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

Andreas Birkendorf
Universität Dortmund
Fachbereich Informatik
D-44221 Dortmund, Germany
birkendo@Ls2.informatik.uni-dortmund.de

Eli Dichterman
Computer Science Department
Technion
Haifa 32000, Israel
eli@cs.technion.ac.il

Jeffrey Jackson
Math and Computer Science Department
Duquesne University
600 Forbes Avenue
Pittsburgh, PA 15282, USA
jackson@mathcs.duq.edu

Norbert Klasner
Universität Dortmund
Fachbereich Informatik
D-44221 Dortmund, Germany
klasner@Ls2.informatik.uni-dortmund.de

Hans Ulrich Simon
Universität Dortmund
Fachbereich Informatik
D-44221 Dortmund, Germany
simon@Ls2.informatik.uni-dortmund.de

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 a 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 $(n-1)$-decision-lists 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 present a uniform-distribution k-RFA learning algorithm for the class of k-decision-lists in which the output value alternates at most j times. The latter algorithm is efficient for j and k constant. For k-TOP we show weak learnability by a k-RFA algorithm (efficient for constant k) and strong uniform-distribution k-RFA learnability of k-TOP with polynomial sample complexity. 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.

1 Introduction

Consider the following learning problem. A number of medical research teams have been studying a certain disease with the goal of developing a rule that can predict circumstances under which the disease is likely to occur. Each separate research team has focused on a few factors potentially relevant to the prediction task, such as age and various aspects of family history; only a few factors are considered in any one study due to limitations such as data availability, subject cooperation, economics, etc. In fact, different teams have focused on different sets of factors (although there may be some overlap). Unfortunately, none of the teams has yet produced a satisfactory prediction rule, probably because of the limited number of factors considered in each study. The question is: Is there a "good" way to combine the results of these separate studies to produce a single rule that operates over the entire set of factors and that produces satisfactory predictions?

Questions such as this one motivate the following model of learning, which is a variation on the well-known PAC model. Specifically, assume that there is a target function to be learned drawn from some class $\mathcal{F}$ of functions each defined over a fixed number of inputs n. As in the PAC model, we will say that $\mathcal{F}$ is learnable if there exists an algorithm that can, with high probability, approximate any function $f \in \mathcal{F}$ given an oracle that provides random examples (input-output pairs) of f. However, unlike the PAC model, the *k-Restricted Focus of Attention (k-RFA)* model interjects a filter between the example oracle and the learner: the learner is only allowed to see k of the assignments to the inputs of f in each example. This filtered example oracle is known as a *k-RFA oracle*. The learner is allowed to choose which k inputs are revealed, and a different set of k inputs may be selected for each example.

A somewhat more general RFA model was introduced by Ben-David and Dichterman [BDD93], who also gave several initial learnability results for the k-RFA model. For example, they have shown 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. They also noted [BDD93, BDD94] that there are close connections between this model and the model of learning probabilistic concepts [KS90] as well as with the Statistical Query model [Kea93]. Furthermore, it follows immediately from their results that, unlike the PAC model, k-RFA learnability of a class is *not* characterized by the VC dimension of the class. In short, the k-RFA model is closely related to several existing learning models, but appears to be sufficiently different that results in these other models do not carry over directly to k-RFA learning.

This paper extends our understanding of k-RFA learnability 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 efficiently PAC-learnable classes currently known (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.

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 for all k, and in fact there is an efficient learning algorithm with respect to the uniform distribution for 1-RFA learning of 1-DL. In yet another contrast, we also prove a hardness result showing, among other things, that distribution-free learnability of even 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 [Jac94]; furthermore, it has also been studied in the context of empirical machine learning [Jac95]. 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. We then use our k-DL results in conjunction with the weak learnability of

k-TOP to argue that, unlike the PAC model, weak and strong learning are *not* equivalent in the k-RFA model (for $k \leq n-2$). 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 k constant) sample complexity but running time which is not necessarily polynomial. Whether or not this algorithm can be made efficient is an interesting open problem that is closely related to the long-standing question of whether or not there is an efficient algorithm for converting the Chow parameter representation of a perceptron into a standard weight-based representing of the perceptron. The paper closes with some thoughts about this problem.

2 Definitions

2.1 The Learning Model

The model introduced in [BDD93] 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, as defined below.

Recall that in the PAC model, nature is represented by an oracle $EX_{f,D}$, which has an access to the target distribution D and the target function f (both unknown to the learner), and supplies the learner with classified examples $\langle x, f(x)\rangle$, where x is drawn randomly from D.

In the RFA model the interaction mechanism between the learner and the oracle is refined. First, 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. The learner can make an observation by invoking the RFA-oracle with some specific projection $w \in W$. The oracle then draws a random instance x from the distribution D (as before), but returns the observation $w(x, f(x))$.

Let the instance space be $\{0,1\}^n$. A special interesting case of the RFA-oracle is the k-RFA-oracle, which supplies the learner with only k bits of each instance (and supplies also the classification bit). We can formally define the interaction as follows. To make an observation, the learner has to choose k indices $\{i_1, \ldots, i_k\} \subset \{1, \ldots, n\}$, and invoke the oracle with these indices. Then the oracle draws a random instance x from the distribution D, and gives the learner the observation $\langle (x_{i_1}, \ldots, x_{i_k}), f(x)\rangle$ (where x_j is the j-th bit of x). Thus, using a k-RFA oracle the user may observe only k bits of each instance x (this is the restriction on the size of his focus of attention), and the classification bit.

Given a target function $f : X \rightarrow \{0,1\}$ and a distribution D over X, we say that a hypothesis $h : X \rightarrow \{0,1\}$ is ϵ-*good* (with respect to D and f), if $\Pr_{x\in D}[f(x) \neq h(x)] < \epsilon$.

Definition 2.1 (k-RFA-learnability) *A class of functions $\mathcal{F}$ is k-RFA learnable using $\mathcal{H}$, if there is an algorithm A, such that for every $f \in \mathcal{F}$, any distribution D, and every $0 < \epsilon, \delta \leq 1$, given inputs ϵ, δ, n and $size(f)$, and given access to the k-RFA oracle, A halts and with probability of at least $1-\delta$ outputs an ϵ-good hypothesis $h \in \mathcal{H}$.*

If $\mathcal{H} = \mathcal{F}$ we say that $\mathcal{F}$ is properly k-RFA learnable. If A halts in time in time $poly(\epsilon^{-1}, \delta^{-1}, n, size(f))$, then we say that $\mathcal{F}$ is efficiently k-RFA learnable.

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 [Riv87]. This class is defined as follows.

