Learning with Limited Visibility

Eli Dichterman
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.
E-mail: eli@cdam.lse.ac.uk

Abstract

This paper surveys recent studies of learning problems in which the learner faces restrictions on the amount of information he can extract from each example he encounters. Our main framework for the analysis of such scenarios is the RFA (Restricted Focus of Attention) model. While being a natural refinement of the PAC learning model, some of the fundamental PAC-learning results and techniques fail in the RFA paradigm; learnability in the RFA model is no longer characterized by the VC dimension, and many PAC learning algorithms are not applicable in the RFA setting. Hence, the RFA formulation reflects the need for new techniques and tools to cope with some fundamental constraints of realistic learning problems. We also present some paradigms and algorithms that may serve as a first step towards answering this need.

Two main types of restrictions can be considered in the general RFA setting: In the more stringent one, called k-RFA, only k of the n attributes of each example are revealed to the learner, while in the more permissive one, called k-wRFA, the restriction is made on the size of each observation (k bits), and no restriction is made on how the observations are extracted from the examples.

We show an information-theoretic characterization of 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 show 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.

We also show a general technique for composing efficient k-RFA algorithms, and apply it to deduce, for instance, the efficient k-RFA learnability of k-DNF formulas, and the efficient 1-RFA learnability of axis-aligned rectangles in the Euclidean space $\mathbf{R}^n$. We also prove the k-RFA learnability of richer classes of Boolean functions (such as k-decision lists) with respect to a given distribution, and the efficient $(n-1)$-RFA learnability (for fixed n), under product distributions, of classes of subsets of $\mathbf{R}^n$ which are defined by mild surfaces.

For the k-wRFA restriction, we show that for $k = O(\log n)$, efficient k-wRFA learning is robust against classification noise. As a straightforward application, we obtain a new simple noise-tolerant algorithm for the class of k-decision lists, by constructing an intuitive k-wRFA algorithm for this task.

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 [42], specific bounds on the sample size needed for such generalization have been proven [14], and efficient learning algorithms have been designed for specific cases (cf. [41]).

It has also been noticed that in many realistic scenarios, the samples from which the learner has to generalize are not fully specified [28, 30]. The importance of this "focus-of-attention" problem has been noticed since the emergence of the computational learning theory [1]. However, the learning models which have been formulated for studying this type of problems usually assume—sometimes implicitly [13]—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 [6], 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 [6], 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. Note that such constraints on the simultaneous accessibility of the relevant parameters may be induced by either practical considerations (*e.g.*, when finding the value of each parameter incurs some cost), or by some inherent constraints (as is the case in particle physics).

The scenario described above is an example for the following common statistical problem: How can a function, which may be deterministic once all the relevant parameters are taken into account, be inferred from samples of its (probabilistic) behavior with respect to a limited number of these parameters? As it is desirable to learn the function itself, we usually seek finer strategies rather than just finding the correlation between variables.

Let us stress two common features of such learning problems: the first one is that the learner is not supplied with the random examples themselves, but rather with some partial information derived from them. The second one is that the learner has some freedom in choosing the type of information which will be extracted from each example.

The RFA framework suggests that, among other characteristics, any learning problem may also be characterized by the amount (and the type) of information that has to be extracted from each training example along its learning process. The model incorporates a mechanism for extracting observations from examples into the theory of learning from randomly drawn examples. Such a mechanism intermediates between the learner and the (labeled) examples provided by the teacher (or the environment) and enables us to focus our attention on the nature and amount of information that a learning algorithm requires.

The RFA framework provides an additional angle along which learning problems may be analyzed; on top of any other characteristics, any learning problem may also be characterized by a set of 'projections' available to the learner. Each projection is a mapping from the space of labeled examples into an 'observation space'. Rather than enjoying the ability to freely exploit every given example, as in previous models of learning, our learner selects one of the projections before each example is being drawn, and observes only the image of the labeled example under this selected projection.

From a philosophical point of view, one may attribute all randomness (or 'noise' phenomena) in nature to the effect of some invisible deterministic variables on the observable environment. As such, models concerning noisy information like that of Kearns and Li [26], as well as those dealing with probabilistic concepts like the Kearns-Schapire p-concepts [28], may be viewed as special cases of our scenario.

Previous models have usually considered some *fixed* mapping between examples and observations. In the standard PAC model this mapping is the identity mapping. However, there are some generalized models which consider a fixed mapping other than the identity mapping.

The notion of limited visibility has been considered in some of the known learning models. A model for learning problems with hidden variables has been considered in [28, 30]. In that model there is a fixed set of variables visible to the learner, while the underlying concept depends also on a set of variables hidden from the learner. Thus, each example (over all the variables) is mapped into its projection on the set of visible variables. With respect to the visible variables, a probabilistic behavior is induced, and the learner tries to learn this behavior.

Another related model is the model of 'switching concepts' introduced by Blum and Chalasani [13]. In that model a probabilistic behavior of the unknown concept is generated by switching between several different (deterministic) concepts. By attributing these 'switches' to the values of some hidden variable, one gets yet another example of a scenario that may be viewed as learning under limited visibility.

However, all previously discussed models fix in advance a unique projection to be applied to the examples. In such models, the most the learner can learn is the probabilistic behavior induced by the underlying concept and distribution on the observation space. Consequently, such settings fail to model the search for relevant attributes—a search that plays a significant role in many realistic learning scenarios.

Another aspect in which the RFA model deviates from the PAC scenario and most of its variants is the nature of communication between the student and the teacher. In the usual PAC scenario the communication is one-way—from the teacher to the student. This is in accordance with viewing the PAC teacher as modelling an 'indifferent environment' of an observing student. The scenario that our framework reflects is somewhat different; the learning process is interactive—the teacher's responses are a function of the student's projection queries. This is closer in spirit to the Learning By Distance (LBD) model [8] and the closely related Statistical Queries (SQ) model of Kearns [25]. In these models the student sends a query to the teacher (about the distance between some hypothesis and the target in the LBD model, or a 'statistical query' in the SQ framework) and receives, in response, an approximated answer from the teacher. This answer may also be viewed as some partial information derived from randomly drawn labeled examples.

The SQ model was motivated by the need to design noise-tolerant algorithms. Kearns proves in [25] that a learning algorithm which is restricted to use statistical queries (and hence cannot be sensitive to specific features of individual examples) is robust against classification noise. It follows from a result in [7] that an $O(\log n)$-RFA oracle can be efficiently simulated by statistical queries, and hence by Kearns' transformation [29] 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 [11]).

Furthermore, it is shown in [18] 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 [7], 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 [6, 7]. Another surprising difference is shown in [12]: 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 [6]. 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 all the learning parameters of the problem).

[12] extends our understanding of k-RFA learnability of Boolean functions in a number of ways. They 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. These classes have been chosen 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, the 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 the interest in k-RFA learnability questions. Also, a combination of some of the 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 the k-DL results, it is shown in [12] 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. [18]). In yet another contrast, it is also shown that, among other things, that distribution-free learnability of 1-DL requires access to at least half of the bits in each example.

The study of k-TOP is motivated in part by the fact that it is known to have useful Fourier properties [23]; furthermore, it has also been studied in the context of empirical machine learning [24]. The Fourier properties of k-TOP have been exploited to show first that k-TOP is weakly k-RFA learnable and that this learning is efficient for constant k. Second, 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.

Combining some of the k-DL and k-TOP results the following is obtained: 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 [35] 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.

This paper is organized as follows. In Section 2 we define our model of learning with Restricted Focus of Attention (RFA), defining also the two main variants of the restrictions considered in this paper. In the more stringent one, called k-RFA, the learner may observe only k attributes of each example (which consists of n attributes), and he may also observe the classification bit. In the more permissive restriction, called k-wRFA, the learner is only restricted to observe k bits of information, where no restriction is made on how they are obtained from the example. Using this formalism, we show how to embed various existing learning models in our generalized model. We also show that some known learning algorithms actually use only part of the available information, and therefore can be naturally reformulated as RFA learning algorithms.

In Section 4 some preliminary definitions of the classes of Boolean functions used in this work are given (parity functions, DNF and CNF representations, k-decision lists, k-TOPs, and AC^0 circuits), as well as some useful statistical theorems (Chernoff bounds, uniform convergence bounds) which we use throughout the paper, and the Fourier Transform which is used in [12] to prove several results. We also define the notion of RFA hardness,

and give a characterization theorem for RFA learnability over a finite domain, a useful tool in proving information-theoretic learnability or non-learnability.

In Section 6 we study learnability under the k-RFA restriction. We first develop in Section 6.1 a technical tool that takes as input PAC learning algorithms of given classes and outputs an efficient k-RFA algorithm for a class composed, in a specific manner, from the given input classes. We conclude that if a class is decomposable (in a formal sense that we define) into PAC-learnable classes, then it is efficiently k-RFA learnable. Applications of the decomposability scheme includes, for instance, an efficient k-RFA learning algorithm for the class of k-DNF formulas, and an efficient 1-RFA learning algorithm for the class of axis-aligned rectangles in $\mathbf{R}^n$.

In Section 6.2 we elaborate on the information-theoretic k-RFA learnability of classes of Boolean functions. Showing that the class of parity functions over $k+1$ Boolean variables is not k-RFA learnable, we conclude that parameterizing learnability with the k-RFA restriction, for $0 \leq k \leq n$, induces a real hierarchy over the set of learnable class. However, when the symmetric behavior of the parity functions with respect to the uniform distribution is broken, a rich class of functions is learnable. Thus, we show that under product distributions in which each Boolean attribute has the same biased probability $p \neq 1/2$ of being 1, the class of *all* Boolean functions is $(n-2)$-RFA learnable. We contrast this result by an information-theoretic hardness result which holds for the same class of distributions, stating that the class of *all* Boolean functions is not $\lfloor \frac{n}{8} \rfloor$-RFA learnable.

We also show k-RFA learning algorithms which learns the class of k-decision list under the uniform distribution (not efficiently), and a quasi-efficient algorithm (originate from [32]) which learns the class of for the rich classes of AC^0 circuits, under the k-RFA restriction, for k logarithmic in the number of attributes. Both algorithms can be generalized to learn under any fixed distribution.

In Section 6.4 we prove $(n-1)$-RFA learnability, under a product distribution, of any class of subsets of $\mathbf{R}^n$ that are defined in terms of their bounding surfaces, subjected to some 'mild behavior' assumptions. (The learnability is efficient for a fixed n).

In Section 7 we study the k-wRFA restriction. We show that visibility (the size of the learner's focus-of-attention) can be traded for sample complexity, the increase in the sample complexity being exponential in the number of bits reduced from the observation size. This implies that efficient wRFA learnability is insensitive to a reduction of $O(\log n)$ bits in visibility.

As for information-theoretic wRFA learnability, we show that every finite class is 1-wRFA learnable (and under a fixed distribution, a result in [8] implies that any class with finite VC dimension is 1-wRFA learnable).

In Section 8 we relate the RFA restrictions to noise-tolerant learnability. We prove that efficient $O(\log n)$-wRFA learnability guarantees robustness against classification noise. Applying this result to a new k-wRFA learning algorithm for the class of k-decision lists, we obtain a noise-tolerant k-wRFA learning algorithm for this class (a noise-tolerant PAC algorithm for this class was first shown in [34]). We also show an equivalence between efficient $O(\log n)$-wRFA learning and learning from statistical queries (Kearns' model). Thus, the RFA restriction may also serve as a simple sufficient condition for verifying robustness against classification-noise. Decatur and Gennaro [19] show that a further restriction on an SQ learner guarantees robustness against both classification noise and attribute noise (simultaneously). Showing that this restriction is essentially equivalent to the $O(\log n)$-RFA restriction, we conclude that the $O(\log n)$-RFA restriction guarantees noise-tolerance in this stronger model of noise.

We close this survey with some directions for further research.

2 The RFA Model

The RFA component may augment any model of learning from examples [10]. We present here the RFA framework as an extension of the well-known PAC model [41].

Let $\mathcal{F}$ be a class of $\{0,1\}$-valued functions (concepts) over an instance space X, and let D be a probability distribution over X. The *target distribution* D is used both for generating the random training examples for the learner and for measuring 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).

On top of these familiar PAC components we add a pair $\langle \mathcal{O}, \mathcal{W} \rangle$; $\mathcal{O}$ is a set called the *observation space*, and $W \subseteq \mathcal{O}^{X\times\{0,1\}}$ is a set of functions which we call *projections*. 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 models the act of focusing the attention on a set of features. This data-collection mechanism is the new element that distinguishes the RFA model from the familiar PAC learning scene.

Formally, we model this mechanism by a function $\Phi : \mathcal{O}^* \to W$, which selects the next projection based on the sequence of observations seen so far. We call such a function a *focusing function*.

Given a sequence of m instances $\overline{x} = (x_1, \ldots, x_m) \in X^m$, a target function $f \in \mathcal{F}$, and a focusing function Φ, the observation sample generated by $\overline{x}$, f and Φ is $sample(\overline{x}, f, \Phi) = (w_1(x_1, f(x_1)), \ldots, w_m(x_m, f(x_m)))$, 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 $error_{f,D}(h) \triangleq \Pr_{x\in D}[h(x) \neq f(x)]$, 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 [14], we model the hypothesis selection by a function $L : \mathcal{O}^* \to \mathcal{H}$, called a *learning function*. Given a sufficiently large sample observations, a successful learning function produces, with high confidence, a good hypothesis. This is formally defined as follows.

Definition 2.1 (learning via a set of projections) *A function class $\mathcal{F} \subseteq \{0,1\}^X$ is learnable via the projection class $W \subseteq \mathcal{O}^{X\times\{0,1\}}$, and using the hypothesis class $\mathcal{H}$, if there is an integer-valued sampling function $m(\cdot,\cdot)$, there is a focusing function $\Phi : \mathcal{O}^* \to W$, 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 $0 < \epsilon, \delta \le 1$*

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

Usually there is a natural complexity parameter n associated with the domain X, implying a parameterization of the other components of the learning problem ($\mathcal{F}, \mathcal{H}, \mathcal{O}$ and W). For instance, n can be the number of Boolean variables, or it can be the number of dimensions in the Euclidean space. Another complexity parameter is the smallest representation length of the target function f, denoted by $size(f)$. In such cases, the sample size may also depend on these complexity parameters.

Since we usually seek a learning *algorithm*, we want the 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: A learning algorithm is efficient if it runs in time $poly(\frac{1}{\epsilon}, \frac{1}{\delta}, n, size(f))$ (note that this also implies a polynomial sample complexity).

When the hypothesis class $\mathcal{H}$ is omitted it is assumed that $\mathcal{H} = \{0,1\}^X$ (*i.e.*, no restriction is made on choosing a hypothesis). The notion *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 the a more permissive setting obtained by requiring successful learning only when D is a member of some predetermined class of distributions over X. One such candidate is the class of product distributions (each variable has a fixed distribution independent of the values of other variables [36]). In some cases we shall even assume that the permissible class contains only one distribution (*i.e.*, the distribution is fixed and known to the learner).

Two main types of restrictions on the learner's focus of attention naturally rise, and are defined below.

The **k-RFA** restriction is a restriction on the way an observation can be obtained from a sampled instance. It is applied to the case where the instance space is parametrized by a complexity parameter which is the number of Boolean attributes ($X = \{0, 1\}^n$) or the number of Real attributes ($X = \mathbf{R}^n$). A k-RFA learner may choose to observe any k attributes out of the n attributes existing in each instance (and may also observe the classification bit). Hence, the set of projections used by a k-RFA learning algorithm is $W_k \triangleq \{w_I : I = \{i_1, \ldots, i_k\} \subseteq [n]\}$, where $w_I(x_1, \ldots, x_n, y) = (x_{i_1}, \ldots, x_{i_k}, y)$ for $I = \{i_1, \ldots, i_k\}$. Learnability under such a restriction is called k-RFA learnability.

Another natural but more permissive restriction is a restriction on the amount of information bits which exist in each observation (without any restriction on how they are extracted from the sampled instances). Hence, this is a restriction on the size of the observation space. We call learnability under such a restriction **k-wRFA** (k-weakly Restricted Focus of Attention) learnability. We say that a function class $\mathcal{F}$ is k-wRFA learnable if it is learnable via $W \subseteq (\{0,1\}^k)^{X \times \{0,1\}}$.

Notice that in the Boolean domain ($X = \{0, 1\}^n$), $(k+1)$-wRFA learnability implies k-RFA learnability and that, for $k = n$, both coincide with the standard PAC model.

The $(k+1)$-wRFA restriction is strictly weaker than the k-RFA restriction, as demonstrated by the following example. Consider a function class $\mathcal{F}$ which contains both the parity function $f_1(x_1, x_2) = x_1 \bigoplus x_2$ and its negation f_2, and let the instance space $\{0,1\}^2$ be uniformly distributed. Clearly, if a learner can observe only the value of one variable at a time (*i.e.*, he can observe either the value of x_1 or the value of x_2 but not both), then he is not capable of distinguishing between f_1 and f_2, and therefore cannot learn the class $\mathcal{F}$. If, however, the learner gets a single bit of information, having the value of $x_1 \bigoplus x_2 \bigoplus f(x_1, x_2)$, then one observation is sufficient to distinguish between the case $f \equiv f_1$ and the case $f \equiv f_2$. Thus, $\mathcal{F}$ is efficiently 1-wRFA learnable but not 1-RFA learnable. This example can be extended to demonstrate that the class of parity functions over $k+1$ Boolean variables is 1-wRFA learnable, but not k-RFA learnable.

3 Other Limited-Visibility Learning Models

We now briefly present several other models in which the learner's visibility is limited, and survey the main results known in these models (formulated in the general RFA notions)..

3.1 Hidden Variable Learning Problems

In some learning problems the learner is restricted to observe only a fixed partial set of the relevant attributes, while all the others are hidden from him. In RFA notions, the learner's attention is restricted to using only a fixed projection—the projection of each instance (over the set of all the relevant variables) onto the set of visible variables.

In such a scenario the learner actually observes a probabilistic behavior with respect to the visible variables; each assignment of values to the visible variables is associated with some probability of being classified as a positive instance, a probability which is induced by the underlying distribution on the instance space, and by the (deterministic) target function. Obviously, in such a case, the best that the learner can do is learning this probabilistic behavior.

A learning model for such scenarios was presented by Kearns and Schapire in [28]. In their model, a concept is not a binary function over the (observed) instance space X, but rather a real function $f : X \to [0, 1]$. An example (x, y) is generated by drawing a random $x \in X$ using the underlying distribution on X, and letting $y = 1$ with probability $f(x)$. The model is known in the literature as the *p-concept* model ('p' stands for 'probabilistic').

