Efficient Noise-Tolerant Learning From Statistical Queries

Michael Kearns *
AT&T Laboratories – Research
Florham Park, New Jersey

June 15, 1998

1 Introduction

In this paper, we study the extension of Valiant's learning model [32] in which the positive or negative classification label provided with each random example may be corrupted by random noise. This extension was first examined in the learning theory literature by Angluin and Laird [1], who formalized the simplest type of white label noise and then sought algorithms tolerating the highest possible rate of noise. In addition to being the subject of a number of theoretical studies [1, 22, 31, 17], the classification noise model has become a common paradigm for experimental machine learning research.

Angluin and Laird provided an algorithm for learning boolean conjunctions that tolerates a noise rate approaching the information-theoretic barrier of 1/2. Subsequently, there have been some isolated instances of efficient noise-tolerant algorithms [20, 27, 29], but little work on characterizing which classes can be efficiently learned in the presence of noise, and no general transformations of Valiant model algorithms into noise-tolerant algorithms. The primary contribution of the present paper is in making significant progress in both of these areas.

We identify and formalize an apparently rather weak sufficient condition on learning algorithms in Valiant's model that permits the immediate derivation of noise-tolerant learning algorithms. More precisely, we define a natural restriction on Valiant model algorithms that allows them to be reliably and efficiently simulated in the presence of arbitrarily large rates of classification noise. This allows us to obtain efficient noise-tolerant learning algorithms for practically every concept class for which an efficient learning algorithm in the

*Author's address: AT&T Laboratories – Research, Room A235, 180 Park Avenue, Florham Park, New Jersey 07932. Electronic mail address: mkearns@research.att.com.

original noise-free Valiant model is known. A notable exception is the class of parity concepts, whose properties we investigate in some detail.

Our sufficient condition is formalized by the introduction of a new model of *learning from statistical queries*, in which the standard Valiant model oracle $EX(f,\mathcal{D})$ (giving random examples of the target concept f with respect to an input distribution $\mathcal{D}$ over X) is replaced by the weaker oracle $STAT(f,\mathcal{D})$. This oracle, rather than supplying the learning algorithm with individual random examples, instead provides accurate estimates for probabilities over the sample space generated by $EX(f,\mathcal{D})$. Taking as input a query of the form (χ,α), where $\chi=\chi(x,\ell)$ is any boolean function over inputs $x\in X$ and $\ell\in\{0,1\}$, $STAT(f,\mathcal{D})$ returns an estimate for the probability that $\chi(x,f(x))=1$ (when x is drawn according to $\mathcal{D}$). This estimate is accurate within additive error $\alpha\in[0,1]$, which we call the *allowed approximation error* of the query.

The natural notion of efficiency in such a model should assign high cost to queries in which χ is computationally expensive to evaluate, and to queries in which α is small. We shall formalize this shortly (Sections 2, 3 and 4 define the Valiant, classification noise and statistical query models respectively), and the result will be a model which is weaker than the standard Valiant model, in the sense that statistical query algorithms can be trivially simulated given access to the noise-free examples oracle $EX(f,\mathcal{D})$.

In the statistical query model, we are effectively restricting the way in which a learning algorithm may use a random sample, and we thus capture the natural notion of learning algorithms that construct a hypothesis based on statistical properties of large samples rather than on the idiosyncrasies of a particular sample. Note that algorithms in this model may also estimate conditional probabilities by expanding the conditional probability as the ratio of two simple probabilities.

One of our main theorems, given in Section 5, is that any class efficiently learnable from statistical queries is also efficiently learnable with classification noise. The theorem holds even with respect to particular distributions or classes of distributions. This latter property is important since many of the most powerful positive results in the Valiant model hold only for special but natural distributions, thus allowing us to obtain efficient noise-tolerant algorithms for these same distributions.

We give many applications of this result in Section 6. In addition to unifying all previous analyses of learning with noise in the Valiant model (since all of the proposed algorithms for noise-tolerant learning can be shown to fall into the statistical query model), we use our new model to obtain efficient noise-tolerant learning algorithms for many concept classes for which no such algorithm was previously known. Examples include learning perceptrons (linear separators) with noise with respect to any radially symmetric distribution; learning conjunctions with noise with only a logarithmic sample size dependence on the number of irrelevant variables; learning n-dimensional axis-aligned rectangles with noise; learning AC^0 with noise with respect to the uniform distribution in time $O(n^{poly(\log n)})$ (for which the algorithm of Linial, Mansour and Nisan [23] can be shown to fall into the statistical query model without modification); and many others.

The fact that practically every concept class known to be efficiently learnable in the Valiant model can in fact be learned from statistical queries (and thus with classification noise) raises the natural question of whether the two models are equivalent. We answer this question negatively in Section 7 by proving that the class of parity concepts, known to be efficiently learnable in the Valiant model, cannot be efficiently learned from statistical queries. The class of parity concepts is also notorious for having no known efficient noise-tolerant algorithm.

In Section 8 we investigate query complexity in our model. Our interest here centers on the tradeoff between the number of statistical queries that must be made, and the required accuracy of these queries. For instance, translation of Valiant model sample size lower bounds [7, 9] into the statistical query model leaves open the possibility that some classes might be learned with just a single statistical query of sufficiently small allowed approximation error. Here we dismiss such possibilities, and provide a much stronger lower bound by proving that for any concept class of Vapnik-Chervonenkis dimension d, there is a distribution on which a statistical query algorithm must make at least $\Omega(d/\log d)$ queries, each with allowed approximation error at most $O(\epsilon)$, in order to obtain a hypothesis with error less than ϵ.

In Section 9 we show the equivalence of learning in the classification noise model and learning in a more realistic model with a variable noise rate, and Section 10 concludes with some open problems.

We note that since the original conference publication of these results, a great many results have been obtained using the statistical query model, including work by Aslam and Decatur [2, 3]. The noise simulation result presented here has also appeared in more tutorial form [21].

2 The Valiant Learning Model

Let $\mathcal{F}$ be a class of $\{0,1\}$-valued functions (also called *concepts*) over an input space X. In trying to design a learning algorithm for the class $\mathcal{F}$, we assume that there is a fixed but arbitrary and unknown *target distribution* $\mathcal{D}$ over X that governs the generation of random examples. More precisely, when executed on the *target concept* $f \in \mathcal{F}$, a learning algorithm will be given access to an oracle $EX(f, \mathcal{D})$ that on each call draws an input x randomly and independently according to $\mathcal{D}$, and returns the *(labeled) example* $\langle x, f(x)\rangle$.

Once we have fixed the target concept f and target distribution $\mathcal{D}$, there is a natural measure of the error of any other concept h with respect to f and $\mathcal{D}$. Thus, we define $error(h) = \mathbf{Pr}_{x\in\mathcal{D}}[f(x) \neq h(x)]$ (throughout the paper, the notation $x \in \mathcal{D}$ indicates that x is drawn randomly according to the distribution $\mathcal{D}$). Notice that we have dropped the dependence of $error(h)$ on f and $\mathcal{D}$ for notational brevity.

We assume that all inputs x are of some common length n. Here the length of inputs is typically measured by the number of components; the most common examples of n are the boolean hypercube $\{0,1\}^n$ and n-dimensional real space R^n. We also assume a mapping $size(f)$ that measures the size or complexity of representing each $f \in \mathcal{F}$ in some fixed encoding scheme. Thus, $size(f)$ will measure the size of the smallest representation (there may be many) of the target concept f in the representation scheme $\mathcal{H}$ used by the learning algorithm [1], and we will allow the algorithm running time polynomial in the input length n and $size(f)$.

Definition 1 *(Learning in the Valiant Model) Let $\mathcal{F}$ be a class of concepts over X, and let $\mathcal{H}$ be a class of representations of concepts over X. We say that $\mathcal{F}$* is efficiently learnable using $\mathcal{H}$ in the Valiant model *if there exists a learning algorithm L and a polynomial $p(\cdot,\cdot,\cdot,\cdot)$ such that for any $f \in \mathcal{F}$ over inputs of length n, for any distribution $\mathcal{D}$ over X, and for any $0 < \epsilon \leq 1$ and $0 < \delta \leq 1$, the following holds: if L is given inputs ϵ, δ, n and $size(f)$, and L is given access to $EX(f,\mathcal{D})$, then L will halt in time bounded by $p(1/\epsilon, 1/\delta, n, size(f))$ and output a representation in $\mathcal{H}$ of a concept h that with probability at least $1-\delta$ satisfies $error(h) \leq \epsilon$. This probability is taken over the random draws from $\mathcal{D}$ made by $EX(f,\mathcal{D})$ and any internal randomization of L. We call ϵ the* accuracy parameter *and δ the* confidence parameter*.*

3 The Classification Noise Model

The well-studied classification noise model [1, 22, 17, 31, 20, 27, 29] is an extension of the Valiant model intended to capture the simplest type of white noise in the labels seen by the learner. We introduce a parameter $0 \leq \eta < 1/2$ called the *noise rate*, and replace the oracle $EX(f,\mathcal{D})$ with the faulty oracle $EX^{\eta}_{CN}(f,\mathcal{D})$ (where the subscript is the acronym for Classification Noise). On each call, $EX^{\eta}_{CN}(f,\mathcal{D})$ first draws an input x randomly according to $\mathcal{D}$ (just as in the noise-free case). The oracle then flips a coin whose probability of *heads* is $1-\eta$ and whose probability of *tails* is η. If the outcome is *heads*, the oracle returns the uncorrupted example $\langle x, f(x)\rangle$; but if the outcome is *tails*, the oracle returns the erroneous example $\langle x, \neg f(x)\rangle$. Note that in this model, errors occur only in the labels given to the learner; the inputs x given to the learner remain independently distributed according to $\mathcal{D}$. Other models allowing corruption of the input as well as the label have been studied previously [33, 17], with considerably less success in finding efficient error-tolerant algorithms. Here we will

[1] The choice of representation used by the learning algorithm can sometimes be quite significant, as previous results have demonstrated concept classes $\mathcal{F}$ for which the choice of hypothesis representation can mean the difference between intractability and efficient learning [25, 18].

concentrate primarily on the classification noise model, although in Section 9 we will examine a more realistic extension of this model.

