Why Skewing Works: Learning Difficult Boolean Functions with Greedy Tree Learners

Bernard Rosell BROSELL@ATT.COM
Lisa Hellerstein HSTEIN@CIS.POLY.EDU
Dept. of Computer and Information Science, Polytechnic University, 5 Metrotech Center, Brooklyn, NY 11201

Soumya Ray SRAY@CS.WISC.EDU
David Page PAGE@BIOSTAT.WISC.EDU
Dept. of Computer Sciences and Dept. of Biostatistics and Medical Informatics, University of Wisconsin, Madison, WI 53706

Abstract

We analyze *skewing*, an approach that has been empirically observed to enable greedy decision tree learners to learn "difficult" Boolean functions, such as parity, in the presence of irrelevant variables. We prove that, in an idealized setting, for any function and choice of skew parameters, skewing finds relevant variables with probability 1. We present experiments exploring how different parameter choices affect the success of skewing in empirical settings. Finally, we analyze a variant of skewing called Sequential Skewing.

1. Introduction

Some Boolean functions, such as parity, are difficult for greedy decision tree learners to learn. These "difficult functions" are such that, even given a complete dataset (the full truth table for the function), every variable has zero gain, according to such standard measures as Information Gain (Quinlan, 1997) or GINI Gain (Breiman et al., 1984). When variables irrelevant to the target are present (as they are in real data), and the target is a difficult function, decision tree learners often have trouble finding the relevant variables. The traditional approach to this problem is to use depth-k lookahead (Norton, 1989), but this approach takes time exponential in k.

Recently, an approach called *Skewing* has been proposed (Page & Ray, 2003), as an alternative to lookahead, to enable greedy decision tree learners to handle difficult Boolean functions efficiently. Skewing works by choosing a "preferred setting" for every variable x used in describing the examples, and a weight factor p where $\frac{1}{2} < p < 1$. Each example is reweighted by p if the value of x matches its preferred setting, and by $1 - p$ otherwise. The final weight of an example is the product over the weights for each variable, resulting in a "skew" of the initial distribution. The reweighting is repeated a small constant number of times, with different preferred settings chosen each time. The gain of every variable is computed after each reweighting, and the variable that shows high gain most often is selected as the split variable. The procedure is designed to cause variables relevant to the target function to be selected, even when the target is difficult and many irrelevant variables are present. Note that no prior knowledge of *which* variables are relevant to the target is required. Further, this procedure increases a standard tree learner's runtime by only a constant factor.

While empirical results demonstrate that skewing enables greedy tree learners to learn difficult functions, there is relatively little understanding of its behavior. In our work, we analyze skewing in an idealized setting, where a complete dataset is available, and prove the following result. Consider a complete dataset labeled according to any Boolean function. If the dataset is reweighted according to any choice of preferred settings and a random weight factor p, then with probability 1 some relevant variable will have non-zero gain on the reweighted set, while all irrelevant variables will have zero gain. Thus, given a complete dataset, skewing will lead to correct choices by a greedy tree learner.

In practice, training data consists of a random sample of examples, rather than a truth table. We provide

Appearing in *Proceedings of the 22nd International Conference on Machine Learning*, Bonn, Germany, 2005.

experiments exploring the behavior of skewing on random samples. First, we consider the case in which a random sample can be drawn from a chosen skewed distribution over the truth table. Second, we consider the case in which we are given a training sample drawn from the uniform distribution, and can only simulate a skewed distribution by reweighting this sample.

We also investigate "Sequential Skewing", a variant of the skewing algorithm (Ray & Page, 2004). Empirically, this algorithm was observed to outperform the original algorithm on randomly chosen difficult functions when large numbers of irrelevant variables were present. Nevertheless, we show that there exist some difficult functions where Sequential Skewing does not cause relevant variables to show gain, regardless of the choice of p. We characterize the difficult functions for which the variant works, in an idealized setting, and show that on those functions, when p is chosen randomly, it works with probability 1.

A related theory paper (Mossel et al., 2003) considered the general problem of finding relevant variables in the presence of many irrelevant ones. Its main result is an algorithm that applies to examples drawn from the uniform distribution; the algorithm is based on deep structural properties of Boolean functions. The paper also included a short proof of a result[1] similar to our main theorem, but for random product distributions, instead of the random skewed distributions treated in our theorem. Since random product distributions have different properties than random skewed distributions, their proof does not suffice for our setting.

2. Theoretical Analysis of Skewing

In this section, we analyze skewing in an idealized setting – when the available data consists of the truth table of a Boolean function. We begin by defining notation. Next, we show that the question of when a variable "has gain" with respect to a function and dataset can be viewed combinatorially. Finally, we prove our main results.

2.1. Notation and Definitions

The integers between 1 and n are denoted by $[1 \ldots n]$, while $(\frac{1}{2}, 1)$ denotes the open real interval from $\frac{1}{2}$ to 1. We consider two-class learning problems, where the features, or variables, are Boolean. Examples are truth assignments over variables, and targets are Boolean functions. Let $f(x_1, \ldots, x_n)$ be a Boolean function that maps $\{0,1\}^n$ to $\{0,1\}$. An *assignment* $a = (a_1, \ldots, a_n)$ to the variables $x_1, \ldots, x_n$ is an element of $\{0,1\}^n$. For $a \in \{0,1\}^n$ and $i \in [1 \ldots n]$, $a(i)$ denotes the ith bit of a and $a_{\neg x_i}$ denotes the assignment obtained from a by negating the ith bit of a. For $a, b \in \{0,1\}^n$, let $d(a,b) = |\{i \in [1, \ldots, n] | a(i) = b(i)\}|$, the number of bits a and b have in common.

