a *random example of f drawn with respect to D* is an example $(x, f(x))$ where x is drawn with respect to D.

A training set T for learning an n-variable Boolean function is a multiset consisting of elements in $\{0,1\}^n \times \{0,1\}$. It defines an associated distribution on $\{0,1\}^n \times \{0,1\}$ sometimes known as the *empirical distribution.* For each $(a, y) \in \{0,1\}^n \times \{0,1\}$, the probability of (a, y) under this distribution is defined to be the fraction of examples in the training set that are equal to (a, y). In the absence of noise, a training set for learning a function $f : \{0,1\}^n \rightarrow \{0,1\}$ is a set of labeled examples $(x, f(x))$. The empirical distribution on such a training set can be viewed as a distribution on $\{0,1\}^n$, rather than on $\{0,1\}^n \times \{0,1\}$.

A product distribution D on $\{0,1\}^n$ is a distribution defined by a parameter vector $[p_1, \ldots, p_n]$ in $[0,1]^n$ where for all $x \in \{0,1\}^n$, $\Pr_D[x] = (\prod_{i: x_i = 1} p_i)(\prod_{i: x_i = 0}(1 - p_i))$. The uniform distribution on $\{0,1\}^n$ is the product distribution defined by $[1/2, 1/2, \ldots, 1/2]$. For fixed $p \in (0,1)$, we use $D[p]$ to denote the product distribution defined by $[p, \ldots, p]$ Distribution $D[p]$ is the *p-biased* distribution.

A *skew* is a pair (σ, p) where $\sigma \in \{0,1\}^n$ is an assignment, and $p \in (0,1)$. We refer to σ as the *orientation* of the skew, and p as the *weighting factor.*

Each skew (σ, p) induces a probability distribution $D_{(\sigma,p)}$ on the 2^n assignments in $\{0,1\}^n$ as follows. Let $\tau_p : \{0,1\} \times \{0,1\} \rightarrow \{p, 1-p\}$ be such that for $b, b' \in \{0,1\}$, $\tau_p(b, b') = p$ if $b = b'$ and $\tau_p(b, b') = 1 - p$ otherwise. For each $a \in \{0,1\}^n$, distribution $D_{(\sigma,p)}$ assigns probability $\Pi_{i=1}^n \tau_p(\sigma_i, a_i)$ to a. Thus distribution $D_{(\sigma,p)}$ is a product distribution in which every variable is set to 1 either with probability p, or with probability $1 - p$. When $\sigma = (1, \ldots, 1)$, the distribution $D_{(\sigma,p)}$ is commonly called a *p-biased* distribution.

Given a skew (σ, p) and a function f, the gain of a variable x_i with respect to f under distribution $D_{(\sigma,p)}$ is thus equivalent to the gain that is calculated by applying skew (σ, p)

(using the procedure described in Section 3) to a training set consisting of the entire truth table for f. We say that variable x_i *has gain for* (f, σ, p) if the gain of x_i with respect to f under $D_{(\sigma,p)}$ is non-zero. We call distributions $D_{(\sigma,p)}$ *skewed distributions*.

We note that in other papers on skewing, p is required to be in $(1/2, 1)$, rather than in $(0, 1)$. Here it is more convenient for us to let p be in $(0, 1)$. Given any orientation σ, and any $p \in (0, 1)$, skew $(\bar{\sigma}, 1-p)$, where $\bar{\sigma}$ is the bitwise complement of σ, induces the same distribution as (σ, p). Thus allowing p to be in $(0, 1)$ does not change the class of skewed distributions, except that we also include the uniform distribution.

Given $a, b \in \{0, 1\}^n$, let $\Delta(a, b) = |\{i \in [1, \dots, n] | a_i \neq b_i\}|$, i.e., $\Delta(a, b)$ is the Hamming distance between a and b. For $a, b \in \{0, 1\}^n$, let $a + b$ denote the componentwise mod 2 sum of a and b. Given $c \in \{0, 1\}^n$, we use $w(c)$ to denote the Hamming weight (number of 1's) of c. Thus $w(a + b) = \Delta(a, b)$.

In the *product distribution choice* (PDC) model, the learning algorithm has access to a special type of random example oracle for a target function $f(x_1, \dots, x_n)$. This random example oracle takes as input the parameters $[p_1, \dots, p_n]$ of a product distribution D over unlabeled examples $(x_1, \dots, x_n)$. The oracle responds with a random example $(x_1, \dots, x_n)$ drawn according to the requested distribution D. We assume the learner knows n.

5.2 Gain

Greedy tree learners partition a data set recursively, choosing a "split variable" at each step. They differ from one another primarily in their measures of "goodness" for split variables. The measure used in the well-known CART system is *Gini gain* (Breiman et al., 1984). Gini gain was also used in the decision tree learners employed in experimental work on skewing (Page and Ray, 2003; Ray and Page, 2004). In this paper, we use the term "gain" to denote Gini gain.

Gini gain is defined in terms of another quantity called *Gini index*. Let S be a (multi) set of labeled examples. Let $S_1 = \{(x, y) \in S | y = 1\}$ and $S_0 = \{(x, y) \in S | y = 0\}$. The Gini index of S is $2\frac{|S_1||S_0|}{|S|^2}$. Let $\tilde{H}(S)$ denote the Gini index of S. Let $T_1 = \{(x, y) \in S_1 | x_i = 1\}$ and $T_0 = \{(x, y) \in S | x_i = 0\}$. For any potential split variable x_i, the Gini index of S *conditional on* x_i is defined to be $\tilde{H}(S|x_i) := \frac{|T_1|}{|S|}\tilde{H}(T_1) + \frac{|T_0|}{|S|}\tilde{H}(T_0)$. The *Gini gain* of a variable x_i with respect to S is

$$G(S, x_i) = \tilde{H}(S) - \tilde{H}(S|x_i). \tag{2}$$

In decision tree terms, this is the weighted sum of the Gini indices of the child nodes resulting from a split on x_i.

Some definitions of Gini gain and Gini index differ from the one above by a factor of 2; our definition follows that of Breiman et al. (1984).

Now suppose that each example in our (multi) set S has an associated *weight*, a real number between 0 and 1. We can define the gain on this weighted set by modifying the above definitions in the natural way: each time the definitions involve the size of a set, we instead use the sum of the weights of the elements in the set.

We can also define Gini index and Gini gain of variable x_i with respect to $f : \{0, 1\}^n \rightarrow \{0, 1\}$ under a distribution D on $\{0, 1\}^n$. The Gini index of f with respect to a probability distribution D on $\{0, 1\}^n$ is $2 \Pr_D[f = 1](1 - \Pr_D[f = 1])$. Let $\tilde{H}_D(f)$ denote the Gini index of f with respect to D. For any potential split variable x_i, the Gini index of f with respect to D, *conditional on* x_i is $\tilde{H}_D(f|x_i) := \Pr_D[x_i = 0]\tilde{H}_D(f_{x_i \leftarrow 0}) + \Pr_D[x_i = 1]\tilde{H}_D(f_{x_i \leftarrow 1})$. The *Gini gain* of a variable x_i with respect to f, under distribution D, is

$$G_D(f, x_i) = \tilde{H}_D(f) - \tilde{H}_D(f|x_i) \tag{3}$$

The Gini gain of x_i with respect to f, under the uniform distribution on $\{0,1\}^n$, is equal to the Gini gain of x_i with respect to the training set T consisting of all entries in the truth table. The Gini gain is always a value in the interval $[0, 1]$.

The following lemma relates the size of the Gini gain with respect to a distribution D to the difference in the conditional probabilities $\Pr_D[f = 1|x_i = 1] - \Pr_D[f = 1|x_i = 0]$.

Lemma 1 *Let f be an n-variable Boolean function, and let D be a distribution on $\{0,1\}^n$ such that $\Pr[x_i = 1]$ is strictly between 0 and 1. Then $G_D(f, x_i)$, the Gini gain of variable x_i with respect to f, under distribution D, is equal to*

$$2p_i(1 - p_i)(Pr_D[f = 1|x_i = 1] - Pr_D[f = 1|x_i = 0])^2 \tag{4}$$

where $p_i = \Pr_D[x_i = 1]$.

Proof. Let $p = p_i$, $\beta = \Pr_D[f = 1]$, $\beta_1 = \Pr_D[f = 1|x_i = 1]$, and $\beta_0 = \Pr_D[f = 1|x_i = 0]$. Thus $\beta = p\beta_1 + (1 - p)\beta_0$.

The Gini gain of x_i with respect to f is

$$\begin{aligned}
&2(\beta(1-\beta) - p(\beta_1(1-\beta_1)) - (1-p)(\beta_0(1-\beta_0))) \\
&\quad = \quad 2(\beta(1-\beta) - (p\beta_1 + (1-p)\beta_0)) + p\beta_1^2 + \beta_0^2(1-p)) \\
&\quad = \quad 2(\beta(1-\beta) - \beta + p\beta_1^2 + \beta_0^2(1-p)) \\
&\quad = \quad 2(-\beta^2 + p\beta_1^2 + \beta_0^2(1-p))
\end{aligned} \tag{5}$$

Substituting $p\beta_1 + (1 - p)\beta_0$ for β, we get that the last quantity is

$$\begin{aligned}
&= \quad 2(-p^2\beta_1^2 - 2p(1-p)\beta_0\beta_1 - (1-p)^2\beta_0^2 + p\beta_1^2 + \beta_0^2(1-p)) \\
&= \quad 2((1-p)p(\beta_1^2 - 2\beta_0\beta_1 + \beta_0^2)) \\
&= \quad 2p(1-p)(\beta_1 - \beta_0)^2
\end{aligned} \tag{6}$$

□

Under distribution D on $\{0,1\}^n$, x_i and (the output of) f are independent iff $G_D(f, x_i) = 0$.

5.3 Fourier Coefficients

Given a Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$, define an associated function $F = 1 - 2f$. That is, $F : \{0,1\}^n \rightarrow \{1,-1\}$ such that $F(x) = 1 - 2f(x)$ for all $x \in \{0,1\}^n$. The function F can be seen as an alternative representation of Boolean function f, using -1 and 1 respectively to represent true and false outputs, rather than 1 and 0.

For every $z \in \{0,1\}^n$, let $\chi_z : \{0,1\}^n \rightarrow \{1,-1\}$ be such that $\chi_z(x) = -1^{(\sum_{i:z_i=1} x_i) \bmod 2}$. Thus χ_z is the alternative representation of the function computing the parity of the variables set to 1 by z. For $z \in \{0,1\}^n$, n-variable Boolean function f, and associated $F = 1-2f$, the *Fourier coefficient* $\hat{f}(z)$ is defined as follows:

$$\hat{f}(z) := E[F(x)\chi_z(x)] \tag{7}$$

here the expectation is with respect to the uniform distribution on $x \in \{0,1\}^n$.

The *degree* of Fourier coefficient $\hat{f}(z)$ is $w(z)$, the Hamming weight of z. The Fourier coefficient *associated with the variable* x_i is $\hat{f}(z)$ where z is the characteristic vector of x_i (i.e. $z_i = 1$ and for $j \neq i$, $z_i = 0$). In an abuse of notation, we will use $\hat{f}(x_i)$ to denote this Fourier coefficient. Thus $\hat{f}(x_i) = E[F(x)(1-2x_i)]$. The function F can be expressed by its Fourier series, $F(x) = \sum_{z \in \{0,1\}^n} \hat{f}(z)\chi_z(x)$.

Fourier coefficients can be generalized from the uniform distribution to product distributions, as described by Furst et al. (1991). Let D be a product distribution on $\{0,1\}^n$ defined by parameters $[p_1, \ldots, p_n]$, all of which are strictly between 0 and 1. For $z \in \{0,1\}^n$, let $\phi_z : \{0,1\}^n \rightarrow \{0,1\}$ be such that $\phi_z(x) = \prod_{i:z_i=1} \frac{\mu_i - x_i}{\sigma_i}$ where $\mu_i = p_i$ is $E_D[x_i]$ and $\sigma_i = \sqrt{p_i(1-p_i)}$ is the standard deviation of x_i under D. The Fourier coefficient $\hat{f}_D(z)$, for product distribution D, is defined as follows:

$$\hat{f}_D(z) := E_D[F(x)\phi_z(x)]. \tag{8}$$

When D is the uniform distribution, this definition is equivalent to the definition of an ordinary Fourier coefficient.

Parseval's identity, applied to the Fourier coefficients of product distributions, states that

$$\sum_{z \in \{0,1\}^n} \hat{f_D}^2(z) = 1. \tag{9}$$

The Fourier coefficient associated with the variable x_i, with respect to product distribution D, is $\hat{f}_D(z)$, where z is the characteristic vector of x_i. Abusing notation as before, we will use $\hat{f}_D(x_i)$ to denote this Fourier coefficient. Thus

$$\hat{f}_D(x_i) = \frac{p_i E_D[F(x)] - E_D[x_i F(x)]}{\sqrt{p_i(1-p_i)}} \tag{10}$$

The next lemma is analogous to Lemma 1, and relates the value of the Fourier coefficient for x_i, for product distribution D, to the difference in conditional probabilities $\Pr_D[f = 1|x_i = 1] - \Pr_D[f = 1|x_i = 0]$.