Despite the noise in the labels, the learning algorithm's goal remains that of finding a hypothesis concept h satisfying $error(h) = \mathbf{Pr}_{x\in\mathcal{D}}[f(x) \neq h(x)] \leq \epsilon$. Furthermore, we would like the algorithm to tolerate the highest possible noise rate. Obviously, as η approaches $1/2$ learning becomes more difficult because the label seen by the learner approaches an unbiased coin flip. Thus we must allow learning algorithms to have a polynomial time dependence on the quantity $1/(1-2\eta)$, which is simply proportional to the inverse of the distance of η from $1/2$.

One final issue is what information the learner should be provided about the exact value of η. For simplicity in the current paper, we adopt the convention of Angluin and Laird [1] and assume that the learning algorithm is given only an upper bound η_b (where $1/2 > \eta_b \geq \eta$), and is given polynomial time dependence on $1/(1-2\eta_b)$. For all of the results presented here, even this assumption can be removed using a technique due to Laird [22].

Definition 2 *(Learning with Noise) Let $\mathcal{F}$ be a class of concepts over X, and let $\mathcal{H}$ be a class of representations of concepts over X. We say that $\mathcal{F}$* is efficiently learnable with noise using *$\mathcal{H}$ if there exists a learning algorithm L and a polynomial $p(\cdot,\cdot,\cdot,\cdot,\cdot)$ such that for any $f \in \mathcal{F}$ over inputs of length n, for any distribution $\mathcal{D}$ over X, for any noise rate $0 \leq \eta < 1/2$, and for any $0 < \epsilon \leq 1$ and $0 < \delta \leq 1$, the following holds: if L is given inputs η_b (where $1/2 > \eta_b \geq \eta$), ϵ, δ, n and $size(f)$, and L is given access to $EX^{\eta}_{CN}(f,\mathcal{D})$, then L will halt in time bounded by $p(1/(1-2\eta_b), 1/\epsilon, 1/\delta, n, size(f))$ and output a representation in $\mathcal{H}$ of a concept h that with probability at least $1-\delta$ satisfies $error(h) \leq \epsilon$. This probability is taken over the random draws from $\mathcal{D}$, the random noise bits of $EX^{\eta}_{CN}(f,\mathcal{D})$ and any internal randomization of L.*

4 The Statistical Query Model

We now introduce a new learning model that is related to but apparently weaker than the Valiant model, and is designed to limit the ways in which the learning algorithm can use the random examples it receives from the oracle $EX(f,\mathcal{D})$. The restriction we would like to enforce is that learning be based not on the particular properties of individual random examples, but instead on the global statistical properties of large samples. Such an approach to learning seems intuitively more robust than algorithms that are willing to make radical alterations to their hypotheses on the basis of individual examples.

To formalize this restriction, we introduce a new oracle $STAT(f,\mathcal{D})$ that will replace the standard examples oracle $EX(f,\mathcal{D})$. It will be helpful throughout the paper to think of $STAT(f,\mathcal{D})$ as an intermediary oracle (standing between the learning algorithm and the examples oracle $EX(f,\mathcal{D})$) whose goal is to enforce the restriction on the learner's use of examples described above. Unlike $EX(f,\mathcal{D})$, $STAT(f,\mathcal{D})$ is an oracle that the learner interacts with. The oracle $STAT(f,\mathcal{D})$ takes as input a *statistical query* of the form (χ,α). Here χ is any mapping of a labeled example to $\{0,1\}$ (thus $\chi : X \times \{0,1\} \to \{0,1\}$) and $\alpha \in [0,1]$.

We interpret a query (χ,α) as a request for the value $P_\chi = \mathbf{Pr}_{x\in\mathcal{D}}[\chi(x,f(x)) = 1]$; we will abbreviate the right side of this equation by $\mathbf{Pr}_{EX(f,\mathcal{D})}[\chi = 1]$ to emphasize that the distribution on examples is simply that generated by the oracle $EX(f,\mathcal{D})$. Thus, each query is a request for the probability of some event on the distribution generated by $EX(f,\mathcal{D})$. However, the oracle $STAT(f,\mathcal{D})$ will not return the exact value of P_χ but only an approximation, and the role of α is to quantify the amount of error the learning algorithm is willing to tolerate in this approximation. More precisely, on query (χ,α) the oracle $STAT(f,\mathcal{D})$ is allowed to return any value $\hat{P}_\chi$ that satisfies $P_\chi - \alpha \leq \hat{P}_\chi \leq P_\chi + \alpha$. We refer to α as the *allowed approximation error* of the query.

At this point, it should be clear that given access to the oracle $EX(f,\mathcal{D})$, it is a simple matter to simulate the behavior of the oracle $STAT(f,\mathcal{D})$ on a query (χ,α) with probability at least $1-\delta$: we draw from $EX(f,\mathcal{D})$ a sufficient number of random labeled examples $\langle x, f(x)\rangle$ and use the fraction of the examples for which $\chi(x,f(x)) = 1$ as our estimate $\hat{P}_\chi$ of P_χ. The number of calls to $EX(f,\mathcal{D})$ required will be polynomial in $1/\alpha$ and $\log(1/\delta)$, and the time required will be polynomial in the time required to evaluate χ, and in $1/\alpha$ and $\log(1/\delta)$. To ensure that efficient algorithms for learning using $STAT(f,\mathcal{D})$ can be efficiently simulated using $EX(f,\mathcal{D})$, we must place natural restrictions on α (namely, that it is an inverse polynomial in the learning problem parameters) and on χ (namely, that it can be evaluated in polynomial time). Thus we require that algorithms only ask $STAT(f,\mathcal{D})$ for estimates of sufficiently simple probabilities, with sufficiently coarse resolution. This is done in the following definition, which formalizes the model of learning from statistical queries. The intuition that algorithms with access to $STAT(f,\mathcal{D})$ can be simulated given access to $EX(f,\mathcal{D})$ is then formalized in greater detail as Theorem 1 below.

Definition 3 *(Learning from Statistical Queries) Let $\mathcal{F}$ be a class of concepts over X, and let $\mathcal{H}$ be a class of representations of concepts over X. We say that $\mathcal{F}$* is efficiently learnable from statistical queries *using $\mathcal{H}$ if there exists a learning algorithm L and polynomials $p(\cdot,\cdot,\cdot)$, $q(\cdot,\cdot,\cdot)$ and $r(\cdot,\cdot,\cdot)$ such that for any $f \in \mathcal{F}$ over inputs of length n, for any distribution $\mathcal{D}$ over X, and for any $0 < \epsilon \leq 1$, the following holds: if L is given inputs ϵ, n and $size(f)$, and L is given access to $STAT(f,\mathcal{D})$, then* (1) *for every query (χ,α) made by L, χ can be evaluated in time $q(1/\epsilon, n, size(f))$ and $1/\alpha$ is bounded by $r(1/\epsilon, n, size(f))$, and* (2) *$L$ will halt in time bounded by $p(1/\epsilon, n, size(f))$ and output a representation in $\mathcal{H}$ of a concept h that satisfies $error(h) \leq \epsilon$.*

Later in the paper, we will also consider the variant of the statistical query model in which the learner is provided with access to unlabeled inputs according to $\mathcal{D}$, in addition to the oracle $STAT(f,\mathcal{D})$. This is because unlabeled inputs are sometimes crucial for learning (for instance, to estimate the important regions of the distribution), and our main theorem (Theorem 3) still holds for this variant. This is most easily seen by noting that algorithms in the noise model still have access to $\mathcal{D}$ simply by ignoring the noisy labels returned by $EX^{\eta}_{CN}(f,\mathcal{D})$.

In the statistical query model, it will sometimes be helpful to identify the class of queries from which a learning algorithm chooses. Thus, we say that *$\mathcal{F}$ is efficiently learnable from statistical queries using $\mathcal{H}$ with query space $\mathcal{Q}$* if the above definition can be met by an algorithm that only makes queries (χ,α) satisfying $\chi \in \mathcal{Q}$.