Definition 2.2 *A k-decision list is a list of pairs $\langle (t_i, b_i)\rangle_{i=1}^r$, in which each t_i is a term of size k (or less), $b_i \in \{0,1\}$, and t_r is the constant 1. A k-decision list L defines a boolean function as follows: for every $x \in \{0,1\}^n$, $L(x) = b_j$ where $j = min\{i \,|\, t_i(x) = 1\}$ (one may think of a decision list as defining a concept by giving a rule with exceptions). E.g. $\langle (x_1, 1), (x_2, 0), (\overline{x}_3, 1), (1, 0)\rangle$ is a 1-decision list. 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 [Riv87] that k-DL properly contain k-DNF $\cup$ k-CNF, and is efficiently PAC learnable.

It is shown in [BDD93] that the class k-DL is k-RFA (non-efficiently) 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 denoted by TOP, which is defined as follows.

Definition 2.3 (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. We assume that the output of the majority gate is undefined if exactly half of its inputs are "voting" in one direction and half in the other. 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 parity appearing once and having an integer weight associated with it. Furthermore, we can then assume that all parities appear in a positive (no negated literals) sense, and use negative weights to represent parities of the opposite sense.*

We also denote by PAR the class which contains the parity function over n variables (parity_n) and its complement ($\overline{\text{parity}_n}$).

2.3 The Fourier transform

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[1]

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

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 $\hat{f}(a) = \mathbf{E}[f\chi_a]$. 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 $|a| \leq k$. 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 Boolean f, $\hat{f}(a) = \mathbf{E}[f\chi_a]$ represents the correlation of f and χ_a with respect to the uniform distribution.

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 Boolean f 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 Characterizing k-RFA Learnability

In this section we give a condition under which a function class is not learnable from a k-RFA oracle, partially characterizing k-RFA learnability (we intend to provide a more complete characterization in the full paper). The condition follows directly from the definition of k-RFA learning; building on this characterization, we develop a general tool for showing k-RFA learnability hardness which we apply in Section 4. It is also possible to characterize k-RFA learning in terms of certain Fourier characteristics of the target function class; this is also left for the full paper.

First we formulate the notion of indistinguishability of two RFA scenarios. A *scenario* S over the instance space $X = \{0,1\}^n$ is defined to be a pair $\langle f, D \rangle$ of a Boolean function f and a distribution D over X. Define a pseudo-metric d_s on the scenario space, as follows. Given two scenarios $S_1 = \langle f_1, D_1 \rangle$ and $S_2 = \langle f_2, D_2 \rangle$, let $d_s(S_1, S_2) \triangleq \sum_{\{x | f_1(x) \neq f_2(x)\}} \min\{\Pr_{x \in D_1}[\{x\}], \Pr_{x \in D_2}[\{x\}]\}$. We say that S_1 and S_2 are *different* if

[1] Expectations and probabilities here and elsewhere are with respect to the uniform distribution over the instance space unless otherwise indicated.

$d_s(S_1, S_2) > 0$. Notice that if S_1 and S_2 are different then for every function h there is an $i \in \{1, 2\}$, such that h is not a $\frac{1}{2}d_s(S_1, S_2)$-good approximation for f_i under the distribution D_i.

Next we define an equivalence relation among scenarios. Given a scenario $S = \langle f, D\rangle$ over $\{0, 1\}^n$, define for $I = \{i_1, \ldots, i_k\} \subset \{1, \ldots, n\}$, $x \in \{0, 1\}^n$, and $b \in \{0, 1\}$, the conditional probability $p_S(I, x, b) \triangleq \Pr_{z\in D}[f(z) = b \mid \forall i \in I\ z_i = x_i]$. One may easily convince himself that if two scenarios S, S' satisfy $p_S = p_{S'}$ for every I, every x and every b, then they are indistinguishable by a k-RFA learner. Thus we call two such scenarios *k-RFA equivalent*, obtaining

Lemma 3.1 *The existence of two different and k-RFA equivalent scenarios $\langle f_1, D_1\rangle$, $\langle f_2, D_2\rangle$ implies k-RFA non-learnability of any class which contains both f_1 and f_2.*

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 for hardness results which hold for all $n \geq n_0$. We now show a construction which inductively expands two different k-RFA equivalent scenarios over the instance space $\{0, 1\}^n$ into two different $(k+1)$-RFA equivalent scenarios over $\{0, 1\}^{n+1}$. Thus, inductively applying this construction (within a family which is closed under the construction), yields a generalization of a non-learnability result from a given n_0 to all $n \geq n_0$.

Lemma 3.2 *Let $S_i^{(n)} = \langle f_i^{(n)}, D_i^{(n)}\rangle$, $i \in \{1, 2\}$, be two different and k-RFA equivalent scenarios over $\{0, 1\}^n$. Then the following construction yields two different and $(k+1)$-RFA equivalent scenarios $S_i^{(n+1)} = \langle f_i^{(n+1)}, D_i^{(n+1)}\rangle$ over $\{0, 1\}^{n+1}$:*

$$f_i^{(n+1)} \triangleq x_{n+1}\, f_i^{(n)} \ + \ \overline{x}_{n+1}\, f_{3-i}^{(n)}$$

$$D_i^{(n+1)}(x_1, \ldots, x_n, 0) \triangleq \frac{1}{2} D_i^{(n)}(x_1, \ldots, x_n)$$

$$D_i^{(n+1)}(x_1, \ldots, x_n, 1) \triangleq \frac{1}{2} D_{3-i}^{(n)}(x_1, \ldots, x_n)$$

Proof: Since $d_s(S_1^{(n+1)}, S_2^{(n+1)}) = d_s(S_1^{(n)}, S_2^{(n)})$, the constructed scenarios are different. Now let $I = \{i_1, \ldots, i_{k+1}\} \subset \{1, \ldots, n+1\}$, $x \in \{0, 1\}^{n+1}$, and $b \in \{0, 1\}$. To verify that $p_{S_1^{(n+1)}}(I, x, b) = p_{S_2^{(n+1)}}(I, x, b)$, consider the following two cases:

- If $n + 1 \in I$, then $p_{S_1^{(n+1)}}(I, x, b) = \frac{1}{2}p_{S_1^{(n)}}(I, x, b) = \frac{1}{2}p_{S_2^{(n)}}(I, x, b) = p_{S_2^{(n+1)}}(I, x, b)$.
- If $n + 1 \notin I$, then $S_1^{(n+1)}(x_1, \ldots, x_n, x_{n+1}) = S_2^{(n+1)}(x_1, \ldots, x_n, \overline{x}_{n+1})$, implying $p_{S_1^{(n+1)}}(I, x, b) = p_{S_2^{(n+1)}}(I, x, b)$.