Lemma 2 *Let f be an n-variable Boolean function, and let D be a product distribution over $\{0,1\}^n$ defined by $\pi = [p_1, \ldots, p_n]$, such that each $p_i \in (0,1)$. Let $F : \{0,1\}^n \rightarrow \{1,-1\}$ be such that $F = 1-2f$. Then the Fourier coefficient associated with x_i, for distribution D, is*

$$2\sqrt{p_i(1-p_i)}(\Pr_D[f = 1|x_i = 1] - \Pr_D[f = 1|x_i = 0]).$$

Proof. By definition,

$$\hat{f}(x_i) = \frac{p_i E_D[F(x)] - E_D[x_i F(x)]}{\sqrt{p_i(1-p_i)}}. \tag{11}$$

Let $\beta = \Pr_D[f = 1]$ (which equals $\Pr_D[F = -1]$), $\beta_1 = \Pr_D[f = 1|x_i = 1]$, and $\beta_0 = \Pr_D[f = 1|x_i = 0]$.

Since $p_i E_D[F(x)] = p_i(1-2\beta)$, $E_D[F(x)x_i] = p_i(1-2\beta_1)$, and $\beta = p\beta_1 + (1-p)\beta_0$, it follows that

$$\begin{aligned} p_i E_D[F(x)] - E_D[x_i F(x)] &= 2p_i(-\beta + \beta_1) \\ &= 2p_i(-p_i\beta_1 - (1-p_i)\beta_0 + \beta_1) \\ &= 2p_i(1-p_i)(\beta_1 - \beta_0). \end{aligned} \tag{12}$$

Dividing by $\sqrt{p_i(1-p_i)}$, the lemma follows. □

The following important facts about first-order Fourier coefficients for product distributions are easily shown. For D a product distribution on $\{0,1\}^n$ where each $p_i \in (0,1)$,

1. If x_i is an irrelevant variable of a Boolean function f, then $\hat{f}_D(x_i) = 0$.

2. $G_D(f, x_i) = 0$ iff $\hat{f}_D(x_i) = 0$.

5.4 Correlation Immune Functions

For $k \geq 1$, a Boolean function is defined to be *k-correlation immune* if for all $1 \leq d \leq k$, all degree-d Fourier coefficients of f are equal to 0. An equivalent definition is as follows (Xiao and Massey, 1988; Brynielsson, 1989). Let $x_1, \ldots, x_n$ be random Boolean variables, each chosen uniformly and independently. Let $y = f(x_1, \ldots, x_n)$. Then f is k-correlation immune if and only if, for any distinct variables $x_{i_1}, \ldots, x_{i_k}$ of f, the variables $y, x_{i_1}, x_{i_2}, \ldots, x_{i_k}$ are mutually independent.

A greedy decision tree learner would have difficulty learning k-correlation immune functions using only k-lookahead; to find relevant variables in the presence of irrelevant ones for such functions, it would need to use $k+1$-lookahead.

A Boolean function is *correlation immune* if it is 1-correlation immune. Equivalently, a Boolean function f is correlation immune if all variables of f have zero gain for f, with respect to the uniform distribution on $\{0,1\}^n$. As can be seen from Lemma 1, this is the case iff for every input variable x_i of the function, $\Pr[f = 1|x_i = 1] = \Pr[f = 1|x_i = 0]$, where probabilities are with respect to the uniform distribution on $\{0,1\}^n$. The following alternative characterization of correlation-immune functions immediately follows: A Boolean function is correlation-immune iff

$$|\{a \in \{0,1\}^n \mid f(a) = 1 \text{ and } a_i = 1\}| = |\{a \in \{0,1\}^n \mid f(a) = 1 \text{ and } a_i = 0\}|. \tag{13}$$

5.5 Correlation immune functions for product distributions

Correlation immune functions are defined with respect to the uniform distribution. Here we extend the definition to apply to arbitrary product distributions with parameters strictly between 0 and 1. In particular, for such a product distribution D, we can define a function to be *correlation immune for D* if either (1) The degree-1 Fourier coefficients with respect to D are all 0, or (2) the gain of every variable with respect to D is 0, or (3) $\Pr_D[f = 1|x_i = 1] - \Pr_D[f = 1|x_i = 0] = 0$ for all variables x_i of f. By the results in Section 5, these definitions are equivalent.[2]

A natural question is whether there are (non-constant) correlation immune functions for non-uniform product distributions D. There are, as illustrated by the following example, which can be easily generalized to other similar product distributions.

Example: Let n be a multiple of 3, and let D be the product distribution defined by $[2/3, 2/3, \ldots, 2/3]$.

For any n that is a multiple of 3, we will show that the following function f is correlation immune with respect to D.

Let f be the n-variable Boolean function such that $f(x) = 1$ if $x = 110110110110\ldots$ (i.e. $n/3$ repetitions of 110), or when x is equal to one of the two right-shifts of that vector. For all other x, $f(x) = 0$.

To prove correlation immunity, it suffices to show that for each x_i, $\Pr_D[f = 1|x_i = 1] = \Pr_D[f = 1]$.

Each positive example of f has the same probability. It is easy to verify that for each x_i, 2/3 of the positive examples have $x_i = 1$. Thus $\Pr_D[f = 1 \text{ and } x = 1] = 2/3 \Pr_D[f = 1]$. So,

$$\begin{aligned} \Pr_D[f = 1|x = 1] &= \Pr_D[f = 1 \text{ and } x = 1]/\Pr_D[x = 1] \\ &= (2/3\Pr_D[f = 1])/(2/3) \\ &= \Pr_D[f = 1] \qquad (14) \end{aligned}$$

□

In Section 8 we will give examples of product distributions D for which there are no correlation-immune functions.

6. Estimating first-order Fourier coefficients and gain

Fourier-based learning algorithms work by computing estimates of selected Fourier coefficients using a sample. Given a training set $S = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$ for a Boolean function f and $z \in \{0, 1\}^n$, the *estimated Fourier coefficient for z, calculated on S, with respect to product distribution D,* is

$$\hat{f}_{S,D}(z) := \frac{1}{m}\sum_{j=1}^{m}(1 - 2y^{(j)})\phi_z(x^{(j)}). \qquad (15)$$

To simplify notation, where D is clear from context, we will often write $\hat{f}_S(z)$ instead of $\hat{f}_{S,D}(z)$. Since ϕ_z depends on D, calculating $\hat{f}_S(z)$ from S requires knowledge of D. In our

[2] We do not extend the definition of correlation-immunity to non-product distributions. With respect to a non-product distribution, it is possible for both relevant and irrelevant variables to have non-zero gain.

work we only consider cases in which D is known. (In cases that D is an unknown product distribution, its parameters can be estimated (see Furst et al., 1991).)

If S is a random sample of f drawn with respect to D, then $\hat{f}_D(z) = E_D[(1-2f(x))\phi_z(x)]$ and $\hat{f}_{S,D}(z)$ is the estimate of the expectation $E_D[(1-2f(x))\phi_z(x)]$ on sample S.

In Section 9, there are situations in which we will know that, with respect to a known product distribution D, there exists a relevant variable of a function f whose first-order Fourier coefficient has magnitude at least q, for some value q. As mentioned earlier, the first-order Fourier coefficients of irrelevant variables are zero. Thus if one can estimate first-order Fourier coefficients of f so the estimates each have additive error less than $q/2$, then a non-empty subset of the relevant variables of f can be constructed by taking all variables whose Fourier coefficient estimates are at least $q/2$. The following lemma gives an upper bound on the sample size that would be needed to produce the desired estimates with high probability (set $\epsilon = q/2$). The lemma is implicit in the paper of Furst et al. (1991), and follows from a standard bound of Hoeffding.

Lemma 3 *Let f be an n-variable Boolean function. and let D be a product distribution over $\{0,1\}^n$ defined by $[p_1, \ldots, p_n]$. Let $\beta = \max_i\{1/p_i, 1/(1-p_i)\}$, $\epsilon > 0$, and $0 < \delta < 1$. If S is a set of*

$$\frac{1}{\epsilon^2} 2(\beta - 1) \ln \frac{2n}{\delta}$$

random examples of f, drawn from distribution D, then with probability at least $1 - \delta$, $|\hat{f}_S(x_i) - \hat{f}_D(x_i)| < \epsilon$ for all variables x_i of f.

Skewing works by estimating gain, rather than by estimating first-order Fourier coefficients. More generally, one can use gain estimates rather than Fourier coefficient estimates to try to identify relevant variables of a function (assuming some have non-zero gain). Below we give a sample-complexity bound, analogous to Lemma 3, for estimating gain. The technique is based on a standard large deviation estimate, which can be thought of as a "vector" version of the Chernoff bound.

Let $Z(0), Z(1), \ldots$ be a discrete-time Markov process in $\mathbb{R}^k$ with differences bounded by c. That is, $Z(0), Z(1), \ldots$ are random variables taking values in $\mathbb{R}^k$, such that the distribution of $Z(t+1)$ given $Z(u)$ for all $u \leq t$ depends only on $Z(t)$, and for each pair $Z(t), Z(t+1)$ the L_2 norm $||Z(t+1) - Z(t)||$ is at most c. We call the process a *martingale* if for all $t \geq 0$, $E[Z(t)]$ exists, and $E[Z(t+1)|Z(t)] = Z(t)$. (More general definitions exist, but this will suffice for our purpose.) A standard result says that martingales are unlikely to wander too far from their initial values.

Lemma 4 *Let $Z(t)$ be a martingale in R^k with differences bounded by c. Then for any $\lambda > 0$,*

$$\Pr[||Z(t) - Z(0)|| \geq \lambda] \leq 2\exp(\frac{-\lambda^2}{2tc^2}). \tag{16}$$

Proof See, e.g. Pinelis (1992). □

Lemma 5 *Let f be an n-variable Boolean function and let D be a product distribution over $\{0,1\}^n$ whose parameters are in $(0,1)$. Let $\epsilon > 0$, and $0 < \delta < 1$. If S is a set of*

$$256 \log(2n/\delta)/\epsilon^2$$

random examples of f, drawn from distribution D, then with probability at least $1-\delta$, $|G(S,x_i)-G_D(f,x_i)| \le \epsilon$ for all variables x_i of f.

Proof Let x_i be a variable, and consider the 2×2 table

	$f=0$	$f=1$
$x_i=0$	a_1	a_2
$x_i=1$	a_3	a_4

In this table, the a_j's are probabilities, so that a_1 denotes the probability (under D) that $x_i=f=0$, and similarly for the others. Therefore, $0 \le a_j \le 1$, and $\sum a_j = 1$.

By drawing a random sample S of f from distribution D, we get counts m_1, m_2, m_3, m_4 corresponding to the a_j's. For example, m_2 is the number of examples in S for which $x_i = 0$ and $f=1$. We can view the sampling procedure as happening over time, where the tth example is drawn at time t.

At times $t=0,1,2,\ldots$, we can observe

$$Z(t) := (m_1 - a_1 t, m_2 - a_2 t, m_3 - a_3 t, m_4 - a_4 t). \tag{17}$$

By the definition of Z,

$$\begin{aligned} E[Z(t+1)-Z(t)|Z(t)] &= a_1(1-a_1,-a_2,-a_3,-a_4)+a_2(-a_1,1-a_2,-a_3,-a_4) \\ &\quad +a_3(-a_1,-a_2,1-a_3,-a_4)+a_4(-a_1,-a_2,-a_3,1-a_4) \\ &= (0,0,0,0) \end{aligned} \tag{18}$$

where the last equation follows because $\sum a_j = 1$. Thus $Z(0), Z(1), \ldots$ is a martingale in $\mathbf{R}^4$. Also, $Z(t+1)-Z(t)$ equals, up to symmetry, $(1-a_1,-a_2,-a_3,-a_4)$. Since $a_2^2+a_3^2+a_4^2 \le 1$,

$$(1-a_1)^2 + a_2^2 + a_3^2 + a_4^2 \le 2, \tag{19}$$

and the martingale has differences bounded by $c = \sqrt{2}$.

The gain of x_i in f with respect to distribution D is

$$G_D(f,x_i) = 2\left[\beta(1-\beta) - p\beta_1(1-\beta_1) - (1-p)\beta_0(1-\beta_0)\right] \tag{20}$$

where

$$\beta = \Pr[f=1] = a_2 + a_4, \tag{21}$$

$$p = \Pr[x_i = 1] = a_3 + a_4, \tag{22}$$

$$\beta_0 = \Pr[f=1|x_i=0] = \frac{a_2}{a_1+a_2}, \tag{23}$$

and

$$\beta_1 = \Pr[f=1|x_i=1] = \frac{a_4}{a_3+a_4}. \tag{24}$$

Substituting these into the above gain formula and simplifying, we get

$$G_D(f,x_i) = 2\left[(a_1+a_3)(a_2+a_4) - \frac{a_3a_4}{a_3+a_4} - \frac{a_1a_2}{a_1+a_2}\right] \tag{25}$$

Define the function $G(a_1, \ldots, a_4)$ to be equal to the right hand side of the above equation. This is a continuous function of the a_j's, on the simplex $a_j \geq 0$, $\sum a_j = 1$.

