Small Journal Name, 9, 275–302 (1992)

Toward Efficient Agnostic Learning

MICHAEL J. KEARNS mkearns@research.att.com
ROBERT E. SCHAPIRE schapire@research.att.com
AT&T Bell Laboratories, 600 Mountain Avenue, Murray Hill, NJ 07974-0636

LINDA M. SELLIE sellie@research.att.com
Department of Computer Science, University of Chicago, Chicago, IL 60637

Editor: Lisa Hellerstein

Abstract. In this paper we initiate an investigation of generalizations of the Probably Approximately Correct (PAC) learning model that attempt to significantly weaken the target function assumptions. The ultimate goal in this direction is informally termed *agnostic learning*, in which we make virtually no assumptions on the target function. The name derives from the fact that as designers of learning algorithms, we give up the belief that Nature (as represented by the target function) has a simple or succinct explanation. We give a number of positive and negative results that provide an initial outline of the possibilities for agnostic learning. Our results include hardness results for the most obvious generalization of the PAC model to an agnostic setting, an efficient and general agnostic learning method based on dynamic programming, relationships between loss functions for agnostic learning, and an algorithm for a learning problem that involves hidden variables.

Keywords: machine learning, agnostic learning, PAC learning, computational learning theory

1. Introduction

One of the major limitations of the Probably Approximately Correct (or PAC) learning model (Valiant, 1984) (and related models) is the strong assumptions placed on the so-called *target function* that the learning algorithm is attempting to approximate from examples. While such restrictions have permitted a rigorous study of the computational complexity of learning as a function of the representational complexity of the target function, the PAC family of models diverges from the setting typically encountered in practice and in empirical machine learning research. Empirical approaches often make few or no assumptions on the target function, but search a limited space of hypothesis functions in an attempt to find the "best" approximation to the target function; in cases where the target function is too complex, even this best approximation may incur significant error.

In this paper we initiate an investigation of generalizations of the PAC model in an attempt to significantly weaken the target function assumptions whenever possible. Our ultimate goal is informally termed *agnostic learning*,[1] in which we make virtually no assumptions on the target function. We use the word "agnostic" — whose root means literally "not known" — to emphasize the fact that as designers of learning algorithms, we may have no prior knowledge about the target function. It is important to note that in this paper we make no attempt to remove the assumption of statistical independence between the examples seen by a learning algorithm, another worthwhile research direction that has been pursued by a number of authors (Aldous & Vazirani, 1990; Helmbold & Long, 1994). .

This paper describes a preliminary study of the possibilities and limitations for efficient agnostic learning. As such, we do not claim to have a definitive model but instead use a rather general model (based on the work of Haussler (1992)) that allows easy consideration of many natural modifications. Perhaps not surprisingly in light of evidence from the standard PAC model, efficient agnostic learning in its purest form (no assumptions on target function or distribution) is hard to come by, as some of our results will demonstrate. Thus, we will consider several variations of these perhaps overly ambitious criteria in an attempt to find positive results with target assumptions that are at least significantly weakened over the standard PAC setting.

There are several prior studies of weakened target assumptions for PAC learning that are relevant to our work. The first is due to Haussler (1992) who describes a powerful generalization of the standard PAC model based on decision theory and uniform convergence results. Haussler's results are of central importance to much of the research described here. Indeed, the agnostic model that we describe is quite similar to Haussler's, differing only in the introduction of a "touchstone" class (see Section 2). However, while Haussler's concern is exclusively on the information-theoretic and statistical issues in agnostic learning, we are here concerned almost exclusively with efficient computation. Also relevant is the large body of research on nonparametric density estimation in the field of statistics (see, for instance, Izenman's (1991) excellent survey).

Another relevant investigation is the work on probabilistic concepts of Kearns and Schapire (1990), as well as the work of Yamanishi (1992a) on stochastic rules. Here, the target function is a conditional probability distribution, typically on a discrete range space, such as $\{0,1\}$. A significant portion of the research described in this paper extends this work. Some of the results presented are also closely related to the work of Pitt and Valiant on *heuristic* learning (Pitt & Valiant, 1988; Valiant, 1985), which can be viewed as a variant of our agnostic PAC model.

The following is a brief overview of the paper: in Section 2 we motivate and develop in detail the general learning framework we will use. In Section 3 we consider the restriction of this general model to the case of agnostic PAC learning and give strong evidence for the intractability of even rather simple learning problems in this model. In Section 4 we discuss the empirical minimization of loss and give a general method for agnostic learning of "piecewise" functions that is based on dynamic programming. Section 5 gives a useful relationship in the agnostic setting between two common loss functions, the quadratic and prediction loss, and gives applications of this relationship. In Section 6 we investigate a compromise between agnostic learning and the strong target assumptions of the standard PAC model by providing an efficient learning algorithm in a model for learning problems involving hidden variables. Finally, in Section 7, we list a few of the many problems that remain open in this area.

2. Definitions and models

In this section we define our notation and the generalized framework we will use in our attempt to weaken the target function assumptions needed for efficient learning. Our approach is strongly influenced by the decision-theoretic learning model that was introduced to the computational learning theory community by Haussler (1992). In giving our definitions, we err on the side of formality — in order to lay the groundwork for future research on agnostic learning, we wish to give a model that is both precise and quite general. For most of the paper, however, we will be using various restrictions of this general model that will be locally specified using less cumbersome notation.

Let X be a set called the *domain*; we refer to points in X as *instances*, and we intuitively think of instances as the inputs to a "black box" whose behavior we wish to learn or to model. Let Y' be a set called the *range*, and let Y be a set called the *observed range*. We think of Y' as the space of possible values that might be output by the black box; however, we introduce Y because we may not have direct access to the output value, but only to some quantity derived from it. In general, we make no assumptions about the relationship between Y and Y'. We call a pair $(x, y) \in X \times Y$ an *observation*.

2.1. The assumption class $\mathcal{A}$

The *assumption class* $\mathcal{A}$ is a class of probability distributions on the observation space $X \times Y$. We use $\mathcal{A}$ to represent our assumptions on the phenomenon we are trying to learn or model, and the nature of our observations of this phenomenon. Note that in this definition of $\mathcal{A}$, there may be no functional relationship between x and y in an observation (x, y). However, there are two special cases of this generalized definition that we wish to define.

In the first special case, there *is* a functional relationship, and an *arbitrary* domain distribution. Thus, consider the case where $Y = Y'$ and there is a class of functions $\mathcal{F}$ mapping X to Y'. Suppose $\mathcal{A}$ is the class obtained by choosing any distribution D over X and any $f \in \mathcal{F}$, and letting $A_{D,f} \in \mathcal{A}$ be the distribution generating observations $(x, f(x))$, where x is drawn randomly from D. Then we say that $\mathcal{A}$ is the *functional decomposition using* $\mathcal{F}$, and we have a familiar distribution-free function learning model.

In the second special case, we have $Y' = [0, 1]$, $Y = \{0, 1\}$ and there is again a class of functions $\mathcal{F}$ mapping X to Y'. Now, however, the functional value is not directly observed. Instead, let $\mathcal{A}$ be the class obtained by choosing any distribution D over X and any $f \in \mathcal{F}$, and letting $A_{D,f} \in \mathcal{A}$ be the distribution generating observations (x, b), where x is drawn randomly from D and $b = 1$ with probability

$f(x)$ and $b = 0$ with probability $1 - f(x)$. We call $\mathcal{F}$ a class of *probabilistic concepts* (or *p-concepts*), and we say that $\mathcal{A}$ is the *p-concept decomposition using* $\mathcal{F}$. Here we have a distribution-free p-concept learning model.

In the case that $\mathcal{A}$ is either the functional or p-concept decomposition using a class $\mathcal{F}$, we refer to $\mathcal{F}$ as the *target class*, and if the distribution $A_{D,f} \in \mathcal{A}$ generates the observations we call f the *target function* or *target p-concept* and D the *target distribution.*

2.2. The hypothesis class $\mathcal{H}$ and the touchstone class $\mathcal{T}$

We next introduce two classes of functions from X to Y': the *hypothesis class* $\mathcal{H}$, and the *touchstone class* $\mathcal{T}$. Usually it will be the case that $\mathcal{T} \subseteq \mathcal{H}$. The intuition is that a learning algorithm will attempt to model the behavior from $\mathcal{A}$ that it observes with a *hypothesis function* $h \in \mathcal{H}$. In our model, where we seek to eliminate restrictions on $\mathcal{A}$ as much as possible, we must ask against what standard the hypothesis function will be measured, since nearness to the target may be impossible or undefined. This is the purpose of the touchstone class $\mathcal{T}$. This class provides a standard of measurement for hypotheses, and we will ask that the performance of the hypothesis $h \in \mathcal{H}$ be "near" the performance of the "best" $t \in \mathcal{T}$, where "near" and "best" will be formalized shortly. Although it seems natural to ask that the hypothesis chosen approach the best performance in the class $\mathcal{H}$ (corresponding to the case $\mathcal{T} = \mathcal{H}$), we will see that in some circumstances it is interesting and important to relax this restriction. By leaving the class $\mathcal{T}$ *fixed* and increasing the power of $\mathcal{H}$, we may overcome certain representational hurdles presented by the choice $\mathcal{T} = \mathcal{H}$, in the same way that k-term DNF (disjunctive normal form) formulas are efficiently learnable in the standard PAC model provided we allow the more expressive k-CNF (conjunctive normal form) hypothesis representation (Kearns, Li, Pitt & Valiant, 1987; Pitt & Valiant, 1988).

2.3. The loss function L

Now we formalize the possible meanings of the "best" function in a class. Given the domain X, the range Y', and the observed range Y, a *loss function* is a mapping $L : Y' \times Y \to [0, M]$ for some positive real number M. Given an observation $(x, y) \in X \times Y$ and a function $h : X \to Y'$, the *loss of h on (x, y)* is denoted $L_h(x, y) = L(h(x), y)$. The loss function measures the "distance" or discrepancy between $h(x)$ and the observed value y. Typical examples include the *prediction loss* (also known as the *discrete loss*), where

$$Z(y', y) = \begin{cases} 0 & \text{if } y' = y \\ 1 & \text{if } y' \neq y \end{cases}$$

and the *quadratic loss*

$$Q(y', y) = (y' - y)^2.$$

Since observations are drawn according to a distribution $A \in \mathcal{A}$, we can define the *expected loss* $\mathbf{E}_{(x,y)\in A}[L_h(x,y)]$ of the function h, which we abbreviate $\mathbf{E}[L_h]$ when A is clear from the context. Now we are prepared to define the best possible performance in a class of functions with respect to the loss function L. For the hypothesis class $\mathcal{H}$, we define $opt(\mathcal{H}) = inf_{h\in\mathcal{H}}\{\mathbf{E}[L_h]\}$. Similarly, for the touchstone class $\mathcal{T}$, we define $opt(\mathcal{T}) = inf_{t\in\mathcal{T}}\{\mathbf{E}[L_t]\}$. Note that $opt(\mathcal{H})$ and $opt(\mathcal{T})$ have an implicit dependence on $A \in \mathcal{A}$ that we omit for notational brevity.