Thus, the constructed scenarios are also $(k + 1)$-RFA equivalent. □

Inductively applying the above construction to the different 0-RFA equivalent scenarios $\langle x_1, \frac{1}{2}\rangle$ and $\langle \overline{x}_1, \frac{1}{2}\rangle$, we obtain the known result ([BDD93]) that PAR is **not** k-RFA learnable for $k < n$. As PAR is PAC learnable using only *one* example, this result also demonstrates the gap between PAC learning (= n-RFA learning) and $(n-1)$-RFA learning (and similarly, between $(k+1)$-RFA learning and k-RFA learning). Also notice that, since PAR $\subseteq$ DNF, this result implies k-RFA non-learnability of DNF, for $k < n$.

In the full paper we also study the following construction:
$f_i^{(n+1)}(x_1, \ldots, x_n, 0) \triangleq 0$, $f_i^{(n+1)}(x_1, \ldots, x_n, 1) \triangleq f_i^{(n)}(x_1, \ldots, x_n)$, and the distributions are constructed as in the previous construction. An application of this construction – which we call *linear construction* – is given in Appendix A.2.

4 Distribution-free Learning of Decision Lists

4.1 Results for $(n-k)$-DL

The following theorem shows $(n - 1 - \lceil k/2\rceil)$-RFA non-learnability of $(n - k)$-DL for $n > k$. The proof of this theorem is based on the indistinguishability argument of Section 3. Because the proof is very long and technical, we present a similar proof for a weaker result, Theorem A.5, in the appendix.

Theorem 4.1 *$(n-k)$-DL is* **not** *$(n-1-\lceil k/2\rceil)$-RFA learnable for $n>k$.*

Like the proof of Theorem A.5, the proof of Theorem 4.1 constructs RFA equivalent scenarios with functions $f_1^{(n)}$, $f_2^{(n)} \in$ 2-alt-$(n-k)$-DL. Hence, we can sharpen our result as follows.

Corollary 4.2 *Even 2-alt-$(n-k)$-DL is* **not** *$(n-1-\lceil k/2\rceil)$-RFA learnable for $n>k$.*

Note that (efficient) k-RFA learnability of 1-alt-k-DL = k-DNF $\cup$ k-CNF has already been shown in [BDD93]. Furthermore, we show that an $(n-1)$-RFA oracle gives enough information to learn $(n-1)$-DL (disregarding complexity issues). We do this by presenting an $(n-1)$-RFA learning algorithm for this class which uses a sample size of $O((N\log^3 N/\epsilon)\log(N/\delta))$ for $N=2^n$. The classes $(n-1)$-DL $\cup$ $\{\text{parity}_n\}$ and $(n-1)$-DL $\cup$ $\{\overline{\text{parity}_n}\}$ are $(n-1)$-RFA learnable by a modified version of this algorithm[2] (with the same sample size). Note that $(n-1)$-DL is the concept class of all Boolean functions except the two functions of PAR. Because $VCdim((n-1)\text{-DL}) = 2^n-1$ every algorithm which PAC learns $(n-1)$-DL needs a sample size exponential in n [EHKV89]. Obviously, this is also a lower bound for the sample size of an $(n-1)$-RFA learning algorithm.

Theorem 4.3 *$(n-1)$-DL is properly $(n-1)$-RFA learnable with a sample size of*

$$m = O\left(\frac{N\log^3 N}{\epsilon}\log\frac{N}{\delta}\right).$$

Proof Sketch: Denote by $x^{(k)}$ the element of the n-dimensional Boolean cube which we get if we flip the k-th bit of x. For each call of the $(n-1)$-RFA oracle an $(n-1)$-RFA learner chooses a dimension k and invokes the $(n-1)$-RFA oracle with the index set $\{1,\ldots,n\}\setminus\{k\}$. The $(n-1)$-RFA oracle randomly draws an instance $x\in\{0,1\}^n$ and returns the example $\langle e, f(x)\rangle$, where e is the edge $\{x, x^{(k)}\}$.

The $(n-1)$-RFA learning algorithm `EDGE` for $(n-1)$-DL will inspect the edges of the boolean cube one by one. Each edge $e=\{x, x^{(k)}\}$ falls into one of three categories:

insignificant Informally, this means that the $(n-1)$-RFA oracle did not return a sample including e a significant number of times.

pure Informally, this means that e was presented by the $(n-1)$-RFA oracle a significant number of times with label b, but not a significant number of times with the other label $1-b$.

impure Informally, this means that e was presented by the the $(n-1)$-RFA oracle a significant number of times with each of the two labels.

After inspection of $e=\{x, x^{(k)}\}$, `EDGE` defines its hypothesis h on $x, x^{(k)}$. An arbitrary assignment can be made if e is insignificant. If e is pure and b is the label that was shown a significant number of times, then h assigns label b to $x, x^{(k)}$. For impure edges, there is no such obvious assignment. In order to handle this case, `EDGE` begins by selecting an edge e_0 that is either insignificant or pure. An edge with this property exists, *e.g.*, the edge corresponding to the first test of the target $(n-1)$-DL f. Let's say this edge has the form $e_0=\{x, x^{(k)}\}$, i.e., e_0 goes along dimension k. The remaining edges along dimension k are partitioned according to their Hamming-distance from e_0. `EDGE` will inspect the edges with smaller Hamming-distance first. When handling an edge $e=\{x_2,x_3\}$ with Hamming-distance i, there is therefore a parallel edge $e'=\{x_1,x_4\}$, with Hamming-distance $i-1$, and hypothesis h has been defined on x_1,x_4 already. Now let's return to the critical case that e is impure. Edges e,e' form a rectangle in the boolean cube as shown in Figure 1. Nodes x_2,x_3 should have opposite labels, because e is impure. It suffices therefore to figure out the true label for x_3. For this purpose, `EDGE` will focus its attention to edge $e_1=\{x_4,x_3\}$. Loosely speaking, the following can happen:

1. If e' is insignificant, then the $(n-1)$-RFA oracle will show e_1 with label $h(x_3)$ more often than with label $h(x_4)$.

[2] $(n-1)$-DL $\cup$ PAR is not $(n-1)$-RFA learnable because even PAR is not $(n-1)$-RFA learnable ([BDD93]).

2. If e' is pure and $h(x_4) = b$, then either e_1 is presented with the other label $1 - b$ a significant number of times (which indicates that $1 - b$ is the true label for x_3) or not (which indicates that b is the true label for x_3).

3. If e' is impure, then $h(x_4)$ is likely to be the true label for x_4. Edge e_1 can now be used in a manner similar to the preceding case.

The formalization of these ideas is involved because small errors of hypothesis h on edge e_0 become larger and larger as the the inspection process passes to edges with increasing Hamming-distance from e_0. For this reason, the thresholds used in the definition of insignificance and pureness are chosen extremely small for e_0 (allowing only extremely small errors in the beginning) and are increased with increasing Hamming-distance. This makes hypothesis h very accurate for edges with small Hamming-distance, but larger and larger errors are tolerated as the Hamming-distance of the inspected edges grows. Of course, the overall fine-tuning makes sure that the total error is at most ϵ with high confidence. The gory details of the Algorithm `EDGE` and its analysis are deferred to Appendix A.1. □

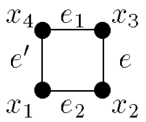

Figure 1: Denotation of edges and instances.