Observe that

$$0 < \frac{\partial}{\partial a_j}\left(\frac{a_j a_k}{a_j + a_k}\right) = \frac{1}{(a_j/a_k + 1)^2} < 1, \tag{26}$$

if $a_j, a_k > 0$, and

$$0 \leq \frac{\partial}{\partial a_j}(a_1 + a_3)(a_2 + a_4) \leq \sum a_i = 1. \tag{27}$$

This implies that $|\partial G/\partial a_j| \leq 2$ in the interior of the simplex.

Suppose that (b_1, b_2, b_3, b_4) and (c_1, c_2, c_3, c_4) are two points on the interior of the simplex with $\max_j\{|c_j - b_j|\} = \mu$. Let $a(t) = b + t(c - b)$ be the the parametric equation of the line from b to c, and let $\tilde{G}(t) = G(a(t))$.

Letting $a_i(t)$ be the ith coordinate of $a(t)$, and applying the chain rule, we get that

$$\frac{\partial \tilde{G}}{\partial t} = \sum_i \frac{\partial \tilde{G}}{\partial a_i}\frac{da_i}{dt} \tag{28}$$

Since $\tilde{G}(0) = G(b)$ and $\tilde{G}(1) = G(c)$, by the mean value theorem, there exists $t^* \in [0, 1]$ such that

$$\frac{\partial \tilde{G}}{\partial t}(t^*) = G(c) - G(b). \tag{29}$$

For $(a_1, \ldots, a_4)$ in the interior of the simplex, $\left|\partial \tilde{G}/\partial a_i\right| \leq 2$. By the definition of $a(t)$, $|da_i/dt| = |c_i - b_i| \leq \mu$. Thus

$$\left|\frac{\partial \tilde{G}}{\partial t}(t^*)\right| = |G(c) - G(b)| \leq 8\mu. \tag{30}$$

Since G is continuous, this holds even for probability vectors b and c on the boundary.

We seek a sample size m large enough that (for all variables x_i)

$$\Pr[\ |G(S, x_i) - G_D(f, x_i)| \geq \epsilon\ ] \leq \frac{\delta}{n}. \tag{31}$$

Let the empirical frequencies be $\hat{a}_j = m_i/m$, $i = 1, \ldots, 4$. By (30), it will suffice to make m large enough that, with probability at least $1 - \delta/n$, we observe $|\hat{a}_j - a_j| < \epsilon/8$ for all j. Let's call a sample "bad" if for some j, $|m_j/m - a_j| \geq \epsilon/8$. Since $Z(0) = \vec{0}$, this implies that $||Z(m) - Z(0)|| \geq \epsilon m/8$. If we take $\lambda = \epsilon m/8$, $c = \sqrt{2}$, and $t = m$ in the Chernoff bound (Equation 16), we see that

$$\Pr[\text{ bad sample }] \leq 2e^{-\frac{\epsilon^2 m}{256}}. \tag{32}$$

This will be less than δ/n as soon as

$$m \geq \frac{256 \log(2n/\delta)}{\epsilon^2}. \tag{33}$$

□

We note that it is possible to obtain a weaker bound than the one in Lemma 5 by using standard Chernoff bound arguments on each of the "counts" m_i used in calculating the gain estimates.

7. A property of non-constant Boolean functions

For $k \in [0, \ldots, n]$, let $W_k(f)$ denote the number of positive assignments of f of Hamming weight k. We will make repeated use of the following lemma.

Lemma 6 *Let f be a non-constant Boolean function on $\{0,1\}^n$. Then there exists a variable x_i of f and a number $k \in [0, \ldots, n-1]$ such that $W_k(f_{x_i \leftarrow 0}) \neq W_k(f_{x_i \leftarrow 1})$.*

Proof. Assume no such variable x_i exists.

We begin by showing that for $i \in [1 \ldots n]$, there exists $k \in [0 \ldots n-1]$ such that $W_k(f_{x_i \leftarrow 1}) > 0$. By the definition of W_k, $W_k(f_{x_i \leftarrow 1}) > 0$ precisely when there exists some assignment $b \in \{0,1\}^n$ such that $f(b) = 1$, $b_i = 1$, and the Hamming weight of b is $k+1$. Since f is not a constant function, there exists $a \in \{0,1\}^n$ such that $f(a) = 1$. Recall that $w(a)$ denotes the Hamming weight of a. If $a_i = 1$, let $k = w(a) - 1$, else let $k = w(a)$. Since $W_k(f_{x_i \leftarrow 0}) = W_k(f_{x_i \leftarrow 1})$, and at least one of these is positive since $f(a) = 1$, it follows that $W_k(f_{x_i \leftarrow 1}) > 0$.

Thus we can define $m = min\{k \mid W_k(f_{x_i \leftarrow 1}) > 0 \text{ for some } i \in [1, \ldots, n]\}$. Let i^* be an index such that $W_m(f_{x_{i^*} \leftarrow 1}) > 0$. For example, suppose f is the function $f(x_1, x_2, x_3) = x_1 x_2 \vee x_3$. Then $f_{x_3 \leftarrow 1}(0,0) = 1$. Since the assignment $(0,0)$ has Hamming weight 0, which is the minimum possible, $m = 0$ and we can take $i^* = 3$.

There are two cases.

Case 1: $0 < m \leq n-1$. Since $W_m(f_{x_{i^*} \leftarrow 1}) > 0$, by assumption $W_m(f_{x_{i^*} \leftarrow 0}) > 0$ also. Thus there exists $t \in \{0,1\}^n$ such that $t_{i^*} = 0$, $f_{x_{i^*} \leftarrow 0}(t^{i^*}) = 1$, and $w(t^{i^*}) = m$. Since $m > 0$, there exists an index $j \neq i^*$ such that $t_j = 1$. Without loss of generality, assume $i^* < j$. Thus $t = (t_1, \ldots, t_{i^*-1}, 0, \ t_{i^*+1}, \ldots, t_{j-1}, 1, t_{j+1}, \ldots, t_n)$. Since $f_{x_{i^*} \leftarrow 0}(t^{i^*}) = 1$, $f(t) = 1$ also, and thus $f_{x_j \leftarrow 1}(t^j) = 1$. However, $w(t^j) = m - 1$ so $W_{m-1}(f_{x_j \leftarrow 1}) > 0$, which contradicts the definition of m.

Case 2: $m = 0$. We claim that for all $a \in \{0,1\}^n$, $f(a) = 1$.

The proof of the claim is by induction on the Hamming weight of a, $w(a)$. For the base case, let $w(a) = 0$. Thus a is the all 0's assignment. By the definition of m, $W_m(f_{x_{i^*} \leftarrow 1}) > 0$ and hence $W_m(f_{x_{i^*} \leftarrow 0}) > 0$ also. But since $m = 0$, the only assignment in $\{0,1\}^{n-1}$ of Hamming weight m is the all 0's assignment, which is equal to a^{i^*} since $w(a) = 0$. Thus $f_{x_{i^*} \leftarrow 0}(a^{i^*}) = 1$, and hence $f(a) = 1$ as claimed.

Now let $j \in [0 \ldots n-2]$. Assume inductively that all assignments x of Hamming weight j satisfy $f(x) = 1$. Let t be an assignment of Hamming weight $j+1$. Choose an index l such that $t_l = 1$; index l exists because $j + 1 > 0$. By the inductive assumption, for every assignment u such that $w(u) = j$, $f(u) = 1$. There are precisely $\binom{n-1}{j}$ assignments u such that $w(u) = j$ and $u_l = 0$. All these assignments u satisfy $f(u) = 1$, and thus $W_j(f_{x_l \leftarrow 0})$ $= \binom{n-1}{j}$. Since $W_j(f_{x_l \leftarrow 0}) = W_j(f_{x_l \leftarrow 1})$, $W_j(f_{x_l \leftarrow 1}) = \binom{n-1}{j}$ also. The quantity $\binom{n-1}{j}$ is equal to the total number of assignments in $\{0,1\}^{n-1}$ containing exactly j 1's. Clearly t^l is one of these assignments, hence $f_{x_l \leftarrow 1}(t^l) = 1$. Since $f(t) = f_{x_l \leftarrow 1}(t^l)$, $f(t) = 1$ also. Since t was an arbitrary assignment of Hamming weight $j+1$, we have that $f(a) = 1$ for all $a \in \{0,1\}^n$ of Hamming weight j+1. Hence we have shown by induction that $f(a) = 1$ for all $a \in \{0,1\}^n$. This contradicts the hypothesis that f is not a constant function, which completes the proof of the lemma. □

Lemma 6 can be restated using the terminology of *weight enumerators*. Given a binary code (i.e. a subset of $\{0,1\}^n$, for some n), the weight enumerator of this code is the polynomial $P(z) = \sum_k W_k z^k$. Lemma 6 states that if f is a non-constant Boolean function, then it has a relevant variable x_i such that codes $C_0 := \{x \in \{0,1\}^{n-1} | f_{x_i \leftarrow 1}(x) = 1\}$, and $C_1 := \{x \in \{0,1\}^{n-1} | f_{x_i \leftarrow 1}(x) = 1\}$ have different weight enumerators.

Lemma 6 proves the existence of a variable x_i with a given property. One might conjecture that all relevant variables of f would share this property, but this is not the case, as shown in the following simple example.

Example: Let $f(x_1, x_2, x_3) = (\neg x_1 \vee \neg x_2 \vee x_3)(x_1 \vee x_2 \vee \neg x_3)$. Let $\sigma = (0,0,0)$. Since $f(1,1,0) \neq f(0,1,0)$, x_1 is a relevant variable of f. It is straightforward to verify that, For $k \in \{0,1,2\}$, $W_k(f_{x_1 \leftarrow 0}) = W_k(f_{x_1 \leftarrow 1})$. The same holds for x_2 by symmetry. Variable x_3 is the only one satisfying the property of Lemma 6.

8. Skewing in an idealized setting

In this section, we analyze skewing in an idealized setting, where the available data consists of the truth table of a Boolean function. We then do an analysis of sequential skewing in the same setting.

8.1 A motivating example

Recall that a correlation immune function $f(x_1, \ldots, x_n)$ is one such that for every variable x_i, the gain of x_i with respect to f is 0 under the uniform distribution on $\{0,1\}^n$. We are interested in the following question: When skewing is applied to a correlation immune function, will it cause a relevant variable to have non-zero gain under the skewed distribution? (Equivalently, will it cause one of the first-order Fourier coefficients to become non-zero?) We show that, in the idealized setting, the answer to this question is "yes" for nearly all skews. The answer is somewhat different for sequential skewing.

When we use a skew (σ, p) to reweight a dataset that consists of an entire truth table, the weight assigned to each assignment a in the truth table by the skewing procedure is $P_{D(\sigma,p)}(a)$, where $D(\sigma, p)$ is the skewed distribution defined by (σ, p). Moreover, the gain of a variable x_i as measured on the weighted truth table is precisely the gain with respect to $D(\sigma, p)$. By Lemma 1, it follows that a variable x_i will have gain on the skewed (weighted) truth table dataset iff $P_D(f = 1|x_i = 1) - P_D(f = 1|x_i = 0) \neq 0$, where $D = D(\sigma, p)$. If x_i is a relevant variable, the difference $P_D(f = 1|x_i = 1) - P_D(f = 1|x_i = 0)$ can be expressed as a polynomial $h(p)$ in p of degree at most $r-1$, where r is the number of relevant variables of f. If x_i is an irrelevant variable, $P_D(f = 1|x_i = 1) - P_D(f = 1|x_i = 0) = 0$. The main work in this section will be to show that for some relevant variable x_i, this polynomial is not identically 0. Having proved that, we will know that for at most $r-1$ values of weight factor p (the roots of h), $h(p) = 0$. For all other values of p, $h(p) \neq 0$, and x_i has gain in f with respect to $D(\sigma, p)$.

We give an example construction of the polynomial $h(p)$ for a particular function and skew. Consider a Boolean function f over 5 variables whose positive examples are $(0,0,0,1,0)$, $(0,0,1,0,0)$, $(1,0,1,1,0)$. Assume a skew (σ, p) where $\sigma = (1, \ldots, 1)$ and p is some arbitrary value in $(0,1)$. Let $D = D_{(\sigma,p)}$. There are two positive examples of f

setting $x_1 = 0$, namely $(0,0,0,1,0),(0,0,1,0,0)$. It is easy to verify that $P_D(f = 1|x_1 = 0) = 2p(1-p)^3$. Similarly, $P_D(f = 1|x_1 = 1) = p^2(1-p)^2$. Let $h(p) = P_D(f = 1|x_1 = 1) - P_D(f = 1|x_1 = 0)$. Then $h(p) = p^2(1-p)^2 - 2p(1-p)^3$, which can be rewritten as a degree-4 polynomial in p. This polynomial has at most 4 roots, and it is not identically 0. It follows that for all but at most 4 choices of p, $h(p)$ is not zero. Thus if we choose p uniformly at random from $(0,1)$, with probability 1, x_1 has gain for (f,σ,p).