We will often need to refer to estimates of these quantities from empirical data. Thus, if S is a sequence of observations, we can estimate $\mathbf{E}[L_h]$ by

$$\hat{\mathbf{E}}_S[L_h] = \frac{1}{|S|} \cdot \sum_{(x,y)\in S} L_h(x,y).$$

This allows us to define the estimated optimal performance for $\mathcal{H}$ and $\mathcal{T}$, defined by $\hat{opt}_S(\mathcal{H}) = inf_{h\in\mathcal{H}}\{\hat{\mathbf{E}}_S[L_h]\}$ and $\hat{opt}_S(\mathcal{T}) = inf_{t\in\mathcal{T}}\{\hat{\mathbf{E}}_S[L_t]\}$. Usually S will be clear from the context, and we will write $\hat{\mathbf{E}}[L_f]$, $\hat{opt}(\mathcal{H})$ and $\hat{opt}(\mathcal{T})$.

2.4. The learning model

We are now ready to give our generalized definition of learning.

Definition. Let X be the domain, let Y' be the range, let Y be the observed range, and let $L : Y' \times Y \to [0, M]$ be the loss function. Let $\mathcal{A}$ be a class of distributions on $X \times Y$, and let $\mathcal{H}$ and $\mathcal{T}$ be classes of functions mapping X to Y'. We say that $\mathcal{T}$ *is learnable by* $\mathcal{H}$ *assuming* $\mathcal{A}$ *(with respect to L)* if there is an algorithm *Learn* and a function $m(\epsilon, \delta)$ that is bounded by a fixed polynomial in $1/\epsilon$ and $1/\delta$ such that for any distribution $A \in \mathcal{A}$, and any inputs $0 < \epsilon, \delta \leq 1$, *Learn* draws $m(\epsilon, \delta)$ observations according to A, halts and outputs a hypothesis $h \in \mathcal{H}$ that with probability at least $1 - \delta$ satisfies $\mathbf{E}[L_h] \leq opt(\mathcal{T}) + \epsilon$. If the running time of *Learn* is bounded by a fixed polynomial in $1/\epsilon$ and $1/\delta$, we say that $\mathcal{T}$ *is efficiently learnable by* $\mathcal{H}$ *assuming* $\mathcal{A}$ *(with respect to L).*

In the case that $\mathcal{A}$ is the functional decomposition using a class $\mathcal{F}$, we replace the phrase "assuming $\mathcal{A}$" with the phrase "assuming the function class $\mathcal{F}$"; in the case that $\mathcal{A}$ is the p-concept decomposition using a class $\mathcal{F}$, we replace it with the phrase "assuming the p-concept class $\mathcal{F}$." If we wish to indicate that the touchstone class $\mathcal{T}$ is learnable by *some* $\mathcal{H}$ assuming $\mathcal{A}$ without reference to a specific $\mathcal{H}$, we will say $\mathcal{T}$ *is (efficiently) learnable assuming* $\mathcal{A}$.

There will often be a natural *complexity parameter n* associated with the domain X, the distribution class $\mathcal{A}$ and the function classes $\mathcal{H}$ and $\mathcal{T}$, in which case it will be understood that $X = \bigcup_{n\geq 1} X_n$, $\mathcal{A} = \bigcup_{n\geq 1} \mathcal{A}_n$, $\mathcal{H} = \bigcup_{n\geq 1} \mathcal{H}_n$, and $\mathcal{T} = \bigcup_{n\geq 1} \mathcal{T}_n$. Standard examples for n are the number of boolean variables or the number of real dimensions. In these cases, we allow the number of observations and the running time of the algorithm in Definition 2.4 to also have a polynomial dependence on n.

2.5. Generating some old and new models

We now define several previously studied and new models of learning by appropriate settings of the parameters $\mathcal{A}$, $\mathcal{H}$, $\mathcal{T}$ and L.

First of all, if $\mathcal{F}$ is any class of boolean functions, $\mathcal{A}$ is the functional decomposition using $\mathcal{F}$, $\mathcal{H} = \mathcal{T} = \mathcal{F}$, and L is the prediction loss function Z, then we obtain the *restricted PAC model* (Valiant, 1984), where the hypothesis class is the same as the target class. If we retain the condition $\mathcal{T} = \mathcal{F}$ but allow $\mathcal{H} \supseteq \mathcal{F}$, we obtain the *standard PAC model* (Kearns et al., 1987), where the hypothesis class may be more powerful than the target class.

Next, if $\mathcal{A}$ is the p-concept decomposition using a class $\mathcal{F}$ of p-concepts, $\mathcal{T} = \mathcal{F}$, and $\mathcal{H} \supseteq \mathcal{F}$, then we obtain the *p-concept learning model* (Kearns & Schapire, 1990), and there are at least two interesting choices of loss functions. If we choose the prediction loss function Z then we ask for the optimal *predictive* model for the $\{0,1\}$ *observations* (also known as the *Bayes optimal decision*), which may be quite different from the actual *probabilities* given by $f \in \mathcal{F}$. This rule has the minimum probability of incorrectly predicting the y-value of a random observation, given the observation's x-value. Alternatively, we may choose the quadratic loss function Q. Here it is known that the quadratic loss will lead us to find a hypothesis h minimizing the quadratic distance between f and h, i.e., $\mathbf{E}[(f-h)^2]$ (Kearns & Schapire, 1990; White, 1989).

Now consider the following generalization of the standard PAC model: let $\mathcal{F}$ be the class of all boolean functions over the domain X, and let $\mathcal{A}$ be the functional decomposition using $\mathcal{F}$. Thus we remove *all* assumptions on the target concept (except the existence of *some* concept consistent with the data). Now if we let $\mathcal{H} = \mathcal{T}$, and choose the prediction loss function Z, then we wish to find a good predictive concept in $\mathcal{H}$ regardless of the nature of the target concept. We will refer to this particular choice of the parameters as the *agnostic PAC model.*

3. Agnostic PAC learning

In this section we examine the agnostic PAC model. Our main results here demonstrate relationships between the agnostic PAC model and some other previously studied variations of the standard PAC model, and provide a strong argument for the need for further restrictions or different models if we wish learning algorithms to be efficient. Related results, also indicating intractability for learning with weakened target concept assumptions, are given by Valiant (1985) and Pitt and Valiant (1988) for a model of *heuristic learning.*

3.1. Agnostic learning and malicious errors

Our first result shows that agnostic PAC learning is at least as hard as PAC learning with malicious errors (Kearns & Li, 1993; Valiant, 1985) (and in fact, a partial converse holds as well). Although we will not formally define the latter model, it is equivalent to the standard PAC model with the addition of a new parameter called the *error rate* β, and now each observation has probability β of being generated by a malicious adversary rather than by the target function and target distribution. The goal in the malicious error model remains that of achieving an arbitrarily good predictive approximation to the underlying target function.

THEOREM 1 *Let $\mathcal{T}$ be a class of boolean functions over X that is efficiently learnable in the agnostic PAC model, and assume that the Vapnik-Chervonenkis dimension of $\mathcal{T}$ is bounded by a polynomial in the complexity parameter n. Then $\mathcal{T}$ is efficiently learnable (using $\mathcal{T}$) in the PAC model by an algorithm tolerating a malicious error rate of $\beta = \Theta(\epsilon)$.*

Proof: The idea is to demonstrate the equivalence of the problem of learning $\mathcal{T}$ in the agnostic PAC model and a natural combinatorial optimization problem based on $\mathcal{T}$, the *disagreement minimization problem* for $\mathcal{T}$, a problem known to be equivalent (up to constant approximation factors) to the problem of learning with malicious errors (Kearns & Li, 1993). In this problem, we are given as input an arbitrary multiset $S = \{(x_1, b_1), \ldots, (x_m, b_m)\}$ of pairs, where $x_i \in X$ and $b_i \in \{0, 1\}$ for all $1 \leq i \leq m$. The correct output for the instance S is the $h^* \in \mathcal{T}$ that minimizes $d_S(h) = |\{i : h(x_i) \neq b_i\}|$ over all $h \in \mathcal{T}$.

It follows from standard arguments (Blumer, Ehrenfeucht, Haussler & Warmuth, 1989) that if the Vapnik-Chervonenkis dimension of $\mathcal{T}$ is polynomially bounded by the complexity parameter n, an algorithm that efficiently solves the disagreement minimization problem for $\mathcal{T}$ can be used as a subroutine by an efficient algorithm for learning $\mathcal{T}$ in the agnostic PAC model. (See Section 4.1 for more details.)

For the other direction of the equivalence, suppose we have an algorithm for efficiently learning $\mathcal{T}$ in the agnostic PAC model, and wish to use this algorithm in order to solve the disagreement minimization problem for $\mathcal{T}$ on a fixed instance S. We first give the argument assuming that no instance x_i appears with two different labels in S; thus, the pairs of S may be thought of as being consistent with a boolean function f, where $f(x_i) = b_i$ for each $1 \leq i \leq n$.

Let us create the distribution D on the instances x_i in the multiset S, giving equal weight $1/m$ to each instance (instances appearing more than once in S will receive proportionally more weight, and instances outside S receive zero weight). We run the agnostic learning algorithm, choosing $\epsilon < 1/m$, and drawing instances from D and labeling them according to the target function f (note that this is equivalent to simply drawing *labeled* pairs randomly from S). The algorithm must then output a hypothesis $h \in \mathcal{T}$ that satisfies

$$\mathbf{Pr}[h \neq f] \leq \mathbf{Pr}[h^* \neq f] + \epsilon < \mathbf{Pr}[h^* \neq f] + \frac{1}{m}$$

where h^* minimizes $d_S(h)$ over all $h \in \mathcal{T}$. This implies $\mathbf{Pr}[h \neq f] = \mathbf{Pr}[h^* \neq f]$ because a single disagreement with f incurs error $1/m$ with respect to D. Since for any h we have $\mathbf{Pr}[h \neq f] = d_S(h)/m$, we have $d_S(h) = d_S(h^*)$, and our optimization problem is solved.

In the case that S contains conflicting labels for some instance and thus is not consistent with any function, we can simply remove from S all pairs of conflicting instances $(x_i, 0)$ and $(x_i, 1)$ until the remaining multiset S' is consistent with a function. Notice that any function disagrees with exactly half of $S - S'$, and thus minimization of $d_S(h)$ reduces to minimization of $d_{S'}(h)$. We now simply perform the above reduction on S'.

Finally, the desired algorithm for learning in the malicious error models follows from the above equivalence of agnostic learning and disagreement minimization, and an equivalence up to constant approximation factors between disagreement minimization and learning $\mathcal{T}$ in the restricted PAC model with malicious errors, a fact proved by Kearns and Li (1993, Theorem 19). In fact, this latter equivalence can be used to obtain a weakened converse to Theorem 1: learning $\mathcal{T}$ with malicious error rate $\beta = \Theta(\epsilon)$ implies an algorithm finding an $h \in \mathcal{T}$ satisfying $\mathbf{Pr}[h \neq f] \leq c \cdot opt(\mathcal{T})$ for some constant c (a weaker multiplicative rather than additive error bound). □

Although there are a number of variations of agnostic PAC learning that may not be directly covered by Theorem 1, we essentially interpret the result as negative evidence for hopes of efficient agnostic PAC learning algorithms, because previous results indicate that a $\Theta(\epsilon)$ malicious error rate can be achieved for only the most limited classes $\mathcal{T}$ (Kearns & Li, 1993) (such as the class of symmetric functions on n boolean variables).