To learn the class $(n-1)$-DL $\cup$ $\{\text{parity}_n\}$ we modify `EDGE` as follows[3]. Now, we cannot guarantee to find an edge e_0 which is insignificant or pure. If all edges are impure, the target function must be the parity function. Otherwise, we do not know whether the target is an $(n-1)$-DL or the parity function, but `EDGE` finds (w.h.p.) an ϵ-good hypothesis. Hence, we get the following corollary.

Corollary 4.4 *The modified Algorithm* `EDGE` *learns* $(n-1)$*-DL* $\cup$ $\{parity_n\}$ *(and also* $(n-1)$*-DL* $\cup$ $\{\overline{parity_n}\}$*) properly in the* $(n-1)$*-RFA learning model with a sample size of*

$$m = O\left(\frac{N \log^3 N}{\epsilon} \log \frac{N}{\delta}\right).$$

4.2 Results for 1-DL

Consider the 1-decision-list $\langle (l_i, b_i)\rangle_{i=1}^r$ of minimum length (therefore $r \leq n+1$). Each pair (l_i, b_i) implies that the function values for the block $B_i \triangleq \{x \in \{0,1\}^n \mid \bar{l}_1 \wedge \bar{l}_2 \wedge \ldots \wedge \bar{l}_{i-1} \wedge l_i = 1\}$ are set to b_i. Block B_i is of size 2^{n-i} for $1 \leq i \leq r-1$ and B_r is of size 2^{n-r+1}.

Theorem 4.5 *The class* 1*-DL is* $(n-k)$*-RFA learnable by* k*-DL for* $1 \leq k \leq n/2$ *with a sample size of*

$$m = O\left(\frac{n^2 k^2 2^k}{\epsilon} \log \frac{n}{\delta}\right).$$

Proof Sketch: We present an algorithm which works in at most n steps divided into 3 phases. In Step i the algorithm appends at least one pair (l', b') to the partial decision list $h_i = \langle (l'_1, b'_1), (l'_2, b'_2), \ldots, (l'_{i-1}, b'_{i-1})\rangle$ to achieve h_{i+1}. Denote by X_i the instances determined by h_i, *i.e.*, the set $\{x \mid \exists 1 \leq j \leq i-1 : l'_j(x) = 1\}$, and by $\overline{X}_i$ the not determined instances. If $\overline{X}_i$ is empty, the algorithm stops and outputs h_i as the hypothesis.

The aim of the first phase is to find blocks B_i for $1 \leq i \leq n-k$. In step i we search for a block B_i as follows. We restrict our focus of attention to the $i-1$ dimensions corresponding to the literals of h_i, such that we can distinguish between examples of X_i and $\overline{X}_i$. We can use the remaining $(n-k)-(i-1)$ dimensions to find a new block B_i in $\overline{X}_i$. Checking each dimension of $\overline{X}_i$ once guarantees that we will find

[3] the modification for $(n-1)$-DL $\cup$ $\{\overline{\text{parity}_n}\}$ is exactly the same

such a B_i. Therefore $\left\lceil \frac{n-(i-1)}{(n-k)-(i-1)} \right\rceil \leq k+1$ samples are sufficient. Since $|\overline{X}_{n-k}| = 2^{n-k}$, the partial decision list h_{n-k+1} can be continued to an k-decision-list for each choice of function values on $\overline{X}_{n-k}$.

While h_i contains less than $n-k$ labels $b'_j = 0$ and less than $n-k$ labels $b'_j = 1$, we perform the second phase. Now, we search for a pair (l'_i, b'_i) as follows. For $b \in \{0, 1\}$ we restrict our focus of attention to the dimensions corresponding to the literals of h_i with label $b'_j \neq b$. This implies that we can recognize examples of blocks already classified with a label $\neq b$. We use one remaining dimension to find (for one value of b) an area which consists of some instances, already covered by blocks labeled b, and a new block. This new block induces (l'_i, b). Here $2(n-i) \leq 2k$ samples are sufficient[4].

If h_i consists of $n-k$ literals labeled 0 (the case "labeled 1" is symmetrical), we start the third phase. Obviously h_i consists of $(i-1)-(n-k)$ literals labeled 1. We restrict our focus of attention to the $n-(i-1)$ dimensions of $\overline{X}_i$ and to the dimensions corresponding to the 1-labeled literals[5]. This implies that 1-labeled instances x_j of $\overline{X}_i$ with relevant probability are recognized (w.h.p.). For each such instance x_j we can construct a pair $(t'_j, 1)$, where t'_j is the $(n-(i-1))$-term regarding to x_j and the dimensions of $\overline{X}_i$. The final hypothesis is built by appending $(t'_1, 1), \ldots, (t'_j, 1), \ldots$, and $(1, 0)$ to h_i.

The sample size can be computed as in Appendix A.1. □

This result implies efficient learnability of 1-DL for $k \in O(\log n)$. Note that Theorem 4.5 is tight in the sense that 1-DL is **not** $(\lceil n/2 \rceil - 1)$-RFA learnable because of Theorem 4.1 with $k = n-1$. Furthermore, for $k = 1$ the used algorithm is proper. For $k \geq 2$ no proper learning algorithm exists, because of the following theorem.

Theorem 4.6 *1-DL is* **not** *properly $(n-k)$-RFA learnable for $k \geq 2$.*

For the proof, which uses the indistinguishability argument of Section 3, see Appendix A.2.

In analogy to Corollary 4.2 we obtain

Corollary 4.7 *Even 2-alt-1-DL is* **not** *properly $(n-k)$-RFA learnable for $k \geq 2$.*

5 Learning Decision Lists under Fixed Distributions

Let D be an arbitrary, but fixed, distribution over $\{0,1\}^n$. 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 $\Pr_{x \in D}[f(x) = 1 | t(x) = 1] \neq \Pr_{x \in D}[g(x) = 1 | t(x) = 1]$. The following result shows that the indistinguishability argument cannot be invoked for k-decision lists in the fixed distribution setting.