Two main results were obtained for hidden variables learning problems in the p-concept model. In [28] it is shown how to learn any Boolean formula of the form $\psi \wedge \theta$, where ψ is a conjunction over the set of visible variables, and θ is any Boolean function over the set of hidden variables. Kearns, Schapire and Sellie show in [30] how to learn any k-term DNF formula (DNF formula with up to k terms; see Section 4 for the definition of DNF formulas), where the formula may depend on both the hidden and the visible variables.

As mentioned above, we are mainly concerned with a different type of hidden variables learning problems, in which the learner can choose the set of visible variables, obeying some restrictions, such as the size of this set. In such scenarios, it seems natural to require that the learner eventually learns the behavior of the target concept with respect to all the relevant variables.

3.2 Learning Switching Concepts

Given an assignment of some values to the hidden variables, a deterministic concept is projected on the visible variables. Thus, assuming independence between the distribution of the hidden variable space and the distribution of the visible variables space, the overall probabilistic behavior of the underlying concept over the visible variables may be regarded as a 'mixture' of the projected (deterministic) concepts; each projected concept c_i is associated with a probability p_i such that $\sum_i p_i = 1$ (the probabilities are determined by the underlying distribution over the hidden variables). In this model, presented by Blum and Chalasani in [13], an example is classified using the concept c_i with probability p_i. Knowing the class from which these concepts are chosen, the learner may try to learn the probabilistic behavior of their mixture. Indeed, Blum and Chalasani show how to learn a mixture of monotone disjunctions. They also demonstrate that in some cases, the learner can learn more than just the probabilistic behavior of the mixture; he can learn the concepts themselves. One such case is the case where there is a 'majority concept' (a c_i for which $p_i > 1/2$), in which case the learner can learn this majority concept.

Another case in which the learner can learn the switching concepts (over the visible variables) is the case where the values of the hidden variables are rarely switched. In such cases, the learner is trying to predict the right classification of the next instance, knowing that the target concept (according to which the instance is classified) may be switched once in a while. It is shown in [13] that in such cases the learner can learn the switching concepts, and use them to make few mistakes on further examples—for each switch, the number of mistakes made by the learner can be bounded by the number of switching concepts.

3.3 Learning By Distances

Ben-David, Itai and Kushilevitz [8] propose a learning model which is characterized by an interaction between a learner and a teacher. In each round the learner proposes a hypothesis, and the teacher responds with an approximation of the distance between the hypothesis and the target, relative to some underlying metric. It is shown in [8], that PAC learnability with respect to a fixed distribution is equivalent to learnability by distances with respect to the metric induced by that distribution (the distance between two concept is the probability of drawing a point in their symmetric difference). Moreover, based on this equivalence, they show that, given a distribution D, rather then learning from the entire classified examples, the learner can settle for just one bit of information, indicating whether the example (randomly drawn according to D) is in the symmetric difference of the learner's current hypothesis and the target. Here is a formulation of this result in the RFA model.

Theorem 3.1 (Ben-David, Itai, Kushilevitz) *If $\mathcal{F}$ is (efficiently) PAC-learnable under the distribution D, then $\mathcal{F}$ is (efficiently) 1-wRFA learnable under the distribution D.*

Note that the above theorem implies that, under a fixed distribution, the hierarchy of k-wRFA learnability ($1 \le k \le n+1$) collapses to 1-wRFA learnability.

3.4 Learning from Statistical Queries

Another model which is strongly related to the RFA model is the model introduced by Kearns in [25] for learning from statistical queries. This model was developed as a conceptual tool for studying the capability of noise-tolerant learning. The idea is to restrict the learner in such a way that he cannot learn from specific examples, but rather has to use only statistics of large samples in order to learn the target concept. By showing that the effect of classification noise on such statistics can be canceled out, Kearns proved that every algorithm which learns from statistical queries is noise-tolerant against classification noise.

In Kearns' statistical model, the learner has no access to classified examples, but rather can get close approximations of efficiently computed events, and may use these approximations to find a good hypothesis. In Section 7 we show that efficient $O(\log n)$-wRFA learning is equivalent to learning from statistical queries. Thus, one may use the easily verifiable RFA condition to ensure noise-tolerant learning. Fortunately, most of the known learning algorithms (or some close variants of them) satisfy this RFA condition [7], and hence we can easily design a noise-tolerant version of them. An exception is the algorithm for learning the class of *all* parity function (a parity function is a binary functions which computes the parity of some fixed subset of its input binary variables). This class is known to be efficiently PAC learnable (by solving a system of linear equations modulo 2 [22]), but is (provably) not efficiently learnable from statistical queries [25]; it is not known whether this class is noise-tolerant learnable. (Note that if it is not, then the SQ model is strictly more restrictive than the model of learning with classification noise).

4 Preliminaries

4.1 Classes of Boolean Functions

We define here the classes of Boolean functions whose learnability with limited visibility is considered in this paper. The instance space is $X_n = \{0,1\}^n$, and we denote the index set $\{1,\ldots,n\}$ by $[n]$.

Definition 4.1 (Parity Functions) *For every set* $\mathrm{I} \subseteq [n]$ *Let $EVEN_I$ be the function which is* 1 *on x if the parity of the set $\{x_i : i \in \mathrm{I}\}$ is even, and* 0 *otherwise. Let* $\mathrm{ODD_I}(x) \triangleq 1 - \mathrm{EVEN_I}(x)$ *for every*

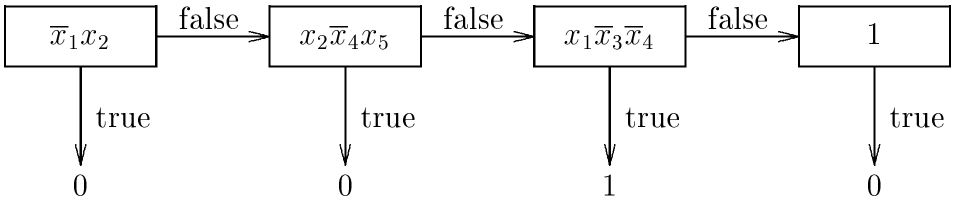

Figure 1: Example of a decision list.

$x \in X_n$. *Also denote:*

$$\begin{aligned}
\mathrm{PARITY_n} &\triangleq \{\mathrm{EVEN_I}, \mathrm{ODD_I} \mid \mathrm{I} \subseteq [\mathrm{n}]\} \\
\mathrm{PARITY_n^k} &\triangleq \{\mathrm{EVEN_{[k]}}, \mathrm{ODD_{[k]}}\} \qquad (\mathrm{k} \leq \mathrm{n})
\end{aligned}$$

Definition 4.2 (DNF and CNF representations) *A literal is a variable x_i or its negation $\bar{x}_i$. A term is a conjunction of literals, and a clause is a disjunction of literals. If the number of literals in a term or in a clause is bounded by k, than it is called a k-term or a k-clause, respectively. A* DNF *formula is a disjunction of terms, and a* CNF *formula is a conjunction of clauses. A k-*DNF *formula is a disjunction of k-terms, and a k-*CNF *formula is a conjunction of k-clauses. We also use the notation k-*DNF *and k-*CNF *to denote the class of all k-*DNF *formulas and k-*CNF *formulas, respectively (over n variables).*

Every function has both DNF and CNF representations. The efficient learnability of these classes of representations is still a main open and intriguing question in learning theory. However, for any fixed k, k-DNF and k-CNF are efficiently learnable [41], and we show in Section 6.1 that this remains true under the k-RFA restriction.

One of the classes whose RFA learnability is studied in this work is the class of decision lists, introduced by Rivest in [33]. 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 a 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 $\langle(\overline{x}_1 x_2, 0), (x_2\overline{x}_4 x_5, 0), (x_1\overline{x}_3\overline{x}_4, 1), (1, 0)\rangle$, 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.

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 4.3 (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 [33] that k-DL properly contains k-DNF $\cup$ k-CNF, and is efficiently PAC learnable for constant k. It is shown in [10] 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.

Definition 4.4 (TOP) *We denote by TOP (Thresholds Of Partities) 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 ($\overline{\text{parity}_n} \equiv \overline{x_1 \oplus x_2 \oplus \ldots \oplus x_n}$).

Finally, we define the class of $\mathbf{AC^0}$ Boolean circuits.

Definition 4.5 ($\mathbf{AC^0}$ Boolean Circuits) *An AC^0 circuit is a Boolean circuit which consists of 'and' and 'or' gates, and whose inputs are the literals $x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n$. The size of the circuit (number of gates) is bounded by a polynomial in n, and its depth is bounded by a constant. The class of functions computable by an AC^0 circuit of depth d is denoted by $AC^0[d]$.*

In Section 6.2, show that this class is 1-wRFA learnable by a quasi-polynomial algorithm.

4.2 Chernoff Bounds

The following form of Chernoff bounds, which we use throughout the paper, originate from [4].

Theorem 4.6 *Let $X_1, \ldots, X_m$ be independent random Boolean variables, with $\Pr[X_i = 1] = p$ for every $1 \le i \le m$. Then, for every $0 \le \gamma \le 1$, the following are bounds on the empirical average $\hat{p} \triangleq \frac{1}{m} \sum_{i=1}^{m} X_i$:*

$$\begin{aligned}
\Pr[\hat{p} > p + \gamma] &\le e^{-2m\gamma^2} \\
\Pr[\hat{p} < p - \gamma] &\le e^{-2m\gamma^2} \\
\Pr[\hat{p} > p(1 + \gamma)] &\le e^{-\frac{1}{3}m\gamma^2} \\
\Pr[\hat{p} < p(1 - \gamma)] &\le e^{-\frac{1}{2}m\gamma^2}
\end{aligned}$$

4.3 VC-Dimension and Uniform Convergence

The following Theorem, due to Vapnik and Chervonenkis [42], is useful in estimating, for a class of events, the number of examples needed to ensure uniform convergence of the empirical probabilities to the true event probabilities.

Theorem 4.7 (Vapnik, Chervonenkis 71) *Let $\mathcal{F} \subseteq \{0,1\}^X$, and let D be a distribution over X. Assume we want to estimate the probabilities $\Pr_{x \in D}[f(x) = 1]$ for all $f \in \mathcal{F}$ simultaneously. Define the VC-dimension of $\mathcal{F}$ to be: $\sup_{A \subseteq X}\{|A| : |\{f^{-1}[1] \cap A : f \in \mathcal{F}\}| = 2^{|A|}\}$. Then, a sample of size $m \geq \frac{16}{\epsilon^2}\left(d \log \frac{16d}{\epsilon^2} + \log \frac{4}{\delta}\right)$ guarantees that, with probability of at least $1 - \delta$, all estimates differ from the true probabilities by less than ϵ, where d is the VC-dimension of $\mathcal{F}$.*

4.4 Statistical Distance

Definition 4.8 (statistical distance) *We define a metric on the class of distributions over a (countable) domain X. Let D and D' be two distributions over X. The* statistical distance *between D and D' is defined by*

$$\Delta_s(D, D') \triangleq \frac{1}{2} \sum_{x \in X} |D[\{x\}] - D'[\{x\}]|$$

It is easy to verify that the statistical distance is indeed a metric (i.e., it is symmetric, it satisfies the triangle inequality, and $\Delta_s(D, D) = 0$ for every distribution D).

Proposition 4.9 *Let $\underline{D} = \langle D_i \rangle_{i=1}^m$ and $\underline{D}' = \langle D_i' \rangle_{i=1}^m$ be two product distribution over X^m. ($\underline{D} = \langle D_i \rangle_{i=1}^m$ is a product distribution if D_i and D_j are independent of each other, for every $1 \leq i < j \leq m$; for a countable domain X this implies $\underline{D}(x_1, \ldots, x_m) = \prod_{i=1}^m D_i(x_i)$). Then $\Delta_s(\underline{D}, \underline{D}') \leq \Sigma_{i=1}^m \Delta_s(D_i, D_i')$.*

Proof: Define $H_i \triangleq \langle D_1, \ldots, D_i, D_{i+1}', \ldots, D_m' \rangle$, for every $0 \leq i \leq m$. Since $H_0 = \underline{D}'$ and $H_m = \underline{D}$, it is sufficient to observe that $\Delta_s(H_{i-1}, H_i) = \Delta_s(D_i, D_i')$ for every $1 \leq i \leq m$, and to use the triangle inequality iteratively. ■

4.5 Efficient Sampling

The following definitions of efficiently computable distribution and efficiently samplable distribution originate from [9].

Definition 4.10 (efficiently computable distribution) *Let D be a distribution over X (with complexity parameter n), where '$\leq$' may is any efficiently computable order on X. D is* efficiently computable *if there exists an efficient (polynomial in n) deterministic algorithm, which on input x produces $D[\{u \in X : u \leq x\}]$.*

Definition 4.11 (efficiently samplable distribution) *Let D be a distribution over X (with complexity parameter n). D is* efficiently samplable *if there exists an efficient (polynomial in n) probabilistic algorithm which produces $x \in X$ with probability $D[\{x\}]$.*

Theorem 4.12 (Ben-David, Chor, Goldreich, Luby) *Every efficiently computable distribution is also efficiently samplable.*

4.6 The Fourier transform

While the Fourier transform has numerous uses in computer science (see, *e.g.*, [2]), we will use a somewhat nonstandard multidimensional version of the transform first applied to learning theory by Linial, Mansour, and Nisan [32]. 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, $\mathbf{E}[f \cdot g] = \sum_a \hat{f}(a)\hat{g}(a)$.

5 Characterizing RFA Learnability

In this chapter we give a full characterization of the function classes over a finite domain X which are learnable via a projection class W. Using this characterization, we develop a general tool for showing k-RFA learnability hardness, which we apply to deduce the k-RFA non-learnability of any class which contains both the even and the odd parity functions over $k+1$ boolean variables. This tool has been recently used by Birkendorf, Dichterman, Jackson, Klasner and Simon [12] to obtain hardness results for the k-RFA learnability of k-decision lists. A Fourier-based characterization of RFA learnability is also shown there.

First we formulate the notion of indistinguishability of two RFA scenarios. A *scenario* over the instance space X is defined to be a pair $\langle f, D\rangle$ of a function f and a distribution D over X. For $f \in \mathcal{F}$, $\langle f, D\rangle$ is also called an $\mathcal{F}$-scenario. The idea of proving hardness of RFA learning a class $\mathcal{F}$, by showing two different $\mathcal{F}$-scenarios which are indistinguishable by an RFA learner, was first used by Ben-David and Dichterman [6] in proving that the class of parity functions (over n variables) is not $(n-1)$-RFA learnable under the uniform distributions. For the case of k-RFA learnability of Boolean functions, the notion of indistinguishability among a set of different scenarios was defined in [12], where it was used to obtain hardness results for the k-RFA learnability of k-decision lists. We give here a full characterization of the function classes over a finite domain X which are learnable via a projection class W. This provides a general tool for either proving or disproving information-theoretic k-RFA learnability.

For a projection class $W \subseteq \mathcal{O}^{X\times\{0,1\}}$, we define a W-*equivalence* relation among scenarios as follows. Given a scenario $S = \langle f, D\rangle$ over X, we define for every projection $w \in W$ a distribution $p_{S,w}$ over the observation space $\mathcal{O}$. For every observation $z \in \mathcal{O}$, $p_{S,w}(z)$ is the probability of observing z via the projection w, when learning the scenario S. That is, $p_S(w,z) \triangleq \Pr_{x\in D}[w(x, f(x)) = z]$.

We say that the scenarios S and S' are W-equivalent if the distributions $p_{S,w}$ and $p_{S',w}$ are identical for every projection $w \in W$ (and hence, a learner which is restricted to use projections from W cannot distinguish between the two scenarios). However, the existence of a set of W-equivalent scenarios $\mathcal{Q}$ does not exclude the possibility of learnability via W, as it might be the case that there is a hypothesis h which is good for all them (*i.e.*, the learner does not have to distinguish between them). Thus, we call such a set W-hard, if there is $\epsilon > 0$, such that every hypothesis $h \in \mathcal{H}$ is ϵ-bad for at least one of the scenarios in the set. (The set is W-hard if it is hard for every $\mathcal{H}$).

Hence, to disprove the learnability of a function class $\mathcal{F}$ via W and using $\mathcal{H}$, it is sufficient to find a set of $\mathcal{F}$-scenarios which is W-hard for $\mathcal{H}$. For a finite instance space, the absence of such a set implies information-theoretic RFA learnability. (hence providing a full characterization of RFA learnability). The following two theorems are generalizations of the corresponding theorems (stated for the Boolean domain) in [12].

Theorem 5.1 *Let $\mathcal{F}$ and $\mathcal{H}$ be function classes, and let W be a projection class. If there is a finite set of $\mathcal{F}$-scenarios which is W-hard for $\mathcal{H}$, then $\mathcal{F}$ is not learnable via W using $\mathcal{H}$.*

Proof: Let $\mathcal{Q}$ be a finite set of $\mathcal{F}$-scenarios which is W-hard for $\mathcal{H}$, and let $\epsilon > 0$ be such that every hypothesis $h \in \mathcal{H}$ is ϵ-bad for some scenario in $\mathcal{Q}$. Assume A is an algorithm which uses a sample of m observations to learn the class $\mathcal{F}$ via W and using $\mathcal{H}$. Being W-equivalent, all the scenarios in $\mathcal{Q}$ induce the same m-fold product distribution P^m over the m-fold observation space. The W-hardness of $\mathcal{Q}$ implies that for each sequence z of m observations drawn from P^m, the hypothesis chosen by A, denoted by $A(z)$, 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, given S as the target scenario, the probability that A fails to find an ϵ-good hypothesis for S is at least $\frac{1}{|\mathcal{Q}|} > 0$. ∎

Theorem 5.2 *A function class $\mathcal{F}$ over a finite domain is learnable via the projection class W and using the hypothesis class $\mathcal{H}$, if and only if there is no set of $\mathcal{F}$-scenarios which is W-hard for $\mathcal{H}$.*

Proof: Assume there is a set $\mathcal{Q}$ of $\mathcal{F}$-scenarios which is W-hard for $\mathcal{H}$. Then there is an $\epsilon > 0$, such that for every $h \in \mathcal{H}$ there is a scenario $S_h \in \mathcal{Q}$ for which h is ϵ-bad. Since the instance space is finite, the hypothesis class $\mathcal{H} \subseteq \{0,1\}^X$ is also finite, hence $\mathcal{Q}' = \{S_h : h \in \mathcal{H}\}$ is a finite class of $\mathcal{F}$-scenarios which is W-hard for $\mathcal{H}$. By Theorem 5.1, $\mathcal{F}$ is not learnable by $\mathcal{H}$ via W.

For the other direction, assume there is no set of $\mathcal{F}$-scenarios which is hard for $\mathcal{H}$, and assume first that the learner knows the exact 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 which 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 RFA estimates via W in order to find a set with small discrepancy. Hence, we need to relate the accuracy of the 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:

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

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 W-resolution of a set $\mathcal{A}$ to be:

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

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

We would like to have a lower bound on the necessary W-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 5.3 () *If there is no set $\mathcal{Q}$ of scenarios which is W-hard for $\mathcal{H}$, then for every $\epsilon > 0$ there is a $\gamma > 0$, 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 W-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-sets 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 infinite sequence in it has a converging subsequence.

