Machine Learning, , 1–37 ()

On Restricted-Focus-of-Attention Learnability of Boolean Functions

ANDREAS BIRKENDORF birkendo@ls2.informatik.uni-dortmund.de

Universität Dortmund, Fachbereich Informatik, D-44221 Dortmund, Germany.

ELI DICHTERMAN eli@cdam.lse.ac.uk

Department of Mathematics, London School of Economics, Houghton Street, London WC2A 2AE, UK. And, Department of Computer Science, Royal Holloway University of London, Egham, Surrey TW20 0EX, UK.

JEFFREY JACKSON jackson@mathcs.duq.edu

Math and Computer Science Department, Duquesne University, 600 Forbes Avenue, Pittsburgh, PA 15282, USA.

NORBERT KLASNER klasner@ls2.informatik.uni-dortmund.de

Universität Dortmund, Fachbereich Informatik, D-44221 Dortmund, Germany.

HANS ULRICH SIMON simon@ls2.informatik.uni-dortmund.de

Universität Dortmund, Fachbereich Informatik, D-44221 Dortmund, Germany.

Received August 20, 1997; Revised August 20, 1997

Editor: Dana Ron

Abstract. In the k-Restricted-Focus-of-Attention (k-RFA) model, only k of the n attributes of each example are revealed to the learner, although the set of visible attributes in each example is determined by the learner. While the k-RFA model is a natural extension of the PAC model, there are also significant differences. For example, it was previously known that learnability in this model is not characterized by the VC-dimension and that many PAC learning algorithms are not applicable in the k-RFA setting.

In this paper we further explore the relationship between the PAC and k-RFA models, with several interesting results. First, we develop an information-theoretic characterization of k-RFA learnability upon which we build a general tool for proving hardness results. We then apply this and other new techniques for studying RFA learning to two particularly expressive function classes, k-decision-lists (k-DL) and k-TOP, the class of thresholds of parity functions in which each parity function takes at most k inputs. Among other results, we prove a hardness result for k-RFA learnability of k-DL, $k \leq n-2$. In sharp contrast, an $(n-1)$-RFA algorithm for learning $(n-1)$-DL is presented. Similarly, we prove that 1-DL is learnable if and only if at least half of the inputs are visible in each instance. In addition, we show that there is a uniform-distribution k-RFA learning algorithm for the class of k-DL. For k-TOP we show weak learnability by a k-RFA algorithm (with efficient time and sample complexity for constant k) and strong uniform-distribution k-RFA learnability of k-TOP with efficient sample complexity for constant k. Finally, by combining some of our k-DL and k-TOP results, we show that, unlike the PAC model, weak learning does *not* imply strong learning in the k-RFA model.

Keywords: Restricted Focus of Attention, PAC-Learning, Learning Algorithms, Boolean Function Classes, Decision Lists, Threshold of Parities, Fourier Transform

1. Introduction

Learning theory has been mainly concerned with the problem of generalizing from a sample of fully-specified classified examples. For this problem classical statistical uniform convergence theorems have been used to characterize scenarios in which a good generalization can be found with high confidence ([28]), specific bounds on the sample size needed for such generalization have been proved [8], and efficient learning algorithms have been designed for specific cases (cf. [27]).

It has also been noticed that in many realistic scenarios, the samples from which the learner has to generalize are not fully specified [21, 22]. The learning models which have been formulated for studying this type of problems usually assume—sometimes implicitly [6]—that there is a fixed set of relevant variables which are invisible to the learner. In such problems, the learner may only attempt to find a good probabilistic prediction rule with respect to the visible attributes. However, as observed by Ben-David and Dichterman [3], there are many cases in which there are no attributes which are inherently invisible, but rather there are other restrictions on the visibility of the attributes, such as the amount of visible attributes in each single example. Since in such cases every attribute is potentially visible, the learner may attempt to find more than just a probabilistic prediction rule; he may try to formulate a full description of the concept with respect to all the relevant attributes.

Consider, for instance, medical research which aims at forming the exact pattern of some disease. Typically, there is some a priori knowledge about the disease, such as the potentially relevant attributes of the disease and the possible patterns of the disease with respect to these attributes. Then, in the course of studying the disease, it is usually possible to sample people from a given population and conduct several tests on each one of them. However, due to practical considerations (*e.g.*, the cost of the tests), or inherent restrictions (*e.g.*, the fact that some blood tests may be destructive, or may not be usable for more than a limited number of tests), the amount of data that is available for each single person is limited.

In such circumstances, researchers face the following problem: They can choose a set of attributes which can be tested on a given sample, and they may choose to test different attributes on different samples. However, they cannot have the full relevant medical record of each sampled person. What type of information can be extracted from such partially-specified samples? Certainly, if the samples are large enough, it is possible to estimate the probability of developing the disease, for each set of attributes, and for every assignment to these attributes (assuming that it is known whether each sampled person has developed the disease or not). Although such estimates are useful in predicting whether a given subject will develop the disease, forming an exact description of the disease with respect to all the relevant attributes may be much more useful in understanding the disease and in finding ways of treating it. This is the main theme of this paper—when and how a learner can use a priori knowledge (*i.e.*, the class of possible concepts) and partially-specified samples to find with high confidence a good approximation of the target concept. For instance, it is implied by the results shown in this paper that, in general, if it is known that the disease may be described as a binary-valued decision list, then in order to find with high confidence a good approximation of the disease at least half of the attributes have to be tested for each sampled person.

The problem of learning in such scenarios motivated the general *restricted-focus-of-attention* learning model [3], in which the learner has no direct access to full examples, but rather may observe each example in one of a limited number of ways. In this work we consider a special type of restriction called *k-RFA*, in which the learner may observe any set of k attributes of each example.

An interesting and useful feature of the RFA restriction is its relation to efficient noise-tolerant learning. It follows from a result in [4] that an $O(\log n)$-RFA oracle can be efficiently simulated by statistical queries, and hence by Kearns' transformation [19] an efficient $O(\log n)$-RFA learner can tolerate classification noise (a simple and direct proof of the noise-robustness of an $O(\log n)$-RFA learner is shown in [5]). Furthermore, it is shown in [11] that if each statistical query uses only a restricted view (*i.e.*, it depends on a logarithmic number of attributes), then the learner can tolerate attribute noise as well. It follows that an $O(\log n)$-RFA learner can tolerate efficiently and simultaneously both attribute and classification noise. Hence, one may view the RFA restriction as a useful conceptual tool in constructing efficient noise-tolerant learning algorithms: Just make sure that the learning algorithm selects no more than a logarithmic number of attributes to be seen in any given input example. As demonstrated in [4], in many cases this is easily accomplished by a slight variation of the well-known learning strategies.

While the k-RFA framework resembles the PAC model in many aspects, there are also some interesting differences. For example, unlike the PAC model, k-RFA learnability of a class is *not* characterized by its VC-dimension [3, 4]. We show in this paper another surprising difference: Weak k-RFA learnability of a class does *not* imply strong k-RFA learnability of that class. Hence, it seems that better understanding of k-RFA learnability can substantially increase our understanding of the extent to which results in other learning models depend on access to complete examples.

A few initial results for k-RFA learning of Boolean functions are given in [3]. For instance, it is shown there that the class of Boolean functions which are representable by k-CNF or k-DNF formulas are efficiently k-RFA learnable (for fixed k), and that the class of k-decision-lists is (inefficiently) k-RFA learnable under the uniform distribution. (We use the notion "efficient" learning when both the time and the sample complexities of the learning algorithm are polynomials in the all the learning parameters of the problem).

This paper extends our understanding of k-RFA learnability of Boolean functions in a number of ways. First, we develop a characterization of k-RFA learnability that forms the basis for a general tool that we later use to prove learnability hardness results. Next, we consider the k-RFA learnability of two specific function classes: k-DL, the class of functions expressible as decision lists in which each test is a k-term; and k-TOP, the class of functions expressible as a threshold of parity functions, where each parity is defined over at most k inputs. We have chosen these classes for several reasons. For constant k, both of these classes are efficiently PAC learnable; in fact, they are among the most expressive classes which are currently known to be efficiently and distribution-free PAC-learnable (both contain k-CNF $\cup$ k-DNF, for example). On the other hand, their learnability in the k-RFA model is not immediately clear. Also, our study of these classes, particularly of k-DL, has shown that seemingly small variations in a question about the class can lead to substantial variation in the resulting answers. This variability adds significantly to our interest in k-RFA learnability questions. Finally, as discussed further below, a combination of some of our results for these two classes produces an interesting result about the relationship between weak and strong learning in the k-RFA model.

As an example of our k-DL results, we show that, in the distribution-free k-RFA model, $(n-1)$-DL is (inefficiently) learnable from an $(n-1)$-RFA oracle. On the other hand, it is information-theoretically impossible to learn $(n-2)$-DL from an $(n-2)$-RFA oracle, even if the decision list has at most two alternations of the labels! Another small change, however, leads to quite a different result: with respect to any known distribution, k-DL *is* k-RFA learnable (not necessarily efficiently) for all k (for $k = 1$ and the uniform distribution there is an efficient learning algorithm; cf. [11]). In yet another contrast, we also prove a hardness result showing, among other things, that distribution-free learnability of 1-DL requires access to at least half of the bits in each example.

Our study of k-TOP is motivated in part by the fact that it is known to have useful Fourier properties [17]; furthermore, it has also been studied in the context of empirical machine learning [18]. We exploit the Fourier properties of k-TOP to show first that k-TOP is weakly k-RFA learnable and that this learning is efficient for constant k. Second, we show that with respect to the uniform distribution, k-TOP is strongly k-RFA learnable with polynomial (in the usual learning parameters, and assuming a constant k) sample complexity, but running time which is not necessarily polynomial.

As indicated earlier, we ultimately combine some of our k-DL and k-TOP results to obtain the following: unlike the PAC model, weak and strong learning are *not* equivalent in the k-RFA model (for $k \leq n-2$). This says that the hypothesis boosting technique introduced by Schapire [25] for transforming weak learning algorithms into strong learners depends in a fundamental way on having access to more of the attributes in an example than the number needed for merely weak learning.

The paper closes with some directions for further research.

2. Definitions

2.1. *The Learning Model*

The model introduced in [3] suggests a general way of extending any learning model by a new mechanism which generates observations (seen by the learner) from examples (drawn by nature). In this work we use the RFA extension of the well-known PAC model [27], as defined below.

Let $\mathcal{F}$ be a class of $\{0,1\}$-valued functions (concepts) over an instance space X, and let D be some probability distribution over X. The distribution D is used both to generate the random training examples for the learner and to define the proximity between a learner's hypothesis and the correct target concept. We use the notation $x \in D$ to denote that x is drawn randomly from the distribution D (over the instance space X).

In the RFA model another characterizing component is added to any learning problem. This is a set W of projections, where a projection is a mapping of classified examples to some observation space $\mathcal{O}$. In the process of learning a *target function* $f \in \mathcal{F}$, the learner can make an observation by selecting a projection $w \in W$, and getting the value of $w(x, f(x))$, where x is a random instance drawn from D. Choosing a projection $w \in W$ models the act of focusing the attention on a set of features.

Let the instance space be $X = \{0,1\}^n$. A special interesting case of the RFA setting is the k-RFA model, $0 \le k \le n$, in which the learner is restricted to choose projections from the class W_k of k-RFA projections; a k-RFA projection $w \in W_k$ is defined by a set of k indices $\{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}$. When x is drawn from D and f is the target function, then the learner observes $\langle (x_{i_1}, \ldots, x_{i_k}),\ f(x)\rangle$ (where x_j is the j-th bit of x). Hence, a k-RFA learner may observe only k bits of each instance x (this is the restriction on the size of the learner's focus of attention), and he can also observe the classification bit.

Formally, this focusing mechanism is modelled by a k-RFA *focusing function* $\Phi : \mathcal{O}^* \to W_k$, which selects the next k-RFA projection based on the sequence of observations seen so far. Given a sequence of m instances $\vec{x} = (x_1, \ldots, x_m) \in X^m$, a target function $f \in \mathcal{F}$, and a k-RFA focusing function Φ, the observation sample generated by $\vec{x}$, f and Φ is $sample(\vec{x}, f, \Phi) = (w_1(x_1, f(x_1)), \ldots, w_m(x_m, f(x_m)))$, where $w_i = \Phi(w_1(x_1, f(x_1)), \ldots, w_{i-1}(x_{i-1}, f(x_{i-1})))$, for $1 \le i \le m$ (w_1 is the value of Φ on the null sequence).

Having a sufficiently large sample of observations, the learner has to choose a hypothesis $h : X \to \{0,1\}$ from the *hypothesis class* $\mathcal{H}$. The error of any h with respect to f and D is measured by $error_{f,D}(h) = \Pr_{x \in D}[h(x) \ne f(x)]$, and a hypothesis h is called ϵ-good (with respect to f and D) if $error_{f,D}(h) \le \epsilon$ (h is ϵ-bad if it is not ϵ-good).

Following [8], our basic definition of learnability in the RFA model, is an information-theoretic one (no computational restrictions). That is, we model the hypothesis selection by a *learning function* $L : \mathcal{O}^* \to \mathcal{H}$. Given a sufficiently large sample of observations, a successful learning function should produce, with high confidence, a good hypothesis. In general, the sample size should be finite, but can be super-polynomial in the parameters of the leaning problem.

Definition 1. [k-RFA Learnability] The function class $\mathcal{F} \subseteq 2^X$ is k-RFA learnable using the hypothesis class $\mathcal{H}$, if there is an integer-valued sampling function $m(\cdot,\cdot,\cdot,\cdot)$, there is a k-RFA focusing function Φ, and there is a learning function $L : \mathcal{O}^* \to \mathcal{H}$, such that for every target function $f \in \mathcal{F}$, for every distribution D on X, and for every $0 < \epsilon, \delta \le 1$

$$\Pr_{\vec{x} \in D^m}\left[error_{f,D}(L(sample(\vec{x}, f, \Phi)) > \epsilon\right] < \delta$$

where $m = m(\epsilon, \delta, n, size(f))$ and $size(f)$ is the minimal representation length of f.

Usually we seek for a learning algorithm, so we want the sampling function m, the focusing function Φ, and the learning function L, to be computable. In fact, we are mainly interested in *efficient* learning algorithms. We say that a learning algorithm is *sample-efficient* if its sampling function m is polynomial in $\frac{1}{\epsilon}$, $\frac{1}{\delta}$, n, and $size(f)$. Also, we say that the algorithm is *efficient* if it is sample-efficient, and both its focusing function and its learning function are computable in polynomial time (in $\frac{1}{\epsilon}$, $\frac{1}{\delta}$, n, and $size(f)$).

When the hypothesis class $\mathcal{H}$ is omitted it is assumed that $\mathcal{H} = \{0, 1\}^X$. However, efficient learning in this case means that the learning algorithm outputs a hypothesis which is computable in polynomial time. The term *proper learnability* is used for the case $\mathcal{H} = \mathcal{F}$.

The above definition models the 'distribution-free' scenario in which the learning algorithm can handle arbitrary generating distributions D (and does not know D in advance). In many cases this requirement appears to be too restrictive. In such cases we shall also consider a more permissive setting obtained by requiring successful learning only with respect to a fixed distribution which is known to the learner.

Finally, a function class is *weakly learnable* (in either the PAC or RFA model) if it is learnable given that ϵ is restricted to be at least $1/p(n, size(f))$ for $p(\cdot, \cdot)$ a fixed polynomial.

2.2. *Classes of Boolean Functions*

One of the classes whose RFA learnability is studied in this work is the class of decision lists, introduced by Rivest in [24]. A decision list is an ordered list of pairs $\langle(t_1, b_1), \ldots, (t_r, b_r)\rangle$, where each t_j is a term (conjunction of literals, where each literal is a Boolean variable or its negation), and each b_j is a Boolean value called label. A pair (t_j, b_j) is satisfied by an assignment $a \in \{0, 1\}^n$ if $t_j(a) = 1$. A decision list L defines a Boolean function as follows. The value of L on the assignment a is determined by the label of the first item in the list which is satisfied by a. To ensure that at least one item is always satisfied, the last item of the list is of the form $(1, b)$, where 1 is the term which is always satisfied. Consider, for instance, the decision list $\langle(\overline{x}_1x_2, 0), (x_2\overline{x}_4x_5, 0), (x_1\overline{x}_3\overline{x}_4, 1), (1, 0)\rangle$, which is illustrated in Figure 1. The values of the list on inputs $(1, 1, 1, 0, 1)$ and $(1, 1, 0, 0, 0)$ are 0 and 1, respectively.