A *truth table* for a function f over n variables is a list of all 2^n assignments over the variables, together with the value $f(a)$ for each assignment a. 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 $i \in [1 \ldots n]$ 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. That is, $f_{x_i \leftarrow b} : \{0,1\}^{n-1} \to \{0,1\}$ such that for all $a \in \{0,1\}^{n-1}$, $f_{x_i \leftarrow b}(a) = f(a_1, a_2, \ldots, a_{i-1}, b, a_i, a_{i+1}, \ldots, a_{n-1})$.

For any probability distribution D over $\{0,1\}^n$ and any $A \subseteq \{0,1\}^n$, we denote by $\Pr_D(A)$ the sum of the probabilities, under D, of assignments in A. Where the distribution D is clear from context, we write $\Pr(A)$.

A *skew* is a pair (σ, p) where $\sigma \in \{0,1\}^n$ is an assignment, and $p \in (\frac{1}{2}, 1)$. We refer to σ as the *orientation* of the skew, and p as the *weight factor*.

2.2. Characterization of Gain

Learners such as C4.5 and CART induce decision trees from a dataset. A dataset of examples from $\{0,1\}^n$ defines a probability distribution over $\{0,1\}^n$, in which the probability of $a \in \{0,1\}^n$ is the relative frequency of a in the dataset, i.e. (Number of occurrences of a in dataset)/(Number of examples in dataset). In this section, we view a dataset as being equivalent to the distribution it defines.

Greedy tree learners partition a dataset recursively, choosing a "split variable" at each step. They differ from one another primarily in their measures of "goodness" for split variables. One such measure is *Information Gain*, which we now review. For any Boolean function f, let $P = \{a \in \{0,1\}^n | f(a) = 1\}$ and $N = \{a \in \{0,1\}^n | f(a) = 0\}$. The *entropy* of f under a distribution D is $H_D(f) = (-\Pr_D(P)\log_2 \Pr_D(P) - \Pr_D(N)\log_2 \Pr_D(N))$. For any potential split variable x_i, the entropy conditional on x_i is the weighted sum of the entropies of the child nodes resulting from a split on x_i: $H_D(f|x_i) = (\Pr_D(x_i = 0)H_D(f_{x_i \leftarrow 0}) + \Pr_D(x_i = 1)H_D(f_{x_i \leftarrow 1}))$. Then the Information Gain of x_i for distribution D and function f is $I_D(f|x_i) = H_D(f) - H_D(f|x_i)$. Where D is understood from context, we denote Information Gain simply by $I(f|x_i)$.

The following lemma provides a characterization of when a variable has non-zero Information Gain un-

[1] The result as stated in their paper is not quite correct, but can be fixed fairly easily.

der distribution D for a function f; it can be shown that the same characterization holds for a variety of other commonly used gain functions.

Lemma 2.1 *Variable x_i has gain with respect to function f and distribution D (that is, $I_D(f|x_i) > 0$) if and only if* $\Pr_D(f = 1|x_i = 1) \neq \Pr_D(f = 1|x_i = 0)$.

Proof. $I(f|x_i) \geq 0$, with equality iff f and x_i are independent (cf. Cover & Thomas, 1991). We show that f and x_i are independent iff $\Pr_D(f = 1|x_i = 1) = \Pr_D(f = 1|x_i = 0)$. Suppose f and x_i are independent. Then $\Pr(f = 1) = \Pr(f = 1|x_i = 1)$ and $\Pr(f = 1) = \Pr(f = 1|x_i = 0)$, and hence $\Pr(f = 1|x_i = 1) = \Pr(f = 1|x_i = 0)$. Conversely, suppose $\Pr(f = 1|x_i = 1) = \Pr(f = 1|x_i = 0)$. Then $\Pr(f = 0|x_i = 1) = \Pr(f = 0|x_i = 0)$ also. For $b \in \{0, 1\}$, we can show that $\Pr(f = b) = \Pr(f = b|x_i = 0)$ as follows: $\Pr(f = b) = \Pr(x_i = 1)\Pr(f = b|x_i = 1) + \Pr(x_i = 0)\Pr(f = b|x_i = 0) = \Pr(x_i = 1)\Pr(f = b|x_i = 1) + \Pr(x_i = 0)\Pr(f = b|x_i = 1) = \Pr(f = b|x_i = 1)$. Similarly, $\Pr(f = b) = \Pr(f = b|x_i = 0)$. Therefore f and x_i are independent. □

In general, the value of the gain is not closely related to the value of the difference $\Pr(f = 1|x_i = 1) - \Pr(f = 1|x_i = 0)$, although one is zero iff the other is.

Let f be a Boolean function on $\{0, 1\}^n$. Let U be the uniform distribution on $\{0, 1\}^n$. We say that f is a *difficult* function if for each variable x_i of f, $I_U(f|x_i) = 0$. By Lemma 2.1, the condition $I_U(f|x_i)$ can be replaced by the combinatorial condition that $|\{a \in \{0, 1\}^n \mid f(a) = 1 \text{ and } a(x_i) = 1\}| = |\{a \in \{0, 1\}^n \mid f(a) = 1 \text{ and } a(x_i) = 0\}|$.