Theorem 5.1 *Let $f, g \in k$-DL. If f, g are not D-equivalent, then there exists a k-term t which separates f, g.*

Proof: Let L be a decision list with items (t_i, b_i) representing f, and L' a decision list with items (t'_j, b'_j) representing g. For the sake of simplicity, we assume that both lists have the same length r (using redundant items for one list if necessary). Let L_k and L'_k be the lists starting both with the sublist $S_k = [(t_1, b_1), (t'_1, b'_1), \ldots, (t_k, b_k), (t'_k, b'_k)]$ and ending with the remaining items of, respectively, L and L'. Let f_k, g_k be the functions represented by these lists, respectively. Note that $L_r = L'_r$. The maximal index k such that f is D-equivalent to f_k and g is D-equivalent to g_k is therefore smaller than r. Let $P \subseteq \{0,1\}^n$ be the set of boolean vectors of positive probability, A the subset of vectors from P which satisfy one of the terms in S_k, $B = P \setminus A$, $T = \{x | \; t_{k+1}(x) = 1\}$, and $T' = \{x | \; t'_{k+1}(x) = 1\}$. Since L_k represents f and L'_k represents g up to D-equivalence, it follows that:

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

The maximality of k implies that:

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

[4] $n-i$ remaining dimensions and 2 values for b

[5] The number of dimensions needed sum up to $k \leq n-k$ ($\Leftrightarrow k \leq n/2$).

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

This theorem implies that for any fixed distribution D, sufficiently close approximations of the parameters $D[f(x) = 1 | t(x) = 1]$, where t ranges over all k-terms, determines f up to D-equivalence.

Corollary 5.2 *For any fixed distribution D over $\{0, 1\}^n$, k-DL is k-RFA learnable for all $k \leq n$.*

The running time of the learning algorithm depends on the specific choice of k and D, and is certainly not polynomial in general. It is however not hard to show that the following holds.

Theorem 5.3 *With respect to the uniform distribution, 1-DL is efficiently 1-RFA learnable and, for constant j, k, j-alt-k-DL is efficiently k-RFA learnable.*

The basic idea is to define ϵ-approximate D-equivalence in a proper way, and to show that the approximations of the relevant probability parameters are sufficiently close if a sample of polynomial size is available. In the full paper, more efficient algorithms for special choices of j, k, D are discussed.

6 k-RFA Learnability of k-TOP

In this section we explore the learnabilty of k-TOP in the k-RFA model. Of particular interest is our observation that k-TOP is weakly k-RFA learnable; in conjunction with earlier results, this shows that weak learning does *not* in general imply strong learning in the k-RFA model (for $k \leq n-2$), in stark contrast with the PAC model. We also show that k-TOP is (inefficiently) k-RFA learnable with respect to the uniform distribution with polynomial sample complexity.

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 efficient for constant k. This follows from the following lemma, which is a slight modification of a similar result in [Jac94].

Lemma 6.1 *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*

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

Because there are only n^k k-parities, and because the correlation of a k-parity with target f can be efficiently and accurately estimated by sampling from a k-RFA oracle, there is an algorithm that is efficient in n (although exponential in k) for weakly learning k-TOP from a k-RFA oracle.

6.2 Weak and Strong k-RFA Learning

In the PAC model of learning, weak learnability implies strong learnability. Existing proofs for this fact are based on the notion of hypothesis boosting. Therefore, an obvious approach to turning the weak learning result above 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. In particular, by the above we know that k-TOP is weakly k-RFA learnable (even efficiently for constant k), but we will now prove that k-TOP is *not* strongly k-RFA learnable (even information theoretically), for $k \leq n-2$. In fact, we have already done most of the work by showing that k-DL is not k-RFA learnable for $k \leq n-2$. All that remains to show is that k-DL is a subclass of k-TOP. And this is easy to see: given a k-DL representation $(t_1, b_1), \ldots, (t_d, b_d)$, we can write this equivalently as a threshold over the following sum of the terms t_i:

$$\sum_{i=1}^{d} 2^{d-i}(-1)^{b_i} t_i.$$

Furthermore, it follows from the definition of the Fourier transform that each of these k-terms can be written as a sum of k-parities. Thus f can be written as a k-TOP (with size exponential in k and d, the size of the k-DL representation of f).

On the other hand, boosting *can* be employed to good use under certain conditions in RFA models. For example, we now use hypothesis boosting to argue that k-TOP functions can be ϵ-approximated efficiently from a K-RFA oracle, where K depends polynomially on the size of the target (and other parameters). 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 6.2 *k-TOP is K-RFA learnable, where K is polynomial in the size s of the target function, in k, and in $\log \epsilon^{-1}$, where ϵ is the accuracy required of the learning algorithm. Note that K does not depend on n. The learning algorithm runs in time n^k but is otherwise polynomial in the usual PAC parameters.*

Proof: As noted earlier, there is an n^k-time weak k-RFA learning algorithm for k-TOP. In fact, the weak hypothesis produced by this learner will be a $(\frac{1}{2} - \Omega(\frac{1}{s}))$-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 [Fre90, Fre93] to this weak learner will produce an ϵ-hypothesis for f consisting of a majority vote over at most $O(s^2 \log \epsilon^{-1})$ weak hypotheses. As each weak hypothesis is defined over only k bits of the input, the algorithm actually only needs access to $K = O(ks^2 \log \epsilon^{-1})$ bits of each instance. □

6.3 Polynomial Sample Size for Uniform k-TOP Learning

Our most general positive result for k-TOP is a (not necessarily efficient) algorithm for learning k-TOP with polynomial sample complexity from a uniform-distribution k-RFA oracle. 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. That k-TOP is learnable from a uniform-distribution k-RFA oracle with polynomial sample complexity then follows by observing that the low-order Fourier coefficients can be efficiently estimated from such an oracle.

Lemma 6.3 *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 $F = \sum_{a \in I_k} \hat{F}(a)\chi_a$ on the domain of f such that $f = \mathrm{sign}(F)$, the coefficients $\hat{F}(a)$ of F are all integer-valued, and $\sum_{a \in I_k} |\hat{F}(a)| \leq s$. Now applying the generalized Parseval's identity we obtain the following:

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

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

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

Now note that again applying Parseval we have

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

Thus $E[|F|] - E[g \cdot F] \leq \epsilon$. Furthermore, since g is Boolean, every one of the terms $g(x)F(x)$ in $E[g \cdot F]$ has magnitude $|F(x)|$. This means that $E[g \cdot F] \leq E[|F|]$, 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 $E[|F|] - E[g \cdot F]$. Therefore f and g can differ on at most an ϵ fraction of the x's. □

Theorem 6.4 *k-TOP is (not necessarily efficiently) learnable from a uniform-distribution k-RFA oracle with sample complexity polynomial in the size of the target function.*

Proof: By Chernoff, a sample of size polynomial in k, s, ϵ^{-1}, and $\log(n/\delta)$ from the k-RFA oracle is sufficient to estimate, with probability at least $1 - \delta$, all of the k-order or less Fourier coefficients of f to within $\epsilon/2s$. By the preceeding 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. □

7 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, can be used further, both in the study of the learnability of other classes and also for other RFA scenarios.