Remark 1: No Confidence Parameter. Note that the confidence parameter δ is absent in this definition of learning. This is because the main purpose of δ in the Valiant model is to allow the learning algorithm a small probability of failure due to an unrepresentative sample from $EX(f,\mathcal{D})$. Since we have now replaced $EX(f,\mathcal{D})$ by the oracle $STAT(f,\mathcal{D})$, whose behavior is always guaranteed to meet the approximation criterion $P_\chi - \alpha \leq \hat{P}_\chi \leq P_\chi + \alpha$, we no longer need to allow this failure probability [2].

Remark 2: Conditional Probabilities. Note that although the statistical query model only provides the learner with an oracle for estimating probabilities, one can also design algorithms that estimate conditional probabilities $\mathbf{Pr}_{EX(f,\mathcal{D})}[\chi_1 = 1|\chi_2 = 1]$, by expanding the conditional probability as a ratio of two simple probabilities, and obtaining sufficiently accurate estimates of the numerator and denominator to yield an additively accurate estimate of the ratio. Such algorithms must be prepared for the case that the probability $\mathbf{Pr}_{EX(f,\mathcal{D})}[\chi_2 = 1]$ of the conditioning event is too small; but this is typically not a restriction, since an algorithm with access to $EX(f,\mathcal{D})$ would also be unable to obtain an estimate for the conditional probability in this case. Some of the algorithms we discuss will take advantage of this way of estimating conditional probabilities. The estimation of conditional probabilities in the statistical query model is also discussed by Aslam and Decatur [2, 3].

Before we proceed with the technical portion of the paper, some final comments regarding all of the models we have defined are in order. First of all, for M representing any of the three models (Valiant, noise or statistical query) we will simply say that *$\mathcal{F}$ is efficiently learnable in model M* to mean that $\mathcal{F}$ is learnable using $\mathcal{H}$ for some $\mathcal{H}$ in which each hypothesis over inputs of length n can be evaluated in time polynomial in n.

Secondly, we will have occasion to study some common variants of these models. For some classes we do not know a polynomial-time learning algorithm but instead have an algorithm with at least a nontrivial time bound; in such cases we drop the modifier "efficient" and instead say that the class is learnable in model M within some explicitly stated time bound. For some classes we

[2] We could still keep δ in order to allow for a probability of failure in randomized learning algorithms, but for simplicity choose not to do so since all the algorithms we discuss are deterministic.

have an efficient algorithm only for a particular distribution $\mathcal{D}$ (or a class of distributions); in such cases we say that the class is learnable *with respect to* $\mathcal{D}$ (or with respect to the class of distributions) in model M.

Finally, we will need the following standard definition. For any concept class $\mathcal{F}$ and a set of inputs $S = \{x_1, \ldots, x_d\}$, we say that $\mathcal{F}$ *shatters* S if for all of the 2^d possible binary labelings of the points in S, there is a concept in $\mathcal{F}$ that agrees with that labeling. The *Vapnik-Chervonenkis dimension* of $\mathcal{F}$ is the cardinality of the largest set shattered by $\mathcal{F}$ [34].

5 Simulating Statistical Queries Using Noisy Examples

Our first theorem formalizes the intuition given above that learning from statistical queries implies learning in the noise-free Valiant model. The proof of this theorem is omitted for brevity, but employs standard Chernoff bound and uniform convergence analyses [7]. The key idea in the simulation is to draw a single large sample with which to estimate all probabilities requested by the statistical query algorithm.

Theorem 1 *Let $\mathcal{F}$ be a class of concepts over X, and let $\mathcal{H}$ be a class of representations of concepts over X. Suppose that $\mathcal{F}$ is efficiently learnable from statistical queries using $\mathcal{H}$ by algorithm L. Then $\mathcal{F}$ is efficiently learnable using $\mathcal{H}$ in the Valiant model, and furthermore:*
• *(Finite $\mathcal{Q}$ case) If L uses a finite query space $\mathcal{Q}$ and α is a lower bound on the allowed approximation error for every query made by L, then the number of calls to $EX(f, \mathcal{D})$ required to learn in the Valiant model is $O(1/\alpha^2 \log(|\mathcal{Q}|/\delta))$.*
• *(Finite VC dimension case) If L uses a query space $\mathcal{Q}$ of Vapnik-Chervonenkis dimension d and α is a lower bound on the allowed approximation error for every query made by L, then the number of calls to $EX(f, \mathcal{D})$ required to learn in the Valiant model is $O(d/\alpha^2 \log(1/\delta))$.*

Note that in the statement of Theorem 1, the sample size dependence on ϵ is hidden in the sense that we expect α and possibly the query class to depend on ϵ.

Theorem 1 shows that the statistical query model identifies one approach to learning in the noise-free Valiant model. We now derive a less obvious and considerably more useful result: namely, that algorithms for learning from statistical queries can in fact be reliably and efficiently simulated given access only to the noisy example oracle $EX^{\eta}_{CN}(f, \mathcal{D})$. The key to this result is the following lemma, which describes how any probability $\mathbf{Pr}_{EX(f,\mathcal{D})}[\chi = 1]$ can be expressed in terms of probabilities over the sample space generated by $EX^{\eta}_{CN}(f, \mathcal{D})$.

Lemma 2 *Let $\chi : X \times \{0,1\} \to \{0,1\}$. Then for any $0 \leq \eta < 1/2$, the probability $\mathbf{Pr}_{EX(f,\mathcal{D})}[\chi = 1]$ can be written as an expression involving only η and probabilities over the sample space generated by $EX^{\eta}_{CN}(f,\mathcal{D})$.*

Proof: The key idea of the proof is to define a partition of the input space X into two disjoint regions X_1 and X_2 as follows: X_1 consists of those points $x \in X$ such that $\chi(x,0) \neq \chi(x,1)$, and X_2 consists of those points $x \in X$ such that $\chi(x,0) = \chi(x,1)$. Thus, X_1 is the set of all inputs such that the label matters in determining the value of χ, and X_2 is the set of all inputs such that the label is irrelevant in determining the value of χ. Note that X_1 and X_2 are disjoint and $X_1 \cup X_2 = X$.

Having defined the regions X_1 and X_2, we can now define the induced distributions on these regions. Thus, if we let $p_1 = \mathcal{D}[X_1]$ and $p_2 = \mathcal{D}[X_2]$ (so $p_1 + p_2 = 1$), then for any subset $S \subseteq X_1$ we define $\mathcal{D}_1[S] = \mathcal{D}[S]/p_1$ and for any subset $S \subseteq X_2$ we define $\mathcal{D}_2[S] = \mathcal{D}[S]/p_2$. Throughout the proof, we will use the following abbreviated notation: P_χ will denote $\mathbf{Pr}_{EX(f,\mathcal{D})}[\chi = 1]$, P_χ^{CN} will denote $\mathbf{Pr}_{EX^{\eta}_{CN}(f,\mathcal{D})}[\chi = 1]$, P_χ^1 will denote $\mathbf{Pr}_{EX(f,\mathcal{D}_1)}[\chi = 1]$, and P_χ^2 will denote $\mathbf{Pr}_{EX(f,\mathcal{D}_2)}[\chi = 1]$. Notice that for P_χ^{CN}, the label given as the second input to χ is potentially noisy.

In a moment we will derive an expression for P_χ (which is the quantity we would like to estimate) involving only η, p_1, p_2, P_χ^{CN}, and P_χ^2. We first argue that all these quantities (excluding η, which we shall deal with separately) can in fact be estimated from the noisy oracle $EX^{\eta}_{CN}(f,\mathcal{D})$.

First, note that it is easy to estimate P_χ^{CN} from calls to $EX^{\eta}_{CN}(f,\mathcal{D})$, because this probability is already defined with respect to the noisy oracle.

Next, note that it is easy to estimate p_1 (and therefore $p_2 = 1 - p_1$) using only calls to $EX^{\eta}_{CN}(f,\mathcal{D})$: given a potentially noisy example $\langle x, \ell \rangle$ from $EX^{\eta}_{CN}(f,\mathcal{D})$, we ignore the label ℓ and test whether $\chi(x,0) \neq \chi(x,1)$. If so, then $x \in X_1$, otherwise $x \in X_2$. Thus for a large enough sample the fraction of the x falling in X_1 will be a good estimate for p_1 via a standard Chernoff bound analysis.

Finally, P_χ^2 can be estimated from $EX^{\eta}_{CN}(f,\mathcal{D})$: we simply sample pairs $\langle x, \ell \rangle$ returned by the noisy oracle, keeping only those inputs x that fall in X_2 (using the membership test $\chi(x,0) = \chi(x,1)$). For such an x, the value of χ is invariant to the label, so we can just compute the fraction of the sampled $x \in X_2$ for which $\chi(x,0) = \chi(x,1) = 1$ as our estimate for P_χ^2.