Intuitively, a decision list is a useful representation for a Boolean function whose value is dominated by terms in some decreasing order of importance; *i.e.*, a term determines the value of the function on a given assignment only if it is true and all of its predecessors in the list are false. In other words, the tail of the list has an influence on the value of the function only for the assignments on which the value of the function has not been already determined by the head of the list.

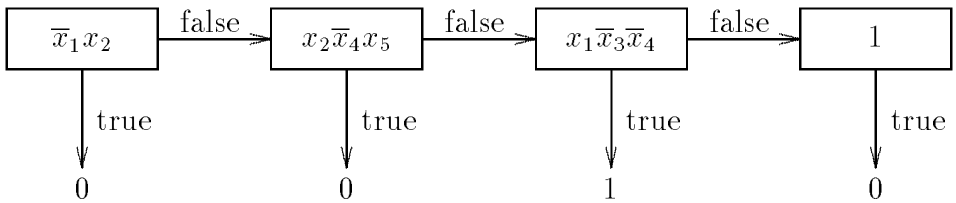

Figure 1. Example of a decision list.

A k-decision-list is a decision list in which each term t_i consists of at most k literals. For example, the decision list given above is a 3-decision-list. Formally, the class of k-decision-lists is defined as follows.

Definition 2. [k-DL] A k-decision-list is a list $\langle (t_i, b_i) \rangle_{i=1}^r$ of pairs, in which each t_i is a k-term, $b_i \in \{0,1\}$, and t_r is the constant 1. The *size* of the k-decision-list is r. A k-decision-list L defines a Boolean function as follows: for every $x \in \{0,1\}^n$, $L(x) = b_j$ where $j = min\{i \,|\, t_i(x) = 1\}$. We denote by k-DL$_n$ the class of all k-decision-lists over n variables.

We also denote by j-alt-k-DL$_n$ the class of all k-decision-lists, in which the number of alternations in each list is bounded by j (an alternation occurs when $b_{i+1} = 1 - b_i$).

We omit the subscript n when it is clear from the context.

It is shown in [24] that k-DL properly contains k-DNF $\cup$ k-CNF, and is efficiently PAC learnable for constant k. It is shown in [3] that the class k-DL is (non-efficiently) k-RFA learnable under the uniform distribution. In this work we further study the RFA learnability of this class, and show some new positive and negative (*i.e.*, non-learnability) results for this class.

Another class studied in this work is the class TOP, which is defined as follows.

Definition 3. [TOP] We denote by TOP the class of Boolean functions expressible as a depth-2 circuit with a majority gate at the root and parity gates at the leaves, and we will require an odd number of parity gates in every TOP expression so that we need not be concerned about the value of the majority gate in the case of half of the parity gates "voting" each way. The inputs to the parity gates are literals, *i.e.*, variables in either an unnegated or a negated sense. All gates in a TOP have unbounded fanin and fanout one. A k-TOP is a TOP in which each parity has fanin at most k; we call such a parity a *k-parity*. The *size of a (k-)TOP r* is the number of parity gates in r.

Note that a parity may appear multiple times in a TOP circuit. It is often convenient to instead think of such a circuit as having each distinct parity appearing just once and associating a positive integer weight with it. Furthermore, as discussed further below, we will find it useful to view parity functions and TOPs as mapping to $\{-1,+1\}$ rather than the standard $\{0,1\}$. In particular, this allows us to view the majority gate at the root of a TOP as a threshold function, which outputs 1 if the weighted sum of the parity functions defining the TOP is positive and -1 if the sum is negative. Put another way, the root node simply takes the sign of the weighted sum of the inputs to the root.

Furthermore, notice that a parity gate defined over a set of variables in which an odd number of the variables are negated is equivalent to the complement of that parity over the same set of variables but with all variables appearing unnegated. For example, $\overline{x_1} \oplus x_2 = \overline{x_1 \oplus x_2}$. Also, given the assumption that parity functions produce values in $\{-1, +1\}$, the effect of complementing a parity function can be achieved within a TOP expression by simply negating the weight associated with that parity function. Thus we have that the TOP expressions $MAJ((x_1 \oplus x_3) + 2(\overline{x_1} \oplus x_2))$ and $MAJ((x_1 \oplus x_3) - 2(x_1 \oplus x_2))$ are equivalent. This view of TOPs as being defined by the sign of the integer-weighted sum of uncomplemented $\{-1, +1\}$-valued parity functions over unnegated variables will be adopted in the remainder of the paper. The size of such a TOP is the sum of the magnitudes of the weights.

We also denote by PAR the class which contains only two functions: the parity function over n variables ($\text{parity}_n \equiv x_1 \oplus x_2 \oplus \ldots \oplus x_n$) and its complement ($\overline{\text{parity}_n} \equiv \overline{x_1 \oplus x_2 \oplus \ldots \oplus x_n}$).

2.3. The Fourier transform

While the Fourier transform has numerous uses in computer science (see, *e.g.*, [1]), we will use a somewhat nonstandard multidimensional version of the transform first applied to learning theory by Linial, Mansour, and Nisan [23]. For each vector $a \in \{0,1\}^n$ we define the function $\chi_a : \{0,1\}^n \to \{-1, +1\}$ as $\chi_a(x) = (-1)^{\sum_i a_i x_i}$. That is, $\chi_a(x)$ is the Boolean function that is 1 when the parity of the bits in x indexed by a is even and is -1 otherwise. These functions have the property that

$$\mathbf{E}_x[\chi_a(x) \cdot \chi_b(x)] = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{otherwise} \end{cases}$$

(Expectations and probabilities here and elsewhere are with respect to the uniform distribution over the instance space unless otherwise indicated). Thus these functions form a basis for the space of all real-valued functions on $\{0,1\}^n$, and every function $f : \{0,1\}^n \to \mathbf{R}$ can be uniquely expressed as a linear combination of the χ functions: $f = \sum_a \hat{f}(a)\chi_a$, where $\hat{f}(a) = \mathbf{E}[f \cdot \chi_a]$. The vector of coefficients $\hat{f}$ is called the (discrete multi-dimensional) *Fourier transform* of f (also known as the Walsh transform). We say that a Fourier coefficient $\hat{f}(a)$ has *order* k if $|a| = k$ and has *bounded order* k if $|a| \leq k$, where $|a|$ represents the Hamming weight of a. Note that $\chi_{\bar{0}}$ is the constant $+1$ function; therefore, $\hat{f}(\bar{0}) = \mathbf{E}[f\chi_{\bar{0}}] = \mathbf{E}[f]$. Also note that for $f \in \{-1, +1\}$, $\hat{f}(a) = \mathbf{E}[f\chi_a]$ represents the correlation of f and χ_a with respect to the uniform distribution. For this and related reasons, in the sections of this paper dealing with Fourier analysis and TOP functions we will assume that $f \in \{-1, +1\}$.

By Parseval's theorem, for every function $f : \{0,1\}^n \to \mathbf{R}$, $\mathbf{E}[f^2] = \sum_a \hat{f}^2(a)$. For $f \in \{-1,+1\}$ it follows that $\sum_a \hat{f}^2(a) = 1$. More generally, for any real-valued functions f and g, $\mathbf{E}[f \cdot g] = \sum_a \hat{f}(a)\hat{g}(a)$.

3. Hardness of k-RFA Learnability

In this section we develop a characterization of the conditions under which a function class is or is not learnable from a k-RFA oracle. (In Appendix A we present an alternative, Fourier-based characterization of k-RFA learnability which, while potentially useful, does not lead directly to any results in this paper.) Building on this characterization, we develop a general tool for showing k-RFA learnability hardness, which we then apply to obtain hardness results for RFA learnability of k-DL.

3.1. *Characterizing k-RFA Learnability*

A general scheme for proving information-theoretic hardness in a given learning model is the following one. Assume we can find a set $\mathcal{Q}$ of scenarios (a scenario here is a setting of all the parameters which are unknown to the learner, typically the target function and the target distribution), satisfying the following two conditions:

1. Any possible hypothesis made by the learner is bad for at least one of the scenarios in $\mathcal{Q}$.

2. A learner in the given model cannot distinguish between the scenarios in $\mathcal{Q}$ (*i.e.*, each scenario in $\mathcal{Q}$ provides the learner with exactly the same information).

Being unable to distinguish between the different scenarios in $\mathcal{Q}$, the learner has to make the same decision in each scenario. However, since any decision is bad for at least one scenario in $\mathcal{Q}$, there must be a scenario in which the learner fails. The exact formulation of this scheme depends on the given learning model.

Such a scheme has been first used by Kearns and Li [20] (and later by others, cf. [16]), in proving the information-theoretic upper bound on the rate of tolerable malicious noise. Specifically, they show that by maliciously corrupting an $\frac{\epsilon}{1+\epsilon}$ fraction of the learner's sample, there are two different scenarios which induce the same distribution over the corrupted sample space, yet any hypothesis made by the leaner is ϵ-bad for at least one of them.

We use a similar idea to formulate a general scheme for proving information-theoretic hardness of learnability in the k-RFA model. It turns out that our formulation also provides a full characterization of k-RFA learnability.

Define a *scenario* over the instance space $X = \{0,1\}^n$ to be a pair $\langle f, D\rangle$ of a Boolean function f and a distribution D over X. If $f \in \mathcal{F}$ then $\langle f, D\rangle$ is called an $\mathcal{F}$-scenario. To formulate the notion of indistinguishability by a k-RFA learner we define an equivalence relation among scenarios as follows. For $I = \{j_1, \ldots, j_k\} \subseteq \{1, \ldots, n\}$ and $x \in \{0,1\}^n$, let $x|_I = (x_{j_1}, \ldots, x_{j_k})$. Given a scenario $S = \langle f, D\rangle$

over $\{0,1\}^n$, define for $I = \subseteq \{1,\ldots,n\}$, $z \in \{0,1\}^k$, and $b \in \{0,1\}$, the probability

$$p_S(I,z,b) \triangleq \Pr_{x\in D}[\, f(x)=b\ ,\ x|_I = z\,]$$

That is, in the scenario S, the probability of observing (z,b) when focusing on the index set I is $p_S(I,z,b)$. The set $\{p_S(I,z,b)\}_{I,z,b}$ is called *k-RFA probabilities* of the scenario S. We say that S_1 and S_2 are *k-RFA equivalent* if $p_{S_1} \equiv p_{S_2}$ (*i.e.*, $p_{S_1}(I,z,b) = p_{S_2}(I,z,b)$ for every I, z, and b). Obviously, this is an equivalence relation. Also, notice that for any k-RFA projection defined by a set of k indices $\{i_1,\ldots,i_k\} \subseteq \{1,\ldots,n\}$, two k-RFA equivalent scenarios induce identical distributions over the observation space, and thus k-RFA equivalent scenarios are indistinguishable by a k-RFA learner.

A hard set for a k-RFA learner is a set of k-RFA equivalent scenarios which has some "discrepancy". A set has an *ϵ-discrepancy* with respect to a hypothesis class $\mathcal{H}$, if every $h \in \mathcal{H}$ is ϵ-bad for at least one of the scenarios (recall that h is ϵ-bad for the scenario $\langle f, D\rangle$ if $error_{f,D}(h) \geq \epsilon$). A set is *$k$-RFA hard* for $\mathcal{H}$ if it has a non-zero discrepancy with respect to $\mathcal{H}$, and all of its scenarios are k-RFA equivalent. Notice that there might be a hard set which does not include any hard pair. We prove that the existence of a k-RFA hard set is sufficient to imply non-learnability in the k-RFA model. Furthermore, we also prove that this condition is weak enough to be necessary (for non-learnability), providing a full characterization of (information-theoretic) learnability in the k-RFA model.

THEOREM 1 *A class $\mathcal{F}$ of boolean functions is k-RFA learnable using the class $\mathcal{H}$ if and only if there is no set of $\mathcal{F}$-scenarios which is k-RFA hard for $\mathcal{H}$.*

Proof: First we prove that the existence of a hard set of scenarios implies non-learnability. Assume that there is a set $\mathcal{Q}$ of k-RFA equivalent $\mathcal{F}$-scenarios, which has an ϵ-discrepancy ($\epsilon > 0$). Let S_h be a scenario in $\mathcal{Q}$ for which h is ϵ-bad. Since the instance space is finite, the hypothesis class is also finite, hence $\mathcal{Q}' = \{S_h : h \in \mathcal{H}\}$ is a finite class of $\mathcal{F}$-scenarios which is k-RFA hard for $\mathcal{H}$.

Let A be a k-RFA learning algorithm which uses a sample of m observations in order to learn the class $\mathcal{F}$ using $\mathcal{H}$. Being k-RFA equivalent, all the scenarios in $\mathcal{Q}'$ induce the same m-fold product distribution P^m over the m-fold observation space. The hardness of $\mathcal{Q}'$ implies that for each sequence z of m observations drawn from P^m, the hypothesis $A(z)$ chosen by A is ϵ-bad for at least one scenario in $\mathcal{Q}'$.

Let $\alpha_S = \Pr_{z\in P^m}[A(z)$ is ϵ-bad for $S]$. As every $A(z)$ is ϵ-bad for some $S \in \mathcal{Q}'$, we have $\sum_{S\in\mathcal{Q}'} \alpha_S \geq 1$, so there must be a scenario $S \in \mathcal{Q}'$ for which $\alpha_S \geq \frac{1}{|\mathcal{Q}'|}$. Hence, for S being the target scenario, the probability that A fails to find an ϵ-good hypothesis is at least $\frac{1}{|\mathcal{Q}'|} > 0$.

Next we prove that if there is no set of $\mathcal{F}$-scenarios which is hard for $\mathcal{H}$, then $\mathcal{F}$ is k-RFA learnable using $\mathcal{H}$. Assume that there is no such hard set, and assume first that the learner knows the exact k-RFA probabilities $\{p_S\}$ of the target scenario S. Let $\mathcal{A} = \{S' : p_{S'} \equiv p_S\}$ (notice that $S \in \mathcal{A}$). Since $\mathcal{A}$ cannot be a hard set, it must have a zero discrepancy. Hence, there must be a hypothesis h which is good for all the scenarios in $\mathcal{A}$, and in particular for the target scenario S. This implies that an infinite sample size is sufficient for finding a good hypothesis. However, we need to show that a *finite* sample size is uniformly sufficient for all the possible target scenarios. Since the number of possible scenarios is infinite (as is the number of possible distributions), it is not immediately obvious why a finite sample size is sufficient.

Here is the main idea of the proof. First, we wish to show that, given the accuracy needed from the learner, it is sufficient to consider a finite cover of the set of all scenarios. Then, by using a finite sample, the learner can choose from this finite cover a set of scenarios which has a small discrepancy, and includes with high probability a good approximation of the target. Once such a set is found, the learner can choose a hypothesis which is good for all the scenarios in the set, and hence also for the target scenario. The crucial point here is to ensure that one can use good k-RFA estimates in order to find a set with small discrepancy. Hence, we need to relate the accuracy of the k-RFA estimates to the discrepancy of a set of scenarios. This is done as follows.

Define the discrepancy of a set of scenarios $\mathcal{A}$ to be:

$$discrepancy(\mathcal{A}) = \min_{h \in \mathcal{H}} \sup_{S \in \mathcal{A}} error_S(h)$$

Notice that if $discrepancy(\mathcal{A}) < \epsilon$ then there is a hypothesis h which is ϵ-good for all the scenarios in $\mathcal{A}$. Also, define the k-RFA resolution of a set $\mathcal{A}$ to be:

$$resolution(\mathcal{A}) = \sup_{S,S' \in \mathcal{A}} \|p_S - p_{S'}\|_\infty = \sup_{S,S' \in \mathcal{A}} \max_{I,z,b} |p_{S_1}(I,z,b) - p_{S_2}(I,z,b)|$$

Obviously, if $resolution(\mathcal{A}) = 0$ then all the scenarios in $\mathcal{A}$ are k-RFA equivalent. Otherwise, it is possible to distinguish between at least two subsets of $\mathcal{A}$ by having close enough estimates of the k-RFA probabilities.