Each skew (σ, p) induces a probability distribution $D_{(\sigma,p)}$ on the 2^n assignments in $\{0, 1\}^n$ as follows. Let $\tau_p : \{0, 1\} \times \{0, 1\} \to \{p, 1 - p\}$ 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. 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 1) to a dataset 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.

2.3. Proof Idea

We are interested in the following question: When skewing is applied to a difficult function, will it cause a relevant variable to have non-zero gain under the skewed distribution? In the next section, we prove that the answer is "yes" for nearly all skews. In this section, we describe the key ideas behind our proof.

We consider skewing a dataset consisting of the full truth table for a Boolean function f. The goal of skewing is to distinguish relevant from irrelevant variables; a skew *works* when some relevant variable x_i has non-zero gain but the irrelevant ones have zero gain. A skew gives x_i non-zero gain iff the weighted fraction of positive assignments is different for $x_i = 0$ than for $x_i = 1$, under that skew (Lemma 2.1). The difference in these fractions can be expressed as a polynomial in the variable p, where p is the weight factor associated with the skew. If the value of this polynomial is not equal to 0, the variable x_i has non-zero gain under the skew. We demonstrate that for a fixed orientation, if x_i is relevant, then for almost all p, the value of the polynomial is not 0. If x_i is irrelevant, the polynomial is identically 0.

As an example of this polynomial, consider the Boolean function f on 5 variables whose positive assignments are $(0, 0, 0, 1, 0)$, $(0, 0, 1, 0, 0)$, $(0, 0, 1, 1, 0)$, and $(1, 0, 0, 0, 1)$. Consider the skew where the preferred setting of every variable is 0 (i.e. $\sigma = (0, 0, 0, 0, 0)$), and p is the weight factor. Then the probabilities (weights) of the positive assignments are $\Pr(0, 0, 0, 1, 0) = \Pr(0, 0, 1, 0, 0) = p^4(1 - p)$ and $\Pr(0, 0, 1, 1, 0) = p^3(1 - p)^2$. Therefore, $\Pr(f = 1|x_1 = 0) = \frac{2p^4(1-p)+p^3(1-p)^2}{p} = 2p^3(1 - p) + p^2(1 - p)^2$, and $\Pr(f = 1|x_1 = 1) = p^3(1 - p)$. The difference $\Pr(f = 1|x_1 = 1) - \Pr(f = 1|x_1 = 0)$ is thus a polynomial in p of degree 4, which has at most 4 roots. Therefore, if the value of p is chosen at random, then with probability 1, $\Pr(f = 1|x_1 = 1) - \Pr(f = 1|x_1 = 0) \neq 0$ and x_1 has non-zero gain under the skew.

Observe that in the polynomial for $\Pr(f = 1|x_1 = 0)$ (respectively, $\Pr(f = 1|x_1 = 1)$), the coefficient of each $p^j(1 - p)^{(n-1-j)}$ is the number of positive assignments where $x_1 = 0$ (respectively, $x_1 = 1$), and exactly j of the remaining variables have the preferred setting of 0. Thus the coefficients of the $p^j(1 - p)^{(n-1-j)}$ count certain positive assignments, a fact we exploit in our proof. These counts are exactly what the skewing approach is modifying in the reweighting process – intuitively, skewing tries to choose favored settings and weights so that the coefficents of the $p^j(1 - p)^{(n-1-j)}$ will not all be equal in the expressions for $\Pr(f = 1|x_1 = 0)$ and $\Pr(f = 1|x_1 = 1)$ respectively. Our proof shows that skewing almost always succeeds in doing this.

2.4. Main Result

We now present our main result. Let x and y be variables, and for $\sigma, a \in \{0,1\}^n$, let $T_{\sigma,a}(x,y)$ be the multiplicative term $x^{d(\sigma,a)}y^{n-d(\sigma,a)}$. So, for example, if $\sigma = (1,1,1)$ and $a = (1,0,0)$, $T_{\sigma,a} = xy^2$. For $p \in (\frac{1}{2},1)$, $T_{\sigma,a}(p,1-p)$ 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 x and y such that $g_{f,\sigma}(x,y) = \sum_{a\in\{0,1\}^n:f(a)=1} T_{\sigma,a}(x,y)$.

For $a \in \{0,1\}^n$, let $a^i = (a_1,\ldots,a_{i-1},a_{i+1},\ldots,a_n)$, that is, a^i denotes a with its ith bit removed. For $f : \{0,1\}^n \to \{0,1\}$ a Boolean function, $j \in [1\ldots n]$, and $\sigma \in \{0,1\}^n$, let $N(f,\sigma,j)$ denote the number of assignments $a \in \{0,1\}^n$ such that $f(a) = 1$ and a has j bits in common with σ. Note that $N(f,\sigma,j)$ is the value of the coefficient of the term x^jy^{n-j} in $g_{f,\sigma}$.

We show that the question of whether variables with non-zero gain exist can be viewed combinatorially.