8.2 Analysis of skewing in an idealized setting

For $f : \{0,1\}^n \to \{0,1\}$ a Boolean function, $k \in [1 \dots n]$, and $\sigma \in \{0,1\}^n$, let $W(f,\sigma,k)$ denote the number of assignments $b \in \{0,1\}^n$ such that $f(b) = 1$ and $\Delta(b,\sigma) = k$.

Using the symmetry of the Boolean hypercube, we can generalize Lemma 6 to obtain the following lemma, which we will use in our analysis of skewing.

Lemma 7 *Let f be a non-constant Boolean function on $\{0,1\}^n$, $\sigma \in \{0,1\}^n$ be an orientation, and $i \in [1 \dots n]$. Then there exists a variable x_i of f and $k \in [0,\dots,n-1]$ such that $W(f_{x_i\leftarrow 1},\sigma^i,k) \neq W(f_{x_i\leftarrow 0},\sigma^i,k)$.*

Proof. Recall that given two assignments a and b, we use $a+b$ to denote componentwise addition mod 2. Let g be the isomorphism on $\{0,1\}^{n-1}$ such that $g(x) = x + \sigma^i$. Let $f' : \{0,1\}^n \to \{0,1\}$ be such that $f'(x) = f(g(x))$.

Applying Lemma 6 to function f', let x_i and k be such that $W_k(f'_{x_i\leftarrow 1}) \neq W_k(f'_{x_i\leftarrow 0})$.

For all $a \in \{0,1\}^{n-1}$, $f'_{x_i\leftarrow 1}(a) = 1$ and $w(a) = k$ iff $f_{x_i\leftarrow 1}(g(a)) = 1$ and $\Delta(g(a),\sigma) = w((a+\sigma)+\sigma) = k$. Since g is an isomorphism, it follows that $W_k(f'_{x_i\leftarrow 1}) = W(f_{x_i\leftarrow 1},\sigma^i,k)$. The analogous statement holds for $W_k(f'_{x_i\leftarrow 0})$. Thus $W(f_{x_i\leftarrow 1},\sigma^i,k) \neq W(f_{x_i\leftarrow 0},\sigma^i,k)$. □

We now show the connection between the above lemma and gain.

Lemma 8 *Let f be a Boolean function on $\{0,1\}^n$, $\sigma \in \{0,1\}^n$ be a fixed orientation, and $i \in [1 \dots n]$. Let r be the number of relevant variables of f. If $W(f_{x_i\leftarrow 1},\sigma^i,j) = W(f_{x_i\leftarrow 0},\sigma^i,j)$ for all $j \in [1 \dots n-1]$, then for all weighting factors $p \in (0,1)$, x_i does not have gain for (f,σ,p). Conversely, if $W(f_{x_i\leftarrow 1},\sigma^i,j) \neq W(f_{x_i\leftarrow 0},\sigma^i,j)$ for some $j \in [1 \dots n-1]$, then for all but at most $r-1$ weighting factors $p \in (0,1)$, x_i has gain for (f,σ,p).*

Proof. Let f_0 denote $f_{x_i\leftarrow 0}$ and f_1 denote $f_{x_i\leftarrow 1}$. Let $\sigma \in \{0,1\}^n$ be a fixed orientation.

For real valued variables y and z and for $a \in \{0,1\}^n$, let $T_{\sigma,a}(y,z)$ be the multiplicative term $y^{n-d}z^d$, where $d = \Delta(\sigma,a)$, the Hamming distance between σ and a. So, for example, if $\sigma = (1,1,1)$ and $a = (1,0,0)$, $T_{\sigma,a} = yz^2$. Note that for $p \in (0,1)$, $T_{\sigma,a}(p,1-p)$ is the probability assigned to a by distribution $D_{(\sigma,p)}$. For $\sigma \in \{0,1\}^n$ and f a Boolean function on $\{0,1\}^n$, let $g_{f,\sigma}$ be the polynomial in y and z such that

$$g_{f,\sigma}(y,z) = \sum_{a\in\{0,1\}^n : f(a)=1} T_{\sigma,a}(y,z) \tag{34}$$

Thus, for example, if f is the two-variable disjunction $f(x_1,x_2) = x_1 \vee x_2$, and $\sigma = (1,1)$, then $g_{f,\sigma} = y^1z^1 + y^1z^1 + y^2z^0 = y^2 + 2yz$.

Define $g'(y,z) = g_{f_1,\sigma^i}(y,z) - g_{f_0,\sigma^i}(y,z)$, where g is as given in Equation 34. The quantity $W(f,\sigma,k)$ is the value of the coefficient of the term $y^{n-k}z^k$ in $g_{f,\sigma}$. Thus $g'(y,z) = \sum_{j=0}^{n-1} c_j y^{n-1-j}z^j$, where for all $j \in [0 \dots n-1]$, $c_j = W(f_1,\sigma^i,j) - W(f_0,\sigma^i,j)$.

Let $p \in (0,1)$. Under distribution $D_{(\sigma,p)}$, $\Pr(f = 1|x_i = 0)$ and $\Pr(f = 1|x_i = 1)$ are equal to $g_{f_0,\sigma^i}(p, 1-p)$ and $g_{f_1,\sigma^i}(p, 1-p)$ respectively. Thus by Lemma 1, x_i has gain for (f, σ, p) iff $g'(p, 1-p) = 0$.

Let $h(p)$ be the polynomial in p such that $h(p) = g'(p, 1-p)$.

If x_i is irrelevant, then for all fixed $p \in (0,1)$, x_i has no gain for (f, σ, p). Further, $W(f_1, \sigma^i, j) = W(f_0, \sigma^i, j)$ for all $j \in [0 \ldots n-1]$. Thus the lemma holds if x_i is irrelevant. In what follows, assume x_i is relevant.

If $W(f_1, \sigma^i, j) = W(f_0, \sigma^i, j)$ for all $j \in [0 \ldots n-1]$, then $h(p)$ is identically 0 and for all fixed $p \in (0,1)$, x_i has no gain for (f, σ, p).

Suppose conversely that $W(f_1, \sigma^i, j) \neq W(f_0, \sigma^i, j)$ for some j. Then $g'(y, z)$ is not identically 0. We will show that $h(p) = g'(p, 1-p)$ is a polynomial of degree at most $r-1$ that is not identically 0.

We begin by showing that $h(p)$ has degree at most $r-1$. Let $x_l \neq x_i$ be an irrelevant variable of f. Assume without loss of generality that $\sigma_l = 1$. Then since $f(a_{x_l \leftarrow 1}) = 1$ iff $f(a_{x_l \leftarrow 0}) = 1$,

$$\begin{aligned}
g_{f,\sigma}(p, 1-p) &= \sum_{a \in \{0,1\}^n : f(a)=1, a_l=1} pT_{\sigma^l, a^l}(p, 1-p) + \sum_{a \in \{0,1\}^n : f(a)=1, a_l=0} (1-p)T_{\sigma^l, a^l}(p, 1-p) \\
&= \sum_{a \in \{0,1\}^n : f(a)=1, a_l=0} T_{\sigma^l, a^l}(p, 1-p) \\
&= \sum_{b \in \{0,1\}^{n-1} : f_{x_l \leftarrow 0}(b)=1} T_{\sigma^l, b}(p, 1-p) \\
&= g_{f_{x_l \leftarrow 0}, \sigma}(p, 1-p) \qquad (35) \\
& \qquad (36)
\end{aligned}$$

That is, $g_{f,\sigma}(p, 1-p)$ is equal to the corresponding polynomial for the function $g_{f_{x_l \leftarrow 0},\sigma}(p, 1-p)$ produced by hardwiring irrelevant variable x_l to 0. By repeating this argument, we get that $g_{f,\sigma} = g_{\tilde{f},\sigma}$ where $\tilde{f}$ is the function obtained from f by hardwiring all of its irrelevant variables to 0. Thus g has degree at most r and $h(p) = g'(p, 1-p)$ has degree at most $r-1$.

Let j' be the smallest j such that $W(f_1, \sigma^i, j) \neq W(f_0, \sigma^i, j)$. Then $a_{j'}$ is non-zero, and all (non-zero) terms of $g'(y, z)$ have the form $a_j y^{r-1-j} z^j$ where $j \geq j'$. We can thus factor out $z^{j'}$ from $g'(y, z)$ to get $g'(y, z) = z^{j'} g''(y, z)$, where $g''(y, z) = \sum_{j=j'}^{r-1} a_j y^{r-1-j} z^{j-j'}$. One term of g'' is $a_{j'} y^{r-1-j'}$, while all other terms have a non-zero power of z. Thus for $p = 1$, $g''(p, 1-p) = a_{j'}$ which is non-zero, proving that $g''(p, 1-p)$ is not identically 0. Hence $h(p) = z^{j'} g''(p, 1-p)$ is the product of two polynomials that are not identically 0, and so $h(p)$ is not identically 0.

Finally, since $h(p)$ is a polynomial of degree at most $r-1$ that is not identically 0, it has at most $r-1$ roots. It follows that there are at most $r-1$ values of p in $(0,1)$ such that x_i does not have gain for (f, σ, p). □

We now present the main theorem of this section.

Theorem 8.1 *Let f be a non-constant Boolean function on $\{0,1\}^n$. Let $\sigma \in \{0,1\}^n$ be an orientation, and let p be chosen uniformly at random from $(0,1)$. Then with probability 1 there exists at least one variable x_i such that x_i has gain for (f, σ, p).*

Proof. Let $\sigma \in \{0,1\}^n$ be a fixed orientation. Let r be the number of relevant variables of f. Let x_i be the variable of f whose existence is guaranteed by Lemma 7. Thus $W(f_{x_i \leftarrow 1}, \sigma^i, j) \neq W(f_{x_i \leftarrow 0}, \sigma^i, j)$ for some j. By Lemma 8, for all but at most $r-1$ weighting factors $p \in (0,1)$, x_i has gain for (f, σ, p). With probability 1, a random p chosen uniformly from $(0,1)$ will not be equal to one of those $r-1$ weighting factors. □

Using the techniques above, we can also show that for certain p-biased distributions $D[p]$, there do not exist any non-constant correlation immune functions with respect to $D[p]$. By Lemma 7 and the proof of Lemma 8, there is some variable x_i such that associated polynomial $h(p)$ (defined with respect to $\sigma = (1,\ldots,1)$) is not identically 0. It follows that for any p that is not a root of h, x_i has gain for $(f,(1,\ldots,1),p)$, and thus f is not correlation immune with respect to distribution $D[p]$. The polynomial $h(p)$ has integer coefficients with magnitude at most 2^r, which restricts its possible roots. For example, all roots of h must be algebraic, and thus for any non-algebraic p, f is not correlation immune with respect to $D[p]$.

With Theorem 8.1 we have shown that for any non-constant function and any orientation σ, there exists at least one variable x_i such that if p is chosen randomly, then, with probability 1, x_i has gain with respect to f under the distribution $D_{(\sigma,p)}$. However, the magnitude of the gain may vary depending on the function and on the skew. The identity of the variable(s) having gain can also depend on the skew. Moreover, there may be other relevant variables that don't have gain for any p. In the example given following the proof of Lemma 6, variables x_1 and x_2 will have no gain for $(f,(0,\ldots,0),p)$ no matter the choice of p.

Theorem 8.1 suggests that skewing is an effective method for finding relevant variables of a non-constant Boolean f, because for nearly all skews, there will be at least one variable with non-zero gain. Equivalently, for nearly all skewed distributions, function f is not correlation immune with respect to that distribution. However, in practice – even in a noiseless situation where examples are all labeled correctly according to a function f – we do not usually have access to the entire truth table, and thus are not able to compute the exact gain of a variable under distribution $D_{(\sigma,p)}$ defined by the skew. We can only estimate that gain. Moreover, in practice we cannot sample from the distribution $D_{(\sigma,p)}$. Instead, we simulate $D_{(\sigma,p)}$ by reweighting our sample.

8.3 Analysis of Sequential Skewing

Sequential skewing is a variant of skewing. In sequential skewing, $n+1$ iterations of reweighting are performed, where n is the number of input variables of the target function. On the j^{th} iteration, examples are reweighted according to the preferred setting of the j^{th} variable alone; if the setting of the j^{th} variable matches the preferred setting, the example is multiplied by p, otherwise the example is multiplied by $1-p$. The reweighting in the jth iteration is designed to simulate the product distribution in which each variable other than x_i is 1 with probability 1/2, and variable x_j has its preferred setting with probability p. As in standard skewing, the algorithm uses the calculated gains to determine which variable to output.