Now to derive the desired expression for P_χ, consider the probability that χ is 1 when the input to χ is obtained from the noisy oracle $EX^{\eta}_{CN}(f,\mathcal{D})$ with noise rate η. We may write

$$
\begin{aligned}
P_\chi^{CN} &= (1-\eta)P_\chi + \eta(p_1 \mathbf{Pr}_{x \in \mathcal{D}_1, \ell \leftarrow \neg f(x)}[\chi(x,\ell) = 1] \\
&\qquad + p_2 \mathbf{Pr}_{x \in \mathcal{D}_2, \ell \leftarrow \neg f(x)}[\chi(x,\ell) = 1]). \qquad (1)
\end{aligned}
$$

The intuition behind this expression is as follows: on a call to $EX^{\eta}_{CN}(f,\mathcal{D})$, with probability $1 - \eta$ there is no misclassification, in which case the call to

$EX^{\eta}_{CN}(f, \mathcal{D})$ behaves identically to a call to $EX(f, \mathcal{D})$. With probability η, however, there is a misclassification. Now given that a misclassification occurs, the label provided is $\neg f(x)$, and there is probability p_1 that the input is drawn from X_1 (and thus is distributed according to $\mathcal{D}_1$), and probability p_2 that the input is drawn from X_2 (and thus is distributed according to $\mathcal{D}_2$). We now derive alternative expressions for three of the terms in Equation (1) for substitution.

First, note that we may write $\mathbf{Pr}_{x \in \mathcal{D}_1, \ell \leftarrow \neg f(x)}[\chi = 1] = \mathbf{Pr}_{EX(f,\mathcal{D}_1)}[\chi = 0] = 1 - P^1_{\chi}$ because in X_1, reversing the label and computing χ is equivalent to leaving the label unaltered and reversing the value of χ.

Second, we may also write $\mathbf{Pr}_{x \in \mathcal{D}_2, \ell \leftarrow \neg f(x)}[\chi = 1] = P^2_{\chi}$ because in X_2 the label is unimportant for the value of χ.

Third, we may make the expansion $P_{\chi} = p_1 P^1_{\chi} + p_2 P^2_{\chi}$.

Making these substitutions into Equation (1), some simple algebra yields

$$\begin{aligned} P^{CN}_{\chi} &= (1-\eta)(p_1 P^1_{\chi} + p_2 P^2_{\chi}) \\ &\quad + \eta(p_1(1 - P^1_{\chi}) + p_2 P^2_{\chi}) \\ &= (1-2\eta) p_1 P^1_{\chi} + p_2 P^2_{\chi} + \eta p_1. \end{aligned} \tag{2}$$

By solving Equation (2) for P^1_{χ} we obtain:

$$P^1_{\chi} = (1/(1-2\eta)p_1)(P^{CN}_{\chi} - p_2 P^2_{\chi} - \eta p_1). \tag{3}$$

Finally, again using the expansion $P_{\chi} = p_1 P^1_{\chi} + p_2 P^2_{\chi}$ and substituting for P^1_{χ} using Equation (3) we obtain

$$\begin{aligned} P_{\chi} &= (1/(1-2\eta)) P^{CN}_{\chi} + (1 - 1/(1-2\eta)) p_2 P^2_{\chi} \\ &\quad - (\eta/(1-2\eta)) p_1. \end{aligned} \tag{4}$$

Equation (4) has the desired form: an expression for P_{χ} in terms of P^{CN}_{χ}, p_1, p_2, P^2_{χ} and η. $\square$(Lemma 2)

Equation (4) suggests an approach for simulating the oracle $STAT(f, \mathcal{D})$ using only the noisy oracle $EX^{\eta}_{CN}(f, \mathcal{D})$: given any query (χ, α) for $STAT(f, \mathcal{D})$, use $EX^{\eta}_{CN}(f, \mathcal{D})$ to obtain sufficiently accurate estimates of each quantity on the right hand side of Equation (4), and then solve to get an accurate estimate for P_{χ}. This is exactly the approach taken in the theorem that follows, which is one of our main results. The main details to be worked out are a sensitivity analysis of Equation (4) to ensure that additively accurate estimates of each quantity on the right hand side provide a sufficiently accurate estimate of P_{χ}, the related issue of guessing a good approximation to the noise rate, and an analysis of the required sample sizes.

Theorem 3 *Let $\mathcal{F}$ be a class of concepts over X, and let $\mathcal{H}$ be a class of representations of concepts over X. Suppose that $\mathcal{F}$ is efficiently learnable from statistical queries using $\mathcal{H}$ by algorithm L. Then $\mathcal{F}$ is efficiently learnable with* *noise using $\mathcal{H}$, and furthermore:*

• *(Finite $\mathcal{Q}$ case) If L uses a finite query space $\mathcal{Q}$ and α is a lower bound on the allowed approximation error for every query made by L, then the number of calls to $EX_{CN}^{\eta}(f,\mathcal{D})$ required to learn with noise is*

$$O((1/\alpha(1-2\eta_b))^2 \log(|\mathcal{Q}|/\delta) + 1/\epsilon^2 \log(1/\delta\alpha(1-2\eta_b))).$$

• *(Finite VC dimension case) If L uses a query space $\mathcal{Q}$ of Vapnik-Chervonenkis dimension d and α is a lower bound on the allowed approximation error for every query made by L, then the number of calls to $EX_{CN}^{\eta}(f,\mathcal{D})$ required to learn with noise is*

$$O(d(1/\alpha(1-2\eta_b))^2 \log(1/\delta) + 1/\epsilon^2 \log(1/\delta\alpha(1-2\eta_b))).$$

Proof: Let η_b be the given bound on η, and suppose we wish to simulate a query (χ, α) for the oracle $STAT(f,\mathcal{D})$. What is needed first is a sensitivity analysis of the right hand side of Equation (4). We have already sketched in the proof of Lemma 2 how to obtain estimates with small additive error for p_1, p_2, P_χ^{CN}, and P_χ^2. We will use η_b to get a good estimate $\hat{\eta}$ for η in a way to be described momentarily; for now we analyze how accurate $\hat{\eta}$ must be in order to allow substituting $1/(1-2\hat{\eta})$ for $1/(1-2\eta)$ without incurring too much error in Equation (4).

Lemma 4 *Let $0 \le \eta, \hat{\eta} < 1/2$ and $0 \le \Delta \le 1$ satisfy $\eta - \Delta \le \hat{\eta} \le \eta + \Delta$. Let $0 \le \alpha \le 1$. Then there exists a constant c such that if $\Delta \le (c\alpha/2)(1-2\eta)^2$, then*

$$1/(1-2\eta) - \alpha \le 1/(1-2\hat{\eta}) \le 1/(1-2\eta) + \alpha. \tag{5}$$

Proof: Taking the extreme allowed values for $\hat{\eta}$ gives

$$1/(1-2(\eta-\Delta)) \le 1/(1-2\hat{\eta}) \le 1/(1-2(\eta+\Delta)).$$

Taking the leftmost inequality of this equation, we see that the leftmost inequality of Equation (5) will be satisfied if we have $1/(1-2\eta) - \alpha \le 1/(1-2(\eta-\Delta))$. Solving for constraints on Δ gives

$$2\Delta \le 1/(1/(1-2\eta) - \alpha) - (1-2\eta).$$

If we set $x = 1/(1-2\eta)$ and $f(x) = 1/x$ we obtain $2\Delta \le f(x-\alpha) - f(x)$. This suggests analysis via the derivative of f. Now $f'(x) = -1/x^2$ and we may write $f(x-\alpha) \le f(x) + c\alpha/x^2$ for some constant $c > 0$, for all $x \in [1, \infty]$. (This is the range of interest for $x = 1/(1-2\eta)$, corresponding to $\eta = 0$ and $\eta = 1/2$.) This gives $\Delta \le c\alpha/2x^2 = (c\alpha/2)(1-2\eta)^2$. An identical analysis gives the same bound on Δ for achieving the rightmost inequality in Equation (5). □(Lemma 4)

Thus, assume for the moment that we have found a value $\hat{\eta}$ satisfying $\eta - \Delta \leq \hat{\eta} \leq \eta + \Delta$ where $\Delta = (c\alpha/2)(1-2\eta_b)^2 \leq (c\alpha/2)(1-2\eta)^2$ as in the statement of Lemma 4. Then provided we have estimates for p_1, p_2, P_χ^{CN} and P_χ^2 that have additive error bounded by $\alpha(1-2\eta_b)$, it can be shown using Lemma 4 and some algebra that solution of Equation (4) using these estimates provides an estimate $\hat{P}_\chi$ of P_χ with additive error $O(\alpha)$. As in Theorem 1, we can use a single shared sample to estimate all queries made to $STAT(f, \mathcal{D})$, resulting in only logarithmic dependence on the query space cardinality or linear dependence on its Vapnik-Chervonenkis dimension; this dependence is obtained via standard techniques [7].

To find the assumed value $\hat{\eta}$, we simply try all values $\hat{\eta} = i\Delta$ for $i = 1, 2, \ldots, 1/2\Delta$ [3]. Clearly for one of these tries, η will be within Δ of the true noise rate η, and the above simulation will yield (with high probability) estimates for all queries to $STAT(f, \mathcal{D})$ accurate to within the desired additive error α. Also note that for each χ, the quantities p_1, p_2, P_χ^{CN} and P_χ^2 do not depend on η, so our simulation needs to estimate these quantities only once. Given these estimates, we then run L repeatedly, each time using the same fixed estimates but a different guess for $\hat{\eta}$ to solve Equation (4) on each query. This will result in a series of hypotheses $h_1, \ldots, h_{1/2\Delta}$ output by the runs of L, one of which has error smaller than ϵ with high probability. It is not difficult to show that given a sufficiently large sample from $EX_{CN}^{\eta}(f, \mathcal{D})$, the h_i that best agrees with the noisy examples has error smaller than ϵ. □(Theorem 3)

We again note that the assumption of an upper bound η_b on the noise rate can be eliminated [22].