Lemma 2.2 *Let f be a Boolean function on $\{0,1\}^n$, $\sigma \in \{0,1\}^n$ be a fixed orientation, and $i \in [1\ldots n]$. If $N(f_{x_i\leftarrow1},\sigma^i,j) = N(f_{x_i\leftarrow0},\sigma^i,j)$ for all $j \in [1\ldots n-1]$, then for all weight factors $p \in (\frac{1}{2},1)$, x_i does not have gain for (f,σ,p). Conversely, if $N(f_{x_i\leftarrow1},\sigma^i,j) \neq N(f_{x_i\leftarrow0},\sigma^i,j)$ for some $j \in [1\ldots n-1]$, then for all but at most $n-1$ weight factors p, x_i has gain for (f,σ,p).*

Proof. Let $\sigma \in \{0,1\}^n$ be a fixed orientation. Let f_0 denote $f_{x_i\leftarrow0}$ and f_1 denote $f_{x_i\leftarrow1}$. Define $g'(x,y) = g_{f_1,\sigma^i}(x,y) - g_{f_0,\sigma^i}(x,y)$. Then $g'(x,y) = \sum_{j=0}^{n-1} c_jx^jy^{n-1-j}$, where for all $j \in [0\ldots n-1]$, $c_j = N(f_1,\sigma^i,j) - N(f_0,\sigma^i,j)$.

Let $p \in (\frac{1}{2},1)$. 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 2.1, x_i has gain for (f,σ,p) iff $g'(p,1-p) \neq 0$.

If $N(f_1,\sigma^i,j) = N(f_0,\sigma^i,j)$ for all $j \in [0\ldots n-1]$, then $g'(x,y)$ is identically 0. So for all $p \in (\frac{1}{2},1)$, $g'(p,1-p) = 0$ and x_i has no gain for (f,σ,p).

If $N(f_1,\sigma^i,j) \neq N(f_0,\sigma^i,j)$ for some j, then $g'(x,y)$ is not identically 0. We show that $g'(p,1-p)$, as a function of p, is not identically 0 either. By multiplying out terms, $g'(p,1-p)$ can be written as a polynomial in p of degree at most $n-1$. Let j' be the largest j such that $N(f_1,\sigma^i,j) \neq N(f_0,\sigma^i,j)$. Then $c_{j'}$ is non-zero, and the non-zero terms of $g'(x,y)$ have the form $c_jx^jy^{n-1-j}$ where $j \leq j'$. Factoring out $y^{n-1-j'}$ from $g'(x,y)$, we get $g'(x,y) = y^{n-1-j'}g''(x,y)$, where $g''(x,y) = \sum_{j=0}^{j'} c_jx^jy^{j'-j}$. The last term of g'' is $c_{j'}x^{j'}$, and all other terms have a non-zero power of y. At $p = 1$, the polynomial $g''(p,1-p)$ is thus equal to $c_{j'}$, which is non-zero, proving that $g''(p,1-p)$ is not identically 0. Hence $g'(p,1-p) = (1-p)^{n-1-j'}g''(p,1-p)$ is the product of two polynomials that are not identically 0, from which it follows that $g'(p,1-p)$ is not identically 0. Since $g'(p,1-p)$ is a polynomial of degree at most $n-1$, it has at most $n-1$ roots. Thus there are at most $n-1$ values of p in $(\frac{1}{2},1)$ such that x_i does not have gain for (f,σ,p). □

We now prove the main theorem.

Theorem 2.1 *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 $(\frac{1}{2},1)$. Then with probability 1 there exists at least one variable x_i such that x_i has gain for (f,σ,p).*

Proof. Assume no such variable exists. Thus, by Lemma 2.2, for all $i,j \in [1\ldots n]$, $N(f_{x_i\leftarrow1},\sigma^i,j) = N(f_{x_i\leftarrow0},\sigma^i,j)$.

We first show that for all $i \in [1\ldots n]$, there exists $j \in [0\ldots n]$ such that $N(f_{x_i\leftarrow\neg\sigma(i)},\sigma^i,j) > 0$. Note that $N(f_{x_i\leftarrow\neg\sigma(i)},\sigma^i,j) > 0$ precisely when there exists some assignment a which differs from σ in its ith bit, and which agrees with σ in exactly j of its bits. Since f is not a constant function, there exists $a \in \{0,1\}^n$ such that $f(a) = 1$. Let $j = d(\sigma^i,a^i)$. If $a(i) = \neg\sigma(i)$, then clearly $N(f_{x_i\leftarrow\neg\sigma(i)},\sigma^i,j) > 0$. Otherwise, $N(f_{x_i\leftarrow\sigma(i)},\sigma^i,j) > 0$, and by Lemma 2.2, $N(f_{x_i\leftarrow\neg\sigma(i)},\sigma^i,j) > 0$ also.

Thus for all $i \in [1\ldots n]$, we can define $j_i = \max\{j \mid N(f_{x_i\leftarrow\neg\sigma(i)},\sigma^i,j) > 0\}$. Let $i^* = \arg\max_i j_i$ and let $m = j_{i^*}$. For example, consider $f(x_1,x_2,x_3) = x_1x_2 \vee x_3$ and $\sigma = (0,0,0)$. Then $\neg\sigma(3) = 1$ and $f_{x_3\leftarrow1}(0,0) = 1$. Since assignment $(0,0)$ has 2 bits in common with $(0,0)$, which is the most possible, $j_3 = 2$. Thus for this f, $m = 2$.

Let $\mathrm{POS}(f) = |\{t : f(t) = 1\}|$. There are two cases.