By applying the same argument iteratively for every $h \in \mathcal{H}$ we obtain a scenario-set 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 W-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 5.3, for every set $\mathcal{A}$ of scenarios, there is a function $\Gamma : \mathbf{R}^+ \to \mathbf{R}^+$, such that $resolution(\mathcal{A}) > \Gamma(discrepancy(\mathcal{A})) > 0$. Given $\epsilon > 0$, let $\gamma = \min\{\Gamma(\frac{\epsilon}{2}), 2\epsilon\}$, where $\mathcal{O}$ is the (finite) observation space, and define:

$$\mathcal{D}' \triangleq \{D \in \mathcal{D} \; : \; \forall x \; D(x) \in \{i/M : i = 0, 1, \ldots, M\}\} \quad ; \quad M = \left\lceil \frac{4|X|}{\gamma} \right\rceil$$

$$\mathcal{T} \triangleq \{\langle f, D \rangle \; : \; f \in \mathcal{F} \; , \; D \in \mathcal{D}'\}$$

where $\mathcal{D}$ is the set of all distributions over X. Notice that $\mathcal{D}'$ is a $\frac{\gamma}{8}$-cover of $\mathcal{D}$, with respect to the statistical-distance metric Δ_s (*i.e.*, for every $D \in \mathcal{D}$ there is $D' \in \mathcal{D}$ such that $\Delta_s(D, D') \le \frac{\gamma}{8}$). Hence, for every two functions f and h,

$$\Pr_{x\in D}[f \ne h] \;\le\; \Pr_{x\in D'}[f \ne h] + 2\Delta_s(D, D') \;\le\; \Pr_{x\in D'}[f \ne h] + \tfrac{\gamma}{4}$$

Taking sufficiently large samples of the target scenario $S = \langle f, D\rangle$, an RFA learner can compute, for every $w \in W$ and every $z \in \mathcal{O}$, an estimate $\hat{p}_S(w, z)$ satisfying (with high confidence) $|\hat{p}_S - p_S| < \frac{\gamma}{4}$. Let

$$\mathcal{B} = \{S' \in \mathcal{T} : \max_{w,z} \; |\hat{p}_S(w,z) - p_{S'}(w,z)| \le \frac{\gamma}{2}\}$$

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 $|D(x) - D'(x)| \le \frac{1}{M}$ for every x, it follows that $|p_S - p_{S'}| \le |X|\frac{\gamma}{4|X|} = \frac{\gamma}{4}$, for every $w \in W$ and every $z \in \mathcal{O}$. Thus, with high confidence, $\|\hat{p}_S - p_{S'}\|_\infty \le \|\hat{p}_S - p_S\|_\infty + \|p_S - p_{S'}\|_\infty \le \frac{\gamma}{4} + \frac{\gamma}{4} = \frac{\gamma}{2}$, implying that $S' \in \mathcal{B}$.

Since *resolution*$(\mathcal{B}) < \gamma$ implies *discrepancy*$(\mathcal{B}) < \frac{\epsilon}{2}$ (recall that $\gamma \le \Gamma(\frac{\epsilon}{2})$), h must be $\frac{\epsilon}{2}$-good for every scenario in $\mathcal{B}$, including the scenario S'. Since $\Delta_s(D, D') \le \frac{1}{2}|X|\frac{\gamma}{4|X|} = \frac{\gamma}{8}$ and since $\gamma \le 2\epsilon$ we conclude that

$$\Pr_{x\in D}[f \ne h] \;\le\; \Pr_{x\in D'}[f \ne h] + \tfrac{\gamma}{4} \;<\; \tfrac{\epsilon}{2} + \tfrac{\epsilon}{2} = \epsilon$$

■

Corollary 5.4 *Every function class over a finite domain is* 1*-wRFA learnable.*

Proof: Note that if $W = \{0,1\}^{X\times\{0,1\}}$, then no set of scenarios is W-hard for $\mathcal{F}$. Assume there is such a W-hard set $\mathcal{Q}$ of $\mathcal{F}$-scenarios. Let $z \in X$, and consider the projections $u_z(x,y) = (x = z)$ and $v_z(x,y) = (x = z \;\wedge\; y = 1)$. For every two scenarios $S = \langle f, D\rangle$ and $T = \langle g, E\rangle$ in $\mathcal{Q}$, it follows from $p_S(u_z, 1) = p_T(u_z, 1)$ that $D(z) = E(z)$, and therefore $p_S(v_z, 1) = p_T(v_z, 1)$ implies $f(z) = g(z)$. Thus, all the scenarios in $\mathcal{Q}$ are identical and cannot form a W-hard set.

■

As a simple application of the above theorem, let us prove that, for a finite instance space, it is sufficient to extract one bit of information from every single example (in Section 8.2 we show that this remains true for finite function classes, even when the domain is infinite).

Theorem 5.5 *Every function class over a finite domain is* 1*-wRFA learnable.*

Proof: Let X be a finite domain. By Theorem 5.2, it is sufficient to show that for a function class $\mathcal{F} \subset \{0,1\}^X$, and for a projection class $W = \{0,1\}^{X\times\{0,1\}}$, no set of $\mathcal{F}$-scenarios is W-hard. Assume there is such a W-hard set $\mathcal{Q}$ of $\mathcal{F}$-scenarios. Let $z \in X$, and consider the projections $u_z(x,y) = (x = z)$ and $v_z(x,y) = (x = z \;\wedge\; y = 1)$. For every two scenarios $S = \langle f, D\rangle$ and $T = \langle g, E\rangle$ in $\mathcal{Q}$, it follows from $p_S(u_z, 1) = p_T(u_z, 1)$ that $D(z) = E(z)$, and therefore $p_S(v_z, 1) = p_T(v_z, 1)$ implies $f(z) = g(z)$. Thus, $\mathcal{Q}$ cannot contain two different scenarios, and hence cannot be W-hard. ■

We also note here that an alternative to the characterization of k-RFA learnability is shown in [12]. Specifically, they 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 5.2 can also be viewed in terms of k-Fourier equivalence. The connection of k-RFA learnability with Fourier analysis, which has proved quite useful in learning theory, seems potentially very useful in the study of learning with limited visibility.

5.1 Hard k-RFA learnability of Boolean Functions

In this section we develop a general scheme for proving hardness results for k-RFA learnability of boolean functions. That is, $X = \{0, 1\}^n$, and the projection class is $W_k = \{w_I : I = \{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}\}$, where $w_I(x, y) = (x_{i_1}, \ldots, x_{i_k}, y)$. We can specialize the hardness condition of Theorem 5.1 to the k-RFA setting, as follows. We say that a pair of scenarios is k-RFA equivalent for $\mathcal{H}$ if they are W_k-equivalent. The pair is also k-RFA hard if there is an $\epsilon > 0$, such that every $h \in \mathcal{H}$ is ϵ-bad for either S_1 or S_2.

For instance, consider the pair of scenarios $S_1 = \langle x_1, \frac{1}{2}\rangle$, $S_2 = \langle \overline{x_1}, \frac{1}{2}\rangle$ over the domain $X = \{0, 1\}$. (I.e., $n = 1$, both distributions are uniform, $f_1 = x_1$ and f_2 is its negation). Then, the pair is 0-RFA equivalent, and for $\epsilon = \frac{1}{2}$, every boolean function is $\frac{1}{2}$-bad for either S_1 or S_2. Hence, the pair is 0-hard, implying that every $\mathcal{F} \supseteq \{x_1, \overline{x}_1\}$ is not 0-RFA learnable.

It is not hard to generalize the above example to any $n \geq 1$, $0 \leq k < n$, where $S_1 = \langle x_1 \oplus \cdots \oplus x_{k+1}, U_n\rangle$, $S_2 = \langle \overline{x_1 \oplus \cdots \oplus x_{k+1}}_1, U_n\rangle$, and U_n is the uniform distribution over $\{0, 1\}^n$, concluding that every every $\mathcal{F}$ which contains both the odd and even parity functions over $k + 1$ boolean variables is not k-RFA learnable. We now develop a general scheme for showing such hardness results.

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 a construction which inductively expands a k-RFA hard pair of scenarios over the instance space $\{0, 1\}^n$ into a $(k + 1)$-RFA hard pair of scenarios over $\{0, 1\}^{n+1}$. 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 $n \geq n_0$.

Definition 5.6 (expansion) *An expansion is a pair (ξ, ρ), where ξ is a function from pairs of scenarios over the domain $\{0, 1\}^n$ to pairs of scenarios over the domain $\{0, 1\}^{n+1}$, $\rho : \mathbf{N} \to \mathbf{N}$, and if (S, T) is a pair of k-RFA hard scenarios, then $\xi(S, T)$ is a pair of $\rho(k)$-RFA hard scenarios.*

It follows directly from the definitions that the following simple construction is an expansion (which we call I-expansion).

Lemma 5.7 (I-expansion) *Let $\rho(k) = k$, and let $\xi(\langle f, D\rangle, \langle g, E\rangle) = (\langle f, \frac{1}{2}D\rangle, \langle g, \frac{1}{2}E\rangle)$. Then (ξ, ρ) is an expansion. ($\frac{1}{2}D$ denotes a distribution over $\{0, 1\}^{n+1}$ which assigns each $(x_1, \ldots, x_n, x_{n+1}) \in \{0, 1\}^{n+1}$ a probability $\frac{1}{2}D(x_1, \ldots, x_n)$).*

Definition 5.8 (expandable) *A family $\{\mathcal{F}_n\}_{n \geq n_0}$ is expandable by (ξ, ρ), if (ξ, ρ) is an expansion, and $\{\mathcal{F}_n\}_{n \geq n_0}$ is closed under ξ.*

For instance, let $\mathcal{F}_n = \{x_1, \overline{x}_1\}$. Then the family $\{\mathcal{F}_n\}_{n \geq n_0}$ is trivially expandable by the I-expansion of Lemma 5.7.

Theorem 5.9 *If there is a pair of $\mathcal{F}_{n_0}$-scenarios which is k_0-RFA hard, and $\{\mathcal{F}_n\}_{n \geq n_0}$ is expandable by (ξ, ρ), then $\mathcal{F}_{n_0+i}$ is not $\rho^i(k_0)$-RFA learnable, for every $i \geq 0$.*

Proof: Assume $(\langle f_{n_0}, D_{n_0}\rangle, \langle g_{n_0}, E_{n_0}\rangle)$ is k_0-RFA hard pair of scenarios, and let (ξ, ρ) be an expansion. Let $\{(f_n, g_n)\}_{n \geq n_0} = \xi^i(f_{n_0}, g_{n_0})$. Then, by induction on $n \geq n_0$, $S_n = \langle f_n, D_n\rangle$ and $T_n = \langle g_n, E_n\rangle$ are $\mathcal{F}_n$-scenarios, and they are $\rho^{n-n_0}(k)$-RFA hard. ∎

Since the pair $\langle x_1, \frac{1}{2}\rangle, \langle \overline{x}_1, \frac{1}{2}\rangle$ is 0-RFA hard, we may use the I-expansion of Lemma 5.7 to show that, the class $\{x_1, \overline{x}_1\}$ (over every domain $\{0, 1\}^n$) is not 0-RFA learnable.

Lemma 5.10 (S-expansion) *The following pair* (ξ, ρ) *is an expansion:* $\rho(k) = k + 1$*, and* $\xi(S_n, T_n) = (S_{n+1}, T_{n+1})$*, where* $S_i = \langle f_i, D_i \rangle$*,* $T_i = \langle g_i, E_i \rangle$*, for* $i \in \{n, n+1\}$*, and*

$$\begin{aligned} f_{n+1}(x_1, \ldots, x_n, x_{n+1}) &\triangleq x_{n+1}\, f_n \;+\; \overline{x}_{n+1}\, g_n \\ g_{n+1}(x_1, \ldots, x_n, x_{n+1}) &\triangleq x_{n+1}\, g_n \;+\; \overline{x}_{n+1}\, f_n \\ D_{n+1}(x_1, \ldots, x_n, 0) &\triangleq \frac{1}{2}\, E_n(x_1, \ldots, x_n) \\ D_{n+1}(x_1, \ldots, x_n, 1) &\triangleq \frac{1}{2}\, D_n(x_1, \ldots, x_n) \\ E_{n+1}(x_1, \ldots, x_n, 0) &\triangleq \frac{1}{2}\, D_n(x_1, \ldots, x_n) \\ E_{n+1}(x_1, \ldots, x_n, 1) &\triangleq \frac{1}{2}\, E_n(x_1, \ldots, x_n) \end{aligned}$$

Proof: Assuming that the pair (S_n, T_n) is a k-RFA hard pair, we prove the $(k+1)$-RFA hardness of the pair (S_{n+1}, T_{n+1}). First we prove that if S_n and T_n are k-RFA equivalent, then S_{n+1} and T_{n+1} are $(k+1)$-RFA equivalent. Let $I = \{i_1, \ldots, i_{k+1}\} \subset \{1, \ldots, n+1\}$, and $z = (x, b) \in \{0,1\}^{k+2}$. To verify that $p_{S_{n+1}}(w_I, z) = p_{T_{n+1}}(w_I, z)$, consider the following two cases:

- If $n + 1 \in I$, then $p_{S_{n+1}}(w_I, z) = \frac{1}{2} p_{S_n}(w_I, z) = \frac{1}{2} p_{T_n}(w_I, z) = p_{T_{n+1}}(w_I, z)$.
- If $n + 1 \notin I$, then $S_{n+1}(x_1, \ldots, x_n, x_{n+1}) = T_{n+1}(x_1, \ldots, x_n, \overline{x}_{n+1})$, implying $p_{S_{n+1}}(w_I, z) = p_{T_{n+1}}(w_I, z)$.

Thus, the pair (S_{n+1}, T_{n+1}) is also $(k+1)$-RFA equivalent.

Finally, if $\epsilon > 0$ is such that every $h : \{0, 1\}^n \to \{0, 1\}$ is ϵ-bad for either S_n or T_n, then every $h : \{0,1\}^{n+1} \to \{0, 1\}$ is ϵ-bad for either S_{n+1} or T_{n+1}, concluding the $(k+1)$-RFA hardness of the pair (S_{n+1}, T_{n+1}).

■

Corollary 5.11 *The class* $\mathrm{PARITY}_{\mathrm{n}}^{\mathrm{k+1}}$ *is not k-RFA learnable, for every* $0 \le k < n$*.*

Proof: The family $\mathrm{PARITY}_{\mathrm{k+1}}^{\mathrm{k+1}}$ is expandable using the S-expansion of Lemma 5.10. Since the pair $\langle x_1, \frac{1}{2} \rangle, \langle \overline{x}_1, \frac{1}{2} \rangle$ is 0-hard, we conclude that for every $k \ge 0$ the class $\mathrm{PARITY}_{\mathrm{k+1}}^{\mathrm{k+1}}$ is not k-RFA learnable. Now we can use the I-expansion of Lemma 5.7 to conclude that $\mathrm{PARITY}_{\mathrm{n}}^{\mathrm{k+1}}$ is not k-RFA learnable, for every $n > k$. ■

Note that a $(k+1)$-RFA learner needs a single example to learn the class $\mathrm{PARITY}_{\mathrm{n}}^{\mathrm{k+1}}$. Thus, the k-RFA restriction induces a strict hierarchy on the learnable classes. Also notice that the above result implies that the class of DNF representations is not k-RFA learnable under the uniform distribution, for any $k < n$.

Another class which is expandable by the S-expansion is the class k-DL. This fact has been recently used in [12] to obtain the following hardness results for k-DL.

Theorem 5.12 (Birkendorf, Dichterman, Jackson, Klasner, Simon 96)
k-DL is not k-RFA learnable, for every $k \le n - 2$*.*

The following Lemma presents another useful expansion, which we call L-expansion.

Lemma 5.13 (L-expansion) *The following pair* (ξ, ρ) *is an expansion:* $\rho(k) = k+1$, *and* $\xi(S_n, T_n) = (S_{n+1}, T_{n+1})$, *where* $S_i = \langle f_i, D_i \rangle$, $T_i = \langle g_i, E_i \rangle$, *for* $i \in \{n, n+1\}$, *and*

$$\begin{aligned}
f_{n+1}(x_1, \ldots, x_n, 1) &\triangleq f_n(x_1, \ldots, x_n) \\
f_{n+1}(x_1, \ldots, x_n, 0) &\triangleq 0 \\
g_{n+1}(x_1, \ldots, x_n, 1) &\triangleq g_n(x_1, \ldots, x_n) \\
g_{n+1}(x_1, \ldots, x_n, 0) &\triangleq 0 \\
D_{n+1}(x_1, \ldots, x_n, 0) &\triangleq \frac{1}{2}\, E_n(x_1, \ldots, x_n) \\
D_{n+1}(x_1, \ldots, x_n, 1) &\triangleq \frac{1}{2}\, D_n(x_1, \ldots, x_n) \\
E_{n+1}(x_1, \ldots, x_n, 0) &\triangleq \frac{1}{2}\, D_n(x_1, \ldots, x_n) \\
E_{n+1}(x_1, \ldots, x_n, 1) &\triangleq \frac{1}{2}\, E_n(x_1, \ldots, x_n)
\end{aligned}$$

We omit the proof which is very similar to the proof of Lemma 5.10. Note that the class of decision lists (1-DL) is expandable using the L-expansion. This fact can be used to prove the following

Theorem 5.14 (Birkendorf, Dichterman, Jackson, Klasner, Simon 96)
1-DL is not properly k-RFA learnable, for every $k \leq n-2$.

Actually, it can be shown that 1-DL is not even $\lceil \frac{n}{2} \rceil$-RFA learnable. Both results are obtained by finding a hard pair of scenario for $k = 1$ and $n = 3$, and using the above S-expansion (for the first result) and the L-expansion (for the second) to generalize for any $n \geq 3$.

Also notice that $(n-1)$-DL is $(n-1)$-RFA learnable (distribution-free), by the following simple strategy: search for a pair of homogeneous adjacent assignments (i.e., assignments which differ in only one bit, and which have the same classification), using a large enough sample; this pair is (or can be) the first term of the $(n-1)$-decision list. Knowing the value of this pair, it is possible to determine the nature of any pair which overlaps with this pair, and so forth, till we find the values of all the assignments. A detailed decription of the learning algorithm and its analysis is shown in [12].

6 k-RFA Learning

6.1 Composing Efficient k-RFA Learning Algorithms

We start with developing a technical tool that takes as input PAC learning algorithms of given classes and outputs an efficient k-RFA algorithm for a class composed, in a specific manner, from the given input classes. We conclude that if a class is decomposable into PAC-learnable classes, then it is efficiently k-RFA learnable.