To summarize, Theorem 3 shows that if we can find an algorithm for efficient learning in the statistical query model, we immediately obtain an algorithm for efficient learning in the noise model. Furthermore, examination of the proof shows that the theorem holds even with respect to specific distributions (that is, efficient statistical query learning for a particular distribution implies efficient learning with noise for the same distribution), and also for the variant of the statistical query model in which the learner is given access to an oracle for unlabeled inputs from $\mathcal{D}$ in addition to access to the oracle $STAT(f, \mathcal{D})$ (see Remark 3 following Definition 3). These stronger statements of the theorem will both be used in the applications given in the following section.

6 Efficient Noise-Tolerant Learning Algorithms

In this section, we give evidence of the power of Theorem 3 by outlining some of its many applications. Perhaps the most important message to be gleaned is that the model of learning from statistical queries appears to be quite general, in the sense that it encompasses practically all of the concept classes known to be efficiently learnable in the Valiant model (and the Valiant model with restricted

[3] An improved method for finding $\hat{\eta}$ has recently been given by Aslam and Decatur [3].

distributions). Thus, practically every class known to be efficiently learnable is in fact efficiently learnable with noise. One of the few and notable apparent exceptions to this phenomenon is examined in the following section.

We first give a partial list of the many algorithms in the Valiant model literature that can be modified to obtain algorithms in the statistical query model with relatively modest effort. Among others, the list includes Valiant's algorithm for conjunctions [32] and Angluin and Laird's noise-tolerant variant [1] of it; the algorithm of Linial, Mansour and Nisan [23] for learning AC^0 in time $O(n^{poly(\log n)})$ with respect to the uniform distribution in the Valiant model (and its subsequent generalization with respect to product distributions due to Furst, Jackson and Smith [11]); several efficient algorithms for learning restricted forms of DNF with respect to the uniform distribution in the Valiant model [18]; and efficient algorithms for learning unbounded-depth read-once formulae with respect to product distributions in the Valiant model [28, 13]. For all of these classes we can obtain efficient algorithms for learning with noise by Theorem 3; in this list, only for conjunctions [1] and Schapire's work on read-once circuits [28] were there previous noise analyses.

As further evidence for the generality of the statistical query model and to give a flavor for the methods involved, we now spend the remainder of this section describing in high-level detail three cases in which new statistical query algorithms can be obtained with more involved analysis than is required for the above algorithms. As mentioned earlier, without loss of generality we assume these algorithms can obtain estimates for conditional probabilities (see Remark 2 following Definition 3).

6.1 Covering Algorithms and Few Relevant Variables

A number of algorithms for learning in the Valiant model use some variant of a fundamental approach that we shall call the *covering method*. Very briefly and informally, the basic idea is to gradually construct an approximation to the target concept by finding a small set of candidate subfunctions with the property that each candidate covers a significant fraction of the current sample, while not incurring too much error on the portion covered. The hypothesis is then obtained by greedy selection of candidate subfunctions. We will see a somewhat detailed example of this approach momentarily.

A partial list of the efficient algorithms employing some version of this approach is: Rivest's algorithm for learning decision lists [26]; Haussler's algorithm for learning boolean conjunctions with few relevant variables [14]; the algorithm of Blumer et al. for learning a union of axis-aligned rectangles in the Euclidean plane; and the algorithm of Kearns and Pitt [19] for learning pattern languages with respect to product distributions.

In its original form, the covering method is not noise-tolerant, and indeed with the exception of decision lists [20, 27], until now there have been no known efficient noise-tolerant algorithms for the above classes. It is possible to give a general variant of the covering method that works in the statistical query model, thus yielding efficient noise-tolerant learning algorithms for all of these problems. For brevity here, we outline only the main ideas for the particular but representative problem of efficiently learning boolean conjunctions with few relevant variables.

In this problem, the target concept f is some conjunction of an unknown subset of the boolean variables $x_1, \ldots, x_n$ (we assume that f is a monotone conjunction without loss of generality [18]). The expectation is that the number of variables k (not necessarily constant) appearing in f is considerably smaller than the total number of variables n ($k << n$), and we would like to find an efficient algorithm whose sample complexity has the mildest possible dependence on n (note that we cannot avoid time complexity that is at least linear in n since it takes this much time just to read an example).

A solution to this problem in the Valiant model was given by Haussler [14], who made use of the covering method and obtained a sample size with only logarithmic dependence on n. This is of some philosophical interest, since it demonstrates that explicit external mechanisms for "focusing the attention" of the learning algorithm on the relevant variables are not required to learn efficiently with small sample sizes. We are interested in knowing if the same statement holds in the presence of large amounts of noise.

The idea of Haussler's covering approach is to take sufficiently large sets S^+ and S^- of positive and negative examples of f, respectively. The algorithm proceeds in two phases, the first to guarantee consistency with S^+ and the second to guarantee consistency with S^-.

In the first phase, the *candidate set* of variables, which is initially all variables, is pruned to eliminate any x_i which is set to 0 in some positive example in S^+; such a variable directly contradicts the data. This phase ensures that *any* conjunction of candidate variables will be consistent with the set S^+.

In the second phase, a subset of the remaining candidates is chosen that "covers" S^-. To do this, we associate with each candidate x_i the set $S_i^- = \{x \in S^- : x_i = 0\}$. Note that by conjuncting x_i to our hypothesis, we guarantee that our hypothesis will correctly label the examples in S_i^- negatively (that is, we cover S_i^-), and thus these can now be removed from S^-. Haussler's algorithm simply greedily covers S^- using the S_i^-; note that the smallest cover has at most k elements. The sample size bound of his analysis depends linearly on k, but only logarithmically on n.

Our goal is to obtain a similar sample size bound even in the presence of noise; to do this we provide an algorithm for learning from statistical queries along with an analysis of the number of queries required and their allowed approximation error (since it is these two quantities that dictate how many noisy examples are required to simulate the statistical query algorithm).

To modify Haussler's algorithm for the statistical query model, note that the first phase of the algorithm may be thought of as computing a coarse estimate of the probability that $x_i = 0$ in a positive example; Haussler's algorithm eliminates any variable with a non-zero estimate. This almost but does not quite fall into the statistical query model, since the implicit allowed approximation error is too small. Instead we will make calls to $STAT(f, \mathcal{D})$ to estimate for each i the conditional probability $\mathbf{Pr}_{EX(f,\mathcal{D})}[x_i = 0 | f(x) = 1]$ with allowed approximation error $O(\epsilon/r)$, where r will be determined by the analysis. Only variables for which the returned estimate is $O(\epsilon/r)$ are retained as candidates.

To obtain such estimates, we take the ratio of estimates in the conditional expansion. Note that we may assume without loss of generality that the denominator $\mathbf{Pr}_{EX(f,\mathcal{D})}[f(x) = 1]$ is at least ϵ (otherwise the trivial hypothesis that always outputs 0 is already sufficiently accurate). After estimating this denominator within approximation error $O(\epsilon)$, it suffices to estimate the numerator $\mathbf{Pr}_{EX(f,\mathcal{D})}[x_i = 0, f(x) = 1]$ within approximation error $O(\epsilon^2/r)$. Thus, in the first phase of the algorithm we require 1 query of approximation error $O(\epsilon)$ and n queries of approximation ϵ^2/r; of course, in the simulation of the latter queries from noisy examples we may use a single sample as suggested in Theorem 3.

To modify the second phase, note that if at stage i Haussler's algorithm has already chosen variables $x_1, \ldots, x_i$ then for each $j > i$ the fraction of the remaining elements of S^- that are covered by S_j^- can be thought of as an estimate of the probability

$$p_{j,i} = \mathbf{Pr}_{EX(f,\mathcal{D})}[x_j = 0 | f(x) = 0, x_1 = \cdots = x_i = 1]$$

(that is, the probability that $x_j = 0$ given that f is negative but the current hypothesis is positive; note that if the conditioning event has too small a probability, then the current hypothesis already suffices). This probability has a natural interpretation: it is the fraction of the currently "uncovered" distribution of negative examples that would *become* covered if we added x_j to the conjunction. Since we know there are at most k variables that would completely cover the distribution of negative examples (namely, the variables appearing in the target conjunction), there must always be a choice of x_j for which this probability $p_{j,i}$ is at least $1/k$. As long as we choose to add an x_j for which $p_{j,i}$ is at least some constant times $1/k$, we will make rapid progress towards covering the negative distribution. Thus, it suffices to estimate the $p_{j,i}$ within approximation error $O(1/k)$. Note that in the simulation from noisy examples, we can use a common sample to simultaneously estimate all of the $p_{j,i}$ for a fixed value of i, but since the conditioning event depends on the variables selected so far, we must draw a fresh sample for each i.