We would like to have a lower bound on the necessary k-RFA resolution of a set $\mathcal{A}$ which guarantees a lower bound on the discrepancy of the set. Assuming that there is no set $\mathcal{Q}$ for which $discrepancy(\mathcal{Q}) > 0$ and $resolution(\mathcal{Q}) = 0$, the following lemma establishes such a relation.

LEMMA 1 *If there is no set of $\mathcal{F}$-scenarios which is k-RFA hard for $\mathcal{H}$, then for every $\epsilon > 0$ there is $\gamma > 0$, such that $discrepancy(\mathcal{A}) \geq \epsilon$ implies $resolution(\mathcal{A}) \geq \gamma$ for every finite set $\mathcal{A}$ of $\mathcal{F}$-scenarios.*

Proof: Assume that there is no set of $\mathcal{F}$-scenarios which is hard for $\mathcal{H}$. If the lemma does not hold then we can find an infinite sequence of $\mathcal{F}$-scenario sets $(\mathcal{A}_j)_{j \in J}$, in which $discrepancy(\mathcal{A}_j) \geq \epsilon$ for every j, but $resolution(\mathcal{A}_j)$ converges to 0. The idea of the proof is to show that in this case there is a sub-sequence of $(\mathcal{A}_j)$ which converges to a hard set, contradicting the assumption that there is no such set.

First we show that there is a sequence of finite scenario set which satisfies the same conditions. For each $h \in \mathcal{H}$, let $S_{h,j} = \langle f_{h,j}, D_{h,j} \rangle \in \mathcal{A}_j$ be an $\mathcal{F}$-scenario for which $error_{S_{h,j}}(h) \geq \epsilon$, and let $\mathcal{B}_j = \{S_{h,j} : h \in \mathcal{H}\}$. Notice that $discrepancy(\mathcal{B}_j) \geq \epsilon$. Also notice that $resolution(\mathcal{B}_j) \leq resolution(\mathcal{A}_j)$ for every j, and hence the sequence $resolution(\mathcal{B}_j)$ converges to 0. Furthermore, since $\mathcal{F}$ and $\mathcal{H}$ are finite, there is an infinite $J' \subseteq J$, and a set of functions $\{g_h : h \in \mathcal{H}\} \subseteq \mathcal{F}$, such that $f_{h,j} = g_h$ for every $j \in J'$ and every $h \in \mathcal{H}$.

Pick some $h \in \mathcal{H}$, and consider the infinite sequence $(D_{h,j})_{j \in J'}$. We claim that it has a converging subsequence. Let $\mathcal{D}$ be the set of all distributions over $\{0,1\}^n$, and let d_s be the *statistical-distance* metric defined by $d_s(D, D') = \sum_{x \in \{0,1\}^n} |D(x) - D'(x)|$. First notice that any distribution D over $\{0,1\}^n$ can be represented as a point $a \in \mathbf{R}^{2^n}$ by letting $a_i = D(\bar{i})$, where $\bar{i}$ is the binary vector-representation of i. Hence, $\langle \mathcal{D}, d_s \rangle$ can be embedded as a subspace in the metric space $\langle \mathbf{R}^{2^n}, d_1 \rangle$, where d_1 is the L^1 metric. Being a bounded and closed subspace, it is also compact, and hence any sequence in it has a converging subsequence.

By applying the same argument iteratively for every $h \in \mathcal{H}$ we obtain a scenario sub-sequence $(\mathcal{B}_j)_{j \in J''}$, such that for every $h \in \mathcal{H}$, the distribution sub-sequence $(D_{h,j})_{j \in J''}$ converges to some distribution D_h. Let $\mathcal{Q}$ be the set of $\mathcal{F}$-scenarios to which the sequence $(B_j)_{j \in J''}$ converges. That is, $\mathcal{Q} = \{\langle g_h, D_h \rangle : h \in \mathcal{H}\}$. We claim that $\mathcal{Q}$ is k-RFA hard for $\mathcal{H}$. First notice that for every $j \in J''$:

$$error_{S_h}(h) \geq error_{S_{h,j}}(h) - d_s(D_h, D_{h,j}) \geq \epsilon - d_s(D_h, D_{h,j})$$

Since $d_s(D_h, D_{h,j})$ converges to 0, we get that $error_{S_h}(h) \geq \epsilon$ for every $h \in \mathcal{H}$, hence $discrepancy(\mathcal{Q}) \geq \epsilon$. Also, for every $h, h' \in \mathcal{H}$ and every $j \in J''$:

$$\|p_{S_h} - p_{S_{h'}}\|_\infty \leq \|p_{S_{h,j}} - p_{S_{h',j}}\|_\infty + d_s(D_h, D_{h,j}) + d_s(D_{h'}, D_{h',j}) \tag{1}$$

Since all the terms in the r.h.s. of Inequality 1 converge to 0, we get $\|P_{S_h} - P_{S_{h'}}\|_\infty = 0$ for every $h, h' \in \mathcal{H}$, and therefore $resolution(\mathcal{Q}) = 0$. ■

By Lemma 1, there is a function $\Gamma : \mathbf{R}^+ \to \mathbf{R}^+$, such that $resolution(\mathcal{A}) > \Gamma(discrepancy(\mathcal{A})) > 0$ for every set $\mathcal{A}$ of scenarios. Given $\epsilon > 0$, let $\gamma = \min\{\Gamma(\frac{\epsilon}{2}), \frac{\epsilon}{2^{k-1}}\}$, and recall that $\mathcal{D}$ is the set of all distributions over $\{0,1\}^n$. Define $\mathcal{D}'$ to be a $(2^{k-2}\gamma)$-cover of $\mathcal{D}$ with respect to the statistical-distance metric d_s:

$$\mathcal{D}' \stackrel{\Delta}{=} \{D \in \mathcal{D} \ : \ \forall x \ D(x) \in \{i/M : i = 0, 1, \ldots, M\}\} \quad ; \quad M = \left\lceil \frac{2^{n-k+2}}{\gamma} \right\rceil$$

$$\mathcal{T} \stackrel{\Delta}{=} \{\langle f, D \rangle \ : \ f \in \mathcal{F} \ , \ D \in \mathcal{D}'\}$$

For every $D \in \mathcal{D}$ there is $D' \in \mathcal{D}$ such that $d_s(D, D') \leq 2^{k-2}\gamma$. Hence, for every two functions f and h,

$$error_{f,D}(h) \ \leq \ error_{f,D'}(h) + d_s(D, D') \ \leq \ error_{f,D'}(h) + 2^{k-2}\gamma$$

Taking sufficiently large samples of the target scenario $S = \langle f, D \rangle$, a k-RFA learner can compute an estimate $\hat{p}_S$ satisfying (with high confidence) $\|\hat{p}_S - p_S\|_\infty < \frac{\gamma}{4}$. Let

$$\mathcal{B} = \{S' \in \mathcal{T} : \ \|\hat{p}_S - p_{S'}\|_\infty \leq \frac{\gamma}{2}\}$$

We claim that a hypothesis h which minimizes $discrepancy(\mathcal{B})$ is with high confidence an ϵ-good hypothesis. Consider the scenario $S' = \langle f, D' \rangle$, where $D'(x) = \lfloor D(x)M \rfloor \frac{1}{M}$. Obviously, $S' \in \mathcal{T}$. Furthermore, since $|D(x) - D'(x)| \leq \frac{1}{M}$ for every x, it follows that $\|p_S - p_{S'}\|_\infty \leq 2^{n-k} \frac{\gamma}{2^{n-k+2}} = \frac{\gamma}{4}$. Thus, with high confidence, $\|\hat{p}_S - p_{S'}\|_\infty \leq \|\hat{p}_S - p_S\|_\infty + \|p_S - p_{S'}\|_\infty \leq \frac{\gamma}{4} + \frac{\gamma}{4} = \frac{\gamma}{2}$, implying that $S' \in \mathcal{B}$.

Since $resolution(\mathcal{B}) < \gamma$ implies $discrepancy(\mathcal{B}) < \frac{\epsilon}{2}$ (recall that $\gamma \leq \Gamma(\frac{\epsilon}{2})$), h must be $\frac{\epsilon}{2}$-good for every scenario in $\mathcal{B}$, including the scenario S'. Since $d_s(D, D') \leq 2^n \frac{\gamma}{2^{n-k+2}} = 2^{k-2}\gamma$, and since $\gamma \leq \frac{\epsilon}{2^{k-1}}$ we conclude that

$$error_{f,D}(h) \ \leq \ error_{f,D'}(h) + 2^{k-2}\gamma \ < \ \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon \qquad \blacksquare$$

3.2. *Hardness of RFA Learnability of k-DL*

Being an *information-theoretic* characterization of k-RFA learnability, the main importance of Theorem 1 is in providing a scheme for proving information-theoretic hardness results in the k-RFA model. We now apply this scheme to obtain hardness RFA results for the learnability of decision lists.

First notice that to disprove the k-RFA learnability of a function class $\mathcal{F}$, it is sufficient to find a *pair* of k-RFA hard $\mathcal{F}$-scenarios. If the k-RFA hardness of the pair is proved for $\mathcal{F}$ itself, then proper k-RFA learnability is disproved.

Now assume $\{\mathcal{F}_n\}_{n \geq n_0}$ is a family of function classes, where $\mathcal{F}_n$ is defined over the instance space $\{0, 1\}^n$. Naturally, we are seeking hardness results which hold for all $n \geq n_0$. We now show few constructions which expand a k-RFA hard pair of scenarios over the instance space $\{0,1\}^n$ into a $(k+1)$-RFA hard pair of scenarios over $\{0,1\}^{n+1}$. By inductively applying this construction (within a family which is closed under the construction), we will obtain a generalization of non-learnability results from a given n_0 to all $n \geq n_0$.

To enable compact descriptions of these constructions, we introduce few additional notations. For $b \in \{0, 1\}$ and $c \in \mathbf{R}^+$, let $\langle b, c \rangle$ be the *constant* scenario $\langle f, D \rangle$, where $f(x) = b$ and $D(x) = c$ for all $x \in \{0, 1\}^n$. For a scenario $S = \langle f, D \rangle$ and $b \in \{0, 1\}$, we denote by $b(S)$ the *projected* scenario $\langle b, D' \rangle$, where

$$D'(x) \triangleq \begin{cases} D(x) & \text{, if } f(x) = b, \\ 0 & \text{, otherwise.} \end{cases}$$

Note that D' is not necessarily a probability distribution over $x_1, \ldots, x_n$. Therefore, we denote by $\langle b(S) \rangle$ the "normalized" scenario, where $D'(\cdot)$ is normalized by $\sum_{x' \in f^{-1}(b)} D'(x')$. However, it will be convenient to abuse our notations by treating the non-normalized form as a scenario.

Notice that for every scenario S, $I \subseteq \{1, \ldots, n\}$, $z \in \{0, 1\}^k$, and $b \in \{0, 1\}$, we have $p_S(I, z, b) = p_{b(S)}(I, z, b)$, and therefore the following holds.

CLAIM 1 *Two scenarios S_1, S_2 are k-RFA equivalent if and only if both pairs $\langle 0(S_1) \rangle$, $\langle 0(S_2) \rangle$, and $\langle 1(S_1) \rangle$, $\langle 1(S_2) \rangle$ are k-RFA equivalent.*

Hence, in order to show equivalence of scenarios, it is sufficient to show the equivalence of their projections.

For scenarios $S_1 = \langle f_1^{(n)}, D_1^{(n)} \rangle$ and $S_2 = \langle f_2^{(n)}, D_2^{(n)} \rangle$, let $S_1 \otimes S_2$ denote the scenario $\langle f^{(n+1)}, D^{(n+1)} \rangle$, where

$$
\begin{aligned}
f^{(n+1)}(x_1, \ldots, x_{n+1}) &\triangleq x_{n+1} f_1^{(n)}(x_1, \ldots, x_n) \; + \; \overline{x}_{n+1} f_2^{(n)}(x_1, \ldots, x_n), \\
D^{(n+1)}(x_1, \ldots, x_n, 1) &\triangleq D_1^{(n)}(x_1, \ldots, x_n), \\
D^{(n+1)}(x_1, \ldots, x_n, 0) &\triangleq D_2^{(n)}(x_1, \ldots, x_n).
\end{aligned}
$$

Again, $S_1 \otimes S_2$ forms not necessarily a scenario over a probability distribution and we denote by $\langle S_1 \otimes S_2 \rangle$ the normalized scenario.

Lemma 2 **(Crossing Construction)** *If S_1, S_2 is a k-RFA hard pair of scenarios over $\{0,1\}^n$, then $\langle S_1 \otimes S_2 \rangle, \langle S_2 \otimes S_1 \rangle$ is a $(k+1)$-RFA hard pair over $\{0,1\}^{n+1}$.*

As a simple example consider the following two scenarios over $\{0,1\}$: Both distributions are uniform, $f_1^{(1)}(x_1) = x_1$, and $f_2^{(1)}(x_1) = \overline{x}_1$. Obviously, this pair of scenarios is 0-RFA hard. Now, if we apply the crossing construction, we get the following two scenarios: Both distributions remain the uniform distribution, $f_1^{(2)}(x_1, x_2) = \overline{x_1 \oplus x_2}$, and $f_2^{(2)} = x_1 \oplus x_2$.

Lemma 2 implies that this is a 1-RFA hard pair of scenarios. By applying the same construction iteratively $n-1$ times, we conclude that the class PAR_n is **not** $(n-1)$-RFA learnable (recall that PAR_n consists of two functions—the parity function over n variables, and its inverse). This result, which has already been shown in [3], demonstrates the gap between PAC learnability ($=$ n-RFA learnability) and $(n-1)$-RFA learnability (and similarly, between $(k+1)$-RFA learnability and k-RFA learnability [5]). It also immediately implies that the class DNF_n (which contains PAR_n) is not $(n-1)$-RFA learnable. We later apply this construction to obtain hardness results for the RFA learnability of k-DL, but first let us prove the lemma.

Proof of Lemma 2: Assume that $S_i = \langle f_i^{(n)}, D_i^{(n)} \rangle$, $i \in \{0,1\}$, is a k-RFA hard pair, and let $S_i' = \langle f_i^{(n+1)}, D_i^{(n+1)} \rangle = \langle S_i \otimes S_{3-i} \rangle$, $i \in \{0,1\}$. We first prove that S_1' and S_2' are $(k+1)$-RFA equivalent. Let $I = \{i_1, \ldots, i_{k+1}\} \subset \{1, \ldots, n+1\}$, $z \in \{0,1\}^{k+1}$, and $b \in \{0,1\}$. To verify that $p_{S_1'}(I, x, b) = p_{S_2'}(I, x, b)$, consider the following two cases:

- $n+1 \in I$. Assume first that $z_{k+1} = x_{n+1} = 0$. Then

$$
\begin{aligned}
f_1^{(n+1)}(x_1, \ldots, x_n, 0) &= f_2^{(n)}(x_1, \ldots, x_n) \\
D_1^{(n+1)}(x_1, \ldots, x_n, 0) &= D_2^{(n)}(x_1, \ldots, x_n)/2.
\end{aligned}
$$

  (Dividing $D_2^{(n)}(\cdot)$ by 2 guarantees that $D_1^{(n+1)}$ is a probability distribution). Hence $p_{S_1'}(I, z, b) = p_{S_2}(I', z', b)/2$ where $I' = I \setminus \{n+1\}$ and $z' = (z_1, \ldots, z_k)$. Similarly, $p_{S_2'}(I, z, b) = p_{S_1}(I', z', b)/2$, and by the assumption $p_{S_1} \equiv p_{S_2}$ we get $p_{S_1'}(I, x, b) = p_{S_2'}(I, x, b)$. The case $z_{k+1} = x_{n+1} = 1$ is symmetric.