Consider, for instance, the class of k-DNF formulas over the domain $X_n = \{0,1\}^n$. A PAC algorithm which efficiently learns this class is shown in [41]. Is it possible to adapt this algorithm to the k-RFA setting, in which only k attributes of each single example are revealed to the learning algorithm? In other words, is this class (efficiently) k-RFA learnable? In this section we develop a general tool which will be useful in answering such questions.

The basic idea is to notice that any k-DNF formula Θ over n variables is a disjunction of DNF formulas $\theta_1, \ldots, \theta_r$, where each θ_i depends on at most k variables. Thus, any negative example for Θ is also a negative example for each θ_i. Since DNF formulas are learnable from negative examples we can learn each DNF formula θ_i by observing only the proper set of k relevant attributes, obtaining a k-RFA learning algorithm for the class of k-DNF. Furthermore, since the number of DNF formulas θ_i is polynomial in n ($r = O(n^k)$), and since the time needed to learn each θ_i does not depend on n (k is fixed), the resulting k-RFA learning algorithm is efficient. We now formulate this intuitive decomposition scheme, applying it to deduce the RFA learnability of several function classes.

Given functions $f_1, \ldots, f_r$ over X_n, and a Boolean function $\psi : \{0,1\}^r \to \{0,1\}$, we define the ψ-composition of $f_1, \ldots, f_r$, denoted $\mathrm{COMPOSE}_\psi(\mathrm{f_1, \ldots, f_r})$, to be

$$\mathrm{COMPOSE}_\psi(\mathrm{f_1, \ldots, f_r})(\mathrm{x}) \triangleq \psi(\mathrm{f_1(x), \ldots, f_r(x)})$$

Applying the composition to a class of functions, we get:

$$\mathrm{COMPOSE}_\psi(\mathcal{F}_1, \ldots, \mathcal{F}_r) \triangleq \{\mathrm{COMPOSE}_\psi(\mathrm{f_1, \ldots, f_r}) \mid \mathrm{f_i} \in \mathcal{F}_\mathrm{i}\, , 1 \leq \mathrm{i} \leq \mathrm{r}\}$$

When $\mathcal{F} = \mathcal{F}_1 = \cdots = \mathcal{F}_r$ we denote their ψ-composition by $\mathrm{COMPOSE}_\psi(\mathcal{F})$.

For instance, $\mathrm{COMPOSE_{disj}(k\text{-}CONJ_n) = k\text{-}DNF_n}$, where 'disj' is the disjunction function over $(2n+1)^k$ literals, k-CONJ$_n$ is the class of k-terms over X_n, and k-DNF$_n$ is the class of k-DNF formulas over X_n.

For a Boolean function $\psi : \{0,1\}^r \to \{0,1\}$, and $b \in \{0,1\}$, we say that ψ is b-invertible if $|\psi^{-1}(b)| = 1$ (i.e. there is a unique assignment $v \in \{0,1\}^r$ satisfying $\psi(v) = b$). Also, if $f(x) = b$, we say that x is an example of type b (for f). We say that $\mathcal{F}$ is b-learnable if $\mathcal{F}$ is learnable from an oracle which supplies only examples of type b (i.e., positive examples for $b = 1$ and negative examples for $b = 0$). $\mathcal{F}$ is k-RFA b-learnable when the oracle is a k-RFA oracle which supplies only observations obtained from examples of type b.

Theorem 6.1 *Let $\mathcal{F}_1, \ldots, \mathcal{F}_r$ be function classes over X_n, where each $\mathcal{F}_i$ is k-RFA b_i-learnable, and let $\psi : \{0,1\}^r \to \{0,1\}$ be b-invertible, where $\psi^{-1}(b) = (b_1, \ldots, b_r)$. Then $COMPOSE_\psi(\mathcal{F}_1, \ldots, \mathcal{F}_r)$ is k-RFA b-learnable, using a sample of size*

$$\max\left\{\frac{4r^2}{\epsilon}\log\frac{2r}{\delta}\, ,\, \frac{8r\Sigma_{i=1}^r \mathrm{VCdim}(\mathcal{F}_i)}{\epsilon}\log\frac{13r}{\epsilon}\right\}$$

Proof: Assume the target function f is decomposable as $f = \mathrm{COMPOSE}_\psi(\mathrm{f_1, \ldots, f_r})$. Since ψ is b-invertible, each example of type b for f is an example of type b_i for f_i ($1 \leq i \leq r$). Thus, the learning algorithm may focus its attention on the k relevant variables of f_i, obtaining examples of type b_i for f_i. Since $\mathcal{F}_i$ is b_i-learnable, $\mathcal{A}$ can learn f_i from a sufficiently large sample for f. Using Theorem 2.1 of Blumer et al. ([14]), we conclude that $\mathcal{A}$ can find with probability of at least $1 - \frac{\delta}{r}$ a hypothesis h_i which is $\frac{\epsilon}{r}$-good for f_i, using a sample of size $\max\left\{\frac{4r}{\epsilon}\log\frac{2r}{\delta}\, ,\, \frac{8r\mathrm{VCdim}(\mathcal{F}_i)}{\epsilon}\log\frac{13r}{\epsilon}\right\}$. Upon having an $\frac{\epsilon}{r}$-good hypothesis for each f_i, $\mathcal{A}$ may use the (known) function ψ to obtain, with probability of at least $1 - \delta$, an ϵ-good hypothesis for f. ∎

Thus, we have a construction scheme which enables us to deduce the k-RFA learnability of a class which is composed of k-RFA learnable classes.

In case each composing class depends on at most k variables, we can take one step further, and deduce the k-RFA learnability of $\mathrm{COMPOSE}_\psi(\mathcal{F}_1, \ldots, \mathrm{F_r})$ from the (PAC) learnability of the composing classes $\mathcal{F}_1, \ldots, \mathcal{F}_r$.

The domain of a function can be expanded from X_k to X_n as follows. Given a function $f : X_k \to \{0,1\}$ and a mapping $m : [k] \to [n]$ (recall that $[j]$ denotes the set $\{1, \ldots, j\}$), we define a new function $f_m : X_n \to \{0,1\}$, by $f_m(x) = f(x_{m(1)}, \ldots, x_{m(k)})$. (Notice that f_m has only k relevant variables; f_m applies f to the variables whose indices are determined by the mapping m). Thus, given a class $\mathcal{F}$ of functions over the domain X_k, we can define a function class over X_n, by $[\mathcal{F}]_{k\to n} \triangleq \{f_m : f \in \mathcal{F}\ ,\ m \in [n]^{[k]}\}$. For instance, $\mathrm{k\text{-}DNF = COMPOSE_{disj}([DNF_k]_{k\to n})}$. Obviously, if $\mathcal{F}$ is learnable then $[\mathcal{F}]_{k\to n}$ is k-RFA learnable, and the learnability is efficient for every fixed k.

Corollary 6.2 *Let $\mathcal{F}_1, \ldots, \mathcal{F}_r$ be function classes over X_k, where each $\mathcal{F}_i$ is k-RFA b_i-learnable, and let $\psi : \{0,1\}^r \to \{0,1\}$ be b-invertible, where $\psi^{-1}(b) = (b_1, \ldots, b_r)$. Then $COMPOSE_\psi([\mathcal{F}_1]_{k\to n}, \ldots, [\mathcal{F}_r]_{k\to n})$ is k-RFA b-learnable, using a sample of size*

$$\max\left\{\frac{4r^2}{\epsilon}\log\frac{2r}{\delta}, \frac{8r\Sigma_{i=1}^r \mathrm{VCdim}(\mathcal{F}_i)}{\epsilon}\log\frac{13r}{\epsilon}\right\}$$

.

For the case $\mathcal{F} = \mathcal{F}_1 = \cdots = \mathcal{F}_r$ we also get the following

Corollary 6.3 *Let $\mathcal{F}$ be a function class over X_k, which is b-learnable, and let $\psi : \{0,1\}^r \to \{0,1\}$ be b'-invertible, where $\psi^{-1}(b') = (b, \ldots, b)$. Then* $\mathrm{COMPOSE}_\psi([\mathcal{F}]_{\mathrm{k}\to\mathrm{n}})$ *is k-RFA b'-learnable, using a sample of size*

$$\max\left\{\frac{4r^2}{\epsilon}\log\frac{2r}{\delta}, \frac{8r^2\mathrm{VCdim}(\mathcal{F})}{\epsilon}\log\frac{13r}{\epsilon}\right\}$$

.

Now the efficient k-RFA learnability of k-DNF and k-CNF formulas is easily obtained by applying Corollary 6.3 to the classes of $\mathrm{DNF_k}$ and $\mathrm{CNF_k}$ formulas, respectively.

Proposition 6.4 *k-DNF and k-CNF are efficiently k-RFA learnable.*

Proof: The classes $\mathrm{DNF_k}$ and $\mathrm{CNF_k}$ are obviously PAC learnable [41], so we can apply Corollary 6.3 to the following compositions:

$$\begin{aligned} \text{k-DNF}_n &= \mathrm{COMPOSE_{disj}}([\mathrm{DNF_k}]_{\mathrm{k}\to\mathrm{n}}) \\ \text{k-CNF}_n &= \mathrm{COMPOSE_{conj}}([\mathrm{CNF_k}]_{\mathrm{k}\to\mathrm{n}}) \end{aligned}$$

Since $\mathrm{VCdim}(\mathrm{DNF_k}) = \mathrm{VCdim}(\mathrm{CNF_k}) = \mathrm{O}(2^\mathrm{k})$, we get by Corollary 6.3 that the sample complexity is $O(\frac{n^{2k}}{\epsilon}\log\frac{n^{2k}}{\epsilon\delta})$, and the learning algorithm is efficient. ■

As another simple application of the decomposability scheme, consider the following classes over the instance space $X_n = \mathbf{R}^n$:

$$\begin{aligned} \mathrm{REC}_n &\triangleq \{\mathrm{rec}_{a,b} \mid a, b \in \mathbf{R}^n\} \\ \mathrm{rec}_{a,b} &\triangleq \{x \in \mathbf{R}^n \mid \forall i\, x_i \in [a_i, b_i]\} \\ \mathrm{CROSS}_n &\triangleq \{\mathrm{cross}_{a,b} \mid a, b \in \mathbf{R}^n\} \\ \mathrm{cross}_{a,b} &\triangleq \{x \in \mathbf{R}^n \mid \exists i\, x_i \in [a_i, b_i]\} \end{aligned}$$

That is, $\mathrm{REC_n}$ is the class of axis-aligned rectangles in $\mathbf{R}^n$, and $\mathrm{CROSS_n}$ is the class of axis-aligned 'crosses' in $\mathbf{R}^n$.

Proposition 6.5 *REC_n and $CROSS_n$ are efficiently* 1*-RFA learnable.*

Proof: Notice that $\mathrm{REC_1} = \mathrm{CROSS_1}$ is the class of intervals in $\mathbf{R}$, which is known to be efficiently learnable from either positive or negative examples ([27]). Therefore we can apply Corollary 6.3 to the compositions

$$\begin{aligned} \mathrm{REC_n} &= \mathrm{COMPOSE_{conj}}(\mathrm{REC_1}) \\ \mathrm{CROSS_n} &= \mathrm{COMPOSE_{disj}}(\mathrm{CROSS_1}) \end{aligned}$$

Since $\mathrm{VCdim}(\mathrm{REC_1}) = \mathrm{VCdim}(\mathrm{CROSS_1}) = 2$, we get by Corollary 6.3 that the sample complexity is $O(\frac{n^2}{\epsilon}\log\frac{n^2}{\epsilon\delta})$, and the learning algorithm is efficient. ■

6.2 k-RFA Learnability of Boolean Functions

Restricting the visibility of the learner makes the question of (information-theoretic) learnability of Boolean function (over the domain $\{0, 1\}^n$) non-trivial. Since the domain is finite, so is the VC dimension, and hence every class of Boolean functions is PAC learnable. However, the k-RFA restriction imposes a strict hierarchy on the learnable classes.

By Theorem 5.2, To prove the hardness of k-RFA learnability for a class $\mathcal{F}$ of Boolean functions, it is sufficient to find a hard W_k-RFA pair of $\mathcal{F}$-scenarios, (where $W_k = \{w_I : I = \{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}\}$, and $w_I(x, y) = (x_{i_1}, \ldots, x_{i_k}, y)$ for $I = \{i_1, \ldots, i_k\}$). We say that a W_k-equivalent pair of scenarios is k-RFA equivalent, and it is W_k-RFA hard if it is W_k-hard.

Let U_n denote the uniform distribution over $\{0, 1\}^n$, and consider the pair of scenarios $S_1 = \langle x_1, U_1\rangle$ and $S_2 = \langle \overline{x_1}, U_1\rangle$ over the domain $\{0, 1\}$. (I.e., $n = 1$, both distributions are uniform, $f_1 = x_1$ and f_2 is its negation). Then, the pair is 0-RFA equivalent, and for $\epsilon = \frac{1}{2}$, every Boolean function is $\frac{1}{2}$-bad for either S_1 or S_2. Hence, the pair is 0-RFA hard, implying that every $\mathcal{F} \supseteq \{x_1, \overline{x}_1\}$ is not 0-RFA learnable.

It is not hard to generalize the above example to any $n \geq 1$, $0 \leq k < n$, where $S_1 = \langle x_1 \oplus \cdots \oplus x_{k+1}, U_n\rangle$, $S_2 = \langle \overline{x_1 \oplus \cdots \oplus x_{k+1}}, U_n\rangle$, obtaining the following hardness result.

Theorem 6.6 *Every class $\mathcal{F}$ containing $PARITY_n^{k+1}$ is not k-RFA learnable.*

Note that a $(k+1)$-RFA learner needs a single example to learn the class $\mathrm{PARITY}_{\mathrm{n}}^{\mathrm{k+1}}$, and hence a k-RFA learner is strictly weaker than a $(k+1)$-RFA learner. Also notice that the above result immediately implies that, for every $k < n$, the class of DNF formulas is not k-RFA learnable.

As the hardness of the RFA learnability of the parity functions seems to be induced by the symmetric behavior of these functions with respect to the uniform distribution, one may wonder what happens when this symmetry is broken. The following two theorems give a partial answer to this question.

We consider the class of distributions, in which the variables are independently and identically distributed (i.i.d.), having a probability p of having the 1, but restrict p to be different from $1/2$. We call such distributions *biased i.i.d. distributions*, and identify it in the sequel by its probability p.

The following theorem show that for under i.i.d. distributions, the class of *all* Boolean functions is $(n-2)$-RFA learnable. We do not know whether in this case the learner's focus-of-attention can be further restricted (it seems hard to generalize the subtle analysis made for $(n-2) - RFA$ learnability to $(n-j)$-RFA learnability, for $j > 2$). However, we show in the sequel a lower bound of $k = n/8$ on the visibility size of the a k-RFA learner (which holds for the same class of distributions).

Theorem 6.7 *The class of all Boolean functions is $(n-2)$-RFA learnable under biased i.i.d. distributions.*

Proof: By Theorem 5.2, it is sufficient to show that there is no set of $(n-2)$-RFA hard scenarios over the domain $\{0, 1\}^n$, in which all the distributions are biased i.i.d.

Consider two scenarios S and S', which consist of two Boolean functions f and g, and two i.i.d. distributions q and r, respectively. First we claim that if the scenarios are k-RFA equivalent, then $q = r$. To see why, consider the $(n-2)$-RFA projection defined by the index set $I = \{1, \ldots, n-2\}$, and recall that for the scenarios to be $(n-2)$-RFA equivalent, the distributions p_{S,w_I} and p_{S',w_I} projected onto the observation space, have to be identical. This implies that $q = \sum_{\{z\in\{0,1\}^{n-1}:z_1=1\}} p_{S,w_I}(z) \sum_{\{z\in\{0,1\}^{n-1}:z_1=1\}} p_{S',w_I}(z) = r$.

Thus, an $(n-2)$-RFA hard set of scenarios have to share the same i.i.d. distribution $p \neq 1/2$. We also claim that in such a set, p cannot be neither 0 nor 1. In both cases, there is only one instance with non-zero probability, and hence all the functions in the set (being $(n-2)$-RFA equivalent) have to agree on that instance ($(0, 0, \ldots, 0)$ when $p = 0$, and $(1, 1, \ldots, 1)$ when $p = 1$). But then, any hypothesis which

Table 1: The functions h_j and the corresponding q_j

j	y_1y_2:	$f_j(y_1,y_2)$ 00	01	10	11	$q_j(p)$
0		0	0	0	0	0
1		0	0	0	1	p^2
2		0	0	1	0	$p-p^2$
3		0	0	1	1	p
4		0	1	0	0	$p-p^2$
5		0	1	0	1	p
6		0	1	1	0	$2p-2p^2$
7		0	1	1	1	$2p-p^2$
8		1	0	0	0	$1-2p+p^2$
9		1	0	0	1	$1-2p+2p^2$
10		1	0	1	0	$1-p$
11		1	0	1	1	$1-p+p^2$
12		1	1	0	0	$1-p$
13		1	1	0	1	$1-p+p^2$
14		1	1	1	0	$1-p^2$
15		1	1	1	1	1

agree with them on that instance is a good hypothesis for all the scenarios in the set, and hence the set cannot be a hard one.

It remains to show that, for $p \notin \{0, 1/2, 1\}$, and for any two different functions, there is an $(n-2)$-RFA projection which projects different distributions (and the functions are distinguishable by an $(n-2)$-RFA learner).

We fix an assignment $(a_4, \ldots, a_n)$ to the variables $x_4, \ldots, x_n$, and consider all the $(n-2)$-RFA probabilities obtained by letting $x_i = b$, for $i \in \{1, 2, 3\}$ and $b \in \{0, 1\}$. Let $f_{i,b}$ be the 2-variables Boolean functions obtained from the target function f, when letting $(x_i, x_4, \ldots, x_n) = (b, a_4, \ldots, a_n)$. The corresponding $(n-2)$-RFA probability is $p_{S,w_I}(z_{i,b})$, where $I = \{i, 4, \ldots, n\}$, and $z_{i,b} = (b, a_4, \ldots, a_n, 1)$. For a given i and b, $f_{i,b}$ can be one of the 2^4 possible functions $h_0, \ldots, h_{15}$ shown in Table 1, where the corresponding $(n-2)$-RFA probabilities are $q_0, \ldots, q_{15}$.