In the reweighting done by sequential skewing, there is a chosen variable x_i, a preferred setting $c \in \{0,1\}$ of that variable, and a weight factor $p \in (0,1)$. We thus define a (sequential) skew to be a triple (i, c, p), where $i \in [1 \ldots n]$, $c \in \{0, 1\}$, and $p \in (0, 1)$. Define the probability distribution $D_{(i,c,p)}$ on $\{0,1\}^n$ such that for $a \in \{0,1\}^n$, $D_{(i,c,p)}$ assigns probability $p \cdot (\frac{1}{2})^{n-1}$ to a if $a_i = c$, and $(1-p) \cdot (\frac{1}{2})^{n-1}$ otherwise. Thus $D_{(i,c,p)}$ is the distribution that would be generated by applying sequential skewing, with parameters x_i, c and p, to the entire truth table.

Let f be a Boolean function on $\{0,1\}^n$. We say that variable x_j *has gain for* (f, i, c, p) if under distribution $D_{(f,i,c,p)}$, $I(f|x_j) > 0$. By Lemma 1, x_j has gain for (f, i, c, p) iff under distribution $D_{(f,i,c,p)}$, $\Pr[f = 1|x_j = 1] \neq \Pr[f = 1|x_j = 0]$.

We will use the following lemma.

Lemma 9 *A Boolean function f is 2-correlation immune iff it is 1-correlation immune, and for all pairs $i < j$, the inputs x_i and x_j are independent given $f(x_1, \ldots, x_n)$.*

Proof. We first prove the forward direction. If f is 2-correlation immune, then it is certainly 1-correlation immune, and all triples (f, x_i, x_j) are mutually independent.

The reverse direction is a calculation. Let $\alpha, \beta, \gamma \in \{0, 1\}$. Using pairwise independence, and then 1-correlation immunity, we get

$$\begin{aligned}
\Pr[f = \alpha, x_i = \beta, x_j = \gamma] &= \Pr[f = \alpha] \Pr[x_i = \beta, x_j = \gamma \mid f = \alpha] \\
&= \Pr[f = \alpha] \Pr[x_i = \beta \mid f = \alpha] \Pr[x_j = \gamma \mid f = \alpha] \\
&= \Pr[f = \alpha] \Pr[x_i = \beta] \Pr[x_j = \gamma] \qquad (37)
\end{aligned}$$

This holds even if $\Pr[f = \alpha] = 0$, for then both sides vanish. □

The constant functions $f = 0$ and $f = 1$ are 2-correlation immune, as is any parity function on 3 or more variables. We have enumerated the 2-correlation immune functions up to $n = 5$ and found that when $n \leq 4$, the only such functions are as above, but for $n = 5$, others begin to appear. Specifically, there are 1058 2-correlation immune functions of 5 variables, but only 128 parity functions and complements of these (with no constraint on the relevant variables). (Our enumeration method works as follows. Vanishing of the relevant Fourier coefficients can be expressed as a linear system with 0-1 solutions, which we can count by a "splitting" process reminiscent of the time-space tradeoff for solving subset sum problems (Odlyzko, 1980).) Denisov (1992) gave an asymptotic formula for the number of 2-correlation immune functions, and from this work it follows that for large n, only a small fraction of the 2-correlation immune functions will be parity functions.

The following theorem shows that, in our idealized setting, sequential skewing can identify a relevant variable of a function, unless that function is 2-correlation immune. It follows that sequential skewing will be ineffective in finding relevant variables of a parity function, even with unlimited sample sizes. In contrast, standard skewing can identify relevant variables of a parity function if the sample size is large enough.

Theorem 8.2 *Let f be a correlation-immune Boolean function on $\{0,1\}^n$ and let $c \in \{0,1\}$. Let p be chosen uniformly at random from $(0, 1)$. If the function f is 2-correlation immune, then for all $j \in [1 \ldots n]$, x_j has no gain for (f, i, c, p). Conversely, if f is not 2-correlation immune, then for some $j \in [1 \ldots n]$, x_j has gain for (f, i, c, p) with probability 1.*

Proof. Let f be a correlation immune function. Let $i \in [1 \dots n]$ and $c \in \{0,1\}$.

Assume $c = 1$. The proof for $c = 0$ is symmetric and we omit it. Consider skew (i, c, p), where $p \in (0,1)$. Let $f^{-1}(1) = \{x \in \{0,1\}^* | f(x) = 1\}$.

Let $j \in [1 \dots n]$. Let $A_1 = |\{a \in f^{-1}(1) \mid a_i = c \text{ and } a_j = 1\}|$, and $B_1 = |\{a \in f^{-1}(1) \mid a_i \neq c \text{ and } a_j = 1\}|$. Similarly, let $A_0 = |\{a \in f^{-1}(1) \mid a_i = c \text{ and } a_j = 0\}|$, $B_0 = |\{a \in f^{-1}(1) \mid a_i \neq c \text{ and } a_j = 0\}|$.

Under distribution $D_{(f,i,c,p)}$, if $j \neq i$, $\Pr[f = 1 | x_j = 1] = (A_1 p + B_1(1-p)) \left(\frac{1}{2}\right)^{n-2}$. If $j = i$, then because $c = 1$, $\Pr[f = 1 | x_j = 1] = A_1 \left(\frac{1}{2}\right)^{n-1}$. Similarly, if $j \neq i$, $\Pr[f = 1 | x_j = 0] = (A_0 p + B_0(1-p)) \left(\frac{1}{2}\right)^{n-2}$. If $j = i$, $\Pr[f = 1 | x_j = 0] = B_0 \left(\frac{1}{2}\right)^{n-1}$.

The difference $\Pr[f = 1 | x_j = 1] - \Pr[f = 1 | x_j = 0]$ is a linear function in p. If $i \neq j$, this function is identically zero iff $A_1 = A_0$ and $B_1 = B_0$. If it is not identically 0, then there is at most one value of $p \in (0,1)$ for which it is 0. If $i = j$, this function is identically zero iff $A_1 = B_0$. Also note that for $i = j$, $A_0 = 0$ and $B_1 = 0$ by definition.

In addition, since f is correlation immune, $A_1 + A_0 = B_1 + B_0$. If $i = j$, then $\Pr[f = 1 | x_j = 1] - \Pr[f = 1 | x_j = 0]$ is therefore identically zero and x_i has no gain for (f, i, c, p). If $j \neq i$, then x_j has no gain for (f, i, c, p) iff $A_1 = A_0 = B_1 = B_0$. This latter condition is precisely the condition that $\Pr[x_i = \alpha \wedge x_j = \beta | f = \gamma] = \Pr[x_i = \alpha | f = \gamma] \Pr[x_j = \beta | f = \gamma]$ under the uniform distribution on $\{0,1\}^n$. If this condition holds for all pairs $i \neq j$, no variable x_j has gain for (f, i, c, p), and by Lemma 9, f is 2-correlation immune. Otherwise for some $i \neq j$, x_j has gain for (f, i, c, p) for all but at most 1 value of p. The theorem follows. □

9. Exploiting product distributions

Until now we have *simulated* alternative product distributions through skewing. But simulating alternative distributions is not the same as sampling directly from them. In particular, skewing can magnify idiosyncracies in the sample in a way that does not occur when sampling from true alternative distributions. Therefore we now consider the PDC model, in which the learning algorithm can specify product distributions and request random examples from those distributions. In practice it might be possible to obtain examples from such alternative distributions by working with a different population or varying an experimental set-up. Intuitively, one might expect a high degree of overhead in making such changes, in which case it would be desirable to keep the number of alternative distributions small.

9.1 FindRel1: Finding a relevant variable using r distributions

Let Boolean$_{r,n}$ denote the Boolean functions on n variables that have at most r relevant variables. We first present a simple algorithm that we call FindRel1, based on Theorem 8.1. It identifies a relevant variable of any target function in Boolean$_{r,n}$, with probability $1 - \delta$, by estimating the first-order Fourier coefficient of x_i for r distinct product distributions. The algorithm assumes that r is known. If not, standard techniques can be used to compensate. For example, one can repeat the algorithm with increasing values of r (perhaps using doubling), until a variable is identified as being relevant.

The algorithm works as follows. For $j \in \{1, \dots, r\}$, let D_j denote the product distribution that sets each of the n input variables to 1 with probability $j/(r+1)$. For each D_j, the algorithm requests a sample S_j of size m_0 (we will specify m_0 in the proof below). Then, for each of the n input variables x_i, it estimates the associated first-order Fourier coefficients from sample S_j. At the end, the algorithm outputs the set of all variables x_i whose gain on some S_j exceeded a threshold θ_0 (also specified below).

Theorem 9.1 *For all non-constant $f \in Boolean_{r,n}$, with probability at least $1-\delta$ FindRel1 will output a non-empty subset of the relevant variables of f. FindRel1 uses a total of $O((r+1)^{2r} \ln \frac{2nr}{\delta})$ examples, drawn from r distinct p-biased distributions. The running time of FindRel1 is polynomial in $2^{O(r \ln r)}$, n, and $\ln \frac{1}{\delta}$.*

Proof. Since $f \in \mathrm{Boolean}_{r,n}$, f has at least one relevant variable. Recall that for distribution D on $\{0,1\}^n$, $G_D(f, x_i)$ denotes the gain of x_i with respect to f under distribution D. Recall also that $D[p]$ denotes the product distribution that sets each variable x_i to 1 with probability p.

By the arguments in Section 8, for each relevant variable x_i, $\Pr_{D[p]}[f=1|x_i=1] - \Pr_{D[p]}[f=1|x_i=0]$ can be written as a polynomial of degree $r-1$ in p. Call this polynomial $h_i(p)$. The degree of $h_i(p)$ depends on r, rather than n, because D is a product distribution and hence the conditional probabilities $\Pr[f=1|x_i=1]$ and $\Pr[f=1|x_i=0]$ depend only on the settings of the relevant variables. For all irrelevant variables x_i of f, $h_i(p)$ is identically 0.

Now let x_i be a relevant variable such that $h_i(p)$ is not identically 0. By Theorem 8.1, f has at least one such relevant variable. The polynomial $h_i(p)$ has degree at most $r-1$ and hence has at most $r-1$ roots. Therefore, for at least one $j \in \{1,\ldots,r\}$, $h_i(j/(r+1)) \neq 0$.

Let $j^* \in \{1,\ldots,r\}$ be such that $h_i(j^*/(r+1)) \neq 0$. Since h_i has integer coefficients and is of degree at most $r-1$, it follows that $h_i(j^*/(r+1)) = b/(r+1)^{r-1}$, for some integer b. Thus the absolute value of $h_i(j^*/(r+1))$ is at least $1/(r+1)^{r-1}$, and by Lemma 2, the first-order Fourier coefficient (for distribution D_{j^*}) associated with x_i has magnitude at least $\frac{2\sqrt{\frac{j^*}{(r+1)}(1-\frac{j^*}{r+1})}}{(r+1)^{(r-1)}}$, which is lower bounded by $q := \frac{2\sqrt{\frac{1}{(r+1)}(1-\frac{1}{r+1})}}{(r+1)^{(r-1)}}$, Set θ_0 in the description of FindRel1 to be $q/2$.

For any single D_j, it follows from Lemma 3 that for some $m_0 = O((r+1)^{2r-1} \ln \frac{2nr}{\delta})$, if we use a sample of size m_0 drawn from D_j and estimate all n first-order Fourier coefficients for distribution D_j using that sample, then with probability at least $1 - \frac{\delta}{r}$, each of the estimates will have additive error less than $q/2$. Thus with probability at least $1-\delta$, this will hold for all r of the D_j. The total number of examples drawn by FindRel1 is rm_0.

Since for some relevant variable, the associated Fourier coefficient is at least q for some D_j, and for all irrelevant variables, the associated Fourier coefficient is 0 for all D_j, the theorem follows. □

Skewing uses gain estimates, rather than estimates of the first-order Fourier coefficients. FindRel1 can be modified to use gain estimates. By a similar argument as above, it follows from Lemma 1 that for distribution D_{j^*}, some relevant variable has gain at least $q' = 2\frac{1}{r+1}(1-\frac{1}{r+1})(\frac{1}{r+1})^{2r-2}$ with respect to that distribution. We could thus modify FindRel1 to output the variables whose gain exceeds $q'/2$. Then Lemma 5 implies that a sample of size $m_0 = O(r^{4r-2} \ln \frac{nr}{\delta})$ would suffice for the modified FindRel1 to output a non-empty subset of relevant variables. This sample complexity bound is higher than the bound for the original FindRel1 based on Fourier coefficients.

9.2 FindRel2: Lowering the sample complexity

We now present our second algorithm, FindRel2. As discussed in the introduction, it has an advantage over FindRel1 in terms of running time and sample complexity, but requires examples from a larger number of distinct distributions. FindRel2 is based on the following lemma.