The RFA model also enables us to reformulate the following open problem, which significantly predates learning theory research. 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 [Bru90] 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. By our Fourier characterization of k-RFA learnability, we know that when learning from a 1-RFA oracle with respect to the uniform distribution, we can obtain good estimates of the Chow parameters, and in fact these parameters capture *all* of the information available from the oracle. Thus, the above open question is equivalent to the following RFA question: is the class of perceptrons efficiently and properly 1-RFA learnable?

Since perceptrons are efficiently (and properly) PAC learnable, the above problem is essentially reducible to the following one. Given (approximations of) the Chow parameters, is there an efficient test which decides the true classification of a given instance, and which succeeds for almost all the instances? One natural candidate for such a test is the following one. Define for every $1 \leq i \leq n$ and every $b \in \{0,1\}$ the conditional probability $p_b^i = \Pr[f(x) = 1 | x_i = b]$, and define $p = \Pr[f(x) = 1]$. Then take x to be a positive instance if $\frac{1}{n}\Sigma_{i=1}^{n} p_{x_i}^i > p$, otherwise treat it as negative. At this time, we have no proof that this test succeeds for most of the instances of every perceptron, nor do we have a counterexample of a perceptron for which the test fails for a significant fraction of the instances.

References

[BDD93] Shai Ben-David and Eli Dichterman. Learning with restricted focus of attention. In *Proceedings of the 6th Annual Workshop on Computational Learning Theory*, pages 287–296. ACM Press, New York, NY, 1993.

[BDD94] Shai Ben-David and Eli Dichterman. Learnability with restricted focus of attention guarantees noise-tolerance. In *5th International Workshop on Algorithmic Learning Theory, ALT'94*, pages 248–259, 1994.

[Bru90] Jehoshua Bruck. Harmonic analysis of polynomial threshold functions. *SIAM Journal of Discrete Mathematics*, 3(2):168–177, May 1990.

[EHKV89] Andrzej Ehrenfeucht, David Haussler, Michael Kearns, and Leslie G. Valiant. A general lower bound on the number of examples needed for learning. *Information and Computation*, 82:247–261, 1989. First appeared in *Proceedings of the 1st Annual Workshop on Computational Learning Theory.*

[Fre90] Yoav Freund. 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, 1990.

[Fre93] Yoav Freund. *Data Filtering and Distribution Modeling Algorithms for Machine Learning.* PhD thesis, University of California, Santa Cruz, September 1993.

[Jac94] Jeffrey C. Jackson. 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, 1994.

[Jac95] Jeffrey Charles Jackson. *The Harmonic Sieve: A Novel Application of Fourier Analysis to Machine Learning Theory and Practice.* PhD thesis, Carnegie Mellon University, August 1995. Available as technical report CMU-CS-95-183.

[Kea93] Michael J. Kearns. Efficient noise-tolerant learning from statistical queries. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, pages 392–401, 1993.

[KS90] Michael J. Kearns and Robert E. Schapire. Efficient distribution-free learning of probabilistic concepts. In *Proceedings of the 31st Annual IEEE Symposium on Foundations of Computer Science*, pages 382–391, 1990.

[Riv87] Ronald L. Rivest. Learning decision lists. *Machine Learning*, 2:229–246, 1987.

Appendix

A Proofs of Section 4

A.1 Proof of Theorem 4.3

Denote by $x^{(k)}$ the element of the n-dimensional Boolean cube which we get if we flip the k-th bit of x. Now, let $E_k = \{\{x, x^{(k)}\} \mid x \in \{0, 1\}^n\}$ the set of edges of the Boolean cube parallel to dimension k.

For $1 \le k \le n$, an edge e of the Boolean cube, and $b \in \{0, 1\}$ let $p_e^b \triangleq \Pr_{x \in D}[x \in e \wedge f(x) = b]$. Also let $p_e^{\min} \triangleq \min\{p_e^0, p_e^1\}$ and $p_e^{\max} \triangleq \max\{p_e^0, p_e^1\}$. By $\hat{p}_e^b(m)$ denote the empirical probability corresponding to p_e^b and a sample m. Let $\hat{p}_e^{\min}(m) \triangleq \min\{\hat{p}_e^0(m), \hat{p}_e^1(m)\}$ and $\hat{p}_e^{\max}(m) \triangleq \max\{\hat{p}_e^0(m), \hat{p}_e^1(m)\}$. We omit the parameter m when it is clear from the context.

Let $N \triangleq 2^n$ and $\beta \in (1, 2]$. For $0 \le i \le n$ define $\tau_i \triangleq \epsilon/(N \cdot \beta^{4(n-i)+1})$.

The following lemma, which can be proven by Chernoff bounds, gives a sufficient bound of the sample size, such that (w.h.p.) for each edge e and each label b the empirical probability of drawing $\langle e, b\rangle$ is a good approximation for the real probability.

Lemma A.1 *For $\beta \in (1, 2]$ and τ_0 set w.r.t. β, let*

$$m' = \left\lceil \frac{\max\{3, 2\beta^2\}}{\tau_0(\beta-1)^2} \ln \frac{2N \log N}{\delta} \right\rceil = \left\lceil \frac{\max\{3, 2\beta^2\} N \beta^{4n+1}}{\epsilon(\beta-1)^2} \ln \frac{2N \log N}{\delta} \right\rceil .$$

If we call the $(n-1)$-RFA oracle m' times for each k, then with probability at least $1-\delta$, the following holds for all k, all $e \in E_k$, and all $b \in \{0, 1\}$

$$\begin{aligned} \hat{p}_e^b &< \max\{\beta\tau_0, \beta p_e^b\} \text{ and} \quad (1)\\ p_e^b &< \max\{\tau_0, \beta \hat{p}_e^b\}. \quad (2) \end{aligned}$$

Note that (1) and (2) (resp.) imply

$$\begin{aligned} \hat{p}_e^b \ge \beta\tau_0 &\Rightarrow p_e^b > \hat{p}_e^b/\beta \text{ and} \quad (3)\\ p_e^b \ge \tau_0 &\Rightarrow \hat{p}_e^b > p_e^b/\beta. \quad (4) \end{aligned}$$

From now on we assume that the inequalities (1) and (2) of Lemma A.1 hold.

Definition A.2 *For a dimension k let $e_1, e_2 \in E_k$. Let $d(e_1, e_2)$ denote the Hamming distance*[6] *between e_1 and e_2. For an edge $e_0 \in E_k$ and $i = 0, \ldots, n-1$ let*

$$E_k^i(e_0) \triangleq \{e \in E_k \mid d(e_0, e) = i\}.$$

Obviously, the sets $E_k^i(e_0)$ for $0 \le i \le n-1$ form a partition of E_k.