Other results for agnostic PAC learning may be obtained via Theorem 1 and the previous work on learning in the presence of malicious errors. For instance, if $\mathcal{T}$ is any class of boolean functions, and $\mathcal{T}$ is (efficiently) learnable in the error-free PAC model, then there is an (efficient) algorithm for finding $h \in \mathcal{T}$ satisfying $\mathbf{Pr}[h \neq f] \leq O(d_{\mathcal{H}} \cdot opt(\mathcal{T}))$ where f is the target function and $d_{\mathcal{H}}$ is the Vapnik-Chervonenkis dimension of the hypothesis class $\mathcal{H}$ (this follows from Theorems 11 and 19 of Kearns and Li (1993).)

3.2. Intractability of agnostic PAC learning of conjunctions

Now we give a reduction indicating the difficulty of learning simple boolean conjunctions in the agnostic PAC model. If we let $X_n = \{0,1\}^n$ and set $\mathcal{T}_n = \mathcal{H}_n$ to be the class of all conjunctions of literals over the boolean variables $x_1, \ldots, x_n$, then in the agnostic PAC model we wish to find an algorithm that can find a conjunction in $\mathcal{T}_n$ that has a near-minimum rate of disagreement with an unknown boolean target function f. We can show this problem to be hard even for rather restricted f:

THEOREM 2 *Let $X_n = \{0,1\}^n$, and let $\mathcal{F}_n$ be the class of polynomial-size disjunctive normal form formulas over $\{0,1\}^n$. Let $\mathcal{T}_n$ be the class of conjunctions of* *literals over the boolean variables $x_1, \ldots, x_n$. Then $\mathcal{T}$ is not efficiently learnable using $\mathcal{T}$ assuming the function class $\mathcal{F}$, unless RP = NP.*

Proof: Suppose to the contrary of the theorem's statement that there exists an efficient algorithm for the stated learning problem. We show how such an algorithm can be used probabilistically to solve the minimum set cover problem (Garey & Johnson, 1979) in polynomial time, thus implying that RP = NP. A similar proof is given in the context of PAC learning with malicious errors by Kearns and Li (1993), and can be used with Theorem 1 to obtain a similar but weaker result than the one we now derive.

An instance of the minimum set cover problem is a set of objects $O = \{o_1, \ldots, o_t\}$ to be covered, and a collection of subsets of the objects $\mathcal{S} = \{S_1, \ldots, S_n\}$. The goal is to find the smallest subset $\mathcal{S}' \subseteq \mathcal{S}$ that covers all objects (so that for all $o_i \in O$, there exists $S_j \in S'$ such that $o_i \in S_j$).

Without loss of generality, we will assume that all objects o_i are contained in more than one set. Without loss of generality, we also assume that all objects are contained in a unique collection of sets: if two objects are contained in exactly the same sets, we remove one of the objects and any valid set cover will cover the removed object.

The reduction chooses the target function to be the n-term DNF formula $f = T_1 \vee \ldots \vee T_n$ over the variable set $\{x_1, \ldots, x_n\}$, where T_i is the conjunction of all variables except x_i. All instances given to the learning algorithm will be labeled according to f.

For each object o_i, $1 \leq i \leq t$, let a_i be the assignment $\langle a_{i1}, \ldots, a_{in} \rangle$ of values to the n boolean variables (so that x_j is assigned a_{ij}) where we define

$$a_{ij} = \begin{cases} 0 & \text{if } o_i \in S_j \\ 1 & \text{otherwise.} \end{cases}$$

By this construction $f(a_i) = 0$ for all i: since every object is in at least two sets, at least two positions of a_i are zero, and therefore a_i does not satisfy any term in f. Thus, the a_i will be the negative examples.

For each set S_j, $1 \leq j \leq n$, let b_j be the assignment $\langle b_{j1}, \ldots, b_{jn} \rangle$ where

$$b_{jk} = \begin{cases} 0 & \text{if } j = k \\ 1 & \text{otherwise.} \end{cases}$$

Finally let $c = \langle 1, \ldots, 1 \rangle$. Note that $f(b_j) = f(c) = 1$ since b_j satisfies exactly one term in f and c satisfies all terms.

Notice that for each variable x_j, if we choose to include x_j in a monotone conjunction then this conjunction is guaranteed to "cover" (that is, have as negative examples) all a_i such that object o_i appears in set S_j. Further, including x_j in a conjunction incurs the single error b_j on the positive examples. Thus, our goal is to force the agnostic learning algorithm to cover all the negative examples (corresponding to covering all of the objects) while incurring the least positive error (corresponding to a minimum cardinality cover).

The distribution we will use is defined by

$$\begin{aligned} D(a_i) &= \frac{1}{2(t+1)} + \frac{1}{4t(t+1)} \\ D(b_j) &= \frac{1}{4n(t+1)} \\ D(c) &= \frac{1}{2} \end{aligned}$$

and $D(x) = 0$ for all other x. Finally, we set $\epsilon = 1/8n(t+1)$, and we run the assumed agnostic learning algorithm using examples drawn according to D and labeled according to f. Clearly, this entire procedure takes time polynomial in the size of the set cover instance (since the target DNF f is only of polynomial size). Moreover, with high probability, we obtain a conjunction h having error bounded by $opt(\mathcal{T}) + \epsilon$ with respect to f and D.

Let $\mathcal{B} = \{S_j \mid x_j \text{ appears in } h\}$. We first show that $\mathcal{B}$ is a cover.

Note that the conjunction of all variables, $x_1 \cdots x_n$, has error equal to $1/4(t+1)$, since it is consistent with f on c and a_i for all i. Thus $opt(\mathcal{T}) \leq 1/4(t+1)$, which implies that

$$opt(\mathcal{T}) + \epsilon \leq \frac{1}{4(t+1)} + \frac{1}{8n(t+1)} < \frac{1}{2(t+1)}.$$

The conjunction h must be monotone, since otherwise it would be inconsistent with the positive example $c = \langle 1, \ldots, 1 \rangle$ giving an error of at least $1/2$. Also, h must be consistent with all the negative instances a_i, since otherwise its error would be at least $1/2(t+1) + 1/4t(t+1)$. Thus $\mathcal{B}$ covers all objects, since for every a_i there is a variable x_j in h that forces a_i to be negative, and this happens only if S_j includes o_i.

It remains to show that $\mathcal{B}$ is a minimum cover. Suppose there exists a smaller set cover $\mathcal{B}'$. Then we can construct a monomial h' from $\mathcal{B}'$ where x_j is in h' if and only if $S_j \in \mathcal{B}'$. By construction h' is monotone so it is consistent with instance c. Because $\mathcal{B}'$ is a set cover, h' is consistent with a_i for all i. For each $S_j \in \mathcal{B}'$, $h'(b_j) = 0$; thus h' is not consistent with $|\mathcal{B}'|$ elements b_j. Therefore, $opt(\mathcal{T}) \leq \mathbf{Pr}[f \neq h'] = |\mathcal{B}'|/4n(t+1)$. On the other hand, $\mathbf{Pr}[f \neq h] = |\mathcal{B}|/4n(t+1)$ which implies that

$$\mathbf{Pr}[f \neq h] \geq opt(\mathcal{T}) + \frac{|\mathcal{B}| - |\mathcal{B}'|}{4n(t+1)} > opt(\mathcal{T}) + \epsilon,$$

by our choice of ϵ, contradicting the assumption that h has error bounded by $opt(\mathcal{T}) + \epsilon$. Therefore $\mathcal{B}$ is indeed a minimum set cover. □

Thus, even if we assume that the target distribution can be functionally decomposed into a distribution on X and a target function that is guaranteed to be a small DNF formula, it is a hard problem to find a conjunction whose predictive power is within a small additive factor of the best conjunction. Even more surprising, Theorem 2 holds even if the learning algorithm is *told* the target DNF formula! This demonstrates an important principle: having a perfect and succinct description of the process generating the observations may not help in finding an even more succinct "rule of thumb" that tolerably explains the observations. Thus the difficulty may arise not so much from the problem of *learning* but from that of *optimization.*

Similar results are given by Valiant (1985) and Pitt and Valiant (1988).

3.3. Agnostic learning and weak learning

We next describe a connection between agnostic PAC learning and weak PAC learning (in which the standard PAC criterion is relaxed to demand hypotheses whose error with respect to the target is bounded only by $1/2-1/p(n)$ for some polynomial $p(n)$ of the complexity parameter (Kearns & Valiant, 1994; Schapire, 1990).)

If $\hat{\mathcal{T}}$ and $\mathcal{T}$ are two classes of boolean functions over a domain X parameterized by n, we say that $\hat{\mathcal{T}}$ *weakly approximates* $\mathcal{T}$ if there is a polynomial $p(n)$ such that for any distribution D on X_n and any $t \in \mathcal{T}_n$ there is a function $\hat{t} \in \hat{\mathcal{T}}_n$ such that $\mathbf{Pr}_{x\in D}[\hat{t}(x) \neq t(x)] \leq 1/2 - 1/p(n)$.

THEOREM 3 *Let $\hat{\mathcal{T}}$ be a class of boolean functions that weakly approximates a class $\mathcal{T}$. Then $\mathcal{T}$ is efficiently learnable in the standard PAC model if $\hat{\mathcal{T}}$ is efficiently learnable in the agnostic PAC model.*

Proof: The idea is that since $\hat{\mathcal{T}}$ weakly approximates $\mathcal{T}$, whenever the target function is from $\mathcal{T}$, $opt(\hat{\mathcal{T}})$ will be significantly smaller than $1/2$, and the agnostic learning algorithm effectively functions as a weak learning algorithm for $\mathcal{T}$. The result then follows from the "boosting" techniques of Schapire (1990) or Freund (1990; 1992) for converting a weak learning algorithm into a strong learning algorithm. □

Since the class of boolean conjunctions weakly approximates the class of polynomial-size DNF formulas (see, for instance, Schapire (1990, Section 5.3)), it immediately follows from Theorem 3 that learning conjunctions in the agnostic PAC model is at least as hard as learning DNF formulas in the standard PAC model; this can be interpreted as further evidence for the difficulty of the problem, based on the assumption that learning DNF is hard in the standard PAC model. Note that unlike Theorem 2 (where we must set $\mathcal{H} = \mathcal{T}$), this result makes no restrictions on $\mathcal{H}$.

In summary, we see that agnostic PAC learning is intimately related to a number of apparently difficult problems in the standard PAC model. This leads us to two preliminary conclusions: that we should look for efficient agnostic learning in other models and with respect to other loss functions, and that we may want to consider *some* restrictions on the assumption class without reverting to the standard PAC model.

4. Tractable agnostic learning problems

Although the results of Section 3 indicate that our prospects of finding efficient agnostic PAC learning algorithms may be bleak, we demonstrate in this section that at least in some non-trivial situations, efficient agnostic learning is in fact tractable. We give a learning method based on dynamic programming applicable to our general learning framework.