Lemma 10 *Let f have $r \geq 1$ relevant variables. Suppose p is chosen uniformly at random from $(0,1)$. Then there exists a relevant variable x_i of f, and a value $\tau = \exp(-2r + o(r))$ such that with probability at least $\tau/2$ (with respect to the choice of p), $G_{D[p]}(f, x_i) \geq \tau/2$.*

Proof By Theorem 8.1 and its proof, there exists a variable x_i of f such that $\Pr_{D[p]}[f = 1|x_i = 1] - \Pr_{D[p]}[f = 0|x_i = 0]$ can be expressed as a polynomial $h_i(p)$, which has integer coefficients and is not identically 0. Let $g(p) = G_{D[p]}(f, x_i)$. By Lemma 1,

$$g(p) = 2p(1-p)h_i(p)^2. \tag{38}$$

Then there are integers $\gamma_0, \ldots, \gamma_{2r}$ such that $g(p) = \sum_{j=0}^{2r} \gamma_j p^j$. Since $g(p)$ is non-negative but not identically 0, we can define

$$\tau := \int_0^1 g(p)dp = \sum_{j=0}^{2r} \frac{\gamma_j}{j+1} > 0. \tag{39}$$

This is at least $1/L$, where L is the least common multiple of $\{1, \ldots, 2r+1\}$. Observe that for each prime, the number of times it appears in the prime factorization of L equals the number of its powers that are $\leq 2r+1$. Therefore, by the prime number theorem (see e.g. Ivić, 2003), we have, for some $c > 0$,

$$\log L = \sum_{\substack{p^k \leq 2r+1 \\ k \geq 1}} \log p = 2r + O(re^{-c\lambda(r)}), \tag{40}$$

where $\lambda(r) = (\log r)^{3/5}(\log\log r)^{-1/5}$. Thus $\tau = \exp(-2r + o(r))$. Now let α be the fraction of $p \in (0,1)$ for which $g(p) \geq \tau/2$. Then,

$$\tau = \int_{g \geq \tau/2} g + \int_{g < \tau/2} g \leq \alpha + (\tau/2)(1-\alpha). \tag{41}$$

This implies $\alpha \geq \tau/(2-\tau) > \tau/2$, and the lemma follows. □

Note that the proof of the above lemma relies crucially on the non-negativity of the gain function, and thus the same proof technique could not be applied to first-order Fourier coefficients, which can be negative.

It is possible that the bounds in the above result could be improved by exploiting how g comes from the Boolean function f. Without such information, however, the bounds are essentially the best possible. Indeed, by properly choosing g, one can use this idea to estimate the density of primes from below, and get within a constant factor of the prime number theorem. See Montgomery (1994) for a discussion of this point.

FindRel2, our second algorithm for finding a relevant variable, follows easily from the above lemma. We describe the algorithm in terms of two parameters m_1 and m_2. The algorithm begins by choosing m_1 values for p, uniformly at random from $(\tau/8, \tau/8)$ (where τ is as in Lemma 10). Let P be the set of chosen values.

For each value $p \in P$, the algorithm requests m_2 random examples drawn with respect to distribution $D[p]$, forming a sample S_p. Then, for each of the n input variables x_i, it computes $G(S_p, x_i)$, the gain of x_i on the sample S_p. At the end, the algorithm outputs all variables x_i such that $G(S_p, x_i) > \theta_1$ for at least one of the generated samples S_p (we will specify θ_1 below).

Using Lemma 10, we can give values to parameters m_1, m_2, and θ_1 in FindRel2 and prove the following theorem.

Theorem 9.2 *For all non-constant $f \in Boolean_{r,n}$, with probability at least $1-\delta$ FindRel2 will output a non-empty subset of the relevant variables of f. FindRel2 uses a total of $O(r^{28r} \log \frac{n}{\delta})$ examples, drawn from $O(e^{4r} \log \frac{1}{\delta})$ product distributions. The running time is polynomial in $2^{O(r)}$, n, and $\log \frac{1}{\delta}$.*

Proof. As in the proof of Theorem 9.1, f has at least one relevant variable x_i for which $h_i(p)$ is not identically 0. Let x_{i^*} denote this variable.

Let $\tau = \exp(-2r(1+O(\log r)^{-1})$ be the value referred to in the statement of Lemma 10. Thus $\frac{1}{\tau} = O(e^{4r})$. By Lemma 10, for at least a $\tau/2$ fraction of the values of $p \in (0,1)$, $h_{i^*}(p) \geq \tau/2$. Thus for more than a $\tau/4$ fraction of the values of $p \in (\tau/8, 1-\tau/8)$, $h_{i^*}(p) \geq \tau/2$. Let us call these "good" values of p. If a single p is chosen uniformly at random from $(\tau/8, 1-\tau/8)$, then the probability p is good is greater than $\tau/4$.

Let $0 < \delta_1 < 1$. If the algorithm chooses $m_1 = \frac{4}{\tau} \ln \frac{1}{\delta_1}$ independent random values of p to form the set P, the probability that P does not contain any good p's is at most $(1-\tau/4)^{m_1} < e^{-m_1\tau/4} = \delta_1$.

Suppose P is such that it does contain at least one good p. Let p^* be such a p. Let $\gamma = G_{D[p^*]}(f, x_{i^*})$. Then by Lemma 1, $\gamma \geq 2p^*(1-p^*)(\tau/2)^2 \geq \frac{\tau^3}{32} - \frac{\tau^4}{256}$, since p^* is good and $p^* \in (\tau/8, 1-\tau/8)$. Set θ_1 in the algorithm to be equal to $\gamma/2$.

Let $0 < \delta_2 < 1$. Set m_2 in the algorithm to be equal to $256 \ln(2nm_1/\delta_2)/\gamma^2$.

Consider any $p \in P$. Then by Lemma 5, with probability at least $1-\delta_2/m_1$, $|G(S_p, x_i) - G_{D[p]}(x_i)| < \gamma/2$ for all variables x_i. Since $|P| = m_1$, it follows that $|G(S_p, x_i) - G_{D[p]}(x_i)| < \gamma/2$ holds for all variables x_i and for all $p \in P$, with probability at least $1-\delta_2$.

Assuming P has at least one good p^*, $G_{D[p^*]}(x_{i^*}) \geq \gamma$ and $G_{D[p]}(x_i) = 0$ for all p and all irrelevant x_i. Thus if $|G(S_p, x_i) - G_{D[p]}(x_i)| < \gamma/2$ holds for every x_i and $p \in P$, and P contains at least one good p, then FindRel2 outputs a non-empty subset of relevant variables of f.

It follows that the the probability that the algorithm does not output a non-empty subset of the relevant variables is at most $\delta_1 + \delta_2$. Setting δ_1 and δ_2 to both equal $\delta/2$, the lemma follows. □

9.3 Two algorithms from the literature

Another approach to finding a relevant variable is implicit in work of Bshouty and Feldman (2002). We present it briefly here.

Bshouty and Feldman's approach is based on the following facts. Variable x_i is relevant to f iff there is some Fourier coefficient $\hat{f}(z)$ with $z_i = 1$ and $\hat{f}(z) \neq 0$. Further, if f has r relevant variables, the absolute value of every non-zero Fourier coefficient of f is at least $1/2^r$.

For $b \in \{0,1\}^{n-1}$, let $1b$ denote the concatenation of 1 with b. Let $w(b)$ denote the Hamming weight of b. Define $R_1(f) = \sum_{b \in \{0,1\}^{n-1}} \hat{f}^2(1b)(\frac{1}{2^{2w(b)}})$. Thus R_1 is a weighted sum of the Fourier coefficients $\hat{f}(z)$ such that $z_1 = 1$. For any $z \in \{0,1\}^n$, the quantity $\hat{f}^2(z)$ is non-zero only if $\{i|z_i = 1\} \subseteq \{i|$ variable x_i is a relevant variable of $f\}$. Therefore, if $\hat{f}^2(1b) \neq 0$, then $w(b) \leq r$. It follows that if x_1 is relevant, $R_1 > 1/2^{4r}$. If x_i is irrelevant, $R_1 = 0$. Let D' be the product distribution specified by the parameter vector $[1/2, 1/4, 1/4, \ldots, 1/4]$ and let $w \in \{0,1\}^n$ be such that $w = [1, 0, \ldots, 0]$. As shown by Bshouty and Feldman Bshouty and Feldman (2002, proof of Lemma 11), $R_1 = E_{x \sim U}[E_{y \sim D'}[f(y)\chi_w(x \oplus y)]]^2$. Here $x \sim U$ denotes that the first expectation is with respect to an x drawn uniformly with U, and $y \sim D'$ denotes that the second expectation is with respect to a y drawn from distribution D'. For any fixed x, $E_{y \sim D'}[f(y)\chi_w(x \oplus y)]]$, can be estimated by drawing random samples $(y, f(y))$ from D'. The quantity S_1 can thus be estimated by uniformly generating values for x, estimating $E_{y \sim D'}[f(y)\chi_w(x \oplus y)]]$ for each x, and then taking the average over all generated values of x. Using arguments of Bshouty and Feldman, which are based on a standard Hoeffding bound, it can be shown that for some constant c_1, a sample of size $O(2^{c_1 r} \log^2(\frac{1}{\delta'}))$ from D' suffices to estimate S_1 to within an additive error of $\frac{1}{2^{4r-1}}$, with probability $1 - \delta'$. If the estimate obtained is within this error, then whether x_i is relevant can be determined by just checking whether the estimate is greater than $\frac{1}{2^{4r-1}}$. We can apply this procedure to all n variables x_i, each time taking a sample of y's from a new distribution. Setting $\delta' = \delta/n$, it follows that a sample of size $O(n 2^{c_1 r} \log^2 \frac{n}{\delta})$ suffices to determine, with probability $1 - \delta$, which of the n variables are relevant.

The above algorithm uses examples chosen from n product distributions. Each product distribution has exactly one parameter set to $1/2$, and all other parameters set to a fixed value $\rho \neq 1/2$ (here $\rho = 1/4$, although this choice was arbitrary).

If the parameters of the product distribution can be set to 0 and 1, membership queries can be simulated. We now briefly describe an algorithm that uses membership queries and uniform random examples to find a relevant variable of a target function with at most r relevant variables. A similar approach is used in a number of algorithms for related problems (see, e.g. Arpe and Reischuk, 2006; Guijarro et al., 1999; Blum et al., 1995; Damaschke, 2000; Bshouty and Hellerstein, 1998).

The algorithm first finds the value of $f(a)$ for some arbitrary a, either by asking a membership query or choosing a random example. Then, the algorithm draws a random sample S of size $2^r \ln \frac{1}{\delta}$. Assuming the function contains at least one relevant variable, a random example has probability at least $1/2^r$ of being negative, and probability at least $1/2^r$ of being positive. Thus if the function has at least 1 relevant variable, with probability at least $1 - \delta$, S contains an example a' such that $f(a') \neq f(a)$. (If it contains no such example, the algorithm outputs the constant function $f(x) = f(a)$.) The algorithm then takes a and a', and using membership queries, executes a standard binary-search procedure for finding a relevant variable of a Boolean function, given a positive and a negative example of that function (cf. Blum et al., 1995, Lemma 4). This procedure makes $O(\log n)$ membership queries.

If we carry out the membership queries in the PDC model by asking for examples from product distribution distributions with parameters 0 and 1, the result is an algorithm that finds a relevant variable with probability at least $1-\delta$ using $O(\log n)$ product distributions and $O(2^r \log \frac{1}{\delta})$ random examples. The random examples can also be replaced by membership queries on (n, r) universal sets (see e.g. Bshouty and Hellerstein, 1998).

10. On the limitations of skewing

One of the motivating problems for skewing was that of learning the parity of r of n variables. The results of Section 8 imply that skewing is effective for learning parity functions if the entire truth table is available as the training set. (Of course, if the entire truth table is available, there are much more straightforward ways of identifying relevant variables.) Equivalently, we can identify relevant variables if we are able to determine the exact gain of each variable with respect to skewed distributions. In practice, though, we need to estimate gain values based on a random sample. The random sample must be large enough so that we can still identify a relevant variable, even though the gain estimates for the variables will have some error. We now consider the following sample complexity question: how large a random sample is needed so that skewing can be used to identify a relevant variable of the parity function, with "high" probability? We would like to know how quickly this sample complexity grows as r and n grow.

As mentioned previously, it is a well-known open question whether it is possible to PAC-learn parity functions in polynomial time, using examples drawn from the uniform distribution, in the presence of random classification noise. Many noise-tolerant PAC learning algorithms can be re-cast as algorithms in a learning model called the *statistical query model*. There are lower bounds on learning parity of r of n variables in the statistical query model (Bshouty and Feldman, 2002; Blum et al., 1994). Skewing is not an algorithm in the statistical query model, so these lower bounds do not apply directly to skewing. However, skewing, like statistical query learning, is based on the estimation of statistics. We use the techniques employed in proving the statistical query lower bounds to prove sample complexity lower bounds for skewing.

It is difficult to analyze the behavior of skewing because the same sample is used and re-used for many gain calculations. This introduces dependences between the resulting gain estimates. Here we consider a modification of the standard skewing procedure, in which we pick a new, independent random sample each time we estimate the gain of a variable with respect to a skew (σ, p). We will call this modification "skewing with independent samples." Intuitively, since the motivation behind skewing is based on estimating statistical quantities, choosing a new sample to make each estimate should not hurt accuracy. (In experiments, we found that skewing with independent samples was slightly more effective in finding relevant variables than standard skewing.)