- $n+1 \notin I$. Let $I' = I \cup \{n+1\}$, and let $z^a = (z_1, \ldots, z_{k+1}, a)$ for $z \in \{0,1\}^{k+1}$ and $a \in \{0,1\}$. Then

$$\begin{aligned} p_{S_1'}(I,z,b) &= p_{S_1'}(I',z^0,b) + p_{S_1'}(I',z^1,b) \\ &= p_{S_2'}(I',z^1,b) + p_{S_1'}(I',z^0,b) = p_{S_2'}(I,z,b) \end{aligned}$$

The first and last equality follows directly from the definitions of the probabilities $p_{S_1'}(I,z,b)$ and $p_{S_2'}(I,z,b)$, whereas the second equality follows from the fact that for every $x \in \{0,1\}^{n+1}$

$$\begin{aligned} f_1^{(n+1)}(x_1,\ldots,x_n,x_{n+1}) &= f_2^{(n+1)}(x_1,\ldots,x_n,\overline{x}_{n+1}), \\ D_1^{(n+1)}(x_1,\ldots,x_n,x_{n+1}) &= D_2^{(n+1)}(x_1,\ldots,x_n,\overline{x}_{n+1}). \end{aligned}$$

We also have to show that the pair S_1', S_2' has a non-zero discrepancy. Let $\epsilon > 0$ be such that every $h : \{0,1\}^n \to \{0,1\}$ is ϵ-bad for either S_1, S_2, and let $h' : \{0,1\}^{n+1} \to \{0,1\}$. Let $h_a(x_1,\ldots,x_n) = h'(x_1,\ldots,x_n,a)$. Then

$$\begin{aligned} error_{S_1'}(h') &= error_{S_2}(h_0)/2 \;+\; error_{S_1}(h_1)/2 \\ error_{S_2'}(h') &= error_{S_1}(h_0)/2 \;+\; error_{S_2}(h_1)/2 \end{aligned}$$

Since both h_0 and h_1 are ϵ-bad for either S_1 or S_2, it follows that h' is $\frac{\epsilon}{2}$-bad for either S_1' or S_2'. ■

Note that the crossing construction also applies to "non-normalized" scenarios, yielding a hard pair of "non-normalized" scenarios. We will use this observation, for instance, in the proof of Lemma 4.

While the pair of scenarios used in the crossing construction was k-RFA hard for every hypothesis class, the following construction yields a pair which is k-RFA hard for 1-DL. This will be used later to obtain a hardness result for proper-learnability of 1-DL.

LEMMA 3 **(Linear Construction)** *If S_1, S_2 is a pair of 1-DL-scenarios over $\{0,1\}^n$ which is k-RFA hard for 1-DL, and $1(S_1) = 1(S_2)$, then $\langle S_1 \otimes \langle 0, D_2^{(n)}\rangle\rangle$, $\langle S_2 \otimes \langle 0, D_1^{(n)}\rangle\rangle$ is a pair of 1-DL-scenarios over $\{0,1\}^{n+1}$ which is $(k+1)$-RFA hard for 1-DL.*

Proof: Let $S_i = \langle f_i^{(n)}, D_i^{(n)}\rangle$, $i \in \{1,2\}$ be a k-RFA equivalent pair of scenarios over $\{0,1\}^n$, and assume $1(S_1) = 1(S_2)$. Let $S_i' = \langle f_i^{(n+1)}, D_i^{(n+1)}\rangle = \langle S_i \otimes \langle 0, D_{3-i}^{(n)}\rangle\rangle$. First notice that if $f_i^{(n)}$ is a 1-decision list, then so is $f_i^{(n+1)}$ (just add the item $(x_{n+1}, 0)$ in front of the list $f_i^{(n)}$).

To prove $(k+1)$-RFA equivalence of S_1', S_2', it is sufficient to prove it for the pair $\langle b(S_i')\rangle$ (by Claim 1). For $b = 1$ the claim is obviously true, since $1(S_1) = 1(S_2)$ implies $\langle 1(S_1')\rangle = \langle 1(S_2')\rangle$. For $b = 0$, consider the scenarios $T_1 = \langle S_1 \otimes S_2\rangle$, $T_2 = \langle S_2 \otimes S_1\rangle$ (obtained by the crossing construction). Note that T_i differs from S_i' only on 1-instances, i.e., an instance labeled by 1. By Lemma 2 the pair T_1, T_2

is $(k+1)$-RFA equivalent, i.e., $p_{T_1}(I,z,0) = p_{T_2}(I,z,0)$, for all $I = \{i_1,\ldots,i_{k+1}\} \subseteq \{1,\ldots,n\}$ and $z \in \{0,1\}^{k+1}$. Hence,

$$\begin{aligned} p_{0(S_1)}(I,z,0) + p_{1(S_1)}(I,z,1) &= p_{T_1}(I,z,0) \\ &= p_{T_2}(I,z,0) \\ &= p_{0(S_2)}(I,z,0) + p_{1(S_2)}(I,z,1), \end{aligned}$$

and since $p_{1(S_1)}(I,z,1) = p_{1(S_2)}(I,z,1)$ (case $b=1$) we have $p_{0(S_1)}(I,z,0) = p_{0(S_2)}(I,z,0)$. Hence, the scenarios $\langle 0(S_1)\rangle$ and $\langle 0(S_2)\rangle$ are $(k+1)$-RFA equivalent.

Finally, we prove that pair S_1', S_2' has a non-zero discrepancy with respect to 1-DL. Let $\epsilon > 0$ be such that every 1-decision list over $\{0,1\}^n$ is ϵ-bad for either S_1 or S_2. Let h be a 1-DL over $\{0,1\}^{n+1}$, and let $h'(x_1,\ldots,x_n) = h(x_1,\ldots,x_n,1)$. Obviously, h' is a 1-decision list over $\{0,1\}^n$, and hence is ϵ-bad for either S_1 or S_2. But $error_{S_i'}(h) \geq error_{S_i}(h')/2$ for both $i=1$ and $i=2$, and hence h is $\frac{\epsilon}{2}$-bad for either S_1' or S_2'. ■

Using the linear construction, we prove the following theorem.

THEOREM 2 *1-DL is* **not** *properly $(n-2)$-RFA learnable.*

Proof: By Lemma 3 it is sufficient to show the existence of a 0-RFA hard pair of scenarios over $\{0,1\}^2$. Let $f_1^{(2)} = \langle (x_2,1),(x_1,1),(1,0)\rangle$, $f_2^{(2)} = \langle (\overline{x}_2,1),(\overline{x}_1,1),(1,0)\rangle$, $D_1^{(2)} = \frac{1}{3}$ for every $x \neq (0,0)$, and $D_2^{(2)}(x) = \frac{1}{3}$ for every $x \neq (1,1)$ (see Table B.1 in Appendix B). It is easy to verify that this forms a pair of scenarios over $\{0,1\}^2$ which is 0-hard for 1-DL. ■

Notice that the linear construction always adds the item $(x_{n+1},0)$ in front of the list, and therefore the lists which are used to obtain the hardness of proper $(n-2)$-RFA learnability have only two alternations of their labels. This implies that even the class 2-alt-1-DL is **not** properly $(n-2)$-RFA learnable.

Next we turn to non-proper learnability of 1-DL. In the next section we present an algorithm that learns this class, if the learner has access to at least half of the attributes (Theorem 6). We now show that this result is tight—no algorithm can learn 1-DL in the k-RFA model when $k < n/2$. We use the following construction.

LEMMA 4 **(Projecting Construction)** *If S_i, $i \in \{0,1\}$ is a pair of 1-DL-scenarios which is k-RFA hard over $\{0,1\}^n$, then the pair $S_i' = \langle U_i \otimes V_i\rangle$, where $U_i = S_i \otimes 0(S_{3-i})$ and $V_i = 1(S_{3-i}) \otimes \langle 1, 2^{-n}\rangle$ (see Table 1), is a pair of 1-DL-scenarios which is $(k+1)$-RFA hard over $\{0,1\}^{n+2}$.*

Before we start the proof let us mention the following simple claim.

CLAIM 2 *Given two k-RFA equivalent scenarios S_1, S_2, an arbitrary scenario S over n variables, and a real number $\alpha > 0$, the scenarios $S_1' = \langle \alpha S_1 \otimes S\rangle$ and*

Table 1. An illustration of the scenarios defined in Lemma 4.

	x_{n+1}					x_{n+1}		
x_{n+2}	1	0	S'_1		x_{n+2}	1	0	S'_2
1	S_1	$0(S_2)$	U_1		1	S_2	$0(S_1)$	U_2
0	$1(S_2)$	$\langle 1, 2^{-n}\rangle$	V_1		0	$1(S_1)$	$\langle 1, 2^{-n}\rangle$	V_2
$\hat{S}_1$	$\hat{U}_1$	$\hat{V}_1$			$\hat{S}_2$	$\hat{U}_2$	$\hat{V}_2$	

$S'_2 = \langle \alpha S_2 \otimes S\rangle$ are also k-RFA equivalent, where αS_i denotes the scenario obtained from S_i by multiplying all the probabilities by $\alpha > 0$.

Proof of Lemma 4: First note that that if S_i is a 1-DL-scenario, then so is S'_i (just add the items $(\overline{x}_{n+2}, 1)$, $(\overline{x}_{n+1}, 0)$ in front of the 1-decision list of the scenario S_i).

The proof that the non-zero discrepancy of the pair S_1, S_2 is preserved by S'_1, S'_2 is similar to the argument made in the proof of Lemma 2. Hence, it remains to show that the scenarios $\langle b(S'_i)\rangle$ are $(k+1)$-RFA equivalent for $b \in \{0, 1\}$. For the case $b = 0$ we get:

$$\begin{aligned} 0(U_i \otimes V_i) &= 0((S_i \otimes 0(S_{3-i})) \otimes (1(S_{3-i}) \otimes \langle 1, 2^{-n}\rangle)) \\ &= 0(S_i \otimes 0(S_{3-i})) \otimes 0(1(S_{3-i}) \otimes \langle 1, 2^{-n}\rangle) \\ &= 0(S_i \otimes S_{3-i}) \otimes \langle 0, 0\rangle \end{aligned}$$

Since the pair S_1, S_2 is k-RFA equivalent we obtain by Lemma 2 and Claim 1 that the scenarios $\langle 0(S_i \otimes S_{3-i})\rangle$, $i \in \{0, 1\}$ are $(k+1)$-RFA equivalent. Hence, by Claim 2, the scenarios $S'_i = \langle 0(U_i \otimes V_i)\rangle$ are $(k+1)$-RFA equivalent.

For the case $b = 1$, consider the following pair $\hat{S}'_i$, $i \in \{1, 2\}$ of scenarios over $\{0, 1\}^{n+2}$:

$$\hat{S}'_i \triangleq \langle \hat{U}_i \otimes \hat{V}_i\rangle, \quad \text{where} \quad \hat{U}_i \triangleq S_i \otimes 1(S_{3-i}), \quad \text{and} \quad \hat{V}_i \triangleq 0(S_{3-i}) \otimes \langle 1, 2^{-n}\rangle$$

(See Table 1). Similarly to the case $b = 0$, we get that the scenarios $\langle 1(\hat{S}'_i)\rangle$, $i \in \{0, 1\}$ are $(k+1)$-RFA equivalent, since

$$\begin{aligned} 1(\hat{U}_i \otimes \hat{V}_i) &= 1((S_i \otimes 1(S_{3-i})) \otimes (0(S_{3-i}) \otimes \langle 1, 2^{-n}\rangle)) \\ &= 1(S_i \otimes 1(S_{3-i})) \otimes 1(0(S_{3-i}) \otimes \langle 1, 2^{-n}\rangle) \\ &= 1(S_i \otimes S_{3-i}) \otimes (\langle 1, 0\rangle \otimes \langle 1, 2^{-n}\rangle) \end{aligned}$$

Note that $\hat{S}'_i$ represent the same scenarios as S'_i up to permutation of the variables x_{n+1}, x_{n+2}, and hence, the scenarios $\langle 1(S'_i)\rangle$, $i \in \{0, 1\}$ are also $(k+1)$-RFA equivalent. ■

The projecting construction can be used to prove the following theorem.

Theorem 3 *1-DL is* **not** $\lfloor \frac{n-1}{2} \rfloor$*-RFA learnable.*

Proof: It is sufficient to proof the theorem for odd n. (For n even, the $(\frac{n}{2}-1)$-RFA non-learnability of 1-DL$_{n-1}$ implies $(\frac{n}{2}-1)$-RFA non-learnability of 1-DL$_n$). By the projecting construction of Lemma 4, it is sufficient to show a pair $\langle f_1^{(3)}, D_1^{(3)}\rangle$, $\langle f_2^{(3)}, D_2^{(3)}\rangle$ of 1-DL–scenarios which is 1-RFA hard. Let:

$$\begin{aligned} f_1^{(3)} &= \langle(\overline{x}_3, 1), (\overline{x}_2, 0), (\overline{x}_1, 0), (1, 1)\rangle \\ f_2^{(3)} &= \langle(\overline{x}_3, 1), (\overline{x}_2, 0), (x_1, 0), (1, 1)\rangle \end{aligned}$$

Let $D_1^{(n)}(x) = \frac{1}{5}$ for $x \in \{(0,1,0), (1,0,1)\}$, otherwise $D_1^{(n)}(x) = \frac{1}{10}$, and let $D_2^{(n)}(x) = \frac{1}{5}$ for $x \in \{(0,0,1), (1,1,0)\}$, otherwise $D_1^{(n)}(x) = \frac{1}{10}$ (see Table B.2 in Appendix B). It is easy to verify the 1-RFA hardness of this pair. ∎

The 1-decision lists $f_i^{(n)}$ used in the above proof have $n-1$ alternations. However, we can define functions $g_i^{(n)} \in$ 2-alt-1-DL such that $\Pr_{x\in D_i}[g_i(x) \neq f_i(x)] = 0$. Hence, $\langle g_i^{(n)}, D_i^{(n)}\rangle$ is also a $\lfloor\frac{n-1}{2}\rfloor$-RFA hard pair, implying that even the class 2-alt-1-DL is **not** $\lfloor\frac{n-1}{2}\rfloor$-RFA learnable. Assuming again that n is odd, $g_1^{(n)}$, $g_2^{(n)}$ will be the following 2-alternating 1-decision lists

$$\begin{aligned} &\langle(\overline{x}_n, 1), (\overline{x}_{n-2}, 1), \ldots, (\overline{x}_3, 1), (\overline{x}_{n-1}, 0), (\overline{x}_{n-3}, 0), \ldots, (\overline{x}_2, 0), (\overline{x}_1, 0), (1, 1)\rangle, \\ &\langle(\overline{x}_n, 1), (\overline{x}_{n-2}, 1), \ldots, (\overline{x}_3, 1), (\overline{x}_{n-1}, 0), (\overline{x}_{n-3}, 0), \ldots, (\overline{x}_2, 0), (x_1, 0), (1, 1)\rangle. \end{aligned}$$

Assume $g_i(x) \neq f_i(x)$. Note, that there exists no $x = (x_1, \ldots, x_n)$ for which $g_i(x) = 0$ and $f_i(x) = 1$. Furthermore, $g_i(x) = 1$ and $f_i(x) = 0$ implies $x_3 = 0 \vee x_5 = 0 \vee \ldots \vee x_n = 0$. Observing that

$$f_i^{(n)}(x) = 0 \wedge D_i^{(n)}(x) > 0 \implies x_3 = x_5 = \ldots = x_n = 1,$$

we get $D_i(x) = 0$.

By combining the crossing and projecting constructions, we obtain yet another hardness result. Assume we have a pair $S_i = \langle f_i^{(c)}, D_i^{(c)}\rangle$ of c'-RFA hard scenarios where $f_i^{(c)} \in a$-alt-c''-DL$_c$. Now the crossing construction yields a pair $S_i' = \langle f_i^{(c+1)}, D_i^{(c+1)}\rangle$ of $(c'+1)$-RFA hard scenarios where $f_i^{(c+1)} \in a$-alt-$(c''+1)$-DL$_{c+1}$. We get by induction that $n-(c-c'')$-DL is **not** $n-(c-c')$-RFA learnable for constant c, c', c''. By Theorem 3 we know that 1-DL$_{k+1}$ is **not** $\lfloor k/2\rfloor$-RFA$_{k+1}$ learnable. Setting $c = k+1$, $c' = \lfloor k/2\rfloor$, $c'' = 1$ we get

THEOREM 4 *$(n-k)$-DL is* **not** *$(n-1-\lceil k/2\rceil)$-RFA learnable.*

Again, the lists used in the proof have at most two alternations, and thus even the class 2-alt-$(n-k)$-DL is **not** $(n-1-\lceil k/2\rceil)$-RFA learnable. Also note that for a fixed k, the class 1-alt-k-DL = k-DNF $\cup$ k-CNF is efficiently k-RFA learnable [3].