4.1. Empirical loss minimization and agnostic learning

One natural technique for designing an agnostic learning algorithm is to first draw a large random sample, and to then find the hypothesis that best fits the observed data. In fact, this canonical approach successfully yields an efficient agnostic learning algorithm in a wide variety of settings, assuming that there exists an efficient algorithm for finding the best hypothesis (with respect to the observed sample).

In this section, we will not make any assumptions on the distributions in $\mathcal{A}$, and will use the expression $\mathcal{T}$ *is agnostically learnable using* $\mathcal{H}$ to indicate that a hypothesis in $\mathcal{H}$ near the best in $\mathcal{T}$ can be found (dropping the reference to $\mathcal{H}$ to indicate that $\mathcal{T}$ is agnostically learnable using some class $\mathcal{H}$).

Let Y be our observed range, let $\mathcal{T}$ and $\mathcal{H}$ be the touchstone and hypothesis classes of functions mapping X into Y', and let L be the loss function. We say that $\mathcal{T}$ is *(efficiently) empirically minimizable* by $\mathcal{H}$ (with respect to L) if there exists a (polynomial-time) algorithm that, given a finite sample $S \in (X \times Y)^*$, computes a hypothesis $h \in \mathcal{H}$ whose empirical loss on S is optimal compared to $\mathcal{T}$; that is, $\hat{\mathbf{E}}_S[L_h] \leq \hat{opt}_S(\mathcal{T})$. (Here, polynomial time means polynomial in the size of the sample S.)

For instance, if $Y \subseteq \mathbb{R}$, and $\mathcal{T}$ is the class of constant real-valued functions on X, then $\mathcal{T}$ is efficiently empirically minimizable with respect to the quadratic loss function since the average of the Y-values observed in S minimizes the empirical loss. More generally, if $f_1, \ldots, f_d$ is a set of d real-valued basis functions on X, then standard regression techniques can be used to efficiently minimize the empirical quadratic loss over the set of all linear combinations of the basis functions (Duda & Hart, 1973; Kearns & Schapire, 1990).

When is empirical minimization sufficient for agnostic learning? This question has been answered in large part by Dudley (1978), Haussler (1992),, Pollard (1984), Vapnik (1982) and others. They show that, in many situations, the hypothesis class $\mathcal{H}$ is such that *uniform convergence* is achieved for reasonably small samples. In such situations, a bound $m(\epsilon, \delta)$ exists such that for any[2] distribution A on $X \times Y$, and any random sample $S \in (X \times Y)^*$ of size $m \geq m(\epsilon, \delta)$ chosen according to A, the probability that the average empirical loss of any $h \in \mathcal{H}$ differs from its true expected loss by more than ϵ is at most δ; that is,

$$\mathbf{Pr}\left[\exists h \in \mathcal{H} : \left|\hat{\mathbf{E}}_S[L_h] - \mathbf{E}[L_h]\right| > \epsilon\right] \leq \delta. \tag{1}$$

Thus, if $\mathcal{T}$ is (efficiently) empirically minimizable by $\mathcal{H}$, and if uniform convergence can be achieved for $\mathcal{H}$, then $\mathcal{T}$ is (efficiently) agnostically learnable using $\mathcal{H}$.

Here is how this is done: Given ϵ and δ, let $t \in \mathcal{T}$ be such that $\mathbf{E}[L_t] \leq opt(\mathcal{T}) + \epsilon/3$. (Since there may not exist a function that achieves the optimum loss, we instead choose any function that is approximately optimal.) Let S be a random sample of size sufficiently large that, with probability at least $1 - \delta$,

$$\left|\hat{\mathbf{E}}_S[L_h] - \mathbf{E}[L_h]\right| \leq \epsilon/3$$

for every $h \in \mathcal{H} \cup \{t\}$. (Note that uniform convergence is not required for the entire touchstone class $\mathcal{T}$, but only for the hypothesis class $\mathcal{H}$ and a single element $t \in \mathcal{T}$ that is close to optimal.) Let $h \in \mathcal{H}$ be the result of applying the assumed empirical minimization algorithm to S. Then, with probability at least $1 - \delta$,

$$\begin{aligned}
\mathbf{E}[L_h] &\leq \hat{\mathbf{E}}[L_h] + \epsilon/3 \\
&\leq \hat{\mathbf{E}}[L_t] + \epsilon/3 \\
&\leq \mathbf{E}[L_t] + 2\epsilon/3 \\
&\leq opt(\mathcal{T}) + \epsilon
\end{aligned}$$

as desired.

Although in this paper we focus primarily on empirical loss minimization, it is worth noting that an alternative approach is to minimize the empirical loss on the data plus some measure of the complexity of the hypothesis (see, for instance, Vapnik (1982)).

4.2. Learning piecewise functions

Thus, in cases where uniform convergence is known to occur, the problem of agnostic learning is largely reduced to that of minimizing the empirical loss on any finite sample. We apply this fact to the problem of agnostically learning families of piecewise functions with domain $X \subseteq \mathbb{R}$. We give a general technique based on dynamic programming for learning such functions (given certain assumptions), and we show, for instance, that this technique can be applied to agnostically learn step functions and piecewise polynomials.

A similar dynamic programming technique is used by Rissanen, Speed and Yu (1992) for finding the "minimum description length" histogram density function; see also Yamanishi (1992b).

We assume below that $X \subseteq \mathbb{R}$. Let $\mathcal{F}$ be a class of functions on X. We say that a function f is an *s-piecewise function over $\mathcal{F}$* if there exist disjoint intervals $I_1, \ldots, I_s$ (called *bins*) whose union is $\mathbb{R}$, and functions $f_1, \ldots, f_s$ in $\mathcal{F}$ such that $f(x) = f_i(x)$ for all $x \in X \cap I_i$. Let $\mathrm{PW}_s(\mathcal{F})$ denote the set of all s-piecewise functions over $\mathcal{F}$.

THEOREM 4 *Let $\mathcal{T}$ be a hypothesis class on $X \subseteq \mathbb{R}$ that is empirically minimizable by $\mathcal{H}$ with respect to L. Then $\mathrm{PW}_s(\mathcal{T})$ is empirically minimizable by $\mathrm{PW}_s(\mathcal{H})$ in time polynomial in s, and the size m of the given sample.*

Proof: We give a general dynamic programming technique for empirically minimizing $\mathrm{PW}_s(\mathcal{T})$. Let $S = \langle (x_1, y_1), \ldots, (x_m, y_m) \rangle$ be the given sample, and assume without loss of generality that $x_1 \le \cdots \le x_m$.

For $1 \le i \le m$ and $1 \le j \le s$, we will be interested in computing a j-piecewise function p_{ij} over $\mathcal{H}$ that, informally, is a "good" j-piecewise hypothesis for S_i, where $S_i = \langle (x_1, y_1), \ldots, (x_i, y_i) \rangle$. More precisely, the empirical loss of p_{ij} on S_i will not exceed that of any hypothesis in $\mathrm{PW}_j(\mathcal{T})$. Then clearly p_{ms} will meet the goal of empirical minimization of $\mathrm{PW}_s(\mathcal{T})$ over the entire sample S.

We use the following straightforward procedure to compute p_{ij}. For $0 \le k \le i$, we consider placing the last k observations in a bin by themselves (that is, we let these k observations belong to the same bin of the piecewise function under construction). We then use our empirical minimization algorithm for $\mathcal{T}$ to compute a hypothesis $h_{ik} \in \mathcal{H}$ whose empirical loss (on the last k observations of S_i) does not exceed that of any hypothesis in $\mathcal{T}$. We next "recursively" compute $p_{i-k,j-1}$, a "good" $(j-1)$-piece hypothesis for the remaining $i-k$ observations. We can combine $p_{i-k,j-1}$ and h_{jk} in the obvious manner to form a j-piece hypothesis p_{ij}^k, and we let $p_{ij} = p_{ij}^{k^*}$ for that k^* giving minimum loss on S_i.

To summarize more formally, the procedure computes p_{ij} as follows:

1. if $j = 1$ then compute $p_{ij} \in \mathcal{H}$ such that $L_{p_{ij}}(S_i) \le L_h(S_i)$ for all $h \in \mathcal{T}$.

2. else for $0 \le k \le i$ do:

   (A) let $T_{ik} = \langle (x_{i-k+1}, y_{i-k+1}), \ldots, (x_i, y_i) \rangle$

   (B) compute $h_{ik} \in H$ such that $L_{h_{ik}}(T_{ik}) \le L_h(T_{ik})$ for all $h \in \mathcal{T}$

   (C) "recursively" compute $p_{i-k,j-1}$

   (D) let

$$p_{ij}^k(x) = \begin{cases} p_{i-k,j-1}(x) & \text{if } x < x_{i-k+1} \\ h_{ik}(x) & \text{otherwise} \end{cases}$$

3. $p_{ij} = p_{ij}^{k*}$ where $k* = \arg\min_k (L_{p_{ij}^k}(S_i))$.

(Here, we use the notation $L_h(S)$ to denote the total loss of h on a sample S: $L_h(S) = \sum_{(x,y)\in S} L_h(x,y)$.)

Although we described the computation of p_{ij} recursively, in fact, we can store these values in a table using standard dynamic programming techniques. That the procedure runs in polynomial time then follows from the fact that only $O(ms)$ piecewise functions p_{ij} are computed and stored in such a table.

To prove the correctness of the procedure, we argue by induction on j that $L_{p_{ij}}(S_i) \le L_h(S_i)$ for $h \in \mathrm{PW}_j(\mathcal{T})$. In the base case that $j = 1$, this follows immediately from our assumption that $\mathcal{T}$ is empirically minimizable by $\mathcal{H}$.

Otherwise, if $j > 1$, then let f be a function in $\mathrm{PW}_j(\mathcal{T})$ defined by bins $I_1, \ldots, I_j$ and functions $f_1, \ldots, f_j \in \mathcal{T}$. Assume without loss of generality that the bins are ordered in the sense that if $u < v$, $r \in I_u$ and $s \in I_v$ then $r < s$.

Choose the largest value of k for which all the points of T_{ik} fall in bin I_j, i.e., for which $\{x_{i-k+1}, \ldots, x_i\} \subseteq I_j$. Then $L_{h_{ik}}(T_{ik}) \leq L_{f_j}(T_{ik})$ by our assumption that $\mathcal{T}$ is empirically minimizable by $\mathcal{H}$. Let f' be the $(j-1)$-piecewise function defined by bins $I_1, \ldots, I_{j-2}, I_{j-1} \cup I_j$ and functions $f_1, \ldots, f_{j-1}$. Then, by the inductive hypothesis, $L_{p_{i-k,j-1}}(S_{i-k}) \leq L_{f'}(S_{i-k})$. Thus,

$$\begin{aligned} L_{p_{ij}}(S_i) &\leq L_{p^k_{ij}}(S_i) \\ &= L_{p_{i-k,j-1}}(S_{i-k}) + L_{h_{ik}}(T_{ik}) \\ &\leq L_{f'}(S_{i-k}) + L_{f_j}(T_{ik}) \\ &= L_f(S_i), \end{aligned}$$

completing the induction and the proof. □

Thus, in the frequent case that empirical minimization of loss is sufficient for learning, Theorem 4 may be used to translate an algorithm for loss minimization over $\mathcal{T}$ into an agnostic learning algorithm for functions that are piecewise over $\mathcal{T}$. As an application, suppose the observed range Y is bounded so that $Y \subseteq [-M, M]$ for some finite M. In such a setting, Theorem 4 implies the efficient agnostic learnability (with respect to the quadratic loss function) of step functions with at most s steps (i.e., piecewise functions in which each piece f_i is a constant function). This follows from the fact that constant functions are empirically minimizable, and the fact that uniform convergence can be achieved for such functions. By a similar though more involved argument, Theorem 4 can be invoked to show more generally that s-piecewise degree-d polynomials can be agnostically learned in polynomial time, as we show below.