How many variables must we select in the second phase before all but ϵ of the negative distribution is covered? Since we cover at least a fraction $1/k$ with each variable added, we solve $(1 - (1/k))^r < \epsilon$ to give $r = O(k \log(1/\epsilon))$. To determine the sample complexity of simulating this statistical query algorithm from noisy examples, we apply Theorem 3 to the following accounting of the required queries:

- In the first phase, 1 query of approximation error ϵ and n queries of approximation error ϵ^2/r, where $r = O(k \log(1/\epsilon))$. These queries may all be estimated from a common noisy sample, as in Theorem 3.

- In the second phase, at most r stages, each of which requires at most n queries of approximation error $1/k$. The queries within a phase may be simulated from a common noisy sample.

Applying Theorem 3 to just the queries from the first phase, we obtain a sample size whose dependence on n is only $\log(n)$, on k is k^2, and on ϵ is $1/\epsilon^4$. (The dependence on the noise rate and confidence parameters are simply those given in Theorem 3.) For the second stage, the dependence on n is $\log(n)$, and on k is k^3. The important aspect of the overall bound is its modest logarithmic dependence on the total number of variables. However, despite the logarithmic dependence on n, our algorithm depends cubically on k, as opposed to Haussler's linear bound. It would be interesting to improve our bound, or prove that it is optimal in the noisy computationally bounded setting. The same remarks apply to the strong dependence on ϵ.

6.2 Learning Perceptrons on Symmetric Distributions

Here the class $\mathcal{F}_n$ consists of all linear half-spaces passing through the origin in R^n. Thus without loss of generality, the target concept can be represented by its normal vector $\vec{u} \in \mathrm{R}^n$ lying on the unit sphere, and $\vec{x} \in \mathrm{R}^n$ is a positive example of $\vec{u}$ if and only if $\vec{u} \cdot \vec{x} \geq 0$ (this is simply the class of perceptrons with threshold 0). The distribution $\mathcal{D}$ we consider is the uniform distribution on the unit sphere (or any other radially symmetric distribution). There is a voluminous literature on learning perceptrons in general (see the work of Minsky and Papert [24] for a partial bibliography) and with respect to this distribution in particular [30, 4, 12]. Here we give a very simple and efficient algorithm for learning from statistical queries (and thus an algorithm tolerating noise). Recent papers have provided more general solutions, again in the statistical query setting [5, 8].

The sketch of the main ideas is as follows: for any vector $\vec{v} \in \mathrm{R}^n$, the error of $\vec{v}$ with respect to the target vector $\vec{u}$ is simply $error(\vec{v}) = \mathbf{Pr}_{EX(\vec{u},\mathcal{D})}[sign(\vec{v} \cdot \vec{x}) \neq sign(\vec{u} \cdot \vec{x})]$. The estimation of such a probability clearly falls into the statistical query model by setting $\chi_{\vec{v}}(\vec{x}, \ell) = 1$ if and only if $sign(\vec{v} \cdot \vec{x})$ agrees with the label ℓ. Now it is not difficult to show that for radially symmetric distributions, $error(\vec{v}) = \rho(\vec{u}, \vec{v})/\pi$, where $\rho(\vec{u}, \vec{v})$ is the angle between $\vec{u}$ and $\vec{v}$. Thus by obtaining accurate estimates of $error(\vec{v})$ we obtain accurate estimates of the projection of the target $\vec{u}$ onto $\vec{v}$. Thus, our algorithm is to choose n linearly independent vectors $\vec{v}_1, \ldots, \vec{v}_n$ and use the oracle $STAT(\vec{u}, \mathcal{D})$ to estimate the coordinates of $\vec{u}$ in the $\vec{v}_i$ system in the way suggested. It is not hard to show that if our estimates are accurate within an additive factor of ϵ/n, then the resulting hypothesis vector $\vec{u}'$ will satisfy $error(\vec{u}') \leq \epsilon$. Since this is an efficient algorithm for learning from statistical queries, we immediately have an efficient algorithm for learning with noise.

6.3 Learning Rectangles in High Dimension

We now give an efficient statistical query algorithm for the class of axis-aligned rectangles in n-dimensional space. This class was first studied by Blumer et al. [7], who analyzed the algorithm that takes the smallest axis-aligned rectangle consistent with a large sample. Note that this algorithm is not noise-tolerant, since in the presence of noise there may be no axis-aligned rectangle separating the positive examples from the negative examples.

Here we need to use the variant of the statistical query model in which we are given access to $\mathcal{D}$ in addition to $STAT(f, \mathcal{D})$ (see Remark 3 following Definition 3 and the comments following the proof of Theorem 3). Our algorithm begins by sampling $\mathcal{D}$ and using the inputs drawn to partition n-dimensional space. More precisely, for each dimension i, we use the sample to divide the x_i-axis into d/ϵ intervals with the property that the x_i component of a random point from $\mathcal{D}$ is approximately equally likely to fall into any of the intervals. This can be done using methods similar to those of Kearns and Schapire [20].

We now estimate the boundary of the target rectangle separately for each dimension using $STAT(f, \mathcal{D})$. Note that if the projection of the target rectangle onto the x_i-axis does not intersect an interval I of that axis, then the conditional probability p_I that the label is positive given that the input has its x_i component in I is 0. On the other hand, if the target's projection onto I is nonzero and there is significant probability that a positive example of the target has its x_i component in I, then p_I must be significantly larger than 0. Thus our algorithm can start from the left, and moving to the right, place the left x_i-boundary of the hypothesis rectangle at the first interval I such that p_I is significant (at least polynomial in ϵ/n); note that estimating p_I can be done solely with calls to $STAT(f, \mathcal{D})$ once the intervals are defined for each coordinate. The analogous computation is done from the right, and for each dimension. The result is an efficient (polynomial in $1/\epsilon$ and n) algorithm for learning n-dimensional rectangles from statistical queries, immediately implying a noise-tolerant learning algorithm.

A combination of the ideas given here and those in the subsection above on covering algorithms yields an efficient noise-tolerant learning algorithm for unions of rectangles in the Euclidean plane.

7 A Hard Class for Learning from Statistical Queries

The results of the last section might tempt us to conjecture that any class efficiently learnable in the Valiant model is efficiently learnable from statistical queries. In this section, we prove this conjecture to be false, by showing that the class of all parity concepts (where each potential target concept is the parity of some unknown subset of the boolean variables $x_1, \ldots, x_n$), which is known to be efficiently learnable in the Valiant model via the solution of a system of linear equations modulo 2 [10, 15], is not efficiently learnable from statistical queries. The fact that the separation of the two models comes via this class is of particular interest, since the parity class has no known efficient noise-tolerant algorithm.

Theorem 5 *Let $\mathcal{F}_n$ be the class of all parity concepts over n boolean variables, and let $\mathcal{F} = \bigcup_{n \geq 1} \mathcal{F}_n$. Then $\mathcal{F}$ is not efficiently learnable from statistical queries.*

Proof: We prove that it is impossible to learn $\mathcal{F}_n$ from statistical queries in time polynomial in n even in the case that the target concept f is drawn randomly from $\mathcal{F}_n$ and the target distribution $\mathcal{D}$ is uniform over $\{0,1\}^n$.

We begin by fixing any mapping $\chi : \{0,1\}^n \times \{0,1\} \to \{0,1\}$. Our immediate goal is to show that a query for $STAT(f, \mathcal{D})$ on any such χ reveals essentially no information about f; this will be accomplished by computing an upper bound on the variance of $P_\chi(f)$. Let us use $P_\chi(f)$ to denote $\mathbf{Pr}_{EX(f,\mathcal{D})}[\chi = 1]$ in order to make explicit the dependence of P_χ on f; in the case of the uniform distribution, we simply have $P_\chi(f) = (1/2^n) \sum_{x \in \{0,1\}^n} \chi(x, f(x))$. Now let $\mathbf{E}_f[P_\chi(f)]$ denote the expected value of $P_\chi(f)$, where the expectation is taken over the random draw of a parity concept f uniformly from $\mathcal{F}_n$. Then by additivity of expectations we may write

$$\begin{aligned}
\mathbf{E}_f[P_\chi(f)] &= (1/2^n) \sum_{x \in \{0,1\}^n} \mathbf{E}_f[\chi(x, f(x))] \\
&= (1/2^n) \sum_{x \in \{0,1\}^n} Q(x) \qquad (6)
\end{aligned}$$

where we define $Q(x) = 0$ if $\chi(x, 0) = \chi(x, 1) = 0$ (let Q_0 denote set of all such x), $Q(x) = 1$ if $\chi(x, 0) = \chi(x, 1) = 1$ (let Q_1 denote the set of all such x), and $Q(x) = 1/2$ if $\chi(x, 0) \neq \chi(x, 1)$ (let $Q_{1/2}$ denote the set of all such x). Equation (6) follows from the fact that for any fixed $x \in \{0,1\}^n$, a randomly chosen parity concept f is equally likely to satisfy $f(x) = 0$ and $f(x) = 1$. Now let q_0, q_1 and $q_{1/2}$ denote the cardinalities of Q_0, Q_1 and $Q_{1/2}$, respectively, so $q_0 + q_1 + q_{1/2} = 2^n$. Then from Equation (6) and the definition of $Q(x)$ we may write

$$\mathbf{E}_f[P_\chi(f)] = (1/2^n)(q_1 + (1/2)q_{1/2}). \qquad (7)$$

We may also write

$$\mathbf{E}_f[P_\chi(f)^2] = (1/2^{2n}) \sum_{x,y \in \{0,1\}^n} \mathbf{E}_f[\chi(x, f(x))\chi(y, f(y))].$$