4. Distribution-free Learning of Decision Lists

In this section we contrast the hardness results, shown in [3] and in Section 3 of this paper, with two positive results for RFA learnability of decision lists. Both results are tight in terms of the amount of visible attributes used by the learning algorithms. In the analysis of both learning algorithms we make use of the following

LEMMA 5 *Let $\mathcal{A} \subseteq \{0,1\}^X$ be finite, and let D be a distribution on X. For every $A \in \mathcal{A}$, let $p(A) = \Pr_{x\in D}[A(x) = 1]$, and let $\hat{p}(A)$ be the empirical estimate of $p(A)$ based on a sample of size m. For every $\tau, \delta \in (0,1)$, and every $\beta \in (1,2]$, let $M(\tau,\beta,\delta,t) = \lceil \frac{\max(3,2\beta^2)}{\tau(\beta-1)^2} \log \frac{t}{\delta} \rceil$. For a sample of size $m \geq M(\tau,\beta,\delta,|\mathcal{A}|)$, the following holds with confidence $1-\delta$: For every $A \in \mathcal{A}$,*

$$p(A) > \beta\tau \implies \frac{p(A)}{\beta} < \hat{p}(A) < \beta p(A)$$
$$p(A) \leq \beta\tau \implies \hat{p}(A) < \beta^2\tau$$

Proof: For a sample of size m, and for every $A \in \mathcal{A}$, the Chernoff bounds imply [2]:

$$\Pr[\hat{p}(A) > \beta p(A)] \leq e^{-mp(A)(\beta-1)^2/3}$$
$$\Pr[\hat{p}(A) < p(A)/\beta] \leq e^{-mp(A)(\beta-1)^2/(2\beta^2)}$$

If we upper bound all inequalities (for every $A \in \mathcal{A}$) by $\frac{\delta}{|\mathcal{A}|}$, then all the estimates are within β of their true values. Solving for m yields $m = M(\tau,\beta,\delta,|\mathcal{A}|)$. ∎

4.1. $(n-1)$-RFA learnability of $(n-1)$-DL

In the PAC model, every class of Boolean function is clearly (information-theoretic) learnable. As implied by the hardness results in [3] and in Section 3 of this paper, this is not the case in the k-RFA model, when $k < n$. In fact, any class which contains both the parity function and its inverse (over n variables) is not $(n-1)$-RFA learnable. One may ask whether by excluding these two functions we gain $(n-1)$-RFA learnability. We answer this question affirmatively. (One may also add either the parity function or its inverse, but not both, without affecting the $(n-1)$-RFA learnability of the class). Notice that this class is actually the class $(n-1)$-DL, and hence by Theorem 4, is not $(n-2)$-RFA learnable. Thus, the result is also tight in terms of the visibility size.

The time and sample complexities of the learning algorithm are $O((n^3 2^n/\epsilon)(n + \ln(1/\delta)))$. Notice that, since $VCdim((n-1)\text{-DL}) = 2^n - 1$, every algorithm which PAC learns $(n-1)$-DL (let alone an RFA one) needs a sample size exponential in n ([13]).

THEOREM 5 *$(n-1)$-DL is properly $(n-1)$-RFA learnable with a sample and time complexity of*

$$O\left(\frac{n^3 2^n}{\epsilon}\left(n + \ln\frac{1}{\delta}\right)\right).$$

Proof: We start by showing information-theoretic learnability of this class, then elaborate on the details needed to construct a learning algorithm. By Theorem 1, the class is properly learnable if there is no $(n-1)$-RFA hard set for the class $(n-1)$-DL. Recall that all the scenarios in such a hard set should be $(n-1)$-RFA equivalent. However, we show that no two $(n-1)$-DL scenarios can be $(n-1)$-RFA equivalent. Furthermore, we show how to construct the target decision list from the $(n-1)$-RFA probabilities; this construction suggests the basic strategy for the algorithm presented later.

An $(n-1)$-RFA observation is made by fixing an index set $I = \{1, \ldots, n\} \setminus \{k\}$. For a target scenario S, we have denoted the probability of observing (z, b) via this index set by $p_S(I, z, b)$, where $z \in \{0, 1\}^{n-1}$ and $b \in \{0, 1\}$ (see Section 3). Notice that I and z identify a pair of instances which differ only in the k'th bit; we call such an unordered pair a k-edge. We use the notation $p_S(e, b) = p_S(I, z, b)$, where e is the edge determined by I and z, and S is the target scenario (omitted in the sequel). We show how to construct the target function f from the set of $(n-1)$-RFA probabilities $\{p(e, b)\}$.

First notice that if an edge is *pure*, *i.e.*, both of its instances have the same label b, then $p(e, 1-b) = 0$. Hence, the label of a pure edge e can be determined from the probabilities $p(e, 0)$ and $p(e, 1)$. Also notice that once the labels of an edge e have been determined, the value of any adjacent edge e' can be determined (e' is adjacent to e if they share a common point), as follows. Assume that $e = (x, y)$ is adjacent to $e' = (y, z)$, and that the label of y has been determined to be b. Then, if $p(e', 1-b) = 0$ (e' is pure) then the label of z is b, otherwise (e' is impure) it is $1-b$. Finally, there is at least one pure edge for every $(n-1)$-decision list—the edge which is determined by the first item in the list. This edge can be used as a *pivotal* edge for determining the labels of all the other edges in $n-1$ stages (in stage i determine the value of an edge whose distance from the pivotal edge is i).

The above argument proves that the exact $(n-1)$-RFA probabilities determine an $(n-1)$-decision list, and Theorem 1 guarantees that estimates based on a finite sample size are sufficient to identify the target list. However, to construct a learning algorithm, we need to refine the basic approach described above.

First we need to refine the notion of being a "pure" edge to allow for a small amount of error. To see why, consider two impure adjacent edges $e = (x, y)$ and $e' = (y, z)$, where the probability of drawing e is low, and the probability of drawing e' is high. Assume further that the label of x and z is 0, while that of y is 1. Having a small probability, the impure edge e might look to the learner like a pure edge. Deciding first the value of e (*i.e.*, of both x and y) to be 0, and knowing that e' is impure, the above approach assigns the wrong label to z, incurring a significant error. Hence, it is preferable in such a case to consider the "almost pure" edge e' as being pure, labelling z with 0.

Furthermore, we have to allow for the amount of impureness in a pure edge to increase throughout the stages of the algorithm. This is due to the fact that this impureness is only estimated. Consider again the previous example, and assume that, in deciding the label of the edge e, we allow for an empirical impureness of size τ. If our estimates are within a factor of β of the real probabilities, then the real impureness could be of size $\beta\tau$. (A good choice for β will be determined later.) Hence, in deciding the label of the edge e', the real impureness that should be tolerated is $\beta\tau$, so the estimated one should be $\beta^2\tau$. Thus, if τ_i is the amount of empirical impureness allowed in stage i, τ_{i+1} should satisfy $\tau_{i+1} = \beta^2\tau_i$. Since $\tau_{n-1} = \beta^{2n-2}\tau_0$, and since the error incurred by each impureness is bounded by $\beta\tau_i$, choosing $\tau_0 = \frac{\epsilon}{\beta^{2n-1}2^n}$ guarantees that the overall error is bounded by ϵ.

The learning algorithm works as follows. For every $1 \leq k \leq n$ it takes a sample of size m' via the index set $\{1, \ldots, n\} \setminus \{k\}$, and, for every k-edge, it estimates the probability $p(e, b)$; let $\hat{p}(e, b)$ be this estimate. We will determine the value of m' later. An edge e is b-pure at stage i if $\hat{p}(e, 1-b) < \tau_i$, (Note that e can be both b-pure and $(1-b)$-pure). e is impure if it neither 0-pure nor 1-pure.

Having the estimates, the algorithm first searches for a pivotal edge—a pure edge $e = (x, y)$ (pure at stage 0), and sets $h(x) = h(y) = b$ if e is b-pure. Then, at any stage $1 \leq i \leq n-1$, the algorithm sets the value of any edge $e' = (y, z)$ which is adjacent to an edge $e = (x, y)$ whose value has been set at stage $i-1$: if e' is $h(y)$-pure at stage i then $h(z) = h(y)$, otherwise $h(z) = 1 - h(y)$.

We now prove that h is ϵ-good for the target function f and the target distribution D, if the following conditions hold for all the estimates $\hat{p}(e, b)$: *if* $p(e, b) \geq \beta\tau_0$ *then* $\hat{p}(e, b)$ *is within a factor of* β *from* $p(e, b)$ *(in both directions), otherwise* $(p(e, b) < \beta\tau_0)$ $\hat{p}(e, b) \leq \beta^2\tau_0$.

Since $\beta\tau_i < \frac{\epsilon}{2^n}$ at any stage i, it is sufficient to prove that for every instance x, if $D[x] \geq \beta\tau_i$, and $h(x)$ has been set in stage i, then $h(x) = f(x)$. The proof is by induction on i. If the pivotal point $e = (x, y)$ is b-pure at stage 0, then it satisfies $\hat{p}(e, 1-b) < \tau_0$, and $p(e, 1-b) < \beta\tau_0$, hence the claim is true for $i = 0$. Assume that the label of $e' = (y, z)$ is set in stage $i+1$, using an edge $e = (x, y)$ whose second label has been set in stage i. If $D[z] \geq \beta\tau_{i+1}$, then $\hat{p}(e', f(z)) \geq \tau_{i+1}$, and therefore e' cannot be $(1 - f(z))$-pure. Hence, $h(z) = 1 - f(z)$ only when e' is impure, and $h(y) = 1 - h(z) = f(z) \neq f(y)$. But then, by the induction hypothesis, $D[y] < \beta\tau_i$, implying $\hat{p}(e', f(y)) < \beta^2\tau_i = \tau_{i+1}$, contradicting the impureness of e'.

It remains do determine appropriate values for β and m', and to show that the above assumptions on parameters $\hat{p}$ and p are valid with probability at least $1 - \delta$. For fixed k, there are 2^{n-1} k-edges, and therefore 2^n probability parameters that have to be empirically estimated from m' examples. Since we have n samples (one for each $1 \leq k \leq n$), we want all estimates based on a sample of size m' to be accurate with confidence $1 - \frac{\delta}{n}$. Hence, by Lemma 5, we need a sample of size $m' = M(\tau_0, \beta, \frac{\delta}{n}, 2^n) = O\left(\frac{n^2 2^n}{\epsilon}\left(n + \ln\frac{1}{\delta}\right)\right)$, and thus $m = nm' = O\left(\frac{n^3 2^n}{\epsilon}\left(n + \ln\frac{1}{\delta}\right)\right)$.

As for the time complexity of the learning algorithm, first notice that the number of edges is $\frac{n}{2}2^n = t$. Computing the estimates from a sample of size m can be done in $O(m)$, and finding the pivotal edge can be done in $O(t)$. Choosing a pivotal edge induces an order on the visit of the other edges, considering each edge only once (at a stage i which is its distance from the pivotal edge). As $t = O(m)$, the overall time complexity is $O(m)$. ■

4.2. $(n-k)$*-RFA Learnability of* $\mathbf{1}$*-DL using* k*-DL*

Theorem 3 shows that 1-DL is not $\lfloor \frac{n-1}{2} \rfloor$-RFA learnable (using any hypothesis class). We now contrast this result by showing that for $k \le n/2$, this class is $(n-k)$-RFA learnable using the hypothesis class k-DL. This shows that 1-DL is learnable if and only if at least half of the attributes are visible in each example. The learning algorithm is efficient for $k = O(\log n)$, and is proper for $k = 1$. Note that, by Theorem 2, 1-DL is not properly $(n-2)$-RFA learnable.

THEOREM 6 *For every every* $1 \le k \le n/2$*, the class* 1*-DL is* $(n-k)$*-RFA learnable using k-DL, with sample complexity of* $O\left(\frac{k^3 n^2 2^{2k}}{\epsilon} \log \frac{n}{\delta}\right)$*, and with time complexity of* $O\left(\frac{k^4 n^2 2^{2k}}{\epsilon} \log \frac{n}{\delta}\right)$

Proof: As in the proof of Theorem 5, we start by showing information-theoretic learnability of this class. That is, given a target 1-decision list, we show how to construct a k-decision list from the exact $(n-k)$-RFA probabilities. Then, we show how to settle for good estimates of these probabilities in order to find a good approximation of the target list.

First notice that for every 1-decision list there is an equivalent one in which each variable appears only once (cf. [26]). Hence, we may assume that in both the target and the hypothesis list each variable appears only once.

We construct the hypothesis h gradually. At any intermediate stage, there are instances for which h is not defined. For any such instance x, we denote $h(x) = \emptyset$. We also denote by $|h|$ the number of items in h. Given a partial decision list h and a term t, define $B(h,t) = \{x \in \{0,1\}^n : h(x) = \emptyset,\ t(x) = 1\}$. That is, $B(h,t)$ is the subset of instances which are not defined by the partial list h, and are satisfied by the term t. We call such a subset *block*. For a target scenario $S = \langle f, D\rangle$, and a block $B = B(h,t)$, let $p_S(B,b) = \Pr_{x\in D}[x \in B, f(x) = b]$ (we henceforth omit the subscript S). Notice that if $p(B(h,t), 1-b) = 0$ and h is consistent with f, then appending (t,b) to h preserves this consistency. In such a case we say that the block $B(h,t)$ is *b-pure*.

The construction of the decision list from the $(n-k)$-RFA probabilities can be done in three phases. The first two phases are based on the following observation (cf. [24]): If f is a 1-decision list, and h is a partial 1-decision list, $|h| < n$, then there is a literal l for which the block $B(h,l)$ is pure (take the first literal l in f which does not appear in h). Hence, the construction is essentially based on searching for pure blocks; if $B(h,l)$ is b-pure, then we can add the item (l,b) to h. However, notice that the probability $p(B(h,l),b)$ is a $(|h|+1)$-RFA probability, and recall that we can only use $(n-k)$-RFA probabilities. Hence, as long as $|h| < n-k$, using the $(n-k)$-RFA probabilities to find pure blocks is straightforward. This forms Phase 1 of the construction.

How can we find pure blocks when $|h| \ge n-k$? Phase 2 of the construction is base on the following observation. Let h_b be the list obtained from h by deleting all the $(1-b)$-items, where a b-item is an item of the form (l,b). We claim that if $|h_b| < n-k$ for both $b = 0$ and $b = 1$, then there is a literal l, and there is

$b \in \{0, 1\}$, such that $B(h_{1-b}, l)$ is a b-pure block. To see why, recall that there is an item (l, b) such that $B(h, l)$ is b-pure. But if $h_{1-b}(x) = \emptyset$ and $h(x) \neq \emptyset$ then necessarily $h(x) = b = f(x)$, and therefore the block $B(h_{1-b}, l)$ is also b-pure. Hence, we can continue the construction as long as $|h_b| < n - k$ for both $b = 0$ and $b = 1$ (searching for a b-pure block $B(h_{1-b}, l)$, for either $b = 0$ or $b = 1$, and appending the item (l, b) to h once such a pure block is found).

Finally, assume that $|h_b| = n - k$ (for either $b = 0$ or $b = 1$), but $|h| < n$ (h is not yet complete). If the next pure block $B(h, l)$ is $(1-b)$-pure, this cannot be revealed using the $(n-k)$-RFA probabilities. However, notice that the number of variables which are not in h_b is is bounded by $k \leq n - k$. Consider a block $B = B(\emptyset, t)$, where $\emptyset$ is undefined for every x, and is a k-term over the variables which are not in h_b. (Notice that the probability $p(B, b)$ is an $(n-k)$-RFA probability). Since $B(h_b, t)$ is a singleton $\vec{x}\}$ (the assignment to every variables is determined by either h_b or t), if $B(\emptyset, t)$ is impure, it must be due to $f(x) = 1 - b$. Hence, in that case $(t, 1-b)$ can be appended to the list. After appending all the $(1-b)$-singletons to the list h (*i.e.*, all the instances for which $h(x) = \emptyset$ and $f(x) = 1 - b$), we can close the list with the item $(1, b)$, concluding with a k-decision list which is equivalent to the target 1-decision list (each step of the construction of h preserves consistency, and the final h is defined over the entire instance space).