Case 1: $0 \leq m < n-1$. By Lemma 2.2, $N(f_{x_{i^*}\leftarrow\sigma(i^*)},\sigma^{i^*},m) > 0$ also. Thus there exists $a \in \{0,1\}^n$ such that $a(i^*) = \sigma(i^*)$, $d(\sigma^{i^*},a^{i^*}) = m$, and $f_{x_{i^*}\leftarrow\sigma(i^*)}(a^{i^*}) = 1$. Since $m < n-1$, there exists an index $k \neq i^*$ such that $a(k) = \neg\sigma(k)$. Since $f(a) = 1$, $f_{x_k\leftarrow\neg\sigma(k)}(a^k) = 1$. However, $d(\sigma^k,a^k) = m+1$ so $N(f_{x_k\leftarrow\neg\sigma(k)},\sigma^k,m+1) > 0$, which contradicts the definition of m.

Case 2: $m = n-1$. We claim that for all $a \in \{0,1\}^n$, $a \in POS(f)$. This contradicts our assumption that f is not a constant function.

The proof of the claim is by induction on r, where

$r = n - d(\sigma, a)$, the number of bits in which σ and a differ. For the base case, let $r = 0$. The only assignment such that $n - d(\sigma, a) = 0$ is σ itself. We will show that $\sigma \in POS(f)$. By the definition of m, $N(f_{x_{i^*} \leftarrow \neg\sigma(i^*)}, \sigma^{i^*}, m) > 0$. By Lemma 2.2, $N(f_{x_{i^*} \leftarrow \sigma(i^*)}, \sigma^{i^*}, m) > 0$ also. Since $m = n - 1$ and $N(f_{x_{i^*} \leftarrow \sigma(i^*)}, \sigma^{i^*}, m) > 0$, there exists $s \in \{0,1\}^n$ such that $s(i^*) = \sigma(i^*)$, $f_{x_{i^*} \leftarrow \sigma(i^*)}(s^{i^*}) = 1$, and $d(\sigma^{i^*}, s^{i^*}) = n-1$. But $s(i^*) = \sigma(i^*)$ and $d(\sigma^{i^*}, s^{i^*}) = n - 1$ implies $s = \sigma$. Hence $f(s) = 1$, i.e, $\sigma \in \mathrm{POS}(f)$.

Now let $r \in [0 \ldots n-2]$ and assume that all assignments differing from σ in exactly r bits are in $POS(f)$. Let a be an assignment that differs from σ in exactly $r+1$ bits. Let l be an index such that $a(l) = \neg\sigma(l)$; index l exists because $r + 1 > 0$. By the inductive hypothesis, for every assignment u such that $n - d(\sigma, u) = r$, $u \in POS(f)$, including those u such that $u(l) = \sigma(l)$. There are $\binom{n-1}{r}$ assignments u such that $n - d(\sigma, u) = r$ and $u(l) = \sigma(l)$. All these assignments are in $POS(f)$, and thus $N(f_{x_l \leftarrow \sigma(l)}, \sigma^l, r) = \binom{n-1}{r}$. By Lemma 2.2, $N(f_{x_l \leftarrow \neg\sigma(l)}, \sigma^l, r) = \binom{n-1}{r}$ also. The quantity $\binom{n-1}{r}$ equals the total number of assignments $a \in \{0,1\}^n$ that differ from σ in the lth bit and in exactly r of the remaining $n-1$ bits. Clearly a is one such assignments. Hence $f(a) \in \mathrm{POS}(f)$. Since a was an arbitrary assignment differing from σ in exactly $r+1$ bits, all assignments differing from σ in exactly $r+1$ bits are in $\mathrm{POS}(f)$. Thus, by induction, all assigments are in $POS(f)$, proving the claim and the theorem. □

With Theorem 2.1 we have shown that for any nonconstant function and any orientation σ, there exists at least one relevant variable x_i such that if p is chosen randomly, then, with probability 1, x_i has gain with respect to f under $D_{(\sigma,p)}$. We note that, since our proof uses only the property of gain given by Lemma 2.1, the skewing technique will work for any gain measure with that property. This includes commonly used measures such as GINI and Information Gain.

3. Empirical Analysis of Skewing

Theorem 2.1 applies when we have a complete dataset for a function f. However, in practice, this is unlikely to be true. In this case, even in a noiseless situation where examples are all labeled correctly according to a function f, we cannot compute the exact gain of a variable with respect to $D_{(\sigma,p)}$ defined by the skew. We can only estimate that gain. Moreover, in practice we cannot sample from $D_{(\sigma,p)}$. Instead, we simulate $D_{(\sigma,p)}$ by reweighting our sample. In this section, we present an empirical analysis designed to explore the behavior of skewing under such conditions. We first present experiments showing the effect of varying orientation σ and weight factor p, assuming we can sample directly from distribution $D_{(\sigma,p)}$. Next, we present results showing how the technique works in practice, when we cannot sample from $D_{(\sigma,p)}$, but only simulate it by skewing the input distribution.

Table 1. Difference between maximum and minimum accuracy of ID3 as the orientation is varied for different sets of difficult functions on k variables. Examples are described by 30 variables. Training sample size is 1000 examples.

k	Random	Antipodal	Parity
5	11.12%	20.35%	11.02%
6	16.82%	39.08%	17.60%

3.1. Effect of varying σ

First, we describe experiments measuring the effect of picking different orientations σ while keeping the weight factor, p, constant. For these experiments, we fix p to be 0.75. We consider difficult Boolean functions of $k = 5$ and 6 variables, with an additional $30-k$ irrelevant variables present in each example. For each function, we perform 2^k trials, one for each of 2^k distinct orientations σ. These σ all have the value 1 for the irrelevant variables, but vary over all 2^k values for the relevant variables. In each trial, we select a sample of 1000 examples from the distribution $D_{(\sigma,p)}$ induced by (σ, p), use standard ID3 to learn a tree from that sample, and then test the resulting tree on a test set of 1000 examples drawn from the uniform distribution. For each k, we report the difference between the largest and smallest test set accuracy obtained on each function over the 2^k trials. If the choice of σ is important, we would expect this difference to be large.