Had the probabilities q_j been all different, it would have been possible to associate every g_j with a unique q_j, hence deducing the target function from the corresponding $(n-2)$-RFA probabilities. However, some probabilities are identical for every p (*e.g.*, $q_2 \equiv q_4$), and others are equal for a specific value of p (*e.g.*, $q_1(2/3) = q_6(2/3)$). Nevertheless, we show that in every case, there is one $(n-2)$-RFA probability on which different functions have to differ. In particular, we show how to identify the function $f_{3,0}$ (and the case $f_{3,1}$ is symmetric). Since the argument holds for every assignment to the variables $x_4, \ldots, x_n$, it implies that the $(n-2)$-RFA probabilities uniquely determine the target function f.

We use the notation $f_{i,b} = (r, s)$ to denote an uncertainty between $f_{i,b} = g_r$ and $f_{i,b} = g_s$ (*e.g.*, $f_{3,0} = (2, 4)$ is such an uncertainty). In case the uncertainty holds only for a specific value of p, say $p = c$, we use the notation $f_{i,b} = (r, s)|_c$ (*e.g.*, $f_{3,0} = (1, 6)|_{2/3}$ is such an uncertainty). Similar notations are used when the uncertainty is among three possible functions (*e.g.*, $f_{3,0} = (3, 5, 8)|_{(3-\sqrt{5})/2}$).

To form a list of all the uncertainties among the functions $h_1, \ldots, h_{15}$, we first introduce three uncertainty-preserving operations. Given a function $h(y_1, y_2)$, the first operation is defined by $\bar{h}(y_1, y_2) = 1 - h(y_1, y_2)$, and the second one is defined by $\tilde{h}(y_1, y_2) = g(1 - y_1, 1 - y_2)$.

To see why the first operation is uncertainty-preserving, assume that $(i, j)|_c$ is an uncertainty, and let $h_k = \bar{h_i}$ and $h_l = \bar{h_j}$. Then, $q_k = 1 - q_i$ and $q_l = 1 - q_j$, and hence $(k, l)|_c$ is also an uncertainty.

Table 2: Uncertainties of $f_{i,b}$

h	$h \Rightarrow \bar{h}$	$h \Rightarrow \tilde{h}$	$h \Rightarrow \tilde{\bar{h}}$
$(2,4)$	(11,13)	(2,4)	(11,13)
$(3,5)$	(10,12)	(3,5)	(10,12)
$(7,8)\|_{1-1/\sqrt{2}}$	$(7,8)\|_{1-1/\sqrt{2}}$	$(1,14)\|_{1/\sqrt{2}}$	$(1,14)\|_{1/\sqrt{2}}$
$(6,8)\|_{1/3}$	$(7,9)\|_{1/3}$	$(1,6)\|_{2/3}$	$(9,14)\|_{2/3}$
$(3,5,8)\|_{(3-\sqrt{5})/2}$	$(7,10,12)\|_{(3-\sqrt{5})/2}$	$(1,10,12)\|_{(\sqrt{5}-1)/2}$	$(3,5,14)\|_{(\sqrt{5}-1)/2}$

As for the second operation, if $h_s = \tilde{h}_i$ and $h_t = \tilde{h}_j$, then $q_s(1-p) = q_i(p)$ and $q_t(1-c) = q_j(c)$, and hence $(s,t)|_{1-c}$ is also an uncertainty. Also note that by combining both operations we get yet another uncertainty-preserving operation $\tilde{\bar{h}}(y_1, y_2) = 1 - h(1-y_1, 1-y_2)$. All the three operations are involutions (*i.e.*, applying them twice we get the identity operation), and hence their inverse are also uncertainty-preserving operations.

All the possible uncertainties (obtained by comparing the functions $h_0, \ldots, h_{15}$ to each other), are shown in Table 2. We call the uncertainties shown in column i of the table uncertainties of type i. The uncertainties of type of type 2, 3, and 4, are obtained from uncertainties of type 1 by applying the operations $h \Rightarrow \bar{h}$, $h \Rightarrow \tilde{h}$, and $h \Rightarrow \tilde{\bar{h}}$, respectively.

We start with the uncertainties of type 1, and in each case, analyze only the remaining uncertainties, until we end up with values which are distinguishable (no uncertainties). To denote several possible uncertainties of a value $v_{i,b}$, say (q,r) or (s,t) (depending on previous uncertainties), we use the set notation $v_{i,b} \in \{(q,r),(s,t)\}$. (and $v_{i,b} \in \{(q,r),(s,t)\}|_c$ when the uncertainty is valid only for $p = c$).

$v_{3,0}(x_1,x_2) = (2,4)$:
$\Rightarrow v_{1,0}(x_3,x_2) \in \{0,1,2,3\} \times \{4,5,6,7\}$
$\Rightarrow v_{1,0}(x_3,x_2) \in \{(1,6)|_{p=2/3}, (2,4), (3,5)\}$
$\Rightarrow v_{2,1}(x_3,x_1) \in \{2,3\} \times \{8,9\}|_{p=2/3} \cup \{0,1\} \times \{8,9\} \cup \{2,3\} \times \{10,11\}$

$v_{3,0}(x_1,x_2) = (3,5)$:
$\Rightarrow v_{1,0}(x_3,x_2) \in \{0,1,2,3\} \times \{4,5,6,7\}$
$\Rightarrow v_{1,0}(x_3,x_2) \in \{(1,6)|_{p=2/3}, (2,4), (3,5)\}$
$\Rightarrow v_{2,1}(x_3,x_1) \in \{6,7\} \times \{12,13\}|_{p=2/3} \cup \{4,5\} \times \{12,13\} \cup \{6,7\} \times \{14,15\}$

$v_{3,0}(x_1,x_2) = (7,8)|_{1-1/\sqrt{2}}$:
$\Rightarrow v_{2,1}(x_3,x_1) \in \{12,13,14,15\} \times \{0,1,2,3\}|_{1-1/\sqrt{2}}$

$v_{3,0}(x_1,x_2) = (6,8)|_{1/3}$:
$\Rightarrow v_{2,1}(x_3,x_1) \in \{8,9,10,11\} \times \{0,1,2,3\}|_{1/3}$

$v_{3,0}(x_1,x_2) = (3,5,8)|_{(3-\sqrt{5})/2}$:
$\Rightarrow v_{1,0}(x_3,x_2) \in \{0,1,2,3\} \times \{4,5,6,7\} \times \{8,9,10,11\}|_{(3-\sqrt{5})/2}$
$\Rightarrow v_{1,0}(x_3,x_2) \in \{(2,4)|_{(3-\sqrt{5})/2}, (3,5,8)|_{(3-\sqrt{5})/2}\}$
$\Rightarrow v_{2,1}(x_3,x_1) \in \{6,7\} \times \{14,15\} \times \{0,1\}|_{(3-\sqrt{5})/2} \cup \{4,5\} \times \{12,13\}|_{(3-\sqrt{5})/2}$

Thus, all the uncertainties of type 1 are resolvable. As for the other uncertainties, recall that each one of them is obtained from an uncertainty of type 1 by an uncertainty-preserving transformation. Assume that there is such an unresolved uncertainty, and let f, g be two different functions which induce that uncertainty. By applying the proper uncertainty-preserving operation, we can get two other functions f', g', which induce an irresolvable uncertainty of type 1, a contradiction.

■

How small could be the visibility of the learner under the conditions of the above theorem (biased i.i.d. distributions). The following theorem shows that, to enable learnability under such distributions, $\Omega(n)$ bits have to be visible.

Theorem 6.8 *The class of all Boolean functions is not $\lfloor \frac{n}{8} \rfloor$-RFA learnable, under the class of non-uniform distributions, in which the variables are independently and identically distributed.*

Proof: Consider the biased i.i.d. distribution $p = 1/4$. We prove that if $k \leq \lfloor \frac{n}{8} \rfloor$, then there must be a k-RFA hard pair of scenarios, implying, by Theorem 5.2, that the class of all Boolean functions is not $\lfloor \frac{n}{8} \rfloor$-RFA learnable.

First notice that if two different scenarios share the same i.i.d.distributions p, then any function h is $(\frac{1}{4})^n$-bad for at least one of them, so it is sufficient to prove that there are two $\lfloor \frac{n}{8} \rfloor$-RFA equivalent scenarios which share the same i.i.d. distribution p.

Any scenario induces a set of $\binom{n}{k}$ k-RFA distributions (one for every k-RFA projection), where each distribution is a set of 2^k k-RFA probabilities (one for each observation). Since $p = 1/4$, each probability can be represented by $2(n-k)$ bits, and thus a k-RFA scenario can be represented by $m = \binom{n}{k} 2^k 2(n-k)$ bits. Since two scenarios are k-RFA equivalent if all of their k-RFA probabilities are identical, the number of k-RFA non-equivalent scenarios is bounded by 2^m. Hence, when 2^m is strictly smaller than 2^{2^n}, the total number of Boolean functions, then there must be at least two different functions which induce the same set of k-RFA probabilities, and hence are not distinguishable by a k-RFA learner.

It remains to show that $m < 2^n$ for $\alpha = \frac{k}{n} < \frac{1}{8}$. By Stirling approximation $\binom{n}{\alpha n} < 2^{nH(\alpha)+c\log n}$, where $H(\alpha) = -\alpha \log \alpha - (1-\alpha)\log(1-\alpha)$, and c is a constant. For large enough n, m is dominated by $2^{n(H(\alpha)+\alpha)}$, which is smaller than 2^n when $2^{H(\alpha)+\alpha} < 2$. Since $-(1-\alpha)\log(1-\alpha) < -\alpha \log \alpha$ for $\alpha < 1/2$, it is left to verify that $-2\frac{1}{8}\log\frac{1}{8} + \frac{1}{8} = \frac{7}{8} < 1$. ∎

Now we study the RFA learnability of some useful classes of Boolean functions. In particular, we show k-RFA learnability of the class of k-decision lists (see Section 4 for its definition). It is shown in [33] that k-DL is a proper superset of k-DNF $\cup$ k-CNF, and is efficiently PAC learnable (this class is among the most expressive classes known to be efficiently and distribution-free learnable). A hardness result for k-RFA learnability of k-DL, for $k \leq n-2$ has been shown in [12]. However, we show here that, under the uniform distribution, this class is (non-efficiently) k-RFA learnable. It is straightforward to generalize the algorithm to learn under any fixed distribution.

Theorem 6.9 *The class of k-DL is k-RFA learnable under the uniform distribution.*

Proof: A k-RFA algorithm which learns the class k-DL is presented in Figure 2.

We now prove its correctness. It is convenient to generalize the definition of decision lists in the following way. We omit the restriction that the last term of the list is 1, and say that if none of the terms of a list L satisfies an instance x, then $L(x) = \emptyset$ (representing an unknown value of $L(x)$).

Let L be the target decision list, and define for each term t of size k:

$$m(t) \triangleq |\{x \in \{0,1\}^n : t(x) = 1, L(x) = 1\}|$$

Note that $m(t)/2^{n-k}$ is the conditional probability of drawing a positive instance x, given that $t(x) = 1$. The first step of $\mathcal{A}$ is to estimate $m(t)$, for each term t of size k. Since the samples are large enough to ensure that (with high probability) $|\hat{m}(t) - m(t)| < 1/2$ (by Theorem 4.7), we may proceed with the assumption that $\mathcal{A}$ has the exact value of $m(t)$. Next observe that if $\hat{L}$ is consistent with L (i.e., $\hat{L}(x) = L(x)$ for every instance classified by $\hat{L}$), and $m(t) = m_b(t)$, then the list obtained by appending (t, b) to $\hat{L}$ is also consistent with L. Furthermore, any partial list which is consistent with L can be extended to a list L' which satisfies $L'(x) = L(x)$ for every instance x. Thus, while T is not empty, there must be $t \in T$ satisfying $m(t) = m_b(t)$ for either $b = 0$ or $b = 1$. It follows that (with high confidence) $\mathcal{A}$ halts after $|T|$ iterations, and outputs a list $\hat{L}$ satisfying $\hat{L}(x) = L(x)$ for every instance x. ∎

Next we show a *distribution-free* $(n-1)$-RFA algorithm which learns the class $(n-1)$-DL is shown in [12].

Initialize:	$\hat{L}$ is an empty list,
	T is the set of all the terms of size k.
Sample:	For each $I \subseteq [n]$, $\|I\| = k$, draw a sample S_I via W_I, of size
	$2^{2n-2k+7}[k(2n-2k+\log k+7)+\log \frac{n^k}{\delta}]$.
	$\hat{m}(t) \leftarrow 2^{n-k} \frac{\|\{(x,1)\in S_I : t(x)=1\}\|}{\|\{(x,y)\in S_I : t(x)=1\}\|}$, for every t over I.
While T $\neq \emptyset$:	For each $t \in T$:
	$m_0(t) \leftarrow \|\{x \in \{0,1\}^n : t(x)=1, \hat{L}(x)=1\}\|$.
	$m_1(t) \leftarrow 2^{n-k} - \|\{x \in \{0,1\}^n : t(x)=1, \hat{L}(x)=0\}\|$.
	Search for $t \in T$ satisfying $\hat{m}(t) = m_b(t)$ for either $b=0$ or $b=1$.
	If such a term t is found, remove it from T and append (t,b) to $\hat{L}$.
	Otherwise, halt (failing to construct the list).
Output:	$\hat{L}$ (with $(1,1)$ appended to its tail).

Figure 2: k-RFA learning of k-DL under the uniform distribution

6.3 Distribution-free Learning of Decision Lists

In this section we contrast the hardness results, shown in [10] and in [12], 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.

Lemma 6.10 () *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 $M(\tau,\beta,\delta,t) = \lceil \frac{\max(3,2\beta^2)}{\tau(\beta-1)^2} \log \frac{|\mathcal{A}|}{\delta} \rceil$. Then, for a sample of size $m \geq M(\tau,\beta,\delta,|\mathcal{A}|)$, the following holds with confidence $1-\delta$: For every $A \in \mathcal{A}$,*

$$\begin{aligned} p(A) > \beta\tau \quad &\Longrightarrow \quad \frac{p(A)}{\beta} < \hat{p}(A) < \beta p(A) \\ p(A) \leq \beta\tau \quad &\Longrightarrow \quad \hat{p}(A) < \beta^2\tau \end{aligned}$$

Proof: For a sample of size m, and for every $A \in \mathcal{A}$, the Chernoff bounds imply [4]:

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

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 $m = M(\tau,\beta,\delta,|\mathcal{A}|)$. ■

6.3.1 $(n-1)$-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 [10] and in [12] 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 5.12 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 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 ([20]).

Theorem 6.11 *$(n-1)$-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 5.2, the class is properly $(n-1)$-RFA 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 5). 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. For simplicity, we use the notation $p_S(e, b)$ rather then $p_S(I, z, b)$, 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 5.2 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 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 $1 \le k \le n$ 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 6.10, 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 $O(m)$. ■

6.3.2 $(n-k)$-RFA Learnability of 1-DL using k-DL

Theorem 5.14 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 $k \leq n/2$, 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 5.14, 1-DL is not properly $(n-2)$-RFA learnable.

Theorem 6.12 *For every every* $1 \leq k \leq n/2$*, 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 6.11, 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. [38]). 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 $B(h,t) \triangleq \{x \in \{0,1\}^n : h(x) = \emptyset,\ t(x) = 1\}$. 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. [33]): 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 $|h| \geq n-k$? Phase 2 of the construction is based 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 bounded by $k \leq n-k$. Consider a block $B = B(\emptyset,t)$, where $\emptyset$ is undefined for every x, and t 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 $\{x\}$ (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 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 6.10 $p(B,b) \leq \beta\tau$, 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 in 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 thus 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

$$error_{f,D}(h) \leq \epsilon_1 + \epsilon_2 + \epsilon_3 = (n - k)\tau\beta(1 + \beta^2 2^k) < (n - k)\tau\beta^2(1 + \beta^2)^k 2^{k+1}$$

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 $\tau = O(\frac{\epsilon}{n 2^{2k}})$ 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 6.10, 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 $\beta\tau$.

Phase 1 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 $\lceil \frac{n-|h|}{n-k-|h|} \rceil \leq k + 1$ 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 6.10, a sample of size $M(\tau, \beta, \frac{\delta}{3(n-k)(k+1)}, 4(n - k))$ is sufficient, hence a sample of size $(n - k)(k + 1)M(\tau, \beta, \frac{\delta}{3(n-k)(k+1)}, 4(n - k)) = O(\frac{k^3 n^2 2^{2k}}{\epsilon} \log \frac{n}{\delta})$ 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 6.10, a sample of size $M(\tau, \beta, \frac{\delta}{6k^2}, k)$ is sufficient for each stage, and $2k^2 M(\tau, \beta, \frac{\delta}{6k^2}, k) = O(\frac{k^3 n^2 2^{2k}}{\epsilon} \log \frac{n}{\delta})$ observations are 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 $M(\tau, \beta, \frac{\delta}{3}, 2^k) = O(\frac{k^3 n 2^k}{\epsilon} \log \frac{1}{\delta})$ is sufficient for this phase, and the overall sample complexity of the algorithm is $O\left(\frac{k^3 n^2 2^{2k}}{\epsilon} \log \frac{n}{\delta}\right)$.

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 in Phases 2 and 3 each example drawn changes only one empirical probability. Hence, the time complexity is $O(km)$. ■

We conclude this section by showing a learning algorithm for the rich class of AC^0 Boolean circuits (see the definition in Section 4.1), which is almost efficient (its time complexity is quasi-polynomial) and learns under the uniform distribution. This result is a direct consequence of the following result, by Linial, Mansour and Nisan [32].

Theorem 6.13 (Linial, Mansour, Nisan) *Under the uniform distribution, the class of $AC^0[d]$ Boolean circuits is PAC-learnable with a sample of size $O(n^{2k} + \log \frac{n}{\delta})$, where ϵ and δ are the accuracy and confidence learning parameters, respectively, and $k = O(\log^d \frac{n}{\epsilon})$.*

Analyzing the proof of the theorem we conclude the following result. Define H_k to be the class of all hypotheses of the form $h(x) = \frac{1}{2}(1 + sign(\sum_{|S| \leq k} \alpha_S \chi_S(x)))$, where each $\alpha_S \in \mathbf{R}$, and $\chi_S(x_1, \ldots, x_n) = 2(\sum_{i \in S} x_i \bmod 2) - 1$.

Corollary 6.14 *Under the uniform distribution, the class of $AC^0[d]$ Boolean functions is k-RFA learnable using H_k, with a sample complexity of $O(n^k(n^{2k} + \log \frac{n}{\delta}))$, where $k = O(\log^d \frac{n}{\epsilon})$, and ϵ, δ are the learning accuracy and confidence learning parameters, respectively.*

Proof: The learning algorithm in [32] actually estimates the coefficients of the Fourier representation of the target function with respect to the basis $\{\chi_S : S \subset [n]\}$. It is shown in [32] that only coefficients corresponding to χ_S for $|S| \leq k$ have strong influence on the value of the function, so it is sufficient to estimate only them. However, since the given basis is orthogonal, each coefficient can be estimated independently (using the internal product of the function and the corresponding χ_S). Thus, a learning algorithm can use the projection w_S to estimate the coefficient of χ_S (recall that, for $S = \{i_1, \ldots, i_k\}$, $w_S(x_1, \ldots, x_n, y) = (x_{i_1}, \ldots, x_{i_k}, y)$).. This implies a multiplicative factor of $O(n^k)$ (the number of relevant coefficients) on the sample complexity. ■

Note that k depends on n,d and ϵ. Thus, a limited visibility of size k implies a lower bound on ϵ for every n and d. However, in Section 7 we show that, in order to learn this class, the learner actually needs to extract only two bits of information from each example.