When using estimates $\hat{p}(B, b)$ of the $(n-k)$-RFA probabilities $p(B, b)$, impure blocks may look like pure blocks to the learner, and thus the constructed list h is only an approximation of the target list. Hence, we have to refine our notion of pureness to allow for a small amount of impureness which might increase throughout the stages of the algorithm. We assume that each estimate is within a factor of β of the true probability whenever the true probability is at least $\beta\tau$ (the value of τ and the sample size needed for that to hold will be determined later), and analyze the error incurred by the the three phases of the construction.

At each stage of Phase 1 ($|h| < n - k$) we search for empirically pure block B ($\hat{p}(B, b) = 0$). If the block is actually impure, then by Lemma 5 $p(B, b) \leq \beta\tau$, so the overall error incurred in this phase is $\epsilon_1 = (n-k)\beta\tau$.

Next consider Phase 2 ($|h| \leq n - k$, $|h_0| < n - k$, $|h_1| < n - k$). Recall that at each stage of this phase we search for a b-pure block $B(h, l)$, but can only estimate the probability $p(B(h_{1-b}, l), b)$. Since h_b is not necessarily consistent with f (due to the error incurred by previous stages), the block $B(h_{1-b}, l)$ may include a small amount of impureness. To recognize that (l, b) is a proper item to append at stage i, we allow for an empirical impureness τ_i, which includes all the errors incurred by previous stages of Phases 1 and 2. That is, $\tau_i = \beta\epsilon_1 + \beta\sum_{j=1}^{i-1}\beta\tau_i$, implying $\tau_i = (n-k)\beta^2\tau(1+\beta^2)^{i-1}$. Hence, The overall error incurred by this phase is bounded by $\epsilon_2 = \sum_{i=1}^{k}\beta\tau_i = (n-k)\beta\tau((1+\beta^2)^k - 1)$.

Each impure block $B(\emptyset, t)$ found in Phase 3 determines the value of a specific instance $x \in B(h_b, t)$. Let $B' = B(\emptyset, t) \setminus \{x\}$. Then $B' \subseteq \{x : h(x) = b\}$. The impureness of $B(\emptyset, t)$ might be due to the errors in B' incurred by previous stages. This error can be bounded by $\epsilon_1 + \epsilon_2$. Hence, we can allow for an the empirical impureness of $\beta(\epsilon_1 + \epsilon_2)$ for the block $B(\emptyset, t)$, and hence appending the item $(t, 1-b)$ to the list incurs an error of at most $\beta^2(\epsilon_1 + \epsilon_2)$. Since at most

2^k instances are determined in this phase, the overall error incurred by Phase 3 is $\epsilon_3 = 2^k\beta^2(\epsilon_1+\epsilon_2) = (n-k)\beta^3\tau(1+\beta^2)^k$.

The overall error is $error_{f,D}(h) \le \epsilon_1+\epsilon_2+\epsilon_3 = (n-k)\tau\beta(1+\beta^2 2^k) < (n-k)\tau\beta^2(1+\beta^2)^k 2^{k+1}$, and choosing $\tau = \frac{\epsilon}{(n-k)\beta^2(1+\beta^2)^k 2^{k+1}}$ guarantees that $error_{f,D}(h) < \epsilon$. Notice that for $\beta = (1+\frac{2}{k})^{\frac{1}{2}}$ we have $\tau = O(\frac{\epsilon}{n2^{2k}})$ and $\frac{1}{(\beta-1)^2} = O(k^2)$.

Now we can determine the sample size needed in each phase, so that any estimate of a probability larger than $\beta\tau$ is within a factor of β of that probability. As the overall confidence in all the estimates should be $1-\delta$, we require a confidence of $1-\frac{\delta}{3}$ for all the estimates used by each one of the three phases. Recall that by Lemma 5, a sample of size $M(\tau,\beta,\delta,t) = \lceil \frac{\max(3,2\beta^2)}{\tau(\beta-1)^2}\log\frac{t}{\delta}\rceil$ is sufficient to ensure (with confidence $1-\delta$), that for a set of t estimates, each estimates is within a factor of β of the true probability, whenever the probability is greater than $\beta\tau$.

The first phase consists of $n-k$ stages. At each stage, having the current hypothesis h, we search for a literal l, for which the block $B(h,l)$ is pure. Let S be the variables which appear in h. We can decompose the variables which are not in h into $\lceil\frac{n-|h|}{n-k-|h|}\rceil \le k+1$ sets, and for each such set T, focus our attention on $S\cup T$. Hence, we need $(n-k)(k+1)$ samples in this phase. For each such sample we need to estimate $4(n-k-|h|)$ probabilities (estimating $p(B(h,l),b)$ for every new literal l and every $b\in\{0,1\}$). By Lemma 5 a sample of size $M(\tau,\beta,\frac{\delta}{3(n-k)(k+1)},4(n-k))$ is sufficient, hence a sample of size $(n-k)(k+1)M(\tau,\beta,\frac{\delta}{3(n-k)(k+1)},4(n-k)) = O(\frac{k^3n^2 2^{2k}}{\epsilon}\log\frac{n}{\delta})$ is sufficient for Phase 1.

At any stage of Phase 2 we search for a $(1-b)$-pure block $B(h_b,l)$, where $|h_b| < n-k$. Let S_b be the set of variables in h_b. Then, for every variable v not in h (at most k) we need to focus our attention on $S_b\cup\{v\}$, for either $b=0$ or $b=1$. Hence, $2k$ samples are sufficient at each stage, and $2k^2$ samples are sufficient for the entire phase. For each sample we need to estimate at most k probabilities, hence by Lemma 5, a sample of size $M(\tau,\beta,\frac{\delta}{6k^2},k)$ is sufficient for each stage, and $2k^2M(\tau,\beta,\frac{\delta}{6k^2},k) = O(\frac{k^3n^2 2^{2k}}{\epsilon}\log\frac{n}{\delta})$ is sufficient for the entire Phase 2.

In Phase 3 we take one sample and estimate at most 2^k probabilities. Hence a sample of size $M(\tau,\beta,\frac{\delta}{3},2^k) = O(\frac{k^3n2^k}{\epsilon}\log\frac{1}{\delta})$ is sufficient for this phase, and the overall sample complexity of the algorithm is $O\left(\frac{k^3n^2 2^{2k}}{\epsilon}\log\frac{n}{\delta}\right)$.

The time complexity is essentially determined by the time needed for estimating the k-RFA probabilities. In the first two phases this is obvious. To see this for the third phase note that an item (t,b) is added only if the corresponding $B(\emptyset,t)$ is not b-pure. But this can happen only if an example (labeled b) in this block is drawn. Hence for each such item at least one example has to be drawn.

In Phase 1 each example drawn changes at most k empirical probabilities, whereas Phases 2 and 3 each example drawn changes only one empirical probability. Hence, the time complexity is $O(km)$. ∎

4.3. Summary of Results about Distribution-free Learning of Decision Lists

Table 3 (Table 4, resp.) summarizes the results shown (or directly implied by them) in this paper and known results from literature ([24]) for proper (improper, resp.) RFA-learning. A legend is given in Table 2.

Table 2. Legend for Tables 3 and 4.

Symbol	Meaning
$+e$	efficiently learnable
$+$	learnable (maybe efficiently)
$+i$	only inefficiently learnable
$-$	not learnable (even not inefficiently)
?	learnability unknown
$-/?$	"$-$" if n is even, and "?" if n is odd

Table 3. Summary of results for proper b-RFA-learning of a-DL.

$a \backslash b$	$0, \ldots, n-2$	$n-1$	n
0	$+e$	$+e$	$+e$
1	$-$	$+e$	$+e$
$2, \ldots, k$	$-$	?	$+e$
$n-k, \ldots, n-2$	$-$	?	$+i$
$n-1$	$-$	$+i$	$+i$
n	$-$	$-$	$+i$

Table 4. Summary of results for improper b-RFA-learning of a-DL.

$a \backslash b$	$0, \ldots, k$	$\lceil n/2 \rceil - 1$	$\lceil n/2 \rceil$	$n - \Theta(\log n)$	$n-k$	$n-4$	$n-3$	$n-2$	$n-1$	n
0	$+e$	$+e$	$+e$	$+e$	$+e$	$+e$	$+e$	$+e$	$+e$	$+e$
1	$-$	$-$	$+$	$+e$	$+e$	$+e$	$+e$	$+e$	$+e$	$+e$
2	$-$	$-$	$-/?$	?	?	?	?	?	$+$	$+e$
3	$-$	$-$	$-$	?	?	?	?	?	$+$	$+e$
4	$-$	$-$	$-$	?	?	?	?	?	$+$	$+e$
$k-1$	$-$	$-$	$-$	?	?	?	?	?	$+$	$+e$
k	$-$	$-$	$-$	?	?	?	?	?	$+$	$+e$
$n-k$	$-$	$-$	$-$	$-$	$-$	?	?	?	$+i$	$+i$
$n-k+1$	$-$	$-$	$-$	$-$	$-$	?	?	?	$+i$	$+i$
$n-6$	$-$	$-$	$-$	$-$	$-$	$-$	?	?	$+i$	$+i$
$n-5$	$-$	$-$	$-$	$-$	$-$	$-$	?	?	$+i$	$+i$
$n-4$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	?	$+i$	$+i$
$n-3$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	?	$+i$	$+i$
$n-2$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$+i$	$+i$
$n-1$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$+i$	$+i$
n	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$+i$

5. Learning Decision Lists under Fixed Distributions

Let D be an arbitrary, but fixed, distribution over $X = \{0,1\}^n$. Two functions f, g in n Boolean variables are called *D-equivalent* if $\Pr_{x \in D}[f(x) \neq g(x)] = 0$. We say that a Boolean term t *separates* f, g if $\Pr_{x \in D}[f(x) = 1 \,|\, t(x) = 1] \neq \Pr_{x \in D}[g(x) = 1 \,|\, t(x) = 1]$. For fixed distribution, the assertion of Theorem 1 can be reformulated as follows:

THEOREM 7 *1. If $\mathcal{F}$ contains two functions f, g which are not D-equivalent and cannot be separated by any k-term t, then there exists no hypothesis class $\mathcal{H}$ such that $\mathcal{F}$ is k-RFA learnable using $\mathcal{H}$.*

2. If every two functions f, g from $\mathcal{F}$ which are not D-equivalent can be separated by some k-term t, then $\mathcal{F}$ is properly k-RFA learnable.

Proof:

1. We claim that $\langle f, D\rangle, \langle g, D\rangle$ form a k-RFA hard set of scenarios even for hypothesis class 2^X. Obviously, both scenarios are k-RFA equivalent, since they cannot be separated by any k-term. Observe next that the symmetric difference of f, g has a strictly positive probability γ because f, g are not D-equivalent. Choose $\epsilon = \gamma/3$. Certainly, no hypothesis can be ϵ-good for both, f and g. Thus the claim follows. By Theorem 1, $\mathcal{F}$ is not k-RFA learnable using 2^X (and thus not k-RFA learnable using any hypothesis class).

2. Assume that $\mathcal{F}$ is not properly k-RFA learnable. We have to show that $\mathcal{F}$ contains two functions f, g that are not D-equivalent and cannot be separated by any k-term t. By Theorem 1, there exists a k-RFA hard set of scenarios for $\mathcal{F}$. Since D is fixed, this hard set has the form $\{\langle f_1, D\rangle, \ldots, \langle f_r, D\rangle\}$. If the functions f_i were pairwise D-equivalent, hypothesis f_1 would be ϵ-good (even 0-good) for all potential targets f_i. This would contradict the k-RFA hardness of the set. Thus there exist two functions f, g in this set which are not D-equivalent. Since $\langle f, D\rangle$ and $\langle g, D\rangle$ are k-RFA equivalent, there cannot exist a separating k-term t. This concludes the proof. ■

The essential message of Theorem 7 is that k-RFA hard sets (if there are any) can always be formed by two hypotheses being D-inequivalent and nonseparable by any k-term.

THEOREM 8 *For any fixed distribution D over $\{0,1\}^n$, k-DL is properly k-RFA learnable for all $1 \leq k \leq n$.*

Proof: According to Theorem 7 it suffices to show that any $f, g \in k$-DL, which are not D-equivalent, can be separated by some k-term t. Let L be a decision list with items (t_i, b_i) for $1 \leq i \leq r$ representing f, and L' a decision list with items (t'_j, b'_j) for $1 \leq j \leq r$ representing g. Here, we assumed, for the sake of simplicity, that both lists have the same length r (using redundant items for one list if necessary). Let L_q and L'_q be the lists starting both with the sublist

$S_q = [(t_1, b_1), (t'_1, b'_1), \ldots, (t_q, b_q), (t'_q, b'_q)]$ and ending with the remaining items of, respectively, L and L'. Let f_q, g_q be the functions represented by these lists, respectively. Note that $L_r = L'_r$. The maximal index q such that f is D-equivalent to f_q and g is D-equivalent to g_q is therefore smaller than r. Let $P \subseteq \{0,1\}^n$ be the set of Boolean vectors of positive probability under D, A the subset of vectors from P which satisfy one of the terms in S_q, $B = P \setminus A$, $T = \{x \,|\, t_{q+1}(x) = 1\}$, and $T' = \{x \,|\, t'_{q+1}(x) = 1\}$. Since L_q represents f and L'_q represents g up to D-equivalence, it follows that:

$$\forall x \in A : f(x) = g(x), \quad \forall x \in B \cap T : f(x) = b_{q+1},$$
$$\forall x \in B \cap T' : g(x) = b'_{q+1}.$$

The maximality of q implies that:

$$\exists x \in B \cap T : g(x) \neq b_{q+1} \text{ or } \exists x \in B \cap T' : f(x) \neq b'_{q+1}.$$

It easily follows that t_{q+1} or t'_{q+1} separates f, g. ∎

The running time and the sample size of the learning algorithm, given implicitly in the proof of Theorem 8, depend on the specific choice of k and D, and are certainly not polynomial in general. However, it is known that with respect to the uniform distribution, 1-DL is efficiently 1-RFA learnable [11].

6. *k*-RFA Learnability of *k*-TOP

In this section we prove two positive results for the learnability of k-TOP in the k-RFA model. First, we prove that k-TOP is weakly k-RFA learnable in a distribution-independent sense. We also show that k-TOP is sample-efficiently k-RFA learnable with respect to the uniform distribution. In the next section we combine the weak learning observation of this section with one of our negative results for decision list learning to show that weak and strong learnability are *not* equivalent in the k-RFA model.

It should be noted that, as is standard in Fourier analysis, we assume throughout this section and the next that Boolean functions map to $\{-1,+1\}$ unless otherwise stated. This includes decision lists, so we will assume a slightly different definition here in which the b_i defining a decision list are in $\{-1,+1\}$ rather than in $\{0,1\}$.

6.1. *Weak Learnability of k-TOP*

Our first observation is that the class k-TOP of thresholds of k-parities is weakly learnable from a k-RFA oracle, and the learning is polynomial-time for constant k. This is a direct result of the following lemma, which is a slight modification of a similar result in [17].

LEMMA 6 *Let f be any k-TOP of size s and D any distribution over the domain of f. Then there exists a parity χ_a with $|a| \leq k$ such that*

$$|\Pr_{x\in D}[f=\chi_a]-\tfrac{1}{2}|\geq \frac{1}{2s}.$$

We use the notation $\tilde{O}(\cdot)$ in the following theorem and elsewhere to represent the standard big-O notation with log factors suppressed.

THEOREM 9 *k-TOP is weakly k-RFA learnable in time $\tilde{O}(n^{k+1}s^2)$.*

Proof Sketch: By standard Chernoff bound arguments (see, e.g., [18]), given an example oracle for f and a fixed χ_a we can produce an estimate of $\Pr_{x\in D}[f=\chi_a]$ that, with probability at least $1-\delta$ over the random draws by the example oracle, is within $1/(8s)$ of the true value. Furthermore, the algorithm producing this estimate runs in time $O(ns^2\log\delta^{-1})$, where the algorithm is assessed unit time for each call to the example oracle. Notice also that a k-RFA oracle suffices rather than a full example oracle if the parity χ_a is a k-parity.

Thus if we know the size s of the target then we can find a weak approximator to k-TOP f by querying a k-RFA oracle in order to estimate the correlation of each of the $O(n^k)$ k-parities with f and choosing as the weak hypothesis any parity having correlation of at least $3/(8s)$ (the δ used in each estimate must of course be set sufficiently small to assure that the overall confidence of the procedure is within that allowed to the weak learner). That such a weak hypothesis exists is guaranteed by the lemma above. Because only $O(n^k)$ estimates are performed and each estimate requires time $\tilde{O}(ns^2)$, this procedure satisfies the claimed time bound.