For a set $S \subseteq \{0,1\}^n$, let us introduce the shorthand notation

$$\Gamma(S) = \sum_{x \in S, y \in \{0,1\}^n} \mathbf{E}_f[\chi(x, f(x))\chi(y, f(y))].$$

Then we may further decompose the above sum by writing

$$\mathbf{E}_f[P_\chi(f)^2] = (1/2^{2n})(\Gamma(Q_0) + \Gamma(Q_1) + \Gamma(Q_{1/2})).$$

The summation $\Gamma(Q_0)$ is simply 0, since $\chi(x, f(x)) = 0$ here. $\Gamma(Q_1)$ simplifies to $q_1 \sum_{y \in \{0,1\}^n} \mathbf{E}_f[\chi(y, f(y))]$ since $\chi(x, f(x)) = 1$, and this simplifies further to be $q_1(q_1 + (1/2)q_{1/2})$ by Equations (6) and (7). For the summation $\Gamma(Q_{1/2})$ we also need to consider the possible cases of y. If $y = x$ (which occurs for only a single value of y), then $\chi(x, f(x))\chi(y, f(y)) = \chi(x, f(x))^2$ will be 1 if and only if $f(x) = b_x$ for the value $b_x \in \{0,1\}$ such that $\chi(x, b) = 1$. This will occur with probability 1/2 for randomly drawn parity concept f. If y falls in Q_0 (which occurs for q_0 values of y), $\chi(x, f(x))\chi(y, f(y)) = 0$. If y falls in Q_1 (which occurs for q_1 of the values of y), then $\chi(x, f(x))\chi(y, f(y)) = \chi(x, f(x))$ and again this is 1 if and only if $f(x) = b_x$, which again will occur with probability 1/2 for a random parity concept f. Finally, if y falls into $Q_{1/2}$ but is not the same as x (which occurs for $q_{1/2} - 1$ of the values of y), then $\chi(x, f(x))\chi(y, f(y)) = 1$ if and only if $f(x) = b_x$ and $f(y) = b_y$, where b_x is as before and $b_y \in \{0,1\}$ is the value satisfying $\chi(y, b_y) = 1$. Since for any fixed x and y, all four labelings of x and y are equally likely for a randomly chosen parity concept f, this will occur with probability 1/4.

Putting this all together, we write

$$\begin{aligned} \mathbf{E}_f[P_\chi(f)^2] &= (1/2^{2n})(q_1(q_1 + (1/2)q_{1/2}) + \\ &\quad q_{1/2}(1/2 + (1/2)q_1 + (1/4)(q_{1/2} - 1))). \end{aligned} \tag{8}$$

Now from Equation (7) we may write

$$\mathbf{E}_f[P_\chi(f)]^2 = (1/2^{2n})(q_1 + (1/2)q_{1/2})^2.$$

By combining this equality with Equation (8), some simple algebra then gives

$$\begin{aligned} \mathbf{Var}_f[P_\chi(f)] &= \mathbf{E}_f[P_\chi(f)^2] - \mathbf{E}_f[P_\chi(f)]^2 \\ &= q_{1/2}/(4 \cdot 2^{2n}) \\ &\leq 1/(2^{n+2}) \end{aligned}$$

since $q_{1/2} \leq 2^n$. Thus we have shown that for any χ, the variance of P_χ is exponentially small with respect to the random draw of target concept. Now suppose for contradiction that parity concepts are efficiently learnable from statistical queries by an algorithm L. Fix ϵ to be any constant smaller than 1/4 (note that with respect to the uniform distribution, any two parity concepts differ with probability 1/2). Assume without loss of generality that α is a lower bound on the allowed approximation error of L's queries, where $\alpha = 1/p(n)$ for some polynomial $p(\cdot)$ since ϵ is constant.

Although the queries made by L may be dynamically chosen, L makes some first query (χ_1, α). Let $\chi_1, \ldots, \chi_{r(n)}$ be the sequence of queries made by L when the answer returned to each query (χ_i, α) is simply $\mathbf{E}_f[P_{\chi_i}(f)]$. Here $r(n)$ is polynomial since L is efficient. Then it is not hard to show using Chebyshev's inequality and the above bound on $\mathbf{Var}_f[P_{\chi_i}(f)]$ that with high probability, a randomly chosen parity concept f' will be consistent with the query responses received by L — that is, with high probability f' satisfies

$$\mathbf{Pr}_f[P_{\chi_i}(f)] - \alpha \leq P_{\chi}(f') \leq \mathbf{Pr}_f[P_{\chi_i}(f)] + \alpha$$

for all $1 \leq i \leq r(n)$. Since many parity concepts are consistent with the responses received by L, the error of L's hypothesis must be large with respect to the random draw of the target f'; this follows from the fact if h agrees with one parity concept with probability at least $1 - \epsilon$, it must disagree with any other parity concept with probability at least $1/2 - \epsilon$. □(Theorem 5)

Note that the proof of Theorem 5 shows that the class of parity concepts is not efficiently learnable from statistical queries for information-theoretic reasons. Thus while it can be shown that in the absence of constraints on computation time or the allowed approximation accuracy, the Valiant and statistical query models are equivalent, Theorem 5 demonstrates that the requirement that an algorithm make only a polynomial number of queries, each of at least inverse polynomial allowed approximation error, separates the models with no unproven complexity assumptions.

Theorem 5 has recently been strengthened and generalized [6] to show that the number of statistical queries required for learning *any* class is determined by the number of "nearly orthogonal" concepts contained in the class.

8 A Lower Bound on Query Complexity

The proof of Theorem 5 is of particular interest because it demonstrates that while the Vapnik-Chervonenkis dimension of a concept class characterizes the number of random examples required for learning in the Valiant model [7], it cannot provide even a rough characterization of the number of queries required for learning in the statistical query model: the Vapnik-Chervonenkis dimension of the class of parity concepts is $\Theta(n)$, and we have shown that the number of statistical queries required is exponential in n. This demonstrates that the Vapnik-Chervonenkis dimension cannot provide good general upper bounds on query complexity, but the possibility of a good general lower bound remains, and is the subject of this section.

It is important to carefully specify what we desire from a lower bound on the number of statistical queries, due to the potential tradeoff between the number of queries made and the allowed approximation error of those queries. More precisely, from Theorem 1 and the lower bound on sample sizes for the Valiant model given by Ehrenfeucht et al. [9], we can easily derive an initial but unsatisfying bound on the number of queries required in the statistical query model: since we know that an algorithm using r queries, each of allowed approximation error at least α, can be simulated to obtain an algorithm in the Valiant model using r/α^2 examples (ignoring the dependence on δ), the Ehrenfeucht et al. bound indicates that $r/\alpha^2 = \Omega(d/\epsilon)$ must hold, where d is the Vapnik-Chervonenkis dimension of the concept class. Thus we have $r = \Omega(d\alpha^2/\epsilon)$. This bound allows the possibility that there is a concept class of VC dimension d which can be learned from just a single statistical query of approximation error $\sqrt{\epsilon/d}$. Similarly, since we have $\alpha = O(\sqrt{\epsilon r/d})$ the bound also allows the possibility that d/ϵ queries of allowed approximation error 1 could always suffice for learning. This latter possibility is ludicrous, since allowed approximation error 1 allows the oracle $STAT(f, \mathcal{D})$ to return arbitrary values in $[0, 1]$, rendering learning impossible in any number of queries.

We now give a considerably better bound, in which the the number of queries made is bounded from below and the allowed approximation error of these queries is bounded from above simultaneously.

Theorem 6 *Let $\mathcal{F}$ be any concept class, let d be the Vapnik-Chervonenkis dimension of $\mathcal{F}$, and let L be an algorithm for learning $\mathcal{F}$ from statistical queries. Then for any ϵ, there is a distribution $\mathcal{D}$ such that L must make at least $\Omega(d/\log d)$ queries with allowed approximation error $O(\epsilon)$ to the oracle $STAT(f, \mathcal{D})$ in order to find a hypothesis h satisfying $error(h) \leq \epsilon$.*

Proof: The proof begins by using the standard hard distribution for learning in the Valiant model [7, 9]. Thus, given the target error value ϵ, we let $\{x_0, x_1, \ldots, x_{d'}\}$ be a shattered set (where $d' = d - 1$), and let $\mathcal{D}$ give weight $1 - 2\epsilon$ to x_0 and weight $2\epsilon/d'$ to each of $x_1, \ldots, x_{d'}$. We let $\mathcal{F}'$ be a finite subclass of $\mathcal{F}$ in which $f(x_0) = 0$ for all $f \in \mathcal{F}'$, and for each of the $2^{d'}$ labelings of $x_1, \ldots, x_{d'}$ there is exactly one representative concept in $\mathcal{F}'$. The target concept f will be chosen randomly from $\mathcal{F}'$.