6.4 k-RFA Learning in the Euclidean Space

In Section 6.1 we have used the decomposition learning tool, in to learn decomposable classes, such as axis-aligned rectangles, under the RFA restriction. For the task of learning non-decomposable classes, such as linear half-spaces in $\mathbf{R}^n$, which appear to be much more subtle, other technique have to be developed. Also, it seems that, for such classes, the distribution-free setting has to be relaxed to a more restricted class of distribution, such as the class of product distribution. Our main result here is a learning algorithm which learns, under product distributions, subspaces of $\mathbf{R}^n$ which are bounded by 'mild' surfaces.

Definition 6.15 (Bounded Envelope Span) *Let $g : \mathbf{R}^n \to \mathbf{R}$ be a continuous function. The* **envelope span** *of g is a function $S_g : \mathbf{R} \to \mathbf{R}$, where $S_g(r)$ is the planar integral of the surface of g over the domain $\{x \in [-r, r]^n : |g(x)| \leq r\}$. That is, $S_g(r)$ is defined by*

$$S_g(r) \triangleq \int \cdots \int_{\{x \in [-r,r]^n : |g(x)| \leq r\}} \left[1 + \sum_{i=1}^{n} (\frac{\partial g}{\partial x_i})^2 \right]^{\frac{1}{2}} dx_1 \cdots dx_n$$

A class $\mathcal{G}$ of continuous functions $g : \mathbf{R}^n \to \mathbf{R}$ has a **bounded envelope span** *if there is a function $S_{\mathcal{G}} : \mathbf{R} \to \mathbf{R}$ such that $S_g(r) \leq S_{\mathcal{G}}(r)$ for every $g \in \mathcal{G}$ and every $r \in \mathbf{R}$.*

Before showing the learning algorithm, let us demonstrate the richness of the class of subspaces of $\mathbf{R}^n$ which have a bounded envelope span. For every positive integer s, let $\mathcal{G}_{\mathbf{s}}$ be the class of all continuous functions, for which every projection on each one of the axes, has no more than s extrema. For instance, all the linear functions are included in $\mathcal{G}_0$, and all the multivariate polynomials of degree d are included in $\mathcal{G}_d$.

Proposition 6.16 *For every $s \geq 0$, $\mathcal{G}_s$ has a bounded envelope span.*

Proof: Given a function $f : \mathbf{R} \to \mathbf{R}$, and a positive constant $c \in \mathbf{R}$, define

$$I_f(r, c) \triangleq \int_{\{|x| \leq r : |f(x)| \leq r\}} (c + (f')^2)^{\frac{1}{2}} \, dx$$

where $f' \equiv \frac{df}{dx}$. We first prove the following lemma.

Lemma 6.17 () *If f has no more than s extrema then $I_f(r,c) \le 2r(s+2)(c+1)^{\frac{1}{2}}$.*

Proof: Consider first a linear function $f(x) = ax+b$, and define for every two closed intervals $A, B \subset \mathbf{R}$

$$I_f(A,B,c) \triangleq \int_{\{x\in A: f(x)\in B\}} (c+(f')^2)^{\frac{1}{2}}\ dx$$

If $|a| < 1$, then $I_f(A,B,c) \le |A|(c+1)^{\frac{1}{2}}$. Otherwise, assume that $a \ge 1$ (the case $a \le -1$ is symmetric), and let $B = [l,u]$. Then, $ax+b \in [l,u]$ implies $x \in [\frac{l-b}{a}, \frac{u-b}{a}]$, and hence

$$I_f(A,B,c) \ \le\ \int_{\frac{l-b}{a}}^{\frac{u-b}{a}} (c+a^2)^{\frac{1}{2}}\, dx \ =\ \frac{u-l}{a}(c+a^2)^{\frac{1}{2}}$$

For every $c > 0$, the function $\frac{(c+a^2)^{\frac{1}{2}}}{a}$ is monotonically decreasing in the domain $a \ge 1$, and therefore $I_f(A,B,c) \le |B|(c+1)^{\frac{1}{2}}$. Thus, $I_f(A,B,c) \le (|A|+|B|)(c+1)^{\frac{1}{2}}$, for every linear function f.

Next consider a continuous function f which has no more than s extrema in the domain $x \in [-r,r]$. For every integer $m \ge 1$, we define a piecewise linear function f_m which approximates f on the domain $x \in [-r,r]$. Let $a_j = \frac{j}{m}r$ for every integer j, $|j| \le m$, and let $f_{m,j}$ be the linear function which is determined by fixing $f_{m,j}(a_j) = f(a_j)$ and $f_{m,j}(a_{j+1}) = f(a_{j+1})$. f_m is defined by letting $f_m(x) = f_{m,j}(x)$ for every $x \in [a_j, a_{j+1}]$.

Let $A_j = [a_j, a_{j+1}]$, and let $B_j = [\min\{f(a_j), f(a_{j+1})\}, \max\{f(a_j), f(a_{j+1})\}]$. Then $I_{f_m}(r,c) = \sum_{j=-m}^{m-1} I_{f_{m,j}}(A_j,B_j,c)$. The linearity of $f_{m,j}$ implies that $I_{f_{m,j}}(A_j,B_j,c) \le (|A_j|+|B_j|)(c+1)^{\frac{1}{2}}$, thus $I_{f_m}(r,c) \le \sum_{j=-m}^{m-1}(|A_j|+|B_j|)(c+1)^{\frac{1}{2}}$. Notice that $\sum_j |A_j| = 2r$, and that since f has no more than s extrema, $\sum_j |B_j| = 2r(s+1)$. It follows that $I_{f_m}(r,c) \le 2r(s+2)(c+1)^{\frac{1}{2}}$ for every $m \ge 1$. Finally, since $f = \lim_{m\to\infty} f_m$, we conclude that $I_f(r,c) = \lim_{m\to\infty} I_{f_m}(r,c) \le 2r(s+2)(c+1)^{\frac{1}{2}}$.

■

Now we claim that for every function $g \in \mathcal{G}_s$, and for every $0 \le i \le n$,

$$S_g(r) \le \max_{x_1,\ldots,x_i} (2r)^i(s+2)^i \int\cdots\int_B \left[i+1+\sum_{j=i+1}^{n} (\frac{\partial g}{\partial x_j})^2 \right]^{\frac{1}{2}} dx_{i+1}\cdots dx_n \tag{2}$$

where $B = \{(x_{i+1},\ldots,x_n) \in [-r,r]^{n-i} : |g(x)| \le r\}$. By the definition of S_g, Inequality 2 holds for $i = 0$. Assuming that it holds for $i-1$, we prove that it holds also for i ($1 \le i \le n$). Fix the values of $x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n$, let $f(x_i) = g(x_1,\ldots,x_n)$, and let $c = (i+\sum_{j=i+1}^{n}(\frac{\partial g}{\partial x_j})^2)^{\frac{1}{2}}$. By Lemma 6.17, $I_f(r,c) \le 2r(s+1)(c+1)$, and thus Inequality 2 holds for i as well. For $i = n$ we get $S_g(r) \le (2r)^n(s+2)^n(n+1)^{\frac{1}{2}}$, concluding that the class $\mathcal{G}_s$ has a bounded envelope span.

■

Let $\mathcal{G}$ be a class of continuous functions from $\mathbf{R}^{n-1}$ to $\mathbf{R}$. We associate with a function $g \in \mathcal{G}$ a surface in $\mathbf{R}^n$, $\{(x_1,\ldots,x_n) : g(x_1,\ldots,x_{n-1}) = x_n\}$, and a subset of $\mathbf{R}^n$, $\{(x_1,\ldots,x_n) : g(x_1,\ldots,x_{n-1}) \ge x_n\}$. The binary-valued function $f_g : \mathbf{R}^n \to \{0,1\}$ is the function which is 1 on the subset below the surface of g. We now show that if the class $\mathcal{G}$ has a bounded envelope span, then the class $\{f_g : g \in \mathcal{G}\}$ is $(n-1)$-RFA learnable under product distributions. That is, the class is learnable even when one of the attributes of each example is invisible. The algorithm is efficient for every fixed dimension n. Thus, for example, we have an efficient algorithm which learns the class of linear half-spaces in $\mathbf{R}^2$ (under product distribution), by observing only one attribute at any specific time.

Theorem 6.18 *Let $\mathcal{G}$ be a class of functions from $\mathbf{R}^{n-1}$ to $\mathbf{R}$ which have a bounded envelope span, and let $\mathcal{F} = \{f_g : g \in \mathcal{G}\}$, where $f_g(x_1,\ldots,x_n) = 1$ iff $g(x_1,\ldots,x_{n-1}) < x_n$. Then $\mathcal{F}$ is $(n-1)$-RFA learnable under the class of product distributions.*

1. For each $1 \le i < n$:

   (a) Draw a sample S_i^1 via w_i, of size $m_1 = \frac{2(n-1)}{\epsilon} \log \frac{3(n-1)}{\delta}$.

   (b) $r_i \leftarrow \max\{|x_i| : x \in S_i^1\}$.

2. $r \leftarrow \max\{r_i : 1 \le i < n\}$; $B = S_{\mathcal{G}}(r)$.

3. For each $1 \le i \le n$:

   (a) Draw a sample S_i^2 via w_i, of size $m_2 = \frac{512B^2}{\epsilon^2} \log \frac{512B^2}{\epsilon^2} + \frac{256B^2}{\epsilon^2} \log \frac{12}{\delta}$.

   (b) Partition the interval $[-r, r]$ of the x_i axis into $\frac{2B}{\epsilon}$ sub-intervals, each with equal number of successive instances from (sorted) S_i^2.

4. Let E_{n-1}, E_n be the grids obtained by these partitions on the axes $(x_1, \ldots, x_{n-1})$ and $(x_1, \ldots, x_n)$, respectively.

5. Draw a sample S_3 via $w_{[n-1]}$, of size
   $m_3 = 32(\frac{2B}{\epsilon})^{2n-2}[2(n-1)\log(64(n-1)(\frac{2B}{\epsilon})^{n-2}) + \log \frac{12}{\delta}]$.

6. For each entry $e \in E_{n-1}$, estimate the probability p_e of drawing a positive instance whose projection on E_{n-1} lies in e.

7. For each entry $e \in E_{n-1}$, mark the $\hat{p}_e(2B)/\epsilon)^{n-1}$ lowest entries in E_n whose projection on E_{n-1} lies in e (the others are marked as negative).

8. Output a hypothesis h which classifies an instance as positive if it lies in a marked entry of E_n.

Figure 3: $(n-1)$-RFA learning of bounded-envelope-span subsets of $\mathbf{R}^n$

Proof: For each $1 \le i \le n$, we choose any projection in which x_i is visible, and denote this projection by w_i. The learning algorithm is described in Figure 3.

For the analysis of the algorithm, note that:

- Choosing $n-1$ samples of size m_1 ensures that, with probability of at least $1-\delta/3$, $\Pr_{x \in D}[\exists i\ |x_i| > r] < \frac{\epsilon}{2}$. Hence, the probability that h errs on an instance which lies outside E_n is at most $\epsilon/2$.
- Using standard uniform convergence analysis (see Theorem 4.7), it can be verified that a sample of size m_2 ensures that, with probability of at least $1 - \frac{\delta}{3}$, each axis is partitioned into $\frac{4B}{\epsilon}$ intervals, such that the probability of an instance to be in each of the intervals is less than $\frac{\epsilon}{2B}$. Since the underlying target distribution is a product distribution, an n-dimensional grid is obtained, for which the probability of drawing a point which lies in some entry of the grid is less than $(\frac{\epsilon}{2B})^n$.
- A sample of size m_3 ensures that, with probability of at least $1 - \frac{\delta}{3}$, for each entry in E_{n-1}, the estimate of the probability of drawing a positive instance which lies in that entry, is within $\frac{1}{2}(\frac{\epsilon}{2B})^{n-1}$ of the true probability. Thus, for a target function f_g, the function g can cross at most $B(2B/\epsilon)^{n-1}$ entries of the grid, incurring an error of at most $(\epsilon/(2B))^n$ in each one of them. Hence the probability that h errs on an instance which lies inside E_n is at most $\epsilon/2$.

■

Notice that the sample size of the learning algorithm shown in the proof of Theorem 6.18 is not uniform for all the product distributions, since the size of the samples m_2 and m_3 depends on the sample m_1. However, in case there is $r \in \mathbf{R}$ such that the distribution on R^n is known to be concentrated in the domain $[-r, r]^n$, the sample size can be uniform for all the product distributions. In such a case the algorithm is efficient for a fixed dimension n (it is polynomial in $\frac{1}{\epsilon}$, in $\frac{1}{\delta}$ and in $S_{\mathcal{G}}(r)$).

We conclude this section, by remarking that for the class $\mathcal{G}_0$, an $(n-1)$-RFA learner can learn a richer class of subset of $\mathbf{R}^n$. Let $\mathcal{F}$ be the class of subsets defined by $\mathcal{G}_0$ as before. In addition, for every $g \in \mathcal{G}_0$, let $\bar{f}_g$ be defined by $\bar{f}_g(x_1, \ldots, x_n) = 1$ iff $g(x_1, \ldots, x_{n-1}) > x_n$, and let $\bar{\mathcal{F}} = \{\bar{f}_g : g \in \mathcal{G}\}$. Observing that f_g and $\bar{f}_g$ induce a distinguishable behavior with respect to the x_n axis (the conditional probabilities are either decreasing or increasing), we conclude that the class is $\mathcal{F} \cup \bar{\mathcal{F}}$ is $(n-1)$-RFA learnable.

7 k-wRFA Learning

In this section we study a variant of the RFA model in which a weaker restriction is made on the learner—he is restricted to observe only a certain amount of bits obtained from each example (no restriction is made on how they are obtained from the example). We call this variant wRFA (weakly Restricted Focus of Attention). We say that a function class $\mathcal{F} \subseteq \{0,1\}^X$ is k-wRFA learnable if there is an algorithm which learns $\mathcal{F}$ via W, where $W \subseteq (\{0,1\}^k)^{X\times\{0,1\}}$.

Notice that the $(k+1)$-wRFA restriction is weaker than the k-RFA restriction: k-RFA learnability implies $(k+1)$-wRFA learnability. Also, for $k = n$, the two variants coincide with the standard PAC model. The following theorem shows that, for every $k < n$, k-RFA learnability is strictly weaker than $(k+1)$-wRFA learnability.

Theorem 7.1 *For every $0 \leq k \leq n$, the class $PARITY_n^{k+1}$ is 1-wRFA learnable but not k-RFA learnable.*

Proof: It is proven in Theorem 6.6, that the class $\mathrm{PARITY}_{\mathrm{n}}^{\mathrm{k+1}} = \{\mathrm{EVEN}_{\mathrm{n}}^{\mathrm{k+1}}, \mathrm{ODD}_{\mathrm{n}}^{\mathrm{k+1}}\}$ is not k-RFA learnable. However, this class is 1-wRFA learnable (given any example $(x_1, \ldots, x_n, y)$, the single-bit observation $x_1 \oplus \cdots \oplus x_{k+1} \oplus y$ is sufficient for distinguishing between the two functions). ■

The next example demonstrates that relaxing an RFA restriction to a wRFA restriction enables a significant reduction in the size of the observation space. (It will be shown in Section 8 that a small observation space guarantees noise-tolerance).

Example In Theorem 6.13 we used a result from [32] to show that, under the uniform distribution, the class of $AC^0[d]$ boolean circuits is k-RFA learnable, for $k = O(\log^d \frac{n}{\epsilon})$. The algorithm learns the target function by estimating the coefficients of its Fourier representation with respect to the basis $B = \{\chi_S : S \subset [n]\}$, where $\chi_S(x_1, \ldots, x_n) = 2(\sum_{i\in S} x_i \bmod 2) - 1$. It is shown in [32] that it is sufficient to estimate the coefficients corresponding to the subset $\{\chi_S \in B : |S| = k\}$. Since the algorithm only need to estimate the correlation between the target function (as a $\{-1,1\}$-valued function) and χ_S, he can use the projection class $\{w'_S : S \subset [n], |S| = k\} \subseteq \{-1,1\}^{X\times\{0,1\}}$, where $w'_S(x_1, \ldots, x_n, y) = \chi_S(x_1, \ldots, x_n) \cdot (2y-1)$. Thus, the class is 1-wRFA learnable (with the same time complexity). In Section 8 we prove that this property guarantees robustness of the learning algorithm against classification noise.

Next we show a tradeoff between the sample complexity and the size of the observation space.

Theorem 7.2 *If $\mathcal{F}$ is k-wRFA learnable with sample complexity $m(\epsilon, \delta, n, size(f))$, then, for every $1 \leq i < k$, $\mathcal{F}$ is $(k-i)$-wRFA learnable with sample complexity $\frac{2^{3i+2} m_1^3}{\delta} \log \frac{m_1 2^{i+2}}{\delta}$, where $m_1 = m(\epsilon, \frac{\delta}{2}, n, size(f))$.*

Proof: Let $\mathcal{A}$ be a k-wRFA learning algorithm which learns $\mathcal{F}$ via a projection class W with sample complexity $m(\epsilon, \delta, n, size(f))$, and let $m_1 = m(\epsilon, \frac{\delta}{2}, n, size(f))$. The target distribution D, the target function f, and the m_1 projections used by $\mathcal{A}$, induce a distribution over the m_1-fold observation space $(\{0,1\}^k)^{m_1}$. Given $j = k - i > 0$, the idea is to construct a j-wRFA algorithm $\mathcal{B}$ which supplies $\mathcal{A}$ with m_1 observations drawn from a distribution which is $\frac{\delta}{2}$-close to the above distribution. Since $\mathcal{A}$ finds a good hypothesis with a probability of at least $1 - \frac{\delta}{2}$, this guarantees that $\mathcal{B}$ finds a good hypothesis with probability of at least $1 - \delta$.