If the size s of the target function f is not known, a standard guess-and-double technique can be applied (see, e.g., [18]). That is, we can start with $s=1$. If no k-parity has correlation $3/8$ with the target, we double s and try again. Notice that this process will converge to a weak approximator in $\log s$ stages, again with high probability for appropriate choices for δ. ■

6.2. Polynomial Sample Size for Uniform $\boldsymbol{k}$-TOP Learning

Our most general positive result for k-TOP is that for constant k, k-TOP is sample-efficiently k-RFA learnable with respect to uniform. To obtain this result, we show that any two noticeably different k-TOP functions will differ noticeably in at least one Fourier coefficient of order k or less. This says that estimates of these low-order Fourier coefficients provide the information necessary to closely approximate a k-TOP function. Since for constant k these low-order Fourier coefficients can be efficiently estimated from a uniform-distribution k-RFA oracle, k-TOP is sample-efficiently k-RFA learnable with respect to uniform.

LEMMA 7 *Let $f:\{0,1\}^n\to\{-1,+1\}$ be a k-TOP of size s and let $I_k=\{a\in\{0,1\}^n : |a|\leq k\}$. Also, let ϵ be any positive constant, and let $g:\{0,1\}^n\to\{-1,+1\}$ be such that for all $a\in I_k$, $|\hat{f}(a)-\hat{g}(a)|\leq\epsilon/s$. Then $\Pr[f=g]\geq 1-\epsilon$.*

Proof: Because f is a k-TOP of size s, there exists a function $F=\sum_{a\in I_k} w_a\chi_a$ on the domain of f such that $f=\operatorname{sign}(F)$, the weights w_a of F are all integer-valued, and $\sum_{a \in I_k} |w_a| \leq s$. But by the definition of the Fourier transform we can also write $F = \sum_a \hat{F}(a)\chi_a$. Thus we see that $\hat{F}(a) = w_a$ for all $|a| \leq k$ and $\hat{F}(a) = 0$ for all $|a| > k$. Now applying the generalized Parseval's identity we obtain the following:

$$\mathbf{E}[|F|] = \mathbf{E}[f \cdot F] = \sum_a \hat{f}(a)\hat{F}(a) = \sum_{a \in I_k} \hat{f}(a)\hat{F}(a).$$

By our assumption about the relation between f and g we then have that

$$\mathbf{E}[|F|] \leq \sum_{a \in I_k} \hat{g}(a)\hat{F}(a) + \sum_{a \in I_k} \frac{\epsilon}{s}|\hat{F}(a)| \leq \sum_{a \in I_k} \hat{g}(a)\hat{F}(a) + \epsilon.$$

Now note that again applying Parseval we have

$$\sum_{a \in I_k} \hat{g}(a)\hat{F}(a) = \sum_a \hat{g}(a)\hat{F}(a) = \mathbf{E}[g \cdot F].$$

Thus $\mathbf{E}[|F|] - \mathbf{E}[g \cdot F] \leq \epsilon$. Furthermore, since g is Boolean, every one of the terms $g(x)F(x)$ in $\mathbf{E}[g \cdot F]$ has magnitude $|F(x)|$. This means that $\mathbf{E}[g \cdot F] \leq \mathbf{E}[|F|]$, with equality achieved if and only if $f \equiv g$. Furthermore, since $|F(x)| \geq 1$ for all x (recall that the $\hat{F}(a)$ are integers and by definition of the sign function $F(x) \neq 0$), each x such that $f(x) \neq g(x)$ adds at least 2^{-n} to the difference $\mathbf{E}[|F|] - \mathbf{E}[g \cdot F]$. Therefore f and g can differ on at most an ϵ fraction of the x's. ■

THEOREM 10 *Let s be the size of a target k-TOP function and ϵ and δ the standard PAC accuracy and confidence parameters, respectively. Then k-TOP is learnable from a uniform-distribution k-RFA oracle with sample complexity $O(n^k ks^2 \log(n/\delta)/\epsilon^2)$ and in time at most singly exponential in n, s, and k.*

Proof: By Chernoff, a sample of size $O(ks^2 \log(n/\delta)/\epsilon^2)$ from the k-RFA oracle is sufficient to estimate, with probability at least $1 - \delta/n^k$, each of the k-order or less Fourier coefficients of f to within $\epsilon/2s$. By the preceding lemma, we then know that any function which has low-order Fourier coefficients within $\epsilon/2s$ of those we have estimated will be an ϵ-approximator to f. And there is at least one such k-TOP—f itself—which satisfies this requirement.

One algorithm for finding this k-TOP then is to systematically construct various k-TOP expressions in such a way that all possible k-TOP functions will eventually be represented. This can be done by writing down lexicographically successive bit strings and checking each to see if it represents a valid encoding (in, say, ASCII) of a k-TOP. $O(n2^{ks})$ is a crude upper bound on the number of strings we will write down before encountering f. For each k-TOP constructed this way we can compute its Fourier coefficients using the Fast Fourier Transform in time singly exponential in n. We then compare the Fourier coefficients of each constructed function with those previously estimated for f until a match is found. ■

7. Weak and Strong k-RFA Learning

In the PAC model of learning, weak learnability implies strong learnability [25]. Existing proofs for this fact are based on the notion of hypothesis boosting. Therefore, an obvious approach to turning the weak learning result of the previous section into a strong learning result is to apply boosting. However, all currently known boosting algorithms work by running the weak learner multiple times, each time on a distribution which is defined in part by the performance of earlier weak hypotheses on instances. This presents a significant problem in the k-RFA model: to determine the appropriate probability weight to assign to an instance, we need to know how earlier hypotheses classify the instance, which requires that each hypothesis have access to enough of the instance to perform the classification. But to the same degree that attention is focused on portions of an instance in order to determine the weight of the instance, attention is not available for performing the weak learning task at hand. This raises an interesting question: does an alternative form of boosting exist that avoids this difficulty? We answer this question negatively.

THEOREM 11 *k-TOP is weakly k-RFA learnable, but is not strongly k-RFA learnable for $1 \leq k \leq n-2$. The weak learning is polynomial-time for constant k, while the strong learning is information-theoretically impossible.*

COROLLARY 1 *Weak k-RFA learnability of a class does not imply strong k-RFA learnability of the class.*

Proof of Theorem 11: By Theorem 9 we know that k-TOP is weakly k-RFA learnable, and in polynomial time for constant k. And we have also shown that it is information theoretically impossible to strongly k-RFA learn k-DL for $1 \leq k \leq n-2$. All that remains to show is that k-DL is a subclass of k-TOP.

To see this, consider a target f with a k-DL representation $(t_1, b_1), \ldots, (t_r, b_r)$. Note that this function can be written equivalently as the sign of the following sum of the terms t_i, which we view as functions with range $\{0, 1\}$ (and recall that we are treating the b_i as $\{-1, +1\}$-valued in this section):

$$\sum_{i=1}^{r} 2^{r-i} b_i t_i.$$

To see that the sign of this sum is equivalent to f, notice that given an input x, each term t_j that is not satisfied contributes nothing to the sum, since $t_j(x) = 0$ for all such t_j. But the first term t_i which *is* satisfied by x will cause $2^{r-i} b_i$ to be added to the sum. Since $2^{r-i} > \sum_{j=i+1}^{r} 2^{r-j}$, the values of b_j, $j > i$, have no effect on the sum. Thus the value of b_i determines the value of the function at x, as desired.

Furthermore, it follows from the definition of the Fourier transform that each of these k-terms t_i can be written as a sum of k-parity functions, since the terms are functions of at most k variables each and every k-variable Boolean function can be written as a linear combination of k-parities. Thus f can be written as the sign of a weighted sum of k-parities, that is, as a k-TOP. ∎

It should be noted that, while the proof above shows that k-DL is a subclass of k-TOP, for a given k-DL representation of size r the construction above may lead to a k-TOP with size exponential in k and in r. Thus if r is, say, linearly related to n, then the k-TOP representation constructed may be exponential-size in n. On the other hand, time-efficient weak learning allows the learner to run in time polynomial in the size of the function within the representation class being learned. Thus, while the scenarios which are hard to learn strongly for k-TOP may not be weakly learnable efficiently with respect to k-DL, they are weakly learnable efficiently with respect to k-TOP.

The above theorem shows that boosting is not applicable in general within the k-RFA model. However, boosting *can* be employed to good use under certain conditions in RFA models. For example, we now use hypothesis boosting to argue that for k constant, k-TOP functions can be ϵ-approximated efficiently from a K-RFA oracle, where K depends polynomially on the size of the target and logarithmically on ϵ^{-1} but does not depend on n. This means that "small" (with respect to n) k-TOP functions are efficiently learnable from an oracle which has focus of attention which, while larger than k, is at least smaller than n.

THEOREM 12 *Let s be the size of a target k-TOP function and ϵ the accuracy required of a learning algorithm. Then k-TOP is K-RFA learnable for $K = 2ks^2 \ln \frac{4}{\epsilon}$. Note that K does not depend on n. The learning algorithm runs in time $\tilde{O}(n^{k+1})$ but is otherwise polynomial in the usual PAC parameters.*

Proof: As noted earlier, there is a $\tilde{O}(n^{k+1}s^2)$-time weak k-RFA learning algorithm for k-TOP. In fact, the weak hypothesis produced by this learner can be made a nearly $(\frac{1}{2} - \frac{1}{2s})$-approximator to the target k-TOP f. Now assume for the moment that we have access to a PAC example oracle rather than a K-RFA oracle. Then applying Freund's boosting-by-majority algorithm [14, 15] to this weak learner will produce an ϵ-hypothesis for f consisting of a majority vote over approximately $2s^2 \ln \frac{4}{\epsilon}$ weak hypotheses. As each weak hypothesis is defined over only k bits of the input, the algorithm actually only needs access to approximately $2ks^2 \ln \frac{4}{\epsilon}$ bits of each instance. ∎

8. Further Research

Being a refinement of the PAC learning model, the formulation of the RFA model stimulates the need for new techniques and approaches in order to cope with new learning problems. Some of the needed tools are developed in this paper, enabling the study of the RFA learnability of interesting classes of boolean functions, such as decision lists and k-TOPs. We believe that these tools, particularly the indistinguishability argument of Section 3.1, can be used further, both in the study of the learnability of other classes and also for other RFA scenarios.

Perhaps the most interesting open problem concerning this work is the following problem, which significantly predates learning theory research but can be naturally reformulated as an RFA problem. Consider the class of linearly-separable half-spaces over $\{0,1\}^n$ (perceptrons). It is well-known that the first-order Fourier coefficients of a perceptron (also called the *Chow parameters* of the perceptron) uniquely determine the perceptron (see [10], or [9] for a more general result). Is it possible to efficiently compute a good weights-based approximation of the perceptron from good approximations of these coefficients?

This question can be naturally formulated as a 1-RFA learning problem, as follows. It can be shown that when learning from a 1-RFA oracle with respect to the uniform distribution, we can obtain good estimates of the Chow parameters. Also, based on our Fourier characterization of k-RFA learning (see Appendix), we know that in fact these parameters capture *all* of the information available from the 1-RFA oracle. Thus, the above open question is equivalent to the following RFA question: is the class of perceptrons efficiently and properly 1-RFA learnable?

Note that since perceptrons are efficiently (and properly) PAC learnable, it is enough to have a good prediction rule which can be computed from approximations of the Chow parameters, and succeeds for almost all the instances. Although it can be shown that one of the Chow parameters is a weak approximator for the target function [12], we currently do not know how to boost weak approximators in the 1-RFA model.

Another intersting question concerns weak and strong learnability in the k-RFA model. We have shown that—in a class that contains functions of size exponentially large in n—weak and strong learnability are not equivalent. Is there also a class of functions all of size polynomial in n for which weak and strong k-RFA learnability differ, or are these learning models equivalent in all such classes?

Appendix A

Fourier Characterization of k-RFA Learnability

We present here an alternative to the characterization of k-RFA learnability developed in Section 3. Specifically, we define k-Fourier equivalence of scenarios and show that two scenarios are k-RFA equivalent precisely when they are k-Fourier equivalent. The definition of k-RFA hardness can therefore be rephrased in terms of k-Fourier equivalence rather than k-RFA equivalence, and thus Theorem 1 can also be viewed in terms of k-Fourier equivalence. While we do not use this characterization to obtain any learnability results in this paper, the connection of k-RFA learnability with Fourier analysis, which has proved quite useful in learning theory, seems potentially very useful.

We will assume in this section that $f \in \{-1,+1\}$; this also means that we assume similar small changes in the definitions of the previous section, such as that the parameter b in the definition of $p_S(I,x,b)$ is in $\{-1,+1\}$.

Definition 4. Two scenarios $S_1 = \langle f_1, D_1\rangle$ and $S_2 = \langle f_2, D_2\rangle$ are k-Fourier equivalent if and only if for all $a \in \{0,1\}^n$ such that $|a| \leq k$,

$$\begin{aligned} \mathbf{E}_{x\in D_1}[\chi_a(x)] &= \mathbf{E}_{x\in D_2}[\chi_a(x)] \text{ and} \\ \mathbf{E}_{x\in D_1}[f_1(x)\cdot\chi_a(x)] &= \mathbf{E}_{x\in D_2}[f_2(x)\cdot\chi_a(x)]. \end{aligned}$$

THEOREM 13 *Two scenarios $S_1 = \langle f_1, D_1\rangle$ and $S_2 = \langle f_2, D_2\rangle$ are k-RFA equivalent if and only if they are k-Fourier equivalent.*

Proof: For a scenario $S = \langle f, D\rangle$ define $E(S,k)$ to be the vector of expectations $\{\mathbf{E}_{x\in D}[\chi_a(x)], \mathbf{E}_{x\in D}[f(x)\cdot\chi_a(x)] \mid |a| \leq k\}$ and let $P(S,k)$ be the vector of probabilities $\{p_S(I,x,b) \mid |I| = k\}$. We will show that the expectations in $E(S,k)$ can be computed given the probabilities in $P(S,k)$ and vice versa. Given this, it follows that if for two scenarios, S_1 and S_2, $p_{S_1}(I,x,b) = p_{S_2}(I,x,b)$ for all $|I| = k$, x, b then the vectors of expectations $E(S_1,k)$ and $E(S_2,k)$—which are functions of the p_{S_1}'s and p_{S_2}'s, respectively—must also be equal. That is, k-RFA equivalence of S_1 and S_2 implies k-Fourier equivalence. Conversely, given that $P(S,k)$ can be computed from $E(S,k)$ then if $E(S_1,k) = E(S_2,k)$ it follows that $P(S_1,k) = P(S_2,k)$, or in other words, k-Fourier equivalence implies k-RFA equivalence. Thus we need only show the claimed functional relationships between the probabilities in $P(S,k)$ and the expectations in $E(S,k)$ to prove the theorem.

Let $S = \langle f, D\rangle$ be a scenario. Consider the expectation $\mathbf{E}_{z\in D}[f(z)\chi_a(z)]$ and assume without loss of generality that a begins with $0 \leq j \leq k$ 1's and ends with $n-j$ 0's. Let $x \in \{0,1\}^j$, and let the notation $\sum_{z=_j x}$ denote the sum over all $z \in \{0,1\}^n$ such that the first j bits of z and x agree. Then

$$\begin{aligned} \mathbf{E}_{z\in D}[f\chi_a] &= \sum_z f(z)\chi_a(z)D(z) \\ &= \sum_x \sum_{z=_j x} f(z)\chi_a(x)D(z) \\ &= \sum_x \chi_a(x) \sum_{z=_j x} f(z)D(z) \\ &= \sum_x \chi_a(x)\left(\Pr_{z\in D}[f(z)=1 \wedge z =_j x] - \Pr_{z\in D}[f(z)=-1 \wedge z =_j x]\right). \end{aligned}$$

Note that the probabilities in the last line above can readily be computed from the probabilities in $P(S,k)$. Thus for all $|a| \leq k$, $\mathbf{E}_D[f\chi_a]$ is a function of the p_S's in $P(S,k)$, and a similar argument shows that $\mathbf{E}_D[\chi_a]$ is as well.