We call an edge e

$$\begin{cases} \textit{insignificant} \text{ of degree } i, & \text{if } \hat{p}_e^{\max} < \tau_i \\ \textit{pure} \text{ of degree } i, & \text{if } \hat{p}_e^{\min} < \tau_i \le \hat{p}_e^{\max} \\ \textit{impure} \text{ of degree } i, & \text{if } \tau_i \le \hat{p}_e^{\min} \end{cases} .$$

Lemma A.3 *Choose an edge $e_0 \in E_k$ for an arbitrary dimension k. Then, a hypothesis h defined as follows is ϵ-good. For an edge $e = \{x, x^{(k)}\} \in E_k^i(e_0)$ let*

$$\begin{cases} h(x) \text{ and } h(x^{(k)}) \text{ arbitrary}, & \text{if } e \text{ is insignificant of degree } i \\ h(x) = b \text{ and } h(x^{(k)}) = b, & \text{if } e \text{ is pure of degree } i \text{ and } \hat{p}_e^{\max} = \hat{p}_e^b \\ h(x) = f(x) \text{ and } h(x^{(k)}) = f(x^{(k)}), & \text{if } e \text{ is impure of degree } i \end{cases} .$$

[6]Define the Hamming distance between two edges parallel to dimension k by the Hamming distance between the endpoints of these edges for which bit k is 0.

Proof: If h makes a mistake on instance x then $\Pr_{x\in D}[x] < \beta\tau_i \le \epsilon/N$ by (2). Thus, h is ϵ-good. ∎

The following algorithm determines an edge e_0 and then iteratively ($i = 0, \ldots, n-1$) computes the values of a hypothesis h of the instances covered by the edges of $E_k^i(e_0)$.

Algorithm `EDGE`

Step 0: Fix an edge e_0 which is insignificant or pure of degree 0. (Note that such an edge exists[7].) Let k be the dimension such that $e_0 \in E_k$. Set the value of the hypothesis h to b for both instances of e_0, where b is such that $\hat{p}_{e_0}^{\max} = \hat{p}_{e_0}^{b}$ (if there is a tie[8] set $b = 0$).

Steps $1 \le i \le n-1$: In Step i we fix hypothesis h on the instances of the edges of $E_k^i(e_0)$. We can consider the edges $e \in E_k^i(e_0)$ in an arbitrary order.

If **e is insignificant or pure of degree i**, set the hypothesis h to b for both instances of e, where b is such that $\hat{p}_e^{\max} = \hat{p}_e^{b}$ (if there is a tie set $b = 0$).

If **e is impure of degree i**, fix an $e' \in E_k^{i-1}(e_0)$ which is a neighbor of e. By e_1 and e_2 denote the two edges of the Boolean cube which connect e and e'. Denote the instances as Figure 1.

Let $b = h(x_4)$. Set

$$h(x_3) = \begin{cases} b, & \text{if } \hat{p}_{e_1}^{\neg b} < \beta^2\tau_{i-1} \\ \neg b, & \text{if } \hat{p}_{e_1}^{\neg b} \ge \beta^2\tau_{i-1} \end{cases} \qquad \text{and} \qquad h(x_2) = \neg h(x_3).$$

Lemma A.4 *Let k and e_0 chosen by Algorithm* `EDGE` *in Step 0 and h the hypothesis produced by the algorithm. For every $0 \le i \le n-1$ and every edge $e = \{x, x^{(k)}\} \in E_k^i(e_0)$ which is impure of degree i, $h(x) = f(x)$ and $h(x^{(k)}) = f(x^{(k)})$ is valid.*

Proof: We prove this lemma by induction on the steps of Algorithm `EDGE`. Because e_0 is not impure (by definition), the lemma holds for $i = 0$.

Let $i \ge 1$ and assume inductively that the lemma is valid for $i-1$. Let e an impure edge of $E_k^i(e_0)$. Now, we make a case distinction according to the classification of e'.

e' is insignificant or pure of degree $i-1$: Choose b such that $\hat{p}_{e'}^{b} = \hat{p}_{e'}^{\max}$ (if there is a tie set $b = 0$). If $f(x_3) = b$, then

$$\hat{p}_{e'}^{\neg b} = \hat{p}_{e'}^{\min} < \tau_{i-1} \overset{(2)}{\Rightarrow} p_{e_1}^{\neg b} \le p_{e'}^{\neg b} < \beta\tau_{i-1} \overset{(1)}{\Rightarrow} \hat{p}_{e_1}^{\neg b} < \beta^2\tau_{i-1}$$

otherwise ($f(x_3) = \neg b$)

$$\hat{p}_e^{\min} \ge \tau_i = \beta^4\tau_{i-1} \overset{(3)}{\Rightarrow} p_{e_1}^{\neg b} \ge \Pr_{x\in D}[x_3] \ge p_e^{\min} > \beta^3\tau_{i-1} \overset{(4)}{\Rightarrow} \hat{p}_{e_1}^{\neg b} > p_{e_1}^{\neg b}/\beta > \beta^2\tau_{i-1}. \tag{5}$$

e' is impure of degree $i-1$: Let $b = h(x_4)$. Then $b = f(x_4)$ because of the induction hypothesis. If $f(x_3) = b$ then $p_{e_1}^{\neg b} = 0$ and thus, $\hat{p}_{e_1}^{\neg b} = 0 < \beta^2\tau_{i-1}$. Otherwise ($f(x_3) = \neg b$) we can use the sequence of implications (5).

Therefore, we get $f(x_3) = h(x_3)$ for both cases. Hence, the predictions of Algorithm `EDGE` for $h(x_3)$ and $h(x_2)$ are correct. ∎

Now we finish the proof of Theorem 4.3. Set $m = m'n (= m' \log N)$ regarding to Lemma A.1. Now, this lemma guarantees that with probability at least $1-\delta$ Lemmas A.3 and A.4 can be applied. Hence, Algorithm `EDGE` produces an ϵ-good hypothesis h with probability at least $1-\delta$. Now, we choose $\beta = 1 + \frac{1}{4n+1}$ such that $\beta^{4n+1} < e$. We get

$$m' = \left\lceil \frac{3(4n+1)^2 N}{\epsilon} \left(1 + \frac{1}{4n+1}\right)^{4n+1} \ln \frac{2N\log N}{\delta} \right\rceil = O\left(\frac{N\log^2 N}{\epsilon} \log \frac{N}{\delta}\right).$$

□

[7] *E.g.*, the first edge of the target $(n-1)$-DL f.

[8] If an edge e is insignificant of degree i, it may happen that $\hat{p}_e^{\max} = \hat{p}_e^{\min} < \tau_i$.

A.2 Proofs of Hardness Results for Decision Lists

Theorem A.5 *$(n-k)$-DL is* **not** *$(n-k)$-RFA learnable for $k \geq 2$.*

Proof: First we will prove the claim for $k = 2$ and then expand the result to $k \geq 2$.