Before proving this theorem, however, we will first need to review some tools for proving uniform convergence. Specifically, we will be interested in the *pseudo dimension* of a class of functions $\mathcal{F}$, a combinatorial property of $\mathcal{F}$ that largely characterizes the uniform convergence over $\mathcal{F}$ (Dudley, 1978; Haussler, 1992; Pollard, 1984).

Let $\mathcal{F}$ be a class of functions $f : X \to \mathbb{R}$, and let $S = \{(x_1, y_1), \ldots, (x_d, y_d)\}$ be a finite subset of $X \times \mathbb{R}$. We say that $\mathcal{F}$ *shatters* S if

$$\{0,1\}^d = \{\langle \mathrm{pos}(f(x_1) - y_1), \ldots, \mathrm{pos}(f(x_d) - y_d)\rangle : f \in \mathcal{F}\}$$

where $\mathrm{pos}(x)$ is 1 if x is positive and 0 otherwise. Thus, $\mathcal{F}$ shatters S if every "above-below" behavior on the points $x_1, \ldots, x_d$ relative to $y_1, \ldots, y_d$ is realized by some function in $\mathcal{F}$.

The *pseudo dimension* of $\mathcal{F}$ is the cardinality of the largest shattered finite subset of $X \times \mathbb{R}$ (or is ∞ if no such maximum exists).

Haussler (1992, Corollary 2) argues that, if the class $L_{\mathcal{H}} = \{L_h : h \in \mathcal{H}\}$ is uniformly bounded and has pseudo dimension $d < \infty$, then a sample of size polynomial in $1/\epsilon$, $1/\delta$ and d is sufficient to guarantee uniform convergence in the sense of Equation (1). Thus, to prove uniform convergence for a hypothesis space $\mathcal{H}$, it suffices to upper bound the pseudo dimension of $L_{\mathcal{H}}$ (and to show that $L_{\mathcal{H}}$ is uniformly bounded).

Since we are here concerned with piecewise functions, the following theorem will be useful for this purpose:

THEOREM 5 *Let $X \subseteq \mathbb{R}$, and let $\mathcal{F}$ be a class of real-valued functions on X with pseudo dimension $d < \infty$. Then the pseudo dimension of $\mathrm{PW}_s(\mathcal{F})$ is at most $s(d+1)-1$.*

Proof: Let S be a subset of $X \times \mathbb{R}$ of cardinality $s(d+1)$. We wish to show that S is not shattered by $\mathrm{PW}_s(\mathcal{F})$.

Let the elements of S be indexed by pairs i, j where $1 \leq i \leq s$ and $1 \leq j \leq d+1$. Further, assume without loss of generality that these elements have been sorted so that $S = \{(x_{ij}, y_{ij})\}_{1 \leq i \leq s, 1 \leq j \leq d+1}$ and $x_{ij} < x_{i'j'}$ if $i < i'$ or if $i = i'$ and $j < j'$. (If the x_{ij}'s are not all distinct, then S cannot possibly be shattered.) Thus, we break the x_{ij}'s into s blocks, each consisting of $d+1$ consecutive points.

Let $S_i = \{(x_{ij}, y_{ij})\}_{1 \leq j \leq d+1}$ be the ith such block. Since $\mathcal{F}$ has pseudo dimension d, S_i cannot be shattered, which means that there must exist a string $\sigma_i \in \{0,1\}^{d+1}$ that is not included in the set

$$\lambda_i = \{\langle \mathrm{pos}(f(x_{i1}) - y_{i1}), \ldots, \mathrm{pos}(f(x_{i,d+1}) - y_{i,d+1})\rangle : f \in \mathcal{F}\}.$$

Let $\sigma = \sigma_1 \sigma_2 \cdots \sigma_s$ be the concatenation of these strings σ_i. We claim that σ is not a member of

$$\begin{aligned}\lambda = \{\langle & \mathrm{pos}(f(x_{11}) - y_{11}), \ldots, \mathrm{pos}(f(x_{1,d+1}) - y_{1,d+1}),\\ & \mathrm{pos}(f(x_{21}) - y_{21}), \ldots, \mathrm{pos}(f(x_{2,d+1}) - y_{2,d+1}),\\ & \qquad\vdots \\ & \mathrm{pos}(f(x_{s,1}) - y_{s,1}), \ldots, \mathrm{pos}(f(x_{s,d+1}) - y_{s,d+1})\rangle : f \in \mathrm{PW}_s(\mathcal{F})\}.\end{aligned}$$

Suppose to the contrary that f witnesses σ's membership in λ. Then f is defined by disjoint intervals $I_1, \ldots, I_s$ whose union is $\mathbb{R}$, and functions $f_1, \ldots, f_s \in \mathcal{F}$. Assume without loss of generality that the intervals have been sorted so that if $i < j$ then every point in I_i is smaller than every point in I_j. Inductively, we show the following invariant holds for f: For $i = 1, 2, \ldots, s$, the set $I_1 \cup \cdots \cup I_i$ does not contain all the elements $x_{1,1}, \ldots, x_{i,d+1}$. The fact that I_1 does not contain all the elements $x_{1,1}, \ldots, x_{1,d+1}$ follows from the definition of σ_1 (otherwise, $f_1 \in \mathcal{F}$ witnesses $\sigma_1 \in \lambda_1$). Suppose that $I_1 \cup \cdots \cup I_i$ contains the elements $x_{1,1}, \ldots, x_{i,d+1}$. By the inductive assumption, $I_1 \cup \cdots \cup I_{i-1}$ contains at most the points $x_{1,1}, \ldots, x_{i-1,d}$; therefore, the interval I_i contains at least the elements $x_{i-1,d+1}, \ldots, x_{i,d+1}$. But then f_i is a witness for $\sigma_i \in \lambda_i$, which contradicts the definition of σ_i.

Thus, in particular, $I_1 \cup \cdots \cup I_s$ does not contain all the points $x_{1,1}, \ldots, x_{s,d+1}$, a clear contradiction since $I_1 \cup \cdots \cup I_s = \mathbb{R}$.

Therefore, as claimed, $\sigma \notin \lambda$, and so S is not shattered, proving the theorem. □

We are now ready to prove the agnostic learnability of piecewise polynomial functions:

THEOREM 6 *Let* $X \subseteq \mathbb{R}$ *and* $Y \subseteq [-M, M]$. *Then there exists an algorithm for agnostically learning the class of s-piecewise degree-d polynomials (with respect to the quadratic loss function Q) in time polynomial in s, d, M,* $1/\epsilon$ *and* $1/\delta$. *The sample complexity of this algorithm is* $(9216M^4/\epsilon^2)(4s(1+d)^2(2+d)\ln(192eM^2/\epsilon)+\ln(16/\delta))$.

Proof: Let $\mathcal{P}$ be the class of real-valued degree-d polynomials on X, and let $\mathcal{P}_s = \mathrm{PW}_s(\mathcal{P})$. Let $\overline{\mathcal{P}}$ be the set of polynomials in $\mathcal{P}$ with range in $[-M, M]$, and similarly define $\overline{\mathcal{P}}_s$. Our goal is to show that $\mathcal{P}_s$ is agnostically learnable.

For any function $f : X \to \mathbb{R}$, let $\mathrm{CLAMP}(f)$ be that function obtained by "clamping" f in the range $[-M, M]$. That is, $\mathrm{CLAMP}(f) = g \circ f$ where

$$g(y) = \begin{cases} -M & \text{if } y \leq -M \\ y & \text{if } -M \leq y \leq M \\ M & \text{if } M \leq y. \end{cases}$$

For a class of real-valued functions $\mathcal{F}$, we also define $\mathrm{CLAMP}(\mathcal{F})$ to be $\{\mathrm{CLAMP}(f) : f \in \mathcal{F}\}$.

As noted above, the collection of all linear combinations of a set of basis functions is empirically minimizable. Thus, choosing basis functions $1, x, \ldots, x^d$, we see that $\mathcal{P}$ is empirically minimizable by $\mathcal{P}$, and therefore, applying Theorem 4, $\mathcal{P}_s$ is empirically minimizable by $\mathcal{P}_s$.

To show that $\mathcal{P}_s$ is agnostically learnable, it would suffice then to prove a uniform-convergence result for $\mathcal{P}_s$. Unfortunately, most of the known techniques (Haussler, 1992; Pollard, 1984) for proving such a result would require that the loss function Q be bounded. In our setting, this would be the case if and only if the functions in the hypothesis space $\mathcal{H}$ were uniformly bounded, which they are not if $\mathcal{H} = \mathcal{P}_s$.

Therefore, rather than output the piecewise polynomial p in $\mathcal{P}_s$ with minimum empirical loss, we instead output $p' = \mathrm{CLAMP}(p)$. Note that the empirical loss of p' is no greater than that of p since our observed range is a subset of $[-M, M]$. Thus, $\mathcal{P}_s$ is empirically minimizable by $\mathrm{CLAMP}(\mathcal{P}_s)$.

We argue next that a polynomial-size sample suffices to achieve uniform convergence for $\mathrm{CLAMP}(\mathcal{P}_s)$ with respect to the loss function Q. As noted above, by Haussler's (1992) Corollary 2, this will be the case if $Q_{\mathrm{CLAMP}(\mathcal{P}_s)}$ is uniformly bounded and has polynomial pseudo dimension. Clearly, every function in $Q_{\mathrm{CLAMP}(\mathcal{P}_s)}$ is bounded between 0 and $4M^2$ so $Q_{\mathrm{CLAMP}(\mathcal{P}_s)}$ is uniformly bounded.

To bound the pseudo dimension of $Q_{\mathrm{CLAMP}(\mathcal{P}_s)}$, we make the following observations:

1. Because every degree-d polynomial p has at most $d-1$ "humps," $\mathrm{CLAMP}(p)$ must be an element of $\overline{\mathcal{P}}_{d+1}$. Thus, $\mathrm{CLAMP}(\mathcal{P}_s) \subseteq \overline{\mathcal{P}}_{s(d+1)} \subseteq \mathcal{P}_{s(d+1)}$, and so $Q_{\mathrm{CLAMP}(\mathcal{P}_s)} \subseteq \mathrm{PW}_{s(d+1)}(Q_{\mathcal{P}})$.

2. Every function $Q_h(x, y)$ in $Q_{\mathcal{P}}$ can be written as a linear combination of the basis functions $1, x^2, \ldots, x^{2d}, y, yx, \ldots, yx^d$ and y^2. This follows from the definition of quadratic loss, and from the fact that h is a degree-d polynomial.