Given a k-wRFA projection w, and given $z \in \{0,1\}^{i+1}$, let w_z be the following j-wRFA projection: $w_z(x,y) = (a_1, \ldots, a_{j-1}, b)$, where $w(x,y) = (a_1, \ldots, a_k)$, and $b = 1$ iff $(a_j, \ldots, a_k) = z$. Whenever $\mathcal{A}$ makes an observation via the projection w, the algorithm $\mathcal{B}$ takes 2^{i+1} large samples, via each of the projections in the set $\{w_z \mid z \in \{0,1\}^{i+1}\}$. Using these samples, $\mathcal{B}$ estimates the distribution which is projected on $\{0,1\}^k$ by D, f and w. Then $\mathcal{B}$ can draw an observation from the estimated distribution, and supplies $\mathcal{A}$ with this observation.

We now compute $\mathcal{B}$'s sample complexity. We want the m_1-fold distribution to be $\frac{\delta}{2}$-close to the real distribution. Thus, for each observation, we need the distribution from which $\mathcal{B}$ draws this observation to be $\frac{\delta}{m_1}$-close to the real distribution. Since $\mathcal{B}$ computes this distribution by estimating 2^{i+1} probabilities, each estimate should be $\frac{\delta}{m_1 2^{i+1}}$-close to the real probability. By Theorem 4.6, a sample of size $\frac{m_1^2 2^{2i+1}}{\delta^2} \log \frac{m_1 2^{i+2}}{\delta}$ is sufficient to guarantee such an estimate with probability of at least $\frac{\delta}{2m_1 2^{i+1}}$. Altogether, the sample complexity of $\mathcal{B}$ is $\frac{m_1^3 2^{3i+2}}{\delta^2} \log \frac{m_1 2^{i+2}}{\delta}$.

■

Corollary 7.3 *If $\mathcal{F}$ is efficiently k-wRFA learnable then $\mathcal{F}$ is efficiently $(k-j)$-wRFA learnable, for every $j < k$, $j = O(\log n)$.*

Does the k-wRFA restriction impose a real hierarchy over the learnable classes of functions? To answer this question, we distinguish between information-theoretic learnability and efficient learnability.

Consider first the case of information-theoretic learnability. In Section 5 we have shown that the hierarchy collapses to 1-wRFA learnability when the domain is finite. We now prove that this remains true for finite function classes, even when the domain is infinite.

Theorem 7.4 *Every finite function class is 1-wRFA learnable.*

Proof: Let $\mathcal{F}$ be a finite function class, and define for every $h \in \mathcal{F}$ a projection

$$w_h(x,y) \triangleq \begin{cases} 1 & h(x) = y \\ 0 & h(x) \neq y \end{cases}$$

We claim that $\mathcal{F}$ is learnable via the projection class $W_{\mathcal{F}} = \{w_h \; : \; h \in \mathcal{F}\}$, as follows. For every $h \in \mathcal{F}$, take a sample S_h of size $\frac{1}{\epsilon} \log \frac{|\mathcal{F}|}{\delta}$ via the projection w_h. Assume f is the target function and D is the target distribution. Then, for every $h \in \mathcal{F}$, with probability of at least $1 - \frac{\delta}{|\mathcal{F}|}$, the sample S_h includes a negative observation if and only if h is ϵ-bad for f under D. Thus, by choosing any h for which S_h includes only positive observations, a learner may obtain, with confidence $1-\delta$, an ϵ-good hypothesis for f under D. ■

Similarly, we can show that if the function class is PAC learnable using a finite hypothesis class $\mathcal{H}$, then it is 1-wRFA learnable using $\mathcal{H}$. Also note that, by Theorem 3.1, under any fixed distribution, every function class $\mathcal{F}$ with finite VC-dimension is 1-wRFA learnable. Thus, it seems that a small observation space is sufficient for learnability. In fact, we do not know of any function class whose learnability imposes a nontrivial lower bound on the size of the observation space.

The situation is different when complexity issues are being considered. In Section 8 we show that efficient $O(\log n)$-wRFA learnability is equivalent to efficient learnability from statistical queries. It is shown in [25] that the class of all parity functions is not efficiently learnable from statistical queries, hence it not efficiently $O(\log n)$-wRFA learnable. However, being efficiently PAC learnable, this class is efficiently $n - O(\log n)$-wRFA learnable. Hence, there is a lower bound of $\omega(\log n)$ on the size of the observation space needed for efficient learnability of the class of parity functions.

Proposition 7.5 *$PARITY_n$ (the class of all parity functions over n variables) is n-wRFA learnable but not $(\log n)$-wRFA learnable.*

7.1 Efficient k-wRFA learnability of k-Decision Lists

It is proven in [12] that k-DL is not k-RFA learnable, for every $k \leq n-2$. In Section 6 we have shown that the class of k-DL is (non-efficiently) k-RFA learnable under the uniform distribution. Here we show a simple and efficient algorithm which learns this class under the 1-wRFA restriction. In Section 8 we use this result to construct a simple and efficient noise-tolerant learning algorithm for this class (another efficient noise-tolerant algorithm which learns decision lists is shown in [34]).

Theorem 7.6 *For a fixed k, k-DL is efficiently 1-wRFA learnable.*

Proof: By Theorem 7.2, it is enough to show efficient k-wRFA learnability. As in the proof of Theorem 6.9, we omit the restriction that the last term in the list is 1, and say that if none of the terms of a list L satisfies an instance x then $L(x) = \emptyset$ (representing an unknown value of $L(x)$). Given a decision list L (which satisfies $L(x) \in \{0, 1, \emptyset\}$ for every $x \in X$), and given $I = \{i_1, \ldots, i_k\} \subseteq [n]$, we define the projection w_I^L by:

$$w_I^L(x,y) = \begin{cases} (x_{i_1}, \ldots, x_{i_k}, \emptyset) & L(x) \in \{0,1\} \\ (x_{i_1}, \ldots, x_{i_k}, y) & L(x) = \emptyset \end{cases}$$

Define $W_{DL} \triangleq \{w_I^L : I \subseteq \{1, \ldots, n\} \ , \ |I| = k \ , \ L \in \text{k-DL}\}$.

Initialize:	$\hat{L}$ is an empty list.
	T is the set of all the terms of size k.
While T $\neq \emptyset$:	For every $V \subseteq \{1, \ldots, n\}$ of size k,
	draw a sample S_V of size $\frac{(2n+1)^k}{\epsilon} \log \frac{(2n+1)^k}{\delta}$ via $w_V^{\hat{L}}$.
	Search for $t \in T$ which satisfies for either $b = 0$ or $b = 1$:
	for every $(x, y) \in S_V$, if $t(x) = 1$ then $y \in \{b, \emptyset\}$.
	If such a term t is found, remove it from T and append (t, b) to $\hat{L}$,
	otherwise halt (failing to construct the list).
Output:	$\hat{L}$ (with $(1,1)$ appended to its tail).

Figure 4: Efficient k-wRFA learning of k-DL

An efficient learning algorithm for k-DL is shown in Figure 7.1. We now analyze the correctness of the algorithm. First note that the number of terms of size at most k , over n variables, is $(2n+1)^k$. Clearly, $|\hat{L}| \leq (2n+1)^k$ at any stage of the algorithm. We prove by induction on the length of $\hat{L}$ that, with probability at least $1 - \frac{|\hat{L}|}{(2n+1)^k}$, $\Pr_{x \in D}[L(x) \neq \phi \ \wedge \ L(x) \neq \hat{L}(x)] < \frac{|\hat{L}|}{(2n+1)^k}\epsilon$, Initially, this is true since $\hat{L}$ is empty (hence, $\hat{L}(x) = \phi$ for every x). Assume that the assertion is true for a list $\hat{L}$. Before appending a new term t to the list, the algorithm draws a sample of size $\frac{(2n+1)^k}{\epsilon} \log \frac{(2n+1)^k}{\delta}$ and tests for homogeneity of the examples with respect to this term. That is, the entry (t, b) is appended to the list $\hat{L}$, if all the examples in the sample which are not determined by $\hat{L}$ are classified as b. This implies that $\Pr_{x \in D}[\{x : L(x) \neq t(x) \ \wedge \ \hat{L}(x) \neq \phi\}] < \frac{\epsilon}{(2n+1)^k}$ with probability of at least $1 - \frac{\delta}{(2n+1)^k}$. Thus, if the entry (t, b) is appended to the list $\hat{L}$, then we get a list which satisfies the inductive assertion.

∎

8 RFA Learning and Noise-Tolerance

In many learning situations the correct classification of some of the examples may be randomly corrupted. A common paradigm for this problem, in both theoretical and experimental research, is the *classification-noise* model [31, 39, 40, 3]. The classification-noise model replaces the (deterministic) classification of the examples supplied to the learner, by a random variable (independently for every drawn instance) that has some fixed probability η of assuming the inverse of the true classification value, and with probability $(1-\eta)$ assumes the correct classification value (for a more stringent noise models, see [26, 16]).

The task of providing methods by which efficient learning algorithms can be strengthened to noise-tolerant ones is one of the most important in computational learning theory. As this task seems to be very difficult, one naturally seeks partial solutions. A natural approach is to consider restrictions on the learner, such that learnability under such restrictions can be shown to entail learnability in the presence of noise.

A significant step in this direction was taken by Kearns, who presented the Statistical Queries (SQ) model [25]. In this model the learning algorithm is restricted to using only statistics of the space of classified examples, rather than having the flexibility of being sensitive to properties of individual examples. It is proven in [25] that any learning algorithm obeying such a restriction can be transformed into a noise-tolerant one (the transformation preserves the complexity of the learning algorithm). Kearns demonstrated the usefulness of the SQ model by showing that many of the known efficient learning algorithms can be naturally adapted to meet the restrictions that the SQ framework imposes upon the learner.

It turns out that the $O(\log n)$-wRFA restriction is also sufficiently restrictive to guarantee robustness against classification noise, and hence provides a sufficient condition for noise-tolerance which may be applied directly to algorithms which learns from random examples. We prove here that both models are essentially equivalent (within polynomial factors).

To formalize the classification-noise variant of the RFA learning scheme (formally defined in Section 2), we fix a *noise rate* η, $0 \le \eta < \frac{1}{2}$, and denote by P_η the distribution over $\{0,1\}$ which is 1 with probability η. Given a sequence of m instances $\bar{x} = (x_1, \ldots, x_m) \in X^m$, a target function $f \in \mathcal{F}$, a focusing function Φ, and a sequence $\bar{z} \in \{0,1\}^m$, the observation sample generated by $\overline{x}$, f, Φ, and $\bar{z}$ is

$$sample(\overline{x}, f, \Phi, \bar{z}) = (w_1(x_1, z_1 \oplus f(x_1)), \ldots, w_m(x_m, z_m \oplus f(x_m)))$$

where $w_i = \Phi(w_1(x_1, f(x_1)), \ldots, w_{i-1}(x_{i-1}, f(x_{i-1})))$, for $1 \le i \le m$.

$\mathcal{F} \subseteq 2^X$ is learnable in the presence of classification noise, via $W \subseteq \mathcal{O}^{X\times\{0,1\}}$ and using $\mathcal{H}$, if there is an integer-valued sampling function $m(\cdot,\cdot)$, there is a focusing function $\Phi : \mathcal{O}^* \to W$, 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, for every noise rate $0 \le \eta < \frac{1}{2}$, and for every $0 < \epsilon, \delta \le 1$

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

As before, an efficient learning algorithm has to find an ϵ-good hypothesis with high probability, but now we let its running time be also polynomial in $(\frac{1}{2} - \eta)^{-1}$. (the closet the noise rate is to $1/2$, the larger is the sample needed to compensate for the noise). For sake of simplicity we assume that the noise rate η is known to the learner (a general scheme for efficiently handling an unknown noise rate is shown in [31, 25]). Note that this model assumes a noisy source of classified examples, while the projecting mechanism is noise-free.

We start by showing noise robustness for a slightly more restrictive RFA setting, which we call **mRFA** (mildly Restricted Focus of Attention). In this model the learner is restricted to observe a certain amount of bits obtained from the attribute vector of each example (no restriction is made on how they are obtained from the attributes), but he may also observe the classification bit of every example. We say that a function class $\mathcal{F}$ is k-mRFA learnable if there is an algorithm which learns $\mathcal{F}$ via W, where each $w \in W$ is a function of the form $w(x,y) = (u(x), y)$, and $u : X \to \{0,1\}^k$. Notice that k-mRFA learnability implies $(k+1)$-wRFA learnability.

The following theorem demonstrates that, once we know the noise rate, it is possible to simulate a 1-bit noiseless mRFA oracle by sampling a noisy mRFA oracle.

Theorem 8.1 *If $\mathcal{F}$ is efficiently 1-mRFA learnable, then $\mathcal{F}$ is efficiently 1-mRFA learnable in the presence of classification noise.*

Proof: Let $\mathcal{A}$ be an efficient 1-mRFA learning algorithm which uses m observations to learn $\mathcal{F}$. For a target function f, and a target distribution D, and a given noise rate, a projection used by $\mathcal{A}$ induces a distribution over the observation space $\{0, 1\}^2$. The idea is to show how an algorithm $\mathcal{B}$, which receives noisy observations, can simulate a noiseless oracle, and feed $\mathcal{A}$ with observations which are are drawn from a distribution which is very close to the true (noiseless) one.

(This basic observation has been noticed before by several authors [31, 5, 37]).

A single observation is made by $\mathcal{B}$ as follows. Assume $\mathcal{A}$ makes an observation via the projection $w(x,y) = (u(x), y)$, where $u(x) \in \{0,1\}$. Let $p_{(a,b)} = \Pr_{x\in D}[w(x,y) = (a,b)]$, for every $(a,b) \in \{0,1\}^2$, and denote by $\tilde{p}_{(a,b)}$ the probability of drawing a noisy observation (a,b). Then

$$\tilde{p}_{(a,b)} = (1-\eta)p_{(a,b)} + \eta p_{(a,\bar{b})} \tag{3}$$

$$\tilde{p}_{(a,\bar{b})} = (1-\eta)p_{(a,\bar{b})} + \eta p_{(a,b)} \tag{4}$$

Knowing η (or a good approximation of it), one may solve the above equations for $p_{(a,b)}$, yielding

$$p_{(a,b)} = \frac{(1-\eta)\tilde{p}_{(a,b)} - \eta\tilde{p}_{(a,\bar{b})})}{1-2\eta} \tag{5}$$

Thus, $\mathcal{B}$ can compute a good estimate of $p_{(a,b)}$ from good estimates of $\tilde{p}_{(a,b)}$ and $\tilde{p}_{(a,\bar{b})}$. (This basic observation has been noticed before by several authors [31, 5, 37]).

It remains to prove the efficiency of $\mathcal{B}$. Assume that $\mathcal{A}$ needs m observations to compute an ϵ-good approximation of the target function with a confidence of at least $1 - \frac{\delta}{2}$. It is sufficient to prove that $\mathcal{B}$ can efficiently compute (and by Theorem 4.12, it can also efficiently sample) a distribution over $\{0,1\}^{2m}$, which is $\frac{\delta}{2}$-close to the true distribution. Hence, the distribution from which $\mathcal{B}$ draws an observation has to be $\frac{\delta}{2m}$-close to the true distribution. Since the distribution is calculated by Equation 5, the estimate of the noisy distribution should be $\frac{\delta}{2m(1-2\eta)}$-close to the true noisy distribution. By Theorem 4.6, the number of observation needed for such an estimate is polynomial in $\left(\frac{\delta}{2m(1-2\eta)}\right)^2$, which is polynomial in all the relevant parameters (including $(1-2\eta)^{-1}$).

■

Hence, the 1-mRFA guarantees noise-tolerance. This restriction can be relaxed to an $O(\log n)$-mRFA restriction as follows. First, the proofs of Theorem 7.2 and Corollary 7.3 can be easily adapted to the mRFA model, yielding the following.

Proposition 8.2 *If $\mathcal{F}$ is efficiently k-mRFA learnable then $\mathcal{F}$ is efficiently $(k-i)$-mRFA learnable, for every $i = O(\log n)$, $i < k$.*

Finally, we notice that one can move from the mRFA restriction to the wRFA restriction by doubling the visibility size, hence the $O(\log n)$-mRFA restriction can be relaxed to an $O(\log n)$-wRFA restriction.

Proposition 8.3 *If the function class $\mathcal{F}$ is k-wRFA learnable from a sample of size m, then $\mathcal{F}$ is $(2k)$-mRFA learnable from a sample of size m.*

Proof: Given a k-wRFA learning algorithm $\mathcal{A}$, transform it to a $2k$-mRFA learning algorithm $\mathcal{B}$, by replacing each k-wRFA projection $w : X \times Y \to \{0,1\}^k$ with the $2k$-mRFA projection $w'(x,y) = (u(x), y)$, where $u(x) = (w(x,0), w(x,1))$. Whenever $\mathcal{A}$ makes an observation via the projection w, $\mathcal{B}$ can make an observation via w'. Having $u(x) = (w(x,0), w(x,1))$ and y, $\mathcal{B}$ can feed $\mathcal{A}$ with either $w(x,0)$ (if $y=0$) or $w(x,1)$ (if $y=1$). ■

Hence, combining Theorem 8.1 with Proposition 8.2 and Proposition 8.3, we get the following sufficient condition for noise-tolerant learnability.

Corollary 8.4 *If $\mathcal{F}$ is efficiently $O(\log n)$-wRFA learnable, then $\mathcal{F}$ is efficiently $O(\log n)$-wRFA learnable in the presence of classification noise.*

Thus, for instance, one can use the new k-wRFA learning algorithm for the class k-DL shown in Figure 7.1 to obtain a simple noise-tolerant algorithm for this class (compare with the algorithm and its analysis given in [34]).

Notice that the noise-tolerant algorithms obtained by the above transformations are randomized, even when the original ones are deterministic. (hence, the failure probability δ is with respect to the random sample and the random choices made by the algorithm). The derandomization schemes shown in [21] can be used to transform the randomized noise-tolerant algorithms into deterministic ones (preserving their efficiency).

Efficient $O(\log n)$-wRFA learnability can serve also as sufficient condition for efficient noise-tolerant PAC learnability. However, in this case we have to restrict every projection used by the RFA algorithm to be efficiently computable. We say that an RFA algorithm $\mathcal{A}$ is *strongly efficient* if it is efficient, and there is a polynomial $p(\cdot,\cdot,\cdot,\cdot)$, such that given the learning parameters ϵ, δ, n, and $size(f)$, every projection used by $\mathcal{A}$ is a function which is computable in time $p(\frac{1}{\epsilon},\frac{1}{\delta},n,size(f))$. This RFA restriction is equivalent to the one implied by Kearns' SQ model (the model of learning from statistical queries; see Section 3.4). Before proving this equivalence, we first give a formal definition of the SQ model.