For simplicity, we will assume that the variable output by the skewing algorithm is the one that exceeds a fixed threshold the maximum number of times. However, as we discuss below, our lower bounds would also apply to implementations using other output criteria.

We prove a sample complexity lower bound for skewing with independent samples, when applied to target function that is the parity of n of r variables. The proof is based based on the fact that the skewing algorithm does not use all the information in the examples. Given a skew (σ, p), and an example $(x, f(x))$, the skewing algorithm weights this example according to $d = \Delta(x, \sigma)$, the Hamming distance between x and σ. The calculation of the gain for a variable x_i on the weighted dataset then depends only on $f(x)$, whether $x_i = \sigma_i$, and on d. These three pieces of information can be viewed as a "summary statistic" for $(x, f(x)$. The skewing algorithm uses only the summary statistics for the examples; it does not use any other information about the examples. We will argue that the summary statistics do not contain enough information to identify relevant variables of a parity function, unless the sample size is "large".

We begin by proving a technical lemma, using techniques of Bshouty and Feldman (2002) and Blum et al. (1994).

In this section, all probabilities are with respect to the uniform distribution.

Let $i \in \{1, \ldots, n\}$. Let $\text{Parity}_{r,n}$ be the set of parity functions on n variables which have r relevant variables. So for each $f \in \text{Parity}_{r,n}$, $f(x_1, \ldots, x_n) = x_{i_1} + x_{i_2} + \ldots + x_{i_r}$ where the sum is taken mod 2, and the x_{i_j} are distinct.

Let $f \in \text{Parity}_{r,n}$. Let $F : \{0,1\}^n \to \{1, -1\}$ be the associated function $F = 1 - 2f$.

Let $d \in \{1, \ldots, n\}$. Let $b, c \in \{0, 1\}$. Let $S_1 = \Pr[x_i = b, \Delta(x, \sigma) = d,$ and $f(x) = c]$ when x_i is a relevant variable of f. Let $S_2 = \Pr[x_i = b, \Delta(x, \sigma) = d,$ and $f(x) = c]$ when x_i is an irrelevant variable of f. Thus S_1 is the probability of obtaining a certain summary statistic for variable x_i, when x_i is relevant, and S_2 is the probability of obtaining that statistic when x_i is irrelevant.

We prove an upper bound on $|S_1 - S_2|$.

Lemma 11 $|S_1 - S_2| \leq \frac{1}{2} min\{\frac{1}{\binom{n-1}{r}}, \frac{1}{\binom{n-1}{r-1}}\}$

Proof. Let $\gamma = |S_1 - S_2|$. Define a function $\psi_i(x, y) : \{0,1\}^n \times \{1, -1\} \to \{1, -1\}$ such that $\psi_i(x, y) = -1$ if $x_i = b$, $\Delta(x, \sigma) = d$ and $y = 1 - 2c$, and $\psi_i(x, y) = 1$ otherwise. Thus $\psi_i(x, F(x)) = -1$ iff $x_i = b$, $\Delta(x, \sigma) = d$, and $f(x) = c$.

For x_i a relevant variable of f, it is easy to verify that $E[\psi_i(x, F(x))] = 1 - 2S_1$. Similarly, for x_i an irrelevant variable of f, $E[\psi_i(x, F(x))] = 1 - 2S_2$.

Let x_j be a relevant variable, and x_k be an irrelevant variable.

Since $|S_1 - S_2| = \gamma$,

$$|E[\psi_j(x, F(x))] - E[\psi_k(x, F(x))]| = 2|S_2 - S_1| = 2\gamma. \tag{42}$$

For any function $\rho : \{0,1\}^n \times \{1, -1\} \to \{1, -1\}$ and any function $G : \{0,1\}^n \to \{1, -1\}$,

$$\begin{aligned} & E[\rho(x, G(x))] \\ = \;& E[\rho(x, -1)\frac{1 - G(x)}{2} + \rho(x, 1)\frac{1 + G(x)}{2}] \\ = \;& \frac{1}{2}E[\rho(x, 1)G(x)] - \frac{1}{2}E[\rho(x, -1)G(x)] + \frac{1}{2}E[\rho(x, 1)] + \frac{1}{2}E[\rho(x, -1)] \end{aligned} \tag{43}$$

Note that the last two terms are independent of G.

Setting $F = G$ and ρ to ψ_j and ψ_k, we get that

$$\begin{aligned} E[\psi_j(x, F(x)) - \psi_k(x, F(x))] &= \frac{1}{2}E[\psi_j(x,1)F(x)] - \frac{1}{2}E[\psi_j(x,-1)F(x)] \\ & \quad -\frac{1}{2}E[\psi_k(x,1)F(x)] + \frac{1}{2}E[\psi_k(x,-1)F(x)] \qquad (44) \end{aligned}$$

Note further that since ψ_j is 1 if its last argument is $-1 + 2c$ and F corresponds to a parity function,

$$E[\psi_j(x, -1 + 2c)F(x)] = E[\psi_k(x, -1 + 2c)F(x)] = E[F(x)] = 0. \qquad (45)$$

Thus

$$\begin{aligned} &|E[\psi_j(x, F(x)) - \psi_k(x, F(x))]| = \\ &\quad |-\frac{1}{2}E[\psi_j(x, 1 - 2c)F(x)] + \frac{1}{2}E[\psi_k(x, 1 - 2c)F(x)]| \qquad (46) \end{aligned}$$

Because $|E[\psi_j(x, F(x)) - \psi_k(x, F(x))]| = 2\gamma$, it follows that

$$|E[\psi_k(x, 1 - 2c)F(x)] - E[\psi_j(x, 1 - 2c)F(x)]| = 4\gamma \qquad (47)$$

Thus either

$$|E[\psi_j(x, 1 - 2c)F(x)]| \geq 2\gamma \qquad (48)$$

or

$$|E[\psi_k(x, 1 - 2c)F(x)]| \geq 2\gamma \qquad (49)$$

Assume the former. Let $g(x) = \psi_j(x, 1 - 2c)$. Note that $g(x) = -1$ iff $x_j = b$ and $\Delta(x, \sigma) = d$.

Consider the discrete Fourier series for g,

$$g(x) = \sum_{a \in \{0,1\}^n} \hat{g}(a)\chi_a(x) \qquad (50)$$

Let $a \in \{0, 1\}^n$ be such that for all $i \in \{1, \ldots, n\}$, $a_i = 1$ iff x_i is a relevant variable of f. Then $F = \chi_a$. By definition of the Fourier coefficient, $\hat{g}(a) = E[g(x)\chi_a(x)] = E[g(x)F(x)]$. Further, $E[g(x)F(x)] = E[\psi_j(x, 1 - 2c)F(x)]$ and thus $|\hat{g}(a)| \geq 2\gamma$.

Note that $g(x)$ has the same value for every function in $\text{Parity}_{r,n}$ having x_i as a relevant variable. The same is therefore true for $\hat{g}(a) = E[g(x)\chi_a(x)]$, for any $a \in \{0, 1\}^n$. We have thus proved that $|\hat{g}(a)| \geq 2\gamma$ for any a such that a is the indicator vector of a set of r variables containing x_j.

Since there are $\binom{n-1}{r-1}$ such vectors a, it follows that in the Fourier expansion of g, there are at least $\binom{n-1}{r-1}$ Fourier coefficients $\hat{g}(a)$ such that $|\hat{g}(a)| \geq 2\gamma$. Thus the sum of the squares of the Fourier coefficients of g is at least $4\gamma^2\binom{n-1}{r-1}$.

By Parseval's identity,

$$4\gamma^2\binom{n-1}{r-1} \leq 1. \tag{51}$$

It follows that

$$\gamma \leq \frac{1}{2\sqrt{\binom{n-1}{r-1}}}. \tag{52}$$

This bound on γ was based on the assumption that $|E[\psi_j(x,-1)F(x)]| \geq 2\gamma$. The other case is that $|E[\psi_k(x,-1)F(x)]| \geq 2\gamma$. The same line of argument shows that in this case

$$\gamma \leq \frac{1}{2\sqrt{\binom{n-1}{r}}}. \tag{53}$$

The lemma follows. □

We now prove a sample complexity lower bound for learning parity functions, using skewing with independent samples.

Theorem 10.1 *Let n, r be such that $\frac{1}{2}\binom{n}{r}^{1/3} > n$. Suppose we use skewing with independent samples to identify a relevant variable of f, where $f \in Parity_{r,n}$. Assuming that the samples are drawn from the uniform distribution, to successfully output a relevant variable with probability at least μ requires that the total number of examples used in making the gain estimates be at least $\frac{(\mu-\frac{r}{n})max\{\binom{n-1}{r-1},\binom{n-1}{r}\}}{2n}$.*

Proof. Consider running skewing with independent samples with a target function $f \in \text{Parity}_{r,n}$ satisfying the conditions of the theorem. Assume that all examples are drawn from the uniform distribution. To estimate the gain of a variable x_i with respect to a skew (σ, p), the skewing algorithm uses a sample drawn from the uniform distribution. In calculating this estimate, the algorithm uses only the following information about each labeled example $(x, f(x))$:

1. The value of x_i.

2. The value of $f(x)$.

3. The value $d = \Delta(x, \sigma)$.

For each labeled example $(x, f(x))$, we can therefore define a corresponding *summary example* $(x_i, f(x), d) \in \{0,1\}^2 \times \{0, \ldots, n\}$ containing the above information. (The value of d in the summary example is dependent on σ, but not on p.) Since skewing uses only the information in the summary examples, we may assume in our analysis that the skewing algorithm is, in fact, given only the summary examples $(x_i, f(x), d)$ for each skew (σ, p) and variable x_i, rather than the raw examples $(x, f(x))$.

The number of distinct possible summary examples is at most $4n$, since there are 2 possible values each for x_i and $f(x)$, and n possible values for d given x_i. The uniform distribution on examples x induces a distribution D on the summary examples $(x_i, f(x), d)$ generated for skew (σ, p) and variable x_i. For fixed σ, distribution D is the same for all relevant variables x_i of f. It is also the same for all irrelevant variables x_i of f. Let D_1^σ be the distribution for the relevant variables, and D_2^σ be the distribution for the irrelevant variables. Let q be the distance between D_1^σ and D_2^σ as measured in the L_1 norm. That is, if K denotes the set of possible summary examples $(x_i, f(x), d)$, then $q = \sum_{z \in K} |\Pr_{D_1^\sigma}[z] - \Pr_{D_2^\sigma}[z]|$.

Since there are at most $4n$ possible summary examples, it follows from Lemma 11 that $q \leq 2n\text{min}\{\frac{1}{\binom{n-1}{r}}, \frac{1}{\binom{n-1}{r-1}}\}$.

Let m be the total number of examples used to estimate the gain of all variables x_i under all skews (σ, p) used by the skewing algorithm. For each x_i, the summary examples used to estimate the gain of x_i with respect to (σ, p) are distributed according to D_1^σ if x_i is relevant, and D_2^σ if x_i is irrelevant. Let s be the L_1 distance between D_1^σ and D_2^σ. Thus $s \leq 2(n-1)\text{min}\{\frac{1}{\binom{n-1}{r}}, \frac{1}{\binom{n-1}{r-1}}\}$.

Since the L_1 distance between D_1^σ and D_2^σ is at most s for every skew (σ, p), it follows that during execution of the algorithm, with probability at least $(1-s)^m$, the summary examples generated for the relevant variables of f are distributed in the same way as the summary examples generated for the irrelevant variables of f. Since the variables were permuted before skewing began, it follows that with probability at least $(1-s)^m$, the final variable output by the skewing algorithm is equally likely to be any of the n input variables of f. Thus the probability that the skewing algorithm outputs an irrelevant variable is at least $(1-s)^m \frac{n-r}{n}$, and its probability that it outputs a relevant variable is at most $1 - (1-s)^m \frac{n-r}{n} < 1 - (1-sm)(1-\frac{r}{n}) < \frac{r}{n} + sm(1-\frac{r}{n}) < \frac{r}{n} + sm$. The first inequality in this sequence holds because $(1-s)^m < (1-sm)$, since $0 < s < 1$.

Since the upper bound of $\frac{r}{n} + sm$ on the success probability (in identifying a random variable) holds when the variables are initially permuted, it holds for the worst case $f \in \text{Parity}_{r,n}$, if the variables are not initially permuted. It follows that if skewing with independent samples outputs a relevant variable of f (for any $f \in \text{Parity}_{r,n}$) with probability at least μ, then the total number of examples used must be at least $\frac{\mu - \frac{r}{n}}{s}$. Since $s \leq 2n\text{min}\{\frac{1}{\binom{n-1}{r}}, \frac{1}{\binom{n-1}{r-1}}\}$, the lemma follows. □

To make the theorem concrete, consider the case where $r = \log n$. Note that if we simply choose one of the n variables at random, the probability of choosing a relevant variable in this case is $\frac{\log n}{n}$. It follows from the theorem that for skewing to output a relevant variable with success "noticably" greater than random guessing, that is, with probability at least $\frac{logn}{n} + \frac{1}{p(n)}$, for some polynomial p, it would need to use more than a superpolynomial number of examples.