3. Thus, $Q_{\mathcal{P}}$ is a subset of a $(2d+3)$-dimensional vector space of functions. Therefore, its pseudo dimension is at most $2d+3$ (Dudley, 1978) (reproved by Haussler (1992, Theorem 4)).

4. By Theorem 5, this implies that the pseudo dimension of $\mathrm{PW}_{s(d+1)}(Q_{\mathcal{P}})$ is at most $s(d+1)(2d+4)$.

Therefore, the pseudo dimension of $Q_{\mathrm{CLAMP}(\mathcal{P}_s)}$ is at most $s(d+1)(2d+4)$.

To complete the proof, we must overcome one final technical difficulty: We must show that there exists a polynomial $q \in \mathcal{P}_s$ whose true expected loss is within $\epsilon/3$ of optimal, and whose empirical loss is within $\epsilon/3$ of its true loss. (See Section 4.1.) Again, this may be difficult or impossible to prove since q may be unbounded.

However, this is not a problem if q has range $[-M, M]$ (i.e., if $q \in \overline{\mathcal{P}}_s$) since in this case a good empirical estimate of q's true loss can be obtained using Hoeffding's (1963) inequality.

Thus, because $\overline{\mathcal{P}}_s \subseteq \mathcal{P}_s$ is empirically minimizable by $\mathrm{CLAMP}(\mathcal{P}_s)$, we have effectively shown that $\overline{\mathcal{P}}_s$ is agnostically learnable using $\mathrm{CLAMP}(\mathcal{P}_s)$.

This is not quite what we set out to prove since our goal was to show that $\mathcal{P}_s$ is agnostically learnable. However, this can now be proved using the fact that every function in $\mathrm{CLAMP}(\mathcal{P}_s)$ is in fact a piecewise polynomial with range in $[-M, M]$.

More specifically, as noted above, $\mathrm{CLAMP}(\mathcal{P}_s) \subseteq \overline{\mathcal{P}}_{s(d+1)}$, so $\mathrm{CLAMP}(\mathcal{P}_s)$ is agnostically learnable using $\mathrm{CLAMP}(\mathcal{P}_{s(d+1)})$. Since the loss of $\mathrm{CLAMP}(p)$ is no worse than that of p, for any function p, it follows that $opt(\mathrm{CLAMP}(\mathcal{P}_s)) \leq opt(\mathcal{P}_s)$. This implies that $\mathcal{P}_s$ is agnostically learnable using $\mathrm{CLAMP}(\mathcal{P}_{s(d+1)})$.

The stated sample complexity bound follows from a combination of the above facts with Haussler's (1992) Corollary 2. □

Thus, we have shown that piecewise polynomials are agnostically learnable when the number of pieces s is fixed. It is natural to ask whether it is truly necessary that s be fixed. In other words, is there an efficient algorithm that "automatically" picks the "right" number of pieces s? Formally, this is asking whether the class $\mathrm{PW}(\mathcal{P}) = \bigcup_{s \geq 1} \mathrm{PW}_s(\mathcal{P})$ is agnostically learnable (with respect to the quadratic loss function). Here, we would allow the learning algorithm time polynomial in $1/\epsilon$, $1/\delta$, and the minimum number of pieces s necessary to have loss at most $opt(\mathrm{PW}(\mathcal{P}))+\epsilon$.

Unfortunately, this is not feasible because we can construct situations in which there is not enough information to determine whether the "right" number of pieces is very large or very small. Specifically, let $X = [0,1]$ be the domain with a uniform distribution on instances in X, let $Y = \{0,1\}$ be the observed range, and assume that the degree of the polynomials we are using is zero (in other words, we are trying to agnostically learn step functions). Consider the following two p-concepts: The first is the constant function $f \equiv 1/2$. In other words, each point x is labeled 0 or 1 with equal probability. In this case, the optimal number of pieces s is one — the quadratic loss is minimized by a single step that is $1/2$ over the entire domain. The second kind of p-concept, denoted g_t, is a deterministic function (i.e., its range is $\{0,1\}$) that has t equal size steps, where t is "large." The value of the function on each of these steps is chosen at random (although, as already mentioned, the function itself is deterministic). In this case, the optimal number of pieces is $s = t$.

Intuitively, it seems clear that the learning algorithm cannot distinguish these two cases until it observes at least two points in the same bin, an event that is unlikely to occur until about $\sqrt{t}$ points are observed. Further, without the ability to distinguish these cases, the learning algorithm cannot find a hypothesis whose loss comes close to optimal. This is because if the learning algorithm stops before having seen $\sqrt{t}$ examples, then it cannot distinguish data produced by f or g_t. Thus, its hypothesis will be far from at least one of these p-concepts, and therefore, the learner has a reasonably high probability of outputting a p-concept that is far from optimal if it chooses a sample significantly smaller than $\sqrt{t}$. On the other hand, if the learner does choose a sample of size $\sqrt{t}$ or larger, then it risks drawing far too many examples when t is large, but the true target p-concept is f (in which case $t = 1$).

Although we omit the details, these arguments can be made rigorous using, for instance, the randomized lower bound techniques of Blumer et al. (1989). Since t is arbitrary, this shows that an arbitrarily large number of observations are needed to agnostically learn piecewise polynomials with any finite number of pieces.

Finally, we mention that the results of this section can be generalized to find piecewise functions that are continuous by only considering a finite set of endpoints for the hypothesis function over each interval and adding the choice of endpoint as a variable in the dynamic program.

5. Relations between loss functions for agnostic learning

Suppose that our assumption class $\mathcal{A}$ is the functional decomposition using some class $\mathcal{F}$ of boolean functions. A common approach to learning under such conditions is to find a *real-valued* hypothesis h instead of a boolean function; the hope is that even given the knowledge that the target $f \in \mathcal{F}$ is boolean, it may be easier to find algorithms that operate in a space of functions characterized by a *continuous* parameterization, and that may thus make incremental changes or pursue hill-climbing methods that do not exist for boolean classes. Indeed, general-purpose learning algorithms such as the well-known backpropogation algorithm for neural networks use exactly such an approach.

However, algorithms searching for a real-valued hypothesis almost invariably attempt to minimize a loss function that incorporates the actual real-valued output $h(x)$ (such as the quadratic loss Q), and as such do not explicitly address performance for the most natural loss function for boolean targets, the prediction loss Z. More precisely, if $f : X \to \{0, 1\}$ is the boolean target function, does finding an $h : X \to [0, 1]$ minimizing $\mathbf{E}[Q_h] = \mathbf{E}[(f - h)^2]$ help us at all in predicting the boolean target value $f(x)$?

One obvious approach is to define $\Theta_h(x) = 1$ if $h(x) \geq 1/2$ and 0 otherwise, and to use Θ_h to make boolean choices from the real-valued h. This works to some degree: it is easy to show that in general,

$$\mathbf{E}[Z_{\Theta_h}] = \mathbf{Pr}[f \neq \Theta_h] \leq 4\mathbf{E}[Q_h].$$

(The proof of the last inequality follows by noting that $4\mathbf{E}[Q_h] = \mathbf{E}[(2f-2h)^2]$, and by observing next that if $f(x) \neq \Theta_h(x)$ (so that $f(x) = 0$ and $h(x) \geq 1/2$, or $f(x) = 1$ and $h(x) < 1/2$) then $(2f-2h)^2 \geq 1$.) This bound is tight in the sense that there exist boolean f and real-valued h for which the equality holds. Thus, in the case that $\mathbf{E}[Q_h]$ is small, the stated bound on the expected prediction loss is nontrivial.

However, in our pursuit of agnostic learning we wish to allow the weakest assumptions on f, in which case we should *not* expect to be able to find a hypothesis h for which $\mathbf{E}[Q_h]$ is small. Further, for $\mathbf{E}[Q_h]$ larger than $1/8$, the bound obtained on $\mathbf{E}[Z_{\Theta_h}]$ is not better than that achieved by random guessing. We would like to find a way of using h to make predictions with a nontrivial probability of mistake even as $\mathbf{E}[Q_h]$ approaches $1/4$ (which is the expected quadratic loss corresponding to "random guessing" achieved by the constant function $1/2$).

For any function $h : X \to [0,1]$, we define $\$_h(x)$ to be a boolean random variable that is 1 with probability $h(x)$ and 0 with probability $1 - h(x)$; thus it is simply the p-concept interpretation of h. We write $\mathbf{E}[Z_{\$_h}]$ to denote $\mathbf{Pr}[f(x) \neq \$_h(x)]$, where this probability is taken over the random draw of x and the coin flip associated with $\$_h$.

THEOREM 7 *Let $f : X \to \{0,1\}$ be any boolean function, and let $h : X \to [0,1]$ be a real-valued function. Then for any distribution D on X,*

$$\mathbf{E}[Z_{\$_h}] = \mathbf{E}[Q_h] + \mathbf{E}[h(1-h)] \leq \mathbf{E}[Q_h] + 1/4.$$

Proof: We have that

$$\begin{aligned} \mathbf{E}[Z_{\$_h}] &= \mathbf{E}[f(1-h) + (1-f)h] \\ &= \mathbf{E}[f - 2fh + h] \end{aligned}$$

and that

$$\begin{aligned} \mathbf{E}[Q_h] &= \mathbf{E}[(f-h)^2] \\ &= \mathbf{E}[f^2 - 2fh + h^2]. \end{aligned}$$

Combining these equations, and noting that $f^2 = f$ (since f is boolean), we have

$$\mathbf{E}[Z_{\$_h}] = \mathbf{E}[Q_h] + \mathbf{E}[h(1-h)]$$

as claimed. The stated upper bound on this quantity follows simply from the fact that $x(1-x) \leq 1/4$ for all real x. □

Thus, provided we have achieved a nontrivial expected quadratic loss with h, we can use $\$_h$ to obtain a nontrivial expected prediction loss. More precisely, if $\mathbf{E}[Q_h] \leq \alpha < 1/4$, then $\mathbf{E}[Z_{\$_h}] \leq 1/4 + \alpha < 1/2$, and may be considerably smaller if h is "almost boolean" in the sense that $\mathbf{E}[h(1-h)]$ is small. Note that in the case of very small expected quadratic loss, we should still use Θ_h for predictions; Theorem 7 covers the agnostic setting where the expected quadratic loss may be large but non-trivial. In either case, since the expected quadratic loss of h is a quantity we can estimate, we can choose which predictor to use, giving us a worst-case expected prediction loss of $\min(4\mathbf{E}[Q_h], \mathbf{E}[Q_h] + \mathbf{E}[h(1-h)])$.

We note that an improved technique was communicated to us by M. Warmuth. This technique replaces $\$_h(x)$ with a rule that predicts 1 with probability $h(x)^2/(h(x)^2 + (1-h(x))^2)$, and 0 otherwise. Using an argument similar to that used in the proof of Theorem 7, it can be shown that this rule has predictive loss $\mathbf{E}[Q_h/(h^2+(1-h)^2)] \leq 2 \cdot \mathbf{E}[Q_h]$.