Now under these settings, a number of simplifying assumptions regarding the nature of L's queries to $STAT(f, \mathcal{D})$ can be made. First, for any χ we must have either $P_\chi \leq 2\epsilon$ or $P_\chi \geq 1 - 2\epsilon$ regardless of the target f due to the large weight given to x_0. Thus we can immediately conclude that any query (χ, α) made by L in which $\alpha \geq 2\epsilon$ reveals no information about f (since an adversary generating the answers of $STAT(f, \mathcal{D})$ can always return either the value 2ϵ or the value $1 - 2\epsilon$ on such queries).

Secondly, if we regard the target concept f as a length d' bit vector $f = (f(x_1), \ldots, f(x_{d'}))$, and we even allow L to make queries with allowed approximation error $\alpha = 0$, then P_χ is determined by the Hamming distances $\rho(f, g^0)$ and $\rho(f, g^1)$, where g^b is the vector of length d' in which the ith bit is 1 if and only if $\chi(x_i, b) = 1$ and $\chi(x_i, \neg b) = 0$.

We can thus reduce the problem of learning from statistical queries in this setting to the following simpler learning problem: the target is a d'-dimension bit vector f, and the learner L makes vector queries g and receives the Hamming distance $\rho(f,g)$ from f to g. The learner must eventually output another bit vector h satisfying $(2\epsilon/d')\sum_{i=1}^{d'} f_i \oplus h_i \leq \epsilon$. To prove the theorem it suffices to lower bound the number of vector queries made by L in this model.

Let $\mathcal{F}_{i-1}$ denote the class of concepts consistent with the answers received by L on its first $i-1$ query vectors $g^1,\ldots,g^{i-1}$, so $\mathcal{F}_{i-1} = \{f' \in \mathcal{F}' : \rho(f',g^j) = \rho(f,g^j), 1 \leq j \leq i-1\}$. Then the ith query vector g^i partitions $\mathcal{F}_{i-1}$ into $d'+1 = d$ pieces $\mathcal{F}_{i-1}^j = \{f' \in \mathcal{F}_{i-1} : \rho(f',g^i) = j\}$ for $0 \leq j \leq d'$.

Now it is easy to show using a Bayesian argument that rather than choosing the target concept f randomly from $\mathcal{F}'$ before L makes its vector queries, it is equivalent to choose a new target concept f^i after every vector query g^{i-1} by drawing f^i randomly from $\mathcal{F}_{i-1}$ (in the sense that for all i, the expected error of L's hypothesis after i vector queries with respect to the current target is the same in both cases). In the latter model, based on the random draw of f^i, the class $\mathcal{F}_i$ is $\mathcal{F}_{i-1}^j$ with probability $|\mathcal{F}_{i-1}^j|/|\mathcal{F}_{i-1}|$.

It is not hard to see that for any natural number $r \geq 1$, we have $\mathbf{Pr}[|\mathcal{F}_i| \geq (1/dr)|\mathcal{F}_{i-1}|] \geq 1 - 1/r$, where this probability is taken over the random choice of f^i. Thus we have that for any sequence of r vector queries, with probability at least $(1-1/r)^r$ (which is lower bounded by a constant for r sufficiently large) we have $|\mathcal{F}_r| \geq (1/dr)^r 2^{d'}$. Solving for conditions on r to satisfy $2^{d'/2} \geq |\mathcal{F}_r| \geq (1/dr)^r 2^{d'}$ yields $r = \Omega(d/\log d)$. For r smaller, the final target concept f^{r+1} is drawn randomly from a set of vectors whose size is (with constant probability) at least $2^{d'/2}$. It can then be shown by a simple counting argument that the expected Hamming distance between L's final hypothesis h and the final target $f = f^{r+1}$ is $\Omega(d)$ (here the expectation is taken over the draw of f^{r+1} from $\mathcal{F}_r$). This implies that the expected error of h is at least a constant times ϵ, so learning cannot be complete. $\square$(Theorem 6)

9 Handling a Variable Noise Rate

One objection to the classification noise model we have investigated is its assumption of the existence of a fixed noise rate η: independent of any previous misclassifications, the probability of the next example being misclassified is always exactly η. In this section, we would like to formalize a more realistic model in which the noise rate η may fluctuate over time, but in which it is still fair to regard any misclassifications as noise in the sense that they are independent of the input drawn. It appears that relaxing this latter condition severely limits the cases for which efficient learning is possible, and results in a perhaps overly pessimistic noise model [31, 17], unless the dependence of the noise on the input has natural structure that can be exploited by the learner [20].

To formalize the new model, we allow an adversary to choose an infinite *bias* *sequence* $\eta_1, \eta_2, \ldots, \eta_m, \ldots$; we require that this sequence be fixed *in advance*, and thus not dependent on the actual examples drawn, as discussed above. Each $\eta_i \in [0,1]$ is interpreted as the probability that the ith example drawn by the learner has its label corrupted. The only restriction on the η_i is that for any value m we must have $1/m \sum_{i=1}^{m} \eta_i \leq \eta$, where $0 \leq \eta < 1/2$ is the *effective noise rate*. Thus, we simply demand that for any sample size m, the effective noise rate for this sample size is bounded by η. As usual, we assume without loss of generality that a learning algorithm is given an upper bound $\eta \leq \eta_b < 1/2$ and is allowed time polynomial in $1/(1-2\eta_b)$ and the usual parameters. Now when learning a target concept f with respect to distribution $\mathcal{D}$, for any i the ith example requested by the learner is chosen as follows: x is drawn randomly according to $\mathcal{D}$, and a coin with probability $1-\eta_i$ of heads is tossed. If the outcome is heads, the example is $\langle x, f(x) \rangle$; otherwise it is $\langle x, \neg f(x) \rangle$. We shall refer to this model as the *variable noise rate model*, and we say that $\mathcal{F}$ can be learned in this model if there is an efficient algorithm tolerating any effective noise rate $\eta < 1/2$.

Several comments regarding this model are in order. First of all, note that the adversary may choose $\eta_i = 0$, $\eta_i = 1$ or any value in between. Thus, the adversary may deterministically specify at which times there will be misclassifications. Secondly, it is no longer true that the probability of misclassification at a given time is independent of the probability at other times, since the bias sequence is arbitrary (subject to the averaging condition). These two properties make variable noise rates a good model for noise bursts, in which a normally functioning system will have no misclassifications, but an occasional malfunction will cause a concentrated stream of consecutive misclassifications. Finally, however, note that despite these allowed dependences, the probability that any particular input is misclassified at any particular time is the same for all inputs, since the bias sequence must be specified by the adversary before the examples are drawn.

The following theorem states that learning in the variable noise rate model is in fact no more difficult than learning in the standard classification noise model.

Theorem 7 *Let $\mathcal{F}$ be a class of concepts over X, and let $\mathcal{H}$ be a class of representations of concepts over X. Then $\mathcal{F}$ is efficiently learnable with noise using $\mathcal{H}$ if and only if $\mathcal{F}$ is efficiently learnable with variable noise rate using $\mathcal{H}$.*

Proof: Variable noise rate learning trivially implies learning in the standard noise model. For the converse, let L be an efficient algorithm for learning $\mathcal{F}$ in the standard noise model. Let m be an appropriate sample size determined by the analysis, and let us first flip m coins of biases $\eta_1, \ldots, \eta_m$ to determine the noise bits $b_1, \ldots, b_m$ used in generating the sample given to L. Now for m sufficiently large, the number of 1's (denoting misclassifications) generated in this sequence is bounded by $(\eta + (1-2\eta)/4)m$ with overwhelming probability via a standard Chernoff or Hoeffding bound analysis. Thus, we can immediately reduce our analysis to that of a binary bias sequence with effective noise rate bounded by $(\eta + (1 - 2\eta)/4) < 1/2$. Let $0 \leq r \leq m$ denote the actual number of misclassifications in the bit sequence.

The main trick is to draw m examples for L (which are then given noisy labels according to the bits b_i), but to give L a random permutation of these m examples. In this way we almost simulate a standard classification noise process with noise rate r/m. The only difference is that whereas such a process would be binomially distributed with a mean of r misclassifications, we are generating only the slice of this distribution with exactly r misclassifications. However, this slice constitutes a significant fraction of the binomial distribution (the probability of falling on the mean is easily seen to be lower bounded by an inverse polynomial in m), and without loss of generality the dependence of L's sample size on the confidence parameter δ is only $\log(1/\delta)$ via standard "confidence boosting" arguments. We can thus set the confidence parameter value given to L to be $\delta' = O(\delta/m)$, which forces L to perform correctly on $1-\delta$ of the r-slice of the binomial distribution with a mean of r misclassifications. The modest $\log 1/\delta$ dependence allows us to do this while keeping the required sample size m polynomial. □(Theorem 7)

As an immediate corollary, we obtain that efficient learning from statistical queries implies efficient learning with variable noise rate. Note that the equivalence given by Theorem 7 holds for distribution-