Now we show that the p_S's in $P(S,k)$ are functions of the expectations in $E(S,k)$; this will also provide some insight into why we chose these expectations for our definition of k-Fourier equivalence. We first want to rewrite $p_S(I,x,b)$ in another form. Let f' be the $\{0,1\}$-valued equivalent of f, specifically the function such that $f(z) = (-1)^{f'(z)}$ for all z. Define b' similarly with respect to b. Now define the $\{0,1\}$-valued function $g_{I,x,b'}(z, f'(z))$ to have value 1 if and only if $f'(z) = b'$ and for all i in I, $z_i = x_i$. Then clearly $\mathbf{E}_{z\in D}[g_{I,x,b'}(z, f'(z))] = p_S(I, x, b)$ for every I, x, and b.

Writing p_S this way allows us to apply an observation of Blum et al. [7] to our analysis. They showed that for any $\{0,1\}$-valued function f' with corresponding $f \in \{-1,+1\}$ and any function $g(z, f'(z))$,

$$\mathbf{E}_{z\in D}[g(z, f'(z))] = \sum_a \hat{g}(a0)\mathbf{E}_{z\in D}[\chi_a(z)] + \sum_a \hat{g}(a1)\mathbf{E}_{z\in D}[f(z)\chi_a(z)],$$

where $a \in \{0,1\}^n$. Now each of the $g_{I,x,b}$ is a deterministic function, and therefore the Fourier coefficients $\hat{g}$ for each of these functions are constants. Furthermore, each $g_{I,x,b}$ depends on only k of the bits in z if $|I| = k$. A standard Fourier argument gives that for such g, $\hat{g}(a0)$ and $\hat{g}(a1)$ will be zero for all $|a| > k$. Thus for all $|I| = k$, x, and b, $p_S(I, x, b) = \mathbf{E}_{z\in D}[g_{I,x,b'}(z, f'(z))]$ is a function of $\mathbf{E}_D[\chi_a]$ and $\mathbf{E}_D[f\chi_a]$ for $|a| \leq k$. ■

Finally, note that $\mathbf{E}_D[\chi_a] = \sum_z \chi_a(z)D(z) = 2^n\mathbf{E}_z[D(z)\chi_a(z)] = 2^n\hat{D}(a)$. Similarly, $\mathbf{E}_D[f\chi_a] = 2^n\widehat{Df}(a)$ (here D is being used to represent both a probability distribution and the real-valued function that returns the weight this distribution assigns to each instance). In other words, the expected values characterizing k-RFA learnability are actually the bounded k-order Fourier coefficients of the target distribution and of the product of the target distribution and the target function. This suggests that k-RFA learnability results for a function class (possibly with respect to a restricted class of distributions) might be obtained by applying Fourier analysis to the class.

Appendix B

Karnaugh diagrams with RFA hard scenario pairs

Tables B.1 and B.2 on the next page show Karnaugh diagrams with the RFA hard scenario pairs we used for the proofs of Theorem 2 and 3, respectively.

The bold numbers are the function values of the input instances, addressed by the rows and columns of the two dimensional tables. The smaller numbers in parentheses represent the probability distributions. For clarity, we avoid fractions. Hence, to get the real distribution these numbers should be divided by the factor given in the titles of the diagrams.

Consider, for instance, in Table B.1 the diagram entitled $\mathbf{f}_2^{(3)}\,{}^{(6\cdot D_2^{(3)})}$. The numbers $\mathbf{0}\,^{(1)}$ in the upper left corner mean that $f_2^{(3)}(0,0,0) = 0$ and that $D_2^{(3)}(0,0,0) = 1/6$. The numbers $\mathbf{1}\,^{(0)}$ in the lower left corner mean that $f_2^{(3)}(0,0,1) = 1$ and that $D_2^{(3)}(0,0,1) = 0$.

Acknowledgments

We would like to thank Paul Fischer for helpful discussions and comments. The comments provided by an anonymous referee have been very useful in improving

Table B.1. Karnaugh-diagrams for the scenarios of Theorem 2.

$\mathbf{f}_1^{(2)}$ $(3 \cdot D_1^{(2)})$

x_1x_2: 00	01	11	10
$\mathbf{0}^{(1)}$	$\mathbf{1}^{(1)}$	$\mathbf{1}^{(0)}$	$\mathbf{1}^{(1)}$

$\mathbf{f}_2^{(2)}$ $(3 \cdot D_2^{(2)})$

x_1x_2: 00	01	11	10
$\mathbf{1}^{(0)}$	$\mathbf{1}^{(1)}$	$\mathbf{0}^{(1)}$	$\mathbf{1}^{(1)}$

$\mathbf{f}_1^{(3)}$ $(6 \cdot D_1^{(3)})$

x_3 \ x_1x_2	00	01	11	10
0	$\mathbf{0}^{(0)}$	$\mathbf{0}^{(1)}$	$\mathbf{0}^{(1)}$	$\mathbf{0}^{(1)}$
1	$\mathbf{0}^{(1)}$	$\mathbf{1}^{(1)}$	$\mathbf{1}^{(0)}$	$\mathbf{1}^{(1)}$

$\mathbf{f}_2^{(3)}$ $(6 \cdot D_2^{(3)})$

x_3 \ x_1x_2	00	01	11	10
0	$\mathbf{0}^{(1)}$	$\mathbf{0}^{(1)}$	$\mathbf{0}^{(0)}$	$\mathbf{0}^{(1)}$
1	$\mathbf{1}^{(0)}$	$\mathbf{1}^{(1)}$	$\mathbf{0}^{(1)}$	$\mathbf{1}^{(1)}$

$\mathbf{f}_1^{(4)}$ $(12 \cdot D_1^{(4)})$

x_3x_4 \ x_1x_2	00	01	11	10
00	$\mathbf{0}^{(1)}$	$\mathbf{0}^{(1)}$	$\mathbf{0}^{(0)}$	$\mathbf{0}^{(1)}$
01	$\mathbf{0}^{(0)}$	$\mathbf{0}^{(1)}$	$\mathbf{0}^{(1)}$	$\mathbf{0}^{(1)}$
11	$\mathbf{0}^{(1)}$	$\mathbf{1}^{(1)}$	$\mathbf{1}^{(0)}$	$\mathbf{1}^{(1)}$
10	$\mathbf{0}^{(0)}$	$\mathbf{0}^{(1)}$	$\mathbf{0}^{(1)}$	$\mathbf{0}^{(1)}$

$\mathbf{f}_2^{(4)}$ $(12 \cdot D_2^{(4)})$

x_3x_4 \ x_1x_2	00	01	11	10
00	$\mathbf{0}^{(0)}$	$\mathbf{0}^{(1)}$	$\mathbf{0}^{(1)}$	$\mathbf{0}^{(1)}$
01	$\mathbf{0}^{(1)}$	$\mathbf{0}^{(1)}$	$\mathbf{0}^{(0)}$	$\mathbf{0}^{(1)}$
11	$\mathbf{1}^{(0)}$	$\mathbf{1}^{(1)}$	$\mathbf{0}^{(1)}$	$\mathbf{1}^{(1)}$
10	$\mathbf{0}^{(1)}$	$\mathbf{0}^{(1)}$	$\mathbf{0}^{(0)}$	$\mathbf{0}^{(1)}$

Table B.2. Karnaugh-diagrams for the scenarios of Theorem 3.

$\mathbf{f}_1^{(3)}$ $(10 \cdot D_1^{(3)})$

x_3 \ x_1x_2	00	01	11	10
0	$\mathbf{1}^{(1)}$	$\mathbf{1}^{(2)}$	$\mathbf{1}^{(1)}$	$\mathbf{1}^{(1)}$
1	$\mathbf{0}^{(1)}$	$\mathbf{0}^{(1)}$	$\mathbf{1}^{(1)}$	$\mathbf{0}^{(2)}$

$\mathbf{f}_2^{(3)}$ $(10 \cdot D_2^{(3)})$

x_3 \ x_1x_2	00	01	11	10
0	$\mathbf{1}^{(1)}$	$\mathbf{1}^{(1)}$	$\mathbf{1}^{(2)}$	$\mathbf{1}^{(1)}$
1	$\mathbf{0}^{(2)}$	$\mathbf{1}^{(1)}$	$\mathbf{0}^{(1)}$	$\mathbf{0}^{(1)}$

$\mathbf{f}_1^{(5)}$ $(28 \cdot D_1^{(5)})$

$x_3x_4x_5$ \ x_1x_2	00	01	11	10
000	$\mathbf{1}^{(1)}$	$\mathbf{1}^{(1)}$	$\mathbf{1}^{(1)}$	$\mathbf{1}^{(1)}$
010	$\mathbf{1}^{(1)}$	$\mathbf{1}^{(1)}$	$\mathbf{1}^{(2)}$	$\mathbf{1}^{(1)}$
110	$\mathbf{1}^{(0)}$	$\mathbf{1}^{(1)}$	$\mathbf{1}^{(0)}$	$\mathbf{1}^{(0)}$
100	$\mathbf{1}^{(1)}$	$\mathbf{1}^{(1)}$	$\mathbf{1}^{(1)}$	$\mathbf{1}^{(1)}$
001	$\mathbf{0}^{(0)}$	$\mathbf{0}^{(0)}$	$\mathbf{0}^{(0)}$	$\mathbf{0}^{(0)}$
011	$\mathbf{1}^{(1)}$	$\mathbf{1}^{(2)}$	$\mathbf{1}^{(1)}$	$\mathbf{1}^{(1)}$
111	$\mathbf{0}^{(1)}$	$\mathbf{0}^{(1)}$	$\mathbf{1}^{(1)}$	$\mathbf{0}^{(2)}$
101	$\mathbf{0}^{(2)}$	$\mathbf{0}^{(0)}$	$\mathbf{0}^{(1)}$	$\mathbf{0}^{(1)}$

$\mathbf{f}_2^{(5)}$ $(28 \cdot D_2^{(5)})$

$x_3x_4x_5$ \ x_1x_2	00	01	11	10
000	$\mathbf{1}^{(1)}$	$\mathbf{1}^{(1)}$	$\mathbf{1}^{(1)}$	$\mathbf{1}^{(1)}$
010	$\mathbf{1}^{(1)}$	$\mathbf{1}^{(2)}$	$\mathbf{1}^{(1)}$	$\mathbf{1}^{(1)}$
110	$\mathbf{1}^{(0)}$	$\mathbf{1}^{(0)}$	$\mathbf{1}^{(1)}$	$\mathbf{1}^{(0)}$
100	$\mathbf{1}^{(1)}$	$\mathbf{1}^{(1)}$	$\mathbf{1}^{(1)}$	$\mathbf{1}^{(1)}$
001	$\mathbf{0}^{(0)}$	$\mathbf{0}^{(0)}$	$\mathbf{0}^{(0)}$	$\mathbf{0}^{(0)}$
011	$\mathbf{1}^{(1)}$	$\mathbf{1}^{(1)}$	$\mathbf{1}^{(2)}$	$\mathbf{1}^{(1)}$
111	$\mathbf{0}^{(2)}$	$\mathbf{1}^{(1)}$	$\mathbf{0}^{(1)}$	$\mathbf{0}^{(1)}$
101	$\mathbf{0}^{(1)}$	$\mathbf{0}^{(1)}$	$\mathbf{0}^{(0)}$	$\mathbf{0}^{(2)}$

the presentation of this work. Andreas Birkendorf, Norbert Klasner, and Hans Ulrich Simon gratefully acknowledge the support of Bundesministerium für Forschung und Technologie grant 01IN102C/2, the support of Deutsche Forschungsgemeinschaft grant Si 498/3-1, and the support of Deutscher Akademischer Austauschdienst grant 322-vigoni-dr. Part of this research was done while Eli Dichterman and Jeffrey Jackson visited Universität Dortmund.

References

1. Aho, A. V., Hopcroft, J. E., and Ullman, J. D. (1974). *The Design and Analysis of Computer Algorithms*. Addison-Wesley.
2. Angluin, D. and Valiant, L. G. (1979). Fast probabilistic algorithms for hamiltonian circuits and matchings. *Journal of Computer Systems and Sciences*, 18:155–193.
3. Ben-David, S. and Dichterman, E. (1993). Learning with restricted focus of attention. In *Proceedings of the 6th Annual Conference on Computational Learning Theory*, pages 287–296. ACM Press, New York, NY.
4. Ben-David, S. and Dichterman, E. (1994). Learnability with restricted focus of attention guarantees noise-tolerance. In *5th International Workshop on Algorithmic Learning Theory, ALT'94*, pages 248–259.
5. Ben-David, S. and Dichterman, E. (1997). Learning with restricted focus of attention. Technical Report LSE-CDAM-97-01, London School of Economics.
6. Blum, A. and Chalasani, P. (1992). Learning switching concepts. In *Proceedings of the 5th Annual Workshop on Computational Learning Theory*, pages 231–242.
7. Blum, A., Furst, M., Jackson, J. C., Kearns, M., Mansour, Y., and Rudich, S. (1994). Weakly learning DNF and characterizing statistical query learning using fourier analysis. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, pages 253–262.
8. Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M. K. (1989). Learnability and the Vapnik-Chervonenkis dimension. *JACM*, 36(4):929–965.
9. Bruck, J. (1990). Harmonic analysis of polynomial threshold functions. *SIAM Journal of Discrete Mathematics*, 3(2):168–177.
10. Chow, C. (1961). On the characterization of threshold functions. In *Proc. Symp. on Switching Circuit Theory and Logical Design*, pages 34–38.
11. Decatur, S. E. and Gennaro, R. (1995). On learning from noisy and incomplete examples. In *Proceedings of the 8th Annual Conference on Computational Learning Theory*, pages 353–360.
12. Domingo, C. Private communication.
13. Ehrenfeucht, A., Haussler, D., Kearns, M., and Valiant, L. G. (1989). A general lower bound on the number of examples needed for learning. *Information and Computation*, 82:247–261. First appeared in *Proceedings of the 1st Annual Workshop on Computational Learning Theory*.
14. Freund, Y. (1990). Boosting a weak learning algorithm by majority. In *Proceedings of the 3rd Annual Workshop on Computational Learning Theory*, pages 202–216. Morgan Kaufmann, San Mateo, CA.
15. Freund, Y. (1993). *Data Filtering and Distribution Modeling Algorithms for Machine Learning*. PhD thesis, University of California, Santa Cruz.
16. Goldman, S. A. and Sloan, R. H. (1995). Can PAC learning algorithms tolerate random attribute noise? *Algorithmica*, 14:70–84.
17. Jackson, J. C. (1994). An efficient membership-query algorithm for learning DNF with respect to the uniform distribution. In *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science*, pages 42–53.
18. Jackson, J. C. (1995). *The Harmonic Sieve: A Novel Application of Fourier Analysis to Machine Learning Theory and Practice*. PhD thesis, Carnegie Mellon University. Available as technical report CMU-CS-95-183.
19. Kearns, M. J. (1993). Efficient noise-tolerant learning from statistical queries. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, pages 392–401.
20. Kearns, M. J. and Li, M. (1993). Learning in the presence of malicious errors. *SIAM J. Comput.*, 22:807–837.
21. Kearns, M. J. and Schapire, R. E. (1990). Efficient distribution-free learning of probabilistic concepts. In *Proceedings of the 31st Annual IEEE Symposium on Foundations of Computer Science*, pages 382–391.
22. Kearns, M. J., Schapire, R. E., and Sellie, L. M. (1992). Towards efficient agnostic learning. In *Proceedings of the 5th Annual Workshop on Computational Learning Theory*, pages 341–352.

23. Linial, N., Mansour, Y., and Nisan, N. (1993). Constant depth circuits, Fourier transform, and learnability. *Journal of the ACM*, 40(3):607–620. Earlier version appeared in *Proceedings of the 30th Annual Symposium on Foundations of Computer Science*, pages 574–579, 1989.
24. Rivest, R. L. (1987). Learning decision lists. *Machine Learning*, 2:229–246.
25. Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning*, 5:197–227.
26. Simon, H. U. (1995). Learning decision lists and trees with equivalence-queries. In *Proceedings of the 2nd European Conference on Computational Learning Theory*, pages 322–336.
27. Valiant, L. G. (1984). A theory of the learnable. *CACM*, 27(11):1134–1142.
28. Vapnik, V. N. and Chervonenkis, A. Y. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its applications*, XVI(2):264–280.