The above proof relies crucially on the fact that skewing uses only the information in the summary examples. The details of how the summary examples are used is not important to the proof. Thus the lower bound applies not only to the implementation of skewing that we assumed (in which the chosen variable is the one whose gain exceeds the fixed threshold the maximum number of times). Assuming independent samples, the lower bound would also apply to other skewing implementations, including, for example, an implementation in which the variable with highest gain over all skews was chosen as the output variableo.

On the other hand, one can also imagine variants of skewing to which the proof would not apply. For example, suppose that we replaced the single parameter p used in skewing by a vector of parameters $[p_1, \ldots, p_n]$, so that in reweighting an example, variable x_i causes the weight to be multiplied by either p_i or $1 - p_i$, depending on whether there is a match with x_i's preferred setting. Our proof technique would not apply here, since we would be using information not present in the summary examples. To put it another way, the proof exploits the fact that the distributions used by skewing are simple ones, defined by a pair (σ, p). Interestingly, it was our focus on such simple distributions that led us to the two new algorithms in Section 9.

The above proof is based on proving an upper bound on s. Another way to prove such an upper bound is to use the fact that a statistical query algorithm could find out if a variable was relevant by requesting a bound on the proportion of examples corresponding to summary examples $(x_i, f(x), d)$, within tolerance $s/2$. Having found the at most r relevant variables, the underlying function could then be found by asking 2^r statistical queries with tolerance $1/2^r$, in order to find out the value of the underlying function on the relevant variables. If s were too large, this would contradict known lower bounds on statistical learning of parity. This approach would result in nearly the same theorem as above, but the proof is less direct. The advantage to this approach is that it that it yields lower bounds not just for parity, but for other functions with high statistical query dimension. (See e.g. Bshouty and Feldman (2002) for the definitions of the statistical query model and statistical query dimension.)

Empirical evidence suggests that parity is a particularly difficult function for skewing to handle; in experiments skewing was much more successful in identifying relevant variables of correlation immune functions other than parity (Page and Ray, 2003). The negative result above depends on the fact that for f a parity function with r relevant variables, $|S_1 - S_2|$ is $O(\frac{1}{\binom{n-1}{r}})$.

Skewing's empirical success in learning other correlation immune functions is likely due to the fact that such an upper bound does not hold for these functions. For example, consider $\text{Consensus}_{r,n}$, the set of all n-variable Boolean functions with r relevant variables, whose value is 1 iff the r relevant variables are all equal. The functions in this set are correlation immune. Assume $n + r$ is even. Let $d = (n + r)/2$ and $\sigma \in \{0,1\}^n$. Let $S_1 = \Pr[x_i = 1 - \sigma_i, \Delta(x, \sigma) = n/2, \text{ and } f(x) = 1]$ when x_i is a relevant variable of f. Let $S_2 = \Pr[x_i = 1 - \sigma_i, \Delta(x, \sigma) = d, \text{ and } f(x) = 1]$ when x_i is an irrelevant variable of f. Then $S_1 = \frac{1}{2^n}\binom{n-r}{\frac{n-r}{2}}$ and $S_2 = \frac{1}{2^n}(\binom{n-r-1}{\frac{n-r}{2}-1} + \binom{n-r-1}{\frac{n+r}{2}-1})$. Then $S_1 - S_2 = \Omega(\frac{1}{2^n}\binom{n-r}{\frac{n-r}{2}})$, since the first term of S_2 is equal to $S_1/2$, and the second term of S_1 is much smaller than the first. Since $\binom{m}{m/2} = \theta(\frac{2^m}{\sqrt{m}})$, $S_1 - S_2 = \Omega(\frac{1}{\sqrt{n-r}2^r})$. Even for r as large as $n/2$, this is $\Omega(\frac{1}{\sqrt{n}2^r})$. Note the difference between this quantity and the analogous upper bound for parity. Specifically, the dependence here is on $\frac{1}{2^r}$ rather than on $\frac{1}{n^r}$.

11. Conclusions and Open Questions

In this paper, we have studied methods of finding relevant variables that are based on exploiting product distributions.

We have provided a theoretical study of skewing, an approach to learning correlation immune functions (through finding relevant variables) that has been shown empirically to be quite successful. On the positive side, we have shown that when the skewing algorithm has access to the complete truth table of a target Boolean function—a case in which standard greedy gain-based learners fail—skewing will succeed in finding a relevant variable of that function. More particularly, under any random choice of skewing parameters, a single round of the skewing procedure will find a relevant variable with probability 1.

In some sense the correlation immune functions are the hardest Boolean functions to learn, and parity functions are among the hardest of these to learn, since a parity function of $k+1$ variables is k-correlation immune. In contrast to the positive result above, we have shown (using methods from statistical query learning) that skewing needs a sample size that is superpolynomial in n to learn parity of $\log n$ relevant variables, given examples from the uniform distribution.

We leave as an open question the characterization of the functions of $\log n$ variables that skewing can learn using a sample of size polynomial in n, given examples from the uniform distribution.

Skewing operates on a sample from a single distribution, and can only *simulate* alternative product distributions. We have used the PDC model to study how efficiently one can find relevant variables, given the ability to sample directly from alternative product distributions. We have presented two new algorithms in the PDC model for identifying a relevant variable of an n-variable Boolean function with r relevant variables. The first algorithm uses only r distinct p-biased distributions. It runs in time polynomial in n and the sample size, $O((r+1)^{2r} \ln \frac{2nr}{\delta})$. The second algorithm uses $O(e^{4r} \ln \frac{1}{\delta})$ p-biased distributions, and runs in time polynomial in n and the sample size, $O(e^{28r} \ln^2 \frac{n}{\delta})$. For $r = \log n$, only the second algorithm runs in time polynomial in n, but at the cost of using a number of distributions that is polynomial in n, instead of $\log n$. We have also briefly described two PDC algorithms that are implicit in the literature.

We leave as an open problem the development of PDC algorithms with improved bounds, and a fuller investigation of the tradeoffs between time and sample complexity, and the number and types of distributions used. As a first step, it would be interesting to show an algorithm whose time complexity is polynomial in n when $r = \log n$, using a number of p-biased distributions that is polynomial in $\log n$. As we have mentioned, it is a major open problem whether there is a polynomial-time algorithm for finding relevant variables of a function of $\log n$ variables, using only examples from the uniform distribution.

Acknowledgments

D. Page, S. Ray, and L. Hellerstein gratefully aknowledge support from the National Science Foundation (NSF IIS 0534908). E. Bach gratefully acknowledges support from the National Science Foundation (NSF CCF-0523680 and CCF-0635355) and from a Vilas Research Associate Award from the University of Wisconsin. Part of this work was performed while L. Hellerstein was visiting the U. of Wisconsin, Madison.

References

T. Akutsu, S. Miyano, and S. Kuhara. A simple greedy algorithm for finding functional relations: Efficient implementation and average case analysis. *Theor. Comput. Sci.*, 292 (2):481–495, 2003.

N. Alon. Derandomization via small sample spaces (abstract). In *SWAT '96: Proceedings of the 5th Scandinavian Workshop on Algorithm Theory*, pages 1–3, London, UK, 1996. Springer-Verlag.

J. Arpe and R. Reischuk. When does greedy learning of relevant features succeed? - a Fourier-based characterization. Technical Report TR06-065, Electronic Colloquium on Computational Complexity Report, May 2006.

E. Bach. Improved asymptotic formulas for counting correlation-immune Boolean functions. Technical Report 1616, University of Wisconsin–Madison, Computer Sciences Department", 2007.

A. Blum. Learning a function of r relevant variables. In *COLT/Kernel '03: Learning Theory and Kernel Machines, Proceedings of the 16th Annual Conference on Learning Theory and 7th Kernel Workshop*, Lecture Notes In Artificial Intelligence: Vol. 2777, pages 731–733. Springer Verlag, 2003.

A. Blum, M. Furst, J. Jackson, M. Kearns, Y. Mansour, and S. Rudich. Weakly learning DNF and characterizing statistical query learning using Fourier analysis. In *STOC '94: Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing*, pages 253–262, New York, NY, USA, 1994. ACM Press.

A. Blum, L. Hellerstein, and N. Littlestone. Learning in the presence of finitely or infinitely many irrelevant attributes. *J. Comput. Syst. Sci.*, 50(1):32–40, 1995.

L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees.* Wadsworth International Group, 1984.

L. Brynielsson. A short proof of the Xiao-Massey lemma. *IEEE Transactions on Information Theory*, 35(6):1344–1344, 1989.

N. H. Bshouty and V. Feldman. On using extended statistical queries to avoid membership queries. *J. Mach. Learn. Res.*, 2:359–395, 2002.

N. H. Bshouty and L. Hellerstein. Attribute-efficient learning in query and mistake-bound models. *J. Comput. Syst. Sci.*, 56(3):310–319, 1998.

P. Camion, C. Carlet, P. Charpin, and N. Sendrier. On correlation-immune functions. In *CRYPTO '91: Advances in Cryptology*, pages 86–100. Springer-Verlag, 1991.

P. Damaschke. Adaptive versus nonadaptive attribute-efficient learning. *Machine Learning*, 41(2):197–215, 2000.

O. V. Denisov. An asymptotic formula for the number of correlation-immune of order k Boolean functions. *Discrete Math. Appls.*, 2:407–426, 1992.

V. Feldman, P. Gopalan, S. Khot, and A. K. Ponnuswami. New results for learning noisy parities and halfspaces. In *FOCS '06: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 563–574, 2006.

D. Fukagawa and T. Akutsu. Performance analysis of a greedy algorithm for inferring Boolean functions. *Inf. Process. Lett.*, 93(1):7–12, 2005.

M. L. Furst, J. C. Jackson, and S. W. Smith. Improved learning of AC^0 functions. In *Proc. 4th Annual Workshop on Comput. Learning Theory (COLT)*, pages 317–325, 1991.

D. Guijarro, J. Tarui, and T. Tsukiji. Finding relevant variables in pac model with membership queries. In *Algorithmic Learning Theory, 10th International Conference, ALT '99, Tokyo, Japan, December 6-8, 1999, Proceedings*, 1999.

A. Ivić. *The Riemann Zeta-Function: Theory and Applications.* Dover, 2003.

E. Lantz, S. Ray, and D. Page. Learning bayesian network structure from correlation immune data. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence (UAI)*, 2007.

Y. Mansour. Learning Boolean functions via the Fourier transform. *Theoretical Advances in Neural Computation and Learning*, pages 391–424, 1994.

H. L. Montgomery. *Ten Lectures on the Interface Between Analytic Number Theory and Harmonic Analysis.* AMS, 1994.

E. Mossel, R. O'Donnell, and R. A. Servedio. Learning juntas. In *Proceedings of the 35th Annual Aymposium on the Theory of Computing*, pages 206–212, 2003.

A. M. Odlyzko. The rise and fall of knapsack cryptosystems. In *Cryptology and Computational Number Theory: Proceedings of Symposia in Applied Mathematics*, volume 42, pages 79–88. AMS, 1980.

D. Page and S. Ray. Skewing: An efficient alternative to lookahead for decision tree induction. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence.* Morgan Kaufmann, San Francisco, CA, 2003.

E. M. Palmer, R. C. Read, and R. W. Robinson. Balancing the n-cube: a census of colorings. *J. Algebraic Combin.*, 1:257–273, 1992.

I. Pinelis. An approach to inequalities for the distributions of infinite-dimensional martingales. In R. M. Dudley, M. G. Hahn, and J. Kuelbs, editors, *Probability in Banach Spaces*, pages 128–134. Birkhauser, 1992.

J.R. Quinlan. *C4.5: Programs for Machine Learning.* Kaufmann, 1997.

S. Ray and D. Page. Sequential skewing: An improved Skewing algorithm. In *Proceedings of the 21st International Conference on Machine Learning.* Morgan Kaufmann, San Francisco, CA, 2004.

Soumya Ray and David Page. Generalized skewing for functions with continuous and nominal variables. In *22nd International Conference on Machine Learning*, 2005. Submitted.

Bernard Rosell, Lisa Hellerstein, Soumya Ray, and David Page. Why skewing works: learning difficult functions with greedy tree learners. In *22nd International Conference on Machine Learning*, pages 728–735, 2005.

B. Roy. A brief outline of research on correlation immune functions. In *Proceedings of the 7th Australian Conference on Information Security and Privacy*, Lecture Notes In Computer Science: Vol. 2384, pages 379–384. Springer Verlag, 2002.

T. Siegenthaler. Correlation-immunity of nonlinear combining functions for cryptographic applications. *IEEE Transactions on Information Theory*, IT-30(5):776–780, 1984.

L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

G.-Z. Xiao and J. L. Massey. A spectral characterization of correlation-immune combining functions. *IEEE Transactions on Information Theory*, 34(3):569–570, 1988.