Definition 8.5 (learning from statistical queries) *A statistical query is a pair (w,α), where $w : X \times \{0,1\} \rightarrow \{0,1\}$ is the query function, and $\alpha \in (0,1)$ is the query accuracy. An $SQ_{f,D}$ oracle is an oracle which, given a query (w,α), returns a value $\hat{p}_w$ which satisfies $|\hat{p}_w - p_w| < \alpha$, where $p_w = \Pr_{x\in D}[w(x,f(x)=1]$.*

The function class $\mathcal{F}$ is efficiently **SQ-learnable** *using the hypothesis class $\mathcal{H}$, if there is an algorithm $\mathcal{A}$, and there are polynomials $p(\cdot,\cdot,\cdot,\cdot)$, $q(\cdot,\cdot,\cdot,\cdot)$, and $r(\cdot,\cdot,\cdot,\cdot)$, such that for every function $f \in \mathcal{F}$, for every distribution D on X, and for every $0 < \epsilon, \delta \leq 1$, given inputs ϵ, δ, n and $size(f)$, and given access to an $SQ_{f,D}$ oracle, $\mathcal{A}$ satisfies:*

1. *For very query (w,α) asked by $\mathcal{A}$, w is computable in time $q(\frac{1}{\epsilon},\frac{1}{\delta},n,size(f))$, and $\frac{1}{\alpha} \leq r(\frac{1}{\epsilon},\frac{1}{\delta},n,size(f))$.*
2. *$\mathcal{A}$ halts in time $p(\frac{1}{\epsilon},\frac{1}{\delta},n,size(f))$, and outputs an ϵ-good hypothesis with confidence of at least $1-\delta$.*

Note that since the statistical oracle always gives a good estimation, any successful deterministic learner succeeds with absolute confidence. This setting is clearly sufficient for deterministic algorithms. However, since randomized algorithms can fail with non-zero probability, we include the confidence parameter δ in the definition of the SQ model. It is easy to verify that Kearns' transformation from efficient statistical algorithms into efficient noise-tolerant PAC algorithms is also applicable to randomized algorithms, implying that the following theorem is valid in that setting.

Theorem 8.6 (Kearns) *If $\mathcal{F}$ is efficiently SQ-learnable (using $\mathcal{H}$), then $\mathcal{F}$ is efficiently PAC learnable (using $\mathcal{H}$) in the presence of classification noise.*

As both the SQ restriction and the $O(\log n)$-wRFA restriction guarantee robustness against classification noise, it is not surprising that they are equivalent.

Theorem 8.7 *A function class $\mathcal{F}$ is learnable by a strongly efficient $O(\log n)$-wRFA algorithm iff it is efficiently learnable from statistical queries.*

Proof: First notice that if there is an efficient $O(\log n)$-wRFA algorithm $\mathcal{A}$ which learns $\mathcal{F}$, then by Corollary 7.3, there is an efficient 1-wRFA algorithm $\mathcal{B}$ which learns $\mathcal{F}$. Furthermore, the reduction shown in the proof of Theorem 7.2 preserves the strong efficiency of the learning algorithm. Hence, it is sufficient to prove the equivalence of strongly efficient 1-wRFA learnability and efficient SQ learnability.

Let $\mathcal{A}$ be a statistical algorithm which produces an ϵ-good hypothesis, with probability of at least $1-\frac{\delta}{2}$, by asking no more than $p(\frac{1}{\epsilon}, \frac{2}{\delta}, n, size(f))$ statistical queries, where each query function is computable in time $q(\frac{1}{\epsilon}, \frac{2}{\delta}, n, size(f))$, and the inverse of each query accuracy is bounded by $r(\frac{1}{\epsilon}, \frac{2}{\delta}, n, size(f))$. p, q, and r are polynomials in their arguments. Let W be the class of all the query functions which may be used by $\mathcal{A}$ during the learning process.

A 1-wRFA algorithm $\mathcal{B}$ can simulate each statistical query (w, α) by drawing a sample of size $O(\frac{1}{\alpha^2}\log\frac{2p}{\delta})$ via the projection w, and computing the fraction $\hat{p}_w$ of the positive observations in that sample. Having a sample of that size, it is implied by Chernoff Bounds (Theorem 4.6) that, with probability of at least $1-\frac{\delta}{2p}$, the estimate $\hat{p}_w$ is close to the true probability $p_w = \Pr[w=1]$, within an additive factor of α. Hence, $\mathcal{B}$ can simulate the statistical oracle and provide $\mathcal{A}$ with valid answers to its statistical queries, using a sample of size $O(\frac{p}{\alpha^2}\log\frac{2p}{\delta})$. Since $\frac{1}{\alpha} \leq r(\frac{1}{\epsilon}, \frac{2}{\delta}, n, size(f))$, and since p and r are polynomials, this sample complexity is polynomial (in ϵ^{-1}, δ^{-1}, n, and $size(f)$). Also notice that each projection used by $\mathcal{B}$ is computable in time $q(\frac{1}{\epsilon}, \frac{2}{\delta}, n, size(f))$, and therefore $\mathcal{B}$ is strongly efficient. With probability of at least $1-\frac{\delta}{2}$, all the estimates of the simulating algorithm $\mathcal{B}$ are sufficiently accurate, enabling $\mathcal{B}$ to produce an ϵ-good hypothesis with probability of at least $1-\delta$.

For the other direction, assume $\mathcal{A}$ learns $\mathcal{F}$ via a projection class $W \subseteq \{0,1\}^{X\times\{0,1\}}$. Assume further that $\mathcal{A}$ needs $m = m(\frac{1}{\epsilon}, \frac{2}{\delta}, n, size(f))$ observations, in order to find an ϵ-good hypothesis for f, with confidence of at least $1-\frac{\delta}{2}$, and that each projection used by $\mathcal{A}$ is computable in time $q(\frac{1}{\epsilon}, \frac{2}{\delta}, n, size(f))$,

A statistical algorithm $\mathcal{B}$ can learn $\mathcal{F}$ by feeding $\mathcal{A}$ with m observations as follows. To supply $\mathcal{A}$ with an observation via a projection w, $\mathcal{B}$ uses the statistical query $(w, \frac{\delta}{2m})$; if $\hat{p}_w$ is the estimate $\mathcal{B}$ receives for the probability $p_w = \Pr_{x\in D}[w(x, f(x)) = 1]$, then it draws the observation 1 with probability $\hat{p}_w$. Notice that the query function w is a projection which is efficiently computable, and that since m is polynomial, the inverse of the accuracy of this query is polynomial.

Assuming $\mathcal{A}$ chooses the sequence of projections $w_1, \ldots, w_m$, let P denote the product distribution $(p_{w_1}, \ldots, p_{w_m})$, and let $\hat{P}$ denote the product distribution $(\hat{p}_{w_1}, \ldots, \hat{p}_{w_m})$. Since $\Delta_s(p_{w_i}, \hat{p}_{w_i}) < \frac{\delta}{2m}$ for every $1 \leq i \leq m$, then by Proposition 4.9, $\Delta_s(P, \hat{P}) < \frac{\delta}{2}$. Hence, the distribution from which $\mathcal{B}$ draws the m observations is $\frac{\delta}{2}$-close to the real distribution, (induced by the target function f, the target distribution D, and the projections chosen by $\mathcal{A}$), implying that the failure probability of $\mathcal{B}$ does not exceed $\frac{\delta}{2} + \frac{\delta}{2} = \delta$. ■

A stronger variant of the SQ restriction [19] has been recently used to obtain a sufficient condition for gaining robustness in a harder noise setting—a noise model in which the attributes of the examples are also noisy. In that model it is assumed that, in addition to the corruption of the classification bit, each attribute bit is also corrupted with some probability ν, and the learning algorithm is allowed to be polynomial in $(1-2\nu)^{-1}$. Defining the *view* of an SQ algorithm to be the maximum number of input bits on which each query depends, the following is proven in [19].

Theorem 8.8 (Decatur, Gennaro) *If $\mathcal{F}$ is efficiently learnable from statistical queries with restricted view $O(\log n)$, then it is efficiently PAC learnable in the presence of both classification noise and attribute noise.*

By the proof scheme of Theorem 8.7, efficient learning from statistical queries with restricted view $O(\log n)$ and strongly efficient $O(\log n)$-RFA learnability are equivalent within polynomial factors. Therefore we get the following

Corollary 8.9 *If $\mathcal{F}$ is learnable by a strongly efficient $O(\log n)$-RFA algorithm, than $\mathcal{F}$ is efficiently learnable in the presence of both attribute and classification noise.*

Thus, for instance, the strongly efficient k-RFA learnability of the class of k-DNF formulas implies that the learning algorithm for this class can tolerate (simultaneously) both classification noise and attribute noise.

9 Discussion

Along the line of making learning theory applicable to realistic learning problems, this work has been concerned with the restrictions made on the visibility of the data which exists in each particular example. The main questions addressed in this work have been: when and how can learnability be made robust against such constraints?

We have formulate a general framework for studying this type of restriction. Making a distinction between the random instances which are distributed in the learning environment, and the observations which are made by the learner, this framework introduces a new mechanism which intermediates between the examples and the observations. Our results show that learnability is strongly related to the capability of this mechanism, suggesting that it is an important characterizing feature of realistic learning problems.

Applying this paradigm to Valiant's PAC model, we get a new setting, in which some of the traditional techniques are still applicable (with minor changes), but many others fail. In particular, the hardness results that have been shown in this work imply that learnability is no longer characterized by the VC dimension. Results in [12] show that the known boosting techniques are not applicable in this setting (i.e., weak learnability does not implies strong learnability). Also, it seems that many learning strategies strongly depend on the learner's ability of being able to observe all the relevant data existing in each example. Hence, new tools and techniques are needed; several were developed and exploited in the paper.

Two main variants of the restriction have been studied; in the first one, called k-RFA, only k of the n relevant attributes are revealed to the learner (which may also observe the classification bit). In the second one, called k-wRFA, the learner is supplied with k-bit observations (and no restriction is made on how they are extracted from the examples).

We have shown a characterization of RFA learnability, showing that RFA learnability via a set of projections W is impossible, if there is a set of different scenarios (where a scenario is a pair of function and distribution) which induce the same probabilistic behavior over every projection in W. This is a full characterization for a finite instance space. On the basis of this characterization we have shown a general constructions for proving hardness results for the k-RFA restriction. Thus, we proved, for instance, that the class of parity functions over $k+1$ boolean variables is not k-RFA learnable, implying that the k-RFA restriction induces a real hierarchy over the set of learnable class. These tools have also been used to prove that, for instance, for every $k < n-2$, the class of k-decision lists is not k-RFA learnable, and that the class of decision lists is not properly $(n-2)$-RFA learnable. We believe this scheme can be used further to obtain more hardness results in limited-visibility models.

For the k-RFA restriction, we defined a notion of decomposability of a class of function with respect to a given set of simpler classes, and show how to use the learnability of these classes to deduce the k-RFA learnability of the decomposable class. We applied the decomposability technique to show that the classes of functions represented by either k-DNF or k-CNF formulas are efficiently k-RFA learnable. Actually, we are not aware of richer boolean classes which are efficiently and distribution-free k-RFA learnable.

We have also shown information-theoretic learnability of richer classes, under a restricted class of distributions. Thus, under the uniform distribution, the class of k-decision lists is k-RFA learnable, and the class of boolean AC^0 circuits is $O(\log n)$-RFA learnable by a quasi-polynomial algorithm. Under product distributions in which each Boolean attribute has the same probability p of being 1, and $p \neq 1/2$, the class of all Boolean functions is $(n-2)$-RFA learnable, but not $\lfloor \frac{n}{8} \rfloor$-RFA learnable.

In the Euclidean n-dimensional space we have used the decomposability tool to demonstrate the efficient learnability of some simple geometric classes, such as axis-aligned rectangles and n-dimensional crosses. We also show that, under a product distribution, the class of n-dimensional subsets which are defined by mild $(n-1)$-dimensional surfaces is $(n-1)$-RFA learnable (efficiently for a fixed n).

For the k-wRFA restriction we have shown how to trade visibility for sample complexity, concluding that efficient learnability is indifferent to a reduction of $O(\log n)$ in the size of the observation space. The only lower bound on visibility we know for efficient k-wRFA learnability is a lower bound of $k = \omega(\log n)$ for the learnability of the class of all parity functions.

It also turns out that robustness against the RFA restrictions is strongly related to robustness against noisy examples. The $O(\log n)$-wRFA restriction, which is also equivalent to the statistical-query restriction, guarantees robustness against classification noise, while the more stringent $O(\log n)$-RFA restriction guarantees robustness against both attribute and classification noise (simultaneously). Thus, for instance, one can easily design an efficient learning algorithm, which learns the class of k-decision lists in the presence of classification noise, based on the simple k-wRFA algorithm presented in Section 8.2.

One of the most interesting open problems concerning limited-visibility learning 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 [17], or [15] 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 [12]), 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 (see [12]), we currently do not know how to boost weak approximators in the 1-RFA model.

Another interesting 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?

References

[1] Summary of the panel discussion. In *Proc. 1st Annu. Workshop on Comput. Learning Theory*, pages 398–401, 1988.

[2] Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.

[3] D. Angluin and P. Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1988.

[4] D. Angluin and L. G. Valiant. Fast probabilistic algorithms for hamiltonian circuits and matchings. *J. of Comput. Syst. Sci.*, 18(2):155–193, 1979.

[5] Javed A. Aslam and Scott E. Decatur. General bounds on statistical query learning and PAC learning with noise via hypothesis boosting. In *Proc. 34th Annu. IEEE Sympos. Found. of Comp. Sci.*, pages 282–291. IEEE Computer Society Press, Los Alamitos, CA, 1993.

[6] S. Ben-David and E. Dichterman. Learning with restricted focus of attention. In *Proc. 6th Annu. Conf. on Comput. Learning Theory*, pages 287–296. ACM Press, New York, NY, 1993.

[7] S. Ben-David and E. Dichterman. Learnability with restricted focus of attention guarantees noise-tolerance. In *Proc. 5th Int. Workshop on Algorithmic Learning Theory*, pages 248–259. Springer-Verlag, 1994.

[8] S. Ben-David, A. Itai, and E. Kushilevitz. Learning by distances. *Inform. Comput.*, 117(2):240–250, 1995.

[9] Shai Ben-David, Benny Chor, Oded Goldreich, and Michael Luby. On the theory of average case complexity. *J. of Comput. and Syst. Sci.*, 44(2):193–219, 1992.

[10] Shai Ben-David and Eli Dichterman. Learning with restricted focus of attention. In *Proc. 6th Annu. Conf. on Comput. Learning Theory*, pages 287–296. ACM Press, New York, NY, 1993.

[11] Shai Ben-David and Eli Dichterman. Learning with restricted focus of attention. Technical Report LSE-CDAM-97-01, London School of Economics, 1997.

[12] Andreas Birkendorf, Eli Dichterman, Jeffrey Jackson, Norbert Klasner, and Hans Ulrich Simon. On restricted-focus-of-attention learnability of Boolean functions. In *Proc. 28th Annu. ACM Sympos. Theory Comput.*, pages 205–216. ACM Press, New York, NY, 1996.

[13] A. Blum and P. Chalasani. Learning switching concepts. In *Proc. 5th Annu. Workshop on Comput. Learning Theory*, pages 231–242. ACM Press, New York, NY, 1992.

[14] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *J. ACM*, 36(4):929–965, 1989.

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

[16] Nicolò Cesa-Bianchi, Eli Dichterman, Paul Fischer, and Hans Ulrich Simon. Noise-tolerant learning near the informationn-theoretic bound. In *Proc. 28th Annu. ACM Sympos. Theory Comput.*, pages 141–150, May 1996.

[17] C.K. Chow. On the characterization of threshold functions. In *Proc. Symp. on Switching Circuit Theory and Logical Design*, pages 34–38, 1961.

[18] Scott E. Decatur and Rosario Gennaro. On learning from noisy and incomplete examples. In *Proc. 8th Annu. Conf. on Comput. Learning Theory*, pages 353–360, 1995.

[19] Scott E. Decatur and Rosario Gennaro. On learning from noisy and incomplete examples. In *Proc. 8th Annu. Conf. on Comput. Learning Theory*, pages 353–360. ACM Press, New York, NY, 1995.

[20] 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.*

[21] D. Haussler, M. Kearns, N. Littlestone, and M. K. Warmuth. Equivalence of models for polynomial learnability. *Inform. Comput.*, 95(2):129–161, December 1991.

[22] D. Helmbold, R. Sloan, and M. K. Warmuth. Learning integer lattices. *SIAM J. Comput.*, 21(2):240–266, 1992.

[23] Jeffrey C. Jackson. An efficient membership-query algorithm for learning DNF with respect to the uniform distribution. In *Proc. 35th Annu. IEEE Sympos. Found. of Comp. Sci.*, pages 42–53, 1994.

[24] 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.

[25] M. Kearns. Efficient noise-tolerant learning from statistical queries. In *Proc. 25th Annu. ACM Sympos. Theory Comput.*, pages 392–401. ACM Press, New York, NY, 1993.

[26] M. Kearns and M. Li. Learning in the presence of malicious errors. *SIAM J. Comput.*, 22:807–837, 1993.

[27] M. Kearns, M. Li, and L. Valiant. Learning boolean formulae. *Journal of the ACM*, 41(6):1298–1328, 1995.

[28] M. J. Kearns and R. E. Schapire. Efficient distribution-free learning of probabilistic concepts. *J. of Comput. and Syst. Sci.*, 48(3), 1994.

[29] 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.

[30] Michael J. Kearns, Robert E. Schapire, and Linda M. Sellie. Toward efficient agnostic learning. *Machine Learning*, 17(2/3):115–142, 1994.

[31] Philip D. Laird. Learning from good and bad data. In *Kluwer international series in engineering and computer science*. Kluwer Academic Publishers, Boston, 1988.

[32] Nathan Linial, Yishai Mansour, and Noam Nisan. Constant depth circuits, Fourier transform, and learnability. *J. ACM*, 40(3):607–620, 1993.

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

[34] Yasubumi Sakakibara. Noise-tolerant occam algorithms and their applications to learning decision trees. *Machine Learning*, 11:37–62, 1993.

[35] Robert E. Schapire. The strength of weak learnability. *Machine Learning*, 5:197–227, 1990.

[36] Robert E. Schapire. Learning probabilistic read-once formulas on product distributions. *Machine Learning*, 14(1):47–81, 1994.

[37] Eli Shamir and Clara Schwartzman. Learning by extended statistical queries and its relation to PAC learning. In *Computational Learning Theory: Eurocolt '95*, pages 357–366. Springer-Verlag, 1995.

[38] Hans U. Simon. Learning decision lists and trees with equivalence-queries. pages 322–336, 1995.

[39] R. H. Sloan. Corrigendum to Types of noise in data for concept learning. In *Proc. 5th Annu. Workshop on Comput. Learning Theory*, page 450. ACM Press, New York, NY, 1992.

[40] R. H. Sloan. Four types of noise in data for PAC learning. *Information Processing Letters*, 54:157–162, 1995.

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

[42] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probab. and its Applications*, 16(2):264–280, 1971.