For $n = 3$ we define scenarios $\langle f_1^{(3)}, D_1^{(3)} \rangle$ and $\langle f_2^{(3)}, D_2^{(3)} \rangle$ as shown in Figure 2. Obviously, $f_1^{(3)}, f_2^{(3)} \in$ 1-DL$_3$. Furthermore, the scenarios are different ($d_s = \frac{1}{5}$) and 1-RFA equivalent. Using Lemma 3.2 we obtain two different $(n-2)$-RFA equivalent scenarios $\langle f_1^{(n)}, D_1^{(n)} \rangle$ and $\langle f_2^{(n)}, D_2^{(n)} \rangle$. Induction over n implies that $f_1^{(n)}, f_2^{(n)}$ are expressible as $(n-2)$-decision-lists

$$\langle (x_1, 1), (x_2, 0), (m_1, 0), \ldots, (m_{2^{n-3}}, 0), (1, 1) \rangle, \ \langle (x_1, 1), (x_2, 0), (m_1', 0), \ldots, (m_{2^{n-3}}', 0), (1, 1) \rangle,$$

where the m_i (m_i') are all $(n-2)$-terms contained in the DNF-formula for the parity-function $x_3 \oplus \ldots \oplus \overline{x}_n$ ($x_3 \oplus \ldots \oplus x_n$).

The generalization of the non-learnability result to $k \geq 2$ is quite simple. Let $n' = n+2-k$. For $k > 2$ we embed the $(n'-2)$-RFA$_{n'}$ equivalent scenarios, constructed as described above, into $(n-k)$-RFA$_n$ equivalent scenarios of higher dimensions $n > n'$ as follows:

$$\langle f_i^{(n)}, D_i^{(n)} \rangle (x_1, \ldots, x_{n'}, \ldots, x_n) \triangleq \langle f_i^{(n')}, 2^{n'-n} D_i^{(n')} \rangle (x_1, \ldots, x_{n'}).$$

□

$f_1^{(3)}$ $(10 \cdot D_1^{(3)})$

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

$f_2^{(3)}$ $(10 \cdot D_2^{(3)})$

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

$f_1^{(4)}$ $(20 \cdot D_1^{(4)})$

x_3x_4 \ x_1x_2	00	01	11	10
00	$1^{(1)}$	$0^{(2)}$	$1^{(1)}$	$1^{(1)}$
01	$0^{(1)}$	$0^{(1)}$	$1^{(2)}$	$1^{(1)}$
11	$1^{(1)}$	$0^{(2)}$	$1^{(1)}$	$1^{(1)}$
10	$0^{(1)}$	$0^{(1)}$	$1^{(2)}$	$1^{(1)}$

$f_2^{(4)}$ $(20 \cdot D_2^{(4)})$

x_3x_4 \ x_1x_2	00	01	11	10
00	$0^{(1)}$	$0^{(1)}$	$1^{(2)}$	$1^{(1)}$
01	$1^{(1)}$	$0^{(2)}$	$1^{(1)}$	$1^{(1)}$
11	$0^{(1)}$	$0^{(1)}$	$1^{(2)}$	$1^{(1)}$
10	$1^{(1)}$	$0^{(2)}$	$1^{(1)}$	$1^{(1)}$

Figure 2: Karnaugh-diagrams for the scenarios from Theorem 4.1.

Proof of Theorem 4.6: It is sufficient to prove the claim for $k = 2$.

We construct the following scenarios $\langle f_1^{(n)}, D_1^{(n)} \rangle$ and $\langle f_2^{(n)}, D_2^{(n)} \rangle$. For $n = 2$ we start with the scenarios defined in Figure 3. For $n > 2$ we obtain $\langle f_1^{(n)}, D_1^{(n)} \rangle$ and $\langle f_2^{(n)}, D_2^{(n)} \rangle$ by applying the *linear construction scheme* defined in Section 3. It follows that $f_1^{(n)}$ and $f_2^{(n)}$ are expressible as 1-decision-lists $\langle (\overline{x_n}, 0), \ldots, (\overline{x_3}, 0), (x_2, 1), (x_1, 1), (1, 0) \rangle$ and $\langle (\overline{x_n}, 0), \ldots, (\overline{x_3}, 0), (\overline{x_2}, 1), (\overline{x_1}, 1), (1, 0) \rangle$. Furthermore, $\langle f_1^{(n)}, D_1^{(n)} \rangle$ and $\langle f_2^{(n)}, D_2^{(n)} \rangle$ are $(n-2)$-RFA equivalent. (For a complete proof we refer the reader to the full paper.)

Notice that $\langle f_1^{(n)}, D_1^{(n)} \rangle$ and $\langle f_2^{(n)}, D_2^{(n)} \rangle$ are not different ($d_s = 0$), so we cannot argue analogously to Section 3. On the other hand, assume, we have a hypothesis h which is consistent with both scenarios, *i.e.*, h and $f_i^{(n)}$ are $D_i^{(n)}$-equivalent for $i \in \{1, 2\}$. Then the subfunction $h'(x_1, x_2) \triangleq h(x)|_{x_3 = \ldots = x_n = 1}$ is uniquely determined:

$$h'(0, 0) = h'(1, 1) = 0, \quad h'(0, 1) = h'(1, 0) = 1.$$

Hence, $h' \in$ PAR$_2$ and therefore, $h \notin$ 1-DL. □

$f_1^{(2)}\ (3 \cdot D_1^{(2)})$

x_2 \ x_1	0	1
0	$0^{(1)}$	$1^{(1)}$
1	$1^{(1)}$	$1^{(0)}$

$f_2^{(2)}\ (3 \cdot D_2^{(2)})$

x_2 \ x_1	0	1
0	$1^{(0)}$	$1^{(1)}$
1	$1^{(1)}$	$0^{(1)}$

$f_1^{(3)}\ (6 \cdot D_1^{(3)})$

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

$f_2^{(3)}\ (6 \cdot D_2^{(3)})$

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

$f_1^{(4)}\ (12 \cdot D_1^{(4)})$

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

$f_2^{(4)}\ (12 \cdot D_2^{(4)})$

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

Figure 3: Karnaugh-diagrams for the scenarios from Theorem 4.6.

B Summary of Results about Decision Lists

Figure 5 (Figure 6, resp.) summarizes the results shown[9] in this paper and known results from literature ([Riv87]) for proper (improper, resp.) RFA-learning. A legend is given in Figure 4.

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

Figure 4: Legend for Figures 5 and 6.

[9] or directly implied by these results

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

Figure 5: Summary of results for proper 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	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$-$	$+i$	$+i$

Figure 6: Summary of results for improper b-RFA-learning of a-DL.