In Table 1, we report the difference between the largest and smallest accuracy for three sets of functions. The first column shows the difference averaged over 100 random difficult k-variable Boolean functions. The second column shows the difference averaged over the k-variable *antipodal* functions (functions having exactly two satisfying assignments, a and $\bar{a}$). The third column shows the difference for k-variable odd parity.

From Table 1, we observe that as σ is varied, the accuracy achieved by ID3 can change dramatically, even when the sample is drawn according to the distribution induced by (σ, p). Therefore, the choice of σ is important, and in fact increases in importance as n increases. Further, some difficult functions, such as the antipodal functions, show more variation than others as σ changes. Given the full truth table of the function,

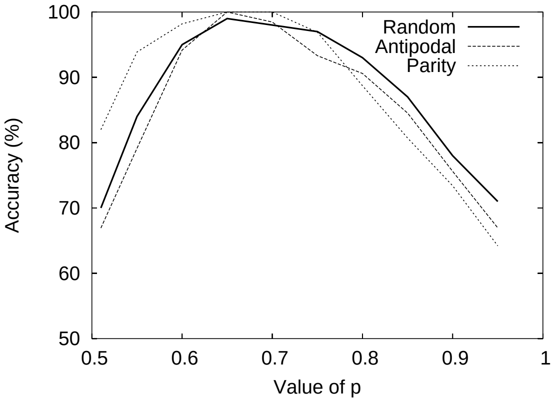

Figure 1. Accuracy as p is varied for different sets of difficult functions on 5 variables. Examples are defined on 30 variables. Training sample size is 1000 examples.

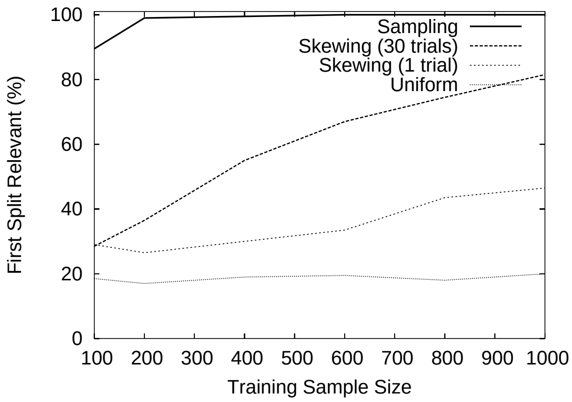

Figure 2. Percentage of times a variable relevant to the target is selected as the first split variable in a tree, as training set size is varied, for random difficult functions on 5 variables. Examples are described by 30 variables.

any σ will isolate the relevant variables, however, with a small training sample and a large number of relevant variables, this may no longer be the case. This leads to overfitting and decreased accuracy.

When k is 4 or less, there is no difference between maximum and minimum accuracy for these experiments. The difference in accuracy for $k = 5$ or more is the result of the relevant variables having varying amounts of gain as σ is changed. With a small training sample, a large number of relevant variables, and a number of irrelevant variables, the variance in gain can translate to a variance in accuracy. However, when these conditions are not satisfied (for example, with few relevant variables and a large sample), any σ will result in gain that is large enough to accurately recover the target function. For this reason, we see no difference between maximum and minimum accuracy for $k = 4$.

3.2. Effect of varying p

In this section, we describe experiments that measure the effect of picking different values of p, the weight factor, while keeping σ fixed. We look at difficult Boolean functions of 5 variables. We generate training sets of 1000 examples, where each example is described by 30 variables (25 irrelevant). We draw the examples from distributions induced by choosing σ to be $111\ldots1$ and letting p range from 0.51 to 0.95. We use ID3 to learn a tree from each training set, and test each tree using a test set of 1000 examples drawn according to the uniform distribution. We track the test set accuracy of ID3 as the value of p changes. If the choice of p is important, we expect some values of p to perform better than other values.

In Figure 1, we show the accuracy as p varies for three sets of functions. First, we show the average accuracy over a random sample of 100 difficult 5-variable Boolean functions. Next, we show the average accuracy for 5-variable antipodal functions. Finally, we show the accuracy for 5-variable odd parity.

From the figure, we observe that the accuracy of ID3 changes significantly as p varies. Values close to 0.5 or 1 result in poor accuracy. Interestingly, a value of p around $\frac{2}{3}$ seems to yield the highest accuracy for all three sets of difficult functions. Determining the reason for this phenomenon is an open problem.

3.3. Skewing versus Sampling from $D_{(\sigma,p)}$

In this section, we describe experiments that evaluate the effect of simulating $D_{(\sigma,p)}$, as done by the skewing algorithm, versus sampling directly from it. We consider difficult Boolean functions of 5 variables, where examples are described by 30 variables. We vary the training set size and compare how often the first split chosen is relevant to the target function, for three methods: (1) ID3 with a sample drawn according to a uniform distribution, (2) ID3 modified to use one iteration of skewing (reweighting), with a sample drawn according to a uniform distribution, and (3) ID3 with a sample drawn according to the distribution induced by the skew used in (2). For comparison purposes, we also show the behavior of using 30 iterations of skewing, as was done in the original skewing algorithm (Page & Ray, 2003). In this case, the skews chosen in each trial are not related to the skew used in (2) above.