5.1. Application: weak agnostic learning of AC^0

We can immediately apply Theorem 7 to some existing algorithms in the standard PAC model to obtain algorithms for "weak" agnostic learning. For instance, Linial, Mansour and Nisan (1993) describe an algorithm in the standard PAC model with the target domain distribution restricted to be uniform over $\{0,1\}^n$. The hypothesis space $\mathcal{H}$ of this algorithm is the class of functions with a Fourier expansion over the so-called *parity basis* whose high-order coefficients (that is, the coefficients of all basis functions whose size exceeds ℓ) are 0. The algorithm runs in time polynomial in n^ℓ, $1/\epsilon$ and $1/\delta$. It is shown that the algorithm finds a real-valued h such that $\mathbf{E}[Z_{\Theta_h}]$ is less than ϵ provided the boolean target function f is "close" to some hypothesis in the restricted hypothesis class $\mathcal{H}$ (that is, the optimal expected prediction loss must be close to zero).

However, $\mathbf{E}[Z_{\Theta_h}]$ is *not* guaranteed to be near the optimal in the agnostic setting where f is unrestricted. Nevertheless, the algorithm of Linial, Mansour and Nisan can be used to find an h that (nearly) minimizes $\mathbf{E}[Q_h]$ even in the agnostic setting; thus we can apply Theorem 7 to show that for *any* boolean target function f, $\min(4\mathbf{E}[Q_h], \mathbf{E}[Q_h] + \mathbf{E}[h(1-h)])$ bounds our expected prediction loss. For instance, this means that if there exists an AC^0 function[3] C that weakly approximates the target function f on the uniform distribution (so that f agrees with C with probability at least $1/2 - 1/p(n)$ for some polynomial p) then the results of Linial, Mansour and Nisan combined with Theorem 7 imply the existence of a quasi-polynomial time algorithm for finding a hypothesis that weakly approximates f. We summarize these ideas with a corollary:

COROLLARY 1 *There exists an algorithm with the following properties. The algorithm is given s, d, ϵ, δ and access to randomly generated examples of a function $f : \{0,1\}^n \to \{0,1\}$. Let γ be such that there exists an AC^0 circuit C of size s and depth d with the property that $\mathbf{Pr}[f \neq C] \leq 1/2 - \gamma$. Then, with probability at least $1-\delta$, the algorithm finds a hypothesis function h such that $\mathbf{Pr}[f \neq h] \leq 1/2 - \gamma^2 + \epsilon$ (where all probabilities are computed with respect to the uniform distribution on $\{0,1\}^n$). The algorithm runs in time polynomial in n^ℓ, $1/\epsilon$, and $\log(1/\delta)$, where $\ell = \left(20\lg(8s/\epsilon^2)\right)^d$.*

Proof sketch: The proof uses properties of the Fourier transform, as described in detail by Linial, Mansour and Nisan (1993). Any function $f : \{0,1\}^n \to \mathbb{R}$ can be written in the form:

$$f(x) = \sum_{S \subseteq \{1,\ldots,n\}} \hat{f}(S)\chi_S(x)$$

where $\chi_S(x) = \prod_{i\in S}(-1)^{x_i}$. A useful fact is Parseval's identity:

$$\mathbf{E}[f^2] = \sum_S \hat{f}(S)^2.$$

Let C be as in the statement of the corollary, and let g be defined by:

$$g(x) = \begin{cases} 1/2 - \gamma & \text{if } C(x) = 0 \\ 1/2 + \gamma & \text{if } C(x) = 1 \end{cases}$$

Then it can be shown that $\mathbf{E}[(f-g)^2] \le 1/4 - \gamma^2$.

Let r be the function defined by

$$r(x) = \sum_{|S|\le \ell} \hat{f}(S)\chi_S(x) + \sum_{|S|>\ell} \hat{g}(S)\chi_S(x).$$

Thus, r is a sort of mixture of f and g.

By Parseval's identity, $\mathbf{E}[(f-r)^2] \le \mathbf{E}[(f-g)^2] \le 1/4 - \gamma^2$.

We can approximate the function r by running the algorithm given in Linial, Mansour and Nisan (1993), with the choices of ℓ and δ as given above, and with ϵ set to $\epsilon^2/4$. We can do this with access to examples of the function f since the algorithm of Linial, Mansour and Nisan approximates the low order coefficients of f (which are the same as for r), and sets the high order coefficients to be zero.

Let h be the resulting hypothesis. Then, by Parseval's identity, and by definition of r,

$$\mathbf{E}[(h-r)^2] = \sum_{|S|\le \ell} (\hat{h}(S) - \hat{f}(S))^2 + \sum_{|S|>\ell} (\hat{g}(S))^2.$$

The first sum is bounded by the accuracy of our approximation of each of the coefficients, and the second sum is bounded using the main lemma of Linial, Mansour and Nisan (1993, Lemma 7). The result is that $\mathbf{E}[(h-r)]^2 \le \epsilon^2/4$.

Since

$$\sqrt{\mathbf{E}[(h-f)^2]} \le \sqrt{\mathbf{E}[(h-r)^2]} + \sqrt{\mathbf{E}[(r-f)^2]},$$

it follows that $\mathbf{E}[Q_h] = \mathbf{E}[(h-f)^2] \le 1/4 - \gamma^2 + \epsilon$. Therefore, by Theorem 7, $\mathbf{E}[Z_{\$_h}] \le 1/2 - \gamma^2 + \epsilon$. □

We conclude this section by mentioning that Theorem 7 can be generalized to a model where the target function f is a discrete function assuming d possible values, and the output of h is a normalized vector in $\mathbb{R}^d$; this is intended to model settings such as character recognition, where we attempt to find a real-valued hypothesis but wish to predict which character is represented in the input with the greatest possible accuracy.

6. Hidden variable problems

Thus far we have been striving for algorithms that find a good hypothesis under the assumption that the target function is arbitrarily complex. An insight that has been made frequently in both the empirical and theoretical machine learning communities, however, is that *no* function is arbitrarily complex over *all* variable sets: if we can somehow define new variables that compute significant subfunctions of the target function, then the representation of the target function may simplify dramatically. This approach to simplifying target functions is sometimes loosely referred to as *feature discovery.*

One difficulty with this approach, of course, is that the right features may be as difficult to discover as the target function itself; in fact, in scientific domains the frontier of research often focuses just on finding the quantities that are relevant to a given phenomenon, and these may be uncovered only after long periods of experimentation and theory. Thus, in this section, we focus not on the problem of discovering features, but rather on the problem of learning when only some of the relevant variables are known or are "visible," while others are "hidden."

We are motivated by the simultaneous realizations that target functions may have simple representations over the appropriate variable set, but that only some of these variables may be known at any given time. This *hidden-variable* model allows an intermediate step between the strong assumptions of the standard PAC model and full agnosticism. This model was previously investigated by Kearns and Schapire (1990).

Let U and V be disjoint sets of variables. We say that the variables in V are *visible*, and that the variables in U are *hidden.* In our setting, the learner observes random examples which are classified according to some deterministic boolean function f over the entire variable set $U \cup V$. However, the learner is allowed to observe only the values of the visible variables. Thus, for a given assignment x to the visible variables V, the label assigned to x appears to be probabilistic. Specifically, the probability that x is labeled 1 is just the probability that an assignment is chosen for the hidden variables that causes f to evaluate to 1. To the learner, it appears that the examples are being labeled according to some p-concept p_f on the visible variables, where $p_f(x)$ is the conditional probability that $f = 1$ given that the visible variables are assigned x; that is, $p_f(x) = \mathbf{Pr}[f = 1 \mid x]$. We can therefore view such hidden-variable problems as p-concept problems where the domain is the set of assignments to the visible variables.

In this section, our goal will be to find the best possible predictor for the induced p-concept p_f when f is chosen from some class of functions $\mathcal{F}$. In other words, we will be interested in finding that rule (called the Bayes optimal predictor) which minimizes the expected prediction loss Z. We assume that the touchstone class is large enough to include the Bayes optimal predictor for any p_f. Finally, it is necessary to assume independence between the distributions of assignments to the hidden and visible variables; without this, it is possible to construct even trivial target functions f for which p_f is arbitrary.

As an easy first example, suppose the function f is chosen from the set of conjunctions of literals over $U \cup V$. In particular, suppose that f is given by the conjunction $M = RS$ where the variables in R and S are hidden and visible, respectively. Then it is not hard to see that $p_f(x)$ is 0 if $S(x) = 0$ and otherwise equals the probability r that $R = 1$. Note that if $r < 1/2$ then the Bayes optimal is the constant function 0; otherwise, it is just the conjunction S. It has been shown (Kearns & Schapire, 1990) that we can approximate the Bayes optimal predictor by applying Valiant's (1984) algorithm for conjunctions to approximate the conjunction S, and by then estimating r using this approximation for S. Our goal in this section is to obtain a similar result for the more general class of k-term DNF.

6.1. An algorithm for k-term DNF hidden variable problems

In the case of conjunctions, the Bayes optimal predictor is either the zero function or the restriction of the conjunction. (The *restriction* of a DNF formula is the formula obtained by syntactically deleting all of the hidden variables.) However, this may not be so in general, as can be seen in the case of k-term DNF formulas. For example, suppose that f is the formula $w_1x_1 \vee w_2x_2$ where w_1 and w_2 are hidden, and x_1 and x_2 are visible. Suppose also that w_1 and w_2 are each 1 independently with probability 0.4. Then in this case, the Bayes optimal predictor is x_1x_2, not the restriction formula $x_1 \vee x_2$.

More generally, let f be the k-term DNF formula $R_1S_1 \vee \cdots \vee R_kS_k$, where the R_i's and S_i's are terms over U and V, respectively. Note that the behavior of the p-concept p_f is exactly determined by the values of $S_1, \ldots, S_k$ (under our assumption that hidden and visible variables are independent). That is, if for $z \in \{0,1\}^k$ we define $q_f(z_1, \ldots, z_k)$ to be the probability that $f = 1$ given that $S_1 = z_1, \ldots, S_k = z_k$, then $p_f(x) = q_f(S_1(x), \ldots, S_k(x))$. Furthermore, it can be seen that q_f is monotone in the sense that $q_f(z) \geq q_f(z')$ whenever $z \geq z'$. (Here, $z \geq z'$ if $z_i \geq z'_i$ for all $1 \leq i \leq k$.) This is because if $z \geq z'$ then

$$q_f(z) = \mathbf{Pr}[\cup_{i:z_i=1} R_i = 1] \geq \mathbf{Pr}[\cup_{i:z'_i=1} R_i = 1] = q_f(z').$$

We have already seen that the Bayes optimal predictor for p_f need not be the restriction of f. In fact, it is not hard to come up with a k-term formula f and a distribution on the hidden variables such that $p_f \geq 1/2$ if and only if more than half of the terms S_i are satisfied. In this case, the Bayes optimal predictor, if expressed as a DNF formula over the visible variables, will be exponentially large (in k). Thus, although the original formula may be quite simple, the Bayes optimal predictor for the induced p-concept may be quite complicated.

Nevertheless, there does exist an efficient algorithm for finding the Bayes optimal predictor when f is a k-term DNF formula. We will show that p_f can be represented as a k-probabilistic decision list with increasing probabilities, a class of p-concepts for which there is known to exist an efficient algorithm for approximating the Bayes optimal predictor (Kearns & Schapire, 1990). A similar technique is used by Blum and Chalasani (1992).