The result of this experiment is shown in Figure 2. As expected, sampling directly from $D_{(\sigma,p)}$ allows ID3 to almost always choose a relevant split. However, given a uniform distribution, the first split is usually chosen randomly (i.e. it is correct about $\frac{5}{30} = 16.67\%$ of the time). Skewing, or simulating $D_{(\sigma,p)}$, increases the likelihood of choosing a relevant split, even if done only once. Further, the chance of selecting a relevant split increases as the number of skewings (reweightings) is increased. However, even with 30 skews (which are used in practice), there is still a drop in accuracy as compared to sampling from $D_{(\sigma,p)}$. Finally, as one might expect, the differences in how often a relevant variable is chosen as the first split translate directly into differences in test-set accuracy for these methods.

4. Analysis of Sequential Skewing

We now analyze Sequential Skewing (Ray & Page, 2004). We again consider the idealized setting in which the dataset consists of the entire truth table.

In Sequential Skewing, the weights for a match or mismatch against the preferred settings are not multiplied over variables. Instead, $n+1$ iterations of reweighting are performed, where n is the number of variables. On the j^{th} iteration, examples are reweighted according to the preferred setting of the j^{th} variable only. The last iteration uses the unweighted data. The variable that achieves maximum gain on any of the $n+1$ weightings of the data is chosen as the split variable. For each iteration, there is a chosen variable x_i, a preferred setting c for x_i, and a weight factor p. We thus define a sequential skew to be a triple (i,c,p), where $i \in [1 \dots n]$, $c \in \{0,1\}$, and $p \in (\frac{1}{2},1)$. Define the 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 f *yields pairwise independence* if under the uniform distribution on $\{0,1\}^n$, variables $x_1,\dots,x_n$ are pairwise independent given f, i.e. $\Pr((x_i=\alpha)\wedge(x_j=\beta)|f=\gamma) = \Pr(x_i=\alpha|f=\gamma)\cdot\Pr(x_j=\beta|f=\gamma)$ for all pairs $i \neq j$, and $\alpha,\beta,\gamma \in \{0,1\}$. Constant functions $f \equiv 1$ and $f \equiv 0$ yield pairwise independence, as does the parity function on $n \geq 3$ variables. Other such functions exist: suppose $S \subseteq \{0,1\}^n$ is such that the uniform distribution on S induces a pairwise independent distribution on $x_1,\dots,x_n$. The function f such that $f(a)=1$ iff $a \in S$ yields pairwise independence. Polynomial-sized sets S of this type are used in derandomization.

We say that variable x_j that *has gain for* f under distribution $D_{(i,c,p)}$ if $I_D(f|x_j) > 0$. By Lemma 2.1, x_j has gain for f under distribution $D_{(i,c,p)}$ iff $\Pr_D(f=1|x_j=1) \neq \Pr_D(f=1|x_j=0)$. The following theorem shows that, in our idealized setting, Sequential Skewing works except when applied to functions that yield pairwise independence. By Theorem 2.1, standard skewing has no such limitation.

Theorem 4.1 *Let f be a difficult Boolean function on $\{0,1\}^n$ and let $c \in \{0,1\}$. Let p be chosen uniformly at random from $(\frac{1}{2},1)$. If the function f yields pairwise independence, then for all $j \in [1\dots n]$, x_j has no gain under $D_{(i,c,p)}$. Conversely, if f does not yield pairwise independence, then for some $j \in [1\dots n]$, x_j has gain for $D_{(i,c,p)}$ with probability 1.*

Proof. Let f be a difficult function. Let $i \in [1\dots n]$ and $c \in \{0,1\}$. Assume $c=1$. The proof for $c=0$ is symmetric. Consider skew (i,c,p), where $p \in (\frac{1}{2},1)$.

Let $j \in [1\dots n]$. Let $r_1 = |\{a \in POS(f) \mid a(i)=c \wedge a(j)=1\}|$, and $s_1 = |\{a \in POS(f) \mid a(i) \neq c \wedge a(j)=1\}|$. Similarly, let $r_0 = |\{a \in POS(f) \mid a(i)=c \wedge a(j)=0\}|$, $s_0 = |\{a \in POS(f) \mid a(i) \neq c \wedge a(j)=0\}|$.

Under $D_{(i,c,p)}$, if $j=i$, then since $c=1$, $\Pr(f=1|x_j=1) = r_1\left(\frac{1}{2}\right)^{n-1}$ and $\Pr(f=1|x_j=0) = s_0\left(\frac{1}{2}\right)^{n-1}$. If $j \neq i$, $\Pr(f=1|x_j=1) = (r_1 p + s_1(1-p))\left(\frac{1}{2}\right)^{n-2}$ and $\Pr(f=1|x_j=0) = (r_0 p + s_0(1-p))\left(\frac{1}{2}\right)^{n-2}$.

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 $r_1 = r_0$ and $s_1 = s_0$. If it is not identically 0, then there is at most one value of $p \in (\frac{1}{2},1)$ for which it is 0. If $i=j$, the function is identically zero iff $r_1 = s_0$. Also, for $i=j$, $r_0 = 0$ and $s_1 = 0$ by definition.

In addition, since f is a difficult function, $r_1 + r_0 = s_1 + s_0$. 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 under $D_{(i,c,p)}$. If $j \neq i$, then x_j has no gain under $D_{(i,c,p)}$ iff $r_1 = r_0 = s_1 = s_0$. 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 $\{0,1\}^n$, for all $\alpha,\beta,\gamma \in \{0,1\}$. If this condition holds for all pairs $i \neq j$, then f yields pairwise independence, and no variable x_j has gain for $D_{(i,c,p)}$. Otherwise for some $i \neq j$, x_j has gain for $D_{(i,c,p)}$ for all but at most 1 value of p. □

5. Bounds on Difficult Functions

It is natural to ask how many n-variable Boolean functions are difficult, since these functions actually *need* skewing. The asymptotic behavior of this number (as a function of n) is unknown. However, the number of difficult functions on n variables has been computed for $n \leq 6$ in previous work (Palmer et al., 1992).

A lower bound of $2^{2^{n-1}}$ is implicit in that work and can be shown as follows. Let f be a Boolean function on $\{0,1\}^n$ such that for all $a \in \{0,1\}^n$, $f(a) = 1$ iff $f(\bar{a}) = 1$, where $\bar{a}$ is the bitwise complement of a. There are 2^{n-1} pairs $\{a, \bar{a}\}$, and hence $2^{2^{n-1}}$ such functions, all of them difficult. We prove the following upper bound.

Theorem 5.1 *The number of difficult functions on n variables is at most 2^{2^n-n}.*

Proof. With each $a \in \{0,1\}^n$, associate the variable $y_{\beta(a)}$, where $\beta(a)$ is the integer represented by a if we interpret a as a binary number. Given a Boolean function f on $\{0,1\}^n$, for all $a \in \{0,1\}^n$, let $y_{\beta(a)} = f(a)$. Thus f can be viewed as a truth assignment to variables $y_0, \ldots, y_{2^n-1}$. Function f is difficult iff for all $j \in [1 \ldots n]$, the number of assignments a such that $f(a) = 1$ and $a(j) = 1$ equals the number of assignments a such that $f(a) = 1$ and $a(j) = 0$. This condition can be expressed as a system of n linear equations over the variables $y_0, \ldots, y_{2^n-1}$. For example, for $n = 2$, one of the equations is $y_1 + y_3 - y_0 - y_2 = 0$, which specifies that the number of $a \in \{0,1\}^n$ such that $f(a) = 1$ and $a(2) = 1$ equals the number of $a \in \{0,1\}^n$ such that $f(a) = 1$ and $a(2) = 0$.

The system can be written in matrix form as $yA = 0$, where y is the row vector $[y_0, \ldots, y_{2^n-1}]$, 0 is a row vector of 0's of length n, and A is the $2^n \times n$ matrix defined as follows. For each $a \in \{0,1\}^n$, let a' be obtained from a by writing a as a row vector, and changing each 0 in a to -1. Then for all $j \in [0 \ldots 2^n - 1]$, row j of A is equal to a', where a is such that $j = \beta(a)$. The system $yA = 0$ is satisfied by a 0/1 assignment to y iff the function corresponding to y is difficult.

Consider the n rows of A corresponding to assignments containing exactly one 1. These rows are linearly independent. If we assign 0 or 1 to each of the $2^n - n$ variables y_i not corresponding to one of these n rows, then there exists a unique way to extend the assignment to the remaining n variables so as to satisfy $yA = 0$; however, the extension may not be 0/1. Thus the number of 0/1 solutions to $yA = 0$ is upper bounded by 2^{2^n-n}, the number of 0/1 assignments to the variables y_i. □

We note that since the number of difficult functions for $n = 3$ is 18, the above upper and lower bounds are not tight even for $n = 3$.

6. Conclusion

In this work, we have analyzed the technique of skewing in an idealized setting. In this setting, we show that the technique will almost always succeed at discovering relevant variables when the target function is difficult for greedy tree learners. We have provided an empirical analysis that complements the theory in the case that we have a small sample from the full truth table. We show a similar theoretical result for Sequential Skewing. Our results provide an initial understanding of why and when skewing is effective. However, much remains to be done to fully understand the capabilities and limitations of this technique. One important extension of this work is to develop a theoretical analysis of the case where a random sample is drawn from the uniform distribution and the sample is reweighted using the skewing procedure. We are also interested in knowing for which functions polynomial-size samples suffice, assuming few relevant variables.

Acknowledgements

The second author was supported by NSF grants CCR-9877122 and ITR-0205647 and by the Othmer Institute for Interdisciplinary Studies. The third author was supported by NIH Grant 1R01 LM07050-01 and by grants from the University of Wisconsin Graduate School.

References

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees.* Wadsworth International Group.

Cover, T. M., & Thomas, J. A. (1991). *Elements of information theory.* Wiley Series in Telecommunications. New York, N.Y.: Wiley-Interscience.

Mossel, E., O'Donnell, R., & Servedio, R. A. (2003). Learning juntas. *Proceedings of the 35th Annual Aymposium on the Theory of Computing* (pp. 206–212).

Norton, S. (1989). Generating better decision trees. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* (pp. 800–805). Los Altos, CA: Morgan Kaufmann.

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

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

Quinlan, J. (1997). *C4.5: Programs for machine learning.* Kaufmann.

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