A *k-probabilistic decision list* (k-PDL) ℓ over variable set V is a sequence of pairs $\langle(d_1, r_1), \ldots, (d_s, r_s)\rangle$ where each d_i is a conjunction of at most k literals from V and each $r_i \in [0, 1]$. We also require that some d_i is the constant function 1 (this is a slightly more convenient requirement than the equivalent requirement that $d_s = 1$). Here, $\ell(x)$ is defined to be r_j where j is the least index for which $d_j(x) = 1$. Such a list is said to have *increasing probabilities* if $r_i \leq r_{i+1}$ for $i < s$. See Kearns and Schapire (1990) and Yamanishi (1992a) for further background on probabilistic decision lists.

Kearns and Schapire (1990) show that k-PDL's with increasing probabilities can be learned with a *model of probability*: they describe an algorithm for finding an approximation h for a given list ℓ such that the expected difference $|h - \ell|$ is small. Thus, it suffices to show that p_f is a k-PDL with increasing probabilities, since we can then use Kearns and Schapire's algorithm to find the Bayes optimal predictor (and furthermore, find a good model of the function p_f itself).

THEOREM 8 *Let f be a k-term DNF formula. Then p_f is equivalent to a k-PDL with increasing probabilities.*

Proof: We show first that q_f is a k-PDL with increasing probabilities. We regard q_f as a function over the variables $s_1, \ldots, s_k$. For each possible assignment $z = \langle z_1, \ldots, z_k\rangle$, let $d_z = \bigwedge_{i:z_i=0} \bar{s}_i$, and let $r_z = q_f(z)$. Let ℓ be a list consisting of exactly the set of pairs (d_z, r_z) for all assignments z and ordered in such a fashion that ℓ has increasing probabilities.

We claim that $\ell(z) = q_f(z)$ for all z. To see that $q_f(z) \geq \ell(z)$, note that $d_z(z) = 1$, and therefore, $\ell(z) \leq r_z = q_f(z)$ since ℓ has increasing probabilities. To see that $q_f(z) \leq \ell(z)$, observe first that $\ell(z) = r_{z'}$ for some z' for which $d_{z'}(z) = 1$. By definition of $d_{z'}$, this means that for each i, if $z'_i = 0$ then $z_i = 0$; that is, $z' \geq z$. So, by monotonicity of q_f, this implies that $q_f(z) \leq q_f(z') = r_{z'} = \ell(z')$. Thus, $q_f(z) = \ell(z)$ as claimed.

By substitution then, $p_f(x) = \ell(S_1(x), \ldots, S_k(x))$. This is a list consisting of pairs (d_z, r_z) where $d_z = \bigwedge_{i:z_i=0} \bar{S}_i$. It is easily seen by DeMorgan's Law that d_z is a k-DNF formula $t_1 \vee \cdots \vee t_w$ over the variables in V. We therefore replace the pair (d_z, r_z) in ℓ with the sequence of pairs $(t_1, r_z), \ldots, (t_w, r_z)$. Applying this operation for each z, it is easily verified that the resulting list is a k-PDL with increasing probabilities that equals p_f. □

As noted above, the algorithm described by Kearns and Schapire (1990) can be applied to prove the following corollary:

COROLLARY 2 *Let f be a k-term DNF formula over the variable set $U \cup V$. Then there exists an efficient algorithm for finding the Bayes optimal predictor for the induced p-concept p_f over the assignments to V.*

6.2. Why k-CNF may be harder than k-term DNF

In this section we give evidence suggesting that learning may be difficult when the target function f is a k-CNF formula. Specifically, we show that for 2-CNF, there exist cases in which the Bayes optimal predictor is arbitrarily complicated, requiring an exponentially large representation.

Let $f = (s_1 \vee r_1) \cdots (s_n \vee r_n)$ where $s_i \in V$ and $r_i \in U$. Let f' be any DNF formula over V, each of whose terms contain exactly $n/2$ of the visible variables. Note that f' may be exponentially large. We show that we can create a distribution D^U on the hidden variables such that f' is the Bayes optimal function for f.

For each term t_i in f' we define an assignment z_i whose jth bit is 1 if and only if s_j does not occur in t_i. Let $\alpha = 1/(4\ell - 2)$ where ℓ is the number of terms of f'. Let $D^U(1^n) = 1/2 - \alpha$, let $D^U(z_j) = 2\alpha$ for all j, and let $D^U(u) = 0$ for all other assignments u.

Let v be an assignment to the visible variables. If a term t_i in f' is satisfied by v then f is satisfied when the hidden variables are assigned either 1^n (the all 1's vector) or z_i. Thus if f' is satisfied then $p_f(v) \geq 1/2 + \alpha$. Otherwise, if f' is not satisfied by v then the only satisfying assignment to the hidden variables that has nonzero probability is 1^n, so $p_f(v) = 1/2 - \alpha$ in this case.

Thus, as claimed, the Bayes optimal predictor for p_f is exactly f'.

Since there exists a doubly exponential number of formulas f' (specifically, there are $2^{\binom{n}{n/2}} = 2^{2^{\Omega(n)}}$ such formulas), this implies that for any representation of the Bayes optimal functions, there is some D^U for which the Bayes optimal predictor has an exponentially long representation.

However, note that most of the functions used in this construction can easily be approximated by a constant-sized representation since when α is small $p_f(v)$ is close to $1/2$ for all assignments v. Thus, it remains open whether the result of Section 6.1 can be extended to handle k-CNF formulas.

7. Open Problems

This paper presented the fruits of an initial investigation into the properties of agnostic learning models. There is much work to be done in this area, and it seems plausible that the "right" model for obtaining powerful positive results should choose a middle ground that balances assumptions on target functions with assumptions on domain distributions, while still remaining applicable to problems arising in practice. Here we have simply studied one extreme set of assumptions in order to obtain some idea of what can and cannot be accomplished efficiently.

The main open research direction is to explore the limits of efficient learning algorithms in agnostic models. Are there other problems for which there exist efficient learning algorithms? For instance, in Section 6, we showed how to learn p-concepts induced by partially visible k-term DNF formulas. Can this result be extended to handle k-CNF formulas? This problem may be harder since the Bayes optimal predictor can be extremely complicated. On the other hand, we have not yet come up with a case where there does not exist a very simple function that *approximates* the Bayes optimal predictor.

Rather than trying to find efficient algorithms for specific learning problems, we might instead explore the theoretical power of known algorithms. That is, we might ask if anything can be proved about the capabilities of various "off-the-shelf" learning algorithms commonly used in practice, such as neural networks and decision-tree algorithms.

We would also like to understand the theoretical properties of some of the models discussed in this paper. For instance, in the fully agnostic PAC model, is there any situation in which membership queries are useful? Intuitively, membership queries should not give us more power since the answers to queries are more or less arbitrary (since the target function is arbitrary). However, we have so far been unable to derive a rigorous proof based on this intuition.

Acknowledgements

Portions of this research were conducted while M. Kearns was at the International Computer Science Institute; while R. Schapire was at Harvard University and supported by AFOSR Grant 89-0506 and NSF Grant CCR-89-02500; and while L. Sellie was visiting AT&T Bell Laboratories.

We are grateful to Avrim Blum for pointing out that a probabilistic function of k terms can be expressed as a k-PDL. We also thank two anonymous referees for their careful reading and helpful comments.

Notes

1. To the best of our knowledge and recollection, the term "agnostic learning" was coined during a discussion among Sally Goldman, Ron Rivest, and the first two authors of this paper.
2. Certain "permissibility" assumptions are required — see Haussler (1992) for details.
3. AC^0 is the class of all boolean functions computed by a constant-depth boolean circuit composed of unbounded fan-in AND, OR and NOT gates.

References

Aldous, D. & Vazirani, U. (1990). A Markovian extension of Valiant's learning model. *31st Annual Symposium on Foundations of Computer Science* (pp. 392–404).

Blum, A. & Chalasani, P. (1992). Learning switching concepts. *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory* (pp. 231–242).

Blumer, A., Ehrenfeucht, A., Haussler, D., & Warmuth, M. K. (1989). Learnability and the Vapnik-Chervonenkis dimension. *Journal of the Association for Computing Machinery, 36*, 929–965.

Duda, R. O. & Hart, P. E. (1973). *Pattern Classification and Scene Analysis.* Wiley.

Dudley, R. M. (1978). Central limit theorems for empirical measures. *The Annals of Probability, 6*, 899–929.

Freund, Y. (1990). Boosting a weak learning algorithm by majority. *Proceedings of the Third Annual Workshop on Computational Learning Theory* (pp. 202–216).

Freund, Y. (1992). An improved boosting algorithm and its implications on learning complexity. *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory* (pp. 391–398).

Garey, M. & Johnson, D. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness.* San Francisco: W. H. Freeman.

Haussler, D. (1992). Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation, 100*, 78–150.

Helmbold, D. P. & Long, P. M. (1994). Tracking drifting concepts by minimizing disagreements. *Machine Learning, 14*, 27–45.

Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association, 58*, 13–30.

Izenman, A. J. (1991). Recent developments in nonparametric density estimation. *Journal of the American Statistical Association, 86*, 205–224.

Kearns, M. & Li, M. (1993). Learning in the presence of malicious errors. *SIAM Journal on Computing, 22*, 807–837.

Kearns, M., Li, M., Pitt, L., & Valiant, L. (1987). On the learnability of Boolean formulae. *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing* (pp. 285–295).

Kearns, M. & Valiant, L. G. (1994). Cryptographic limitations on learning Boolean formulae and finite automata. *Journal of the Association for Computing Machinery, 41*, 67–95.

Kearns, M. J. & Schapire, R. E. (1990). Efficient distribution-free learning of probabilistic concepts. *31st Annual Symposium on Foundations of Computer Science* (pp. 382–391). To appear, *Journal of Computer and System Sciences.*

Linial, N., Mansour, Y., & Nisan, N. (1993). Constant depth circuits, Fourier transform, and learnability. *Journal of the Association for Computing Machinery, 40*, 607–620.

Pitt, L. & Valiant, L. G. (1988). Computational limitations on learning from examples. *Journal of the Association for Computing Machinery, 35*, 965–984.

Pollard, D. (1984). *Convergence of Stochastic Processes.* Springer-Verlag.

Rissanen, J., Speed, T. P., & Yu, B. (1992). Density estimation by stochastic complexity. *IEEE Transactions on Information Theory, 38*, 315–323.

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning, 5*, 197–227.

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

Valiant, L. G. (1985). Learning disjunctions of conjunctions. *Proceedings of the 9th International Joint Conference on Artificial Intelligence* (pp. 560–566).

Vapnik, V. N. (1982). *Estimation of Dependences Based on Empirical Data.* Springer-Verlag.

White, H. (1989). Learning in artificial neural networks: A statistical perspective. *Neural Computation, 1*, 425–464.

Yamanishi, K. (1992a). A learning criterion for stochastic rules. *Machine Learning, 9*, 165–203.

Yamanishi, K. (1992b). Learning nonparametric densities in terms of finite dimensional parametric hypotheses. *IEICE Transactions: D Information and Systems, E75D*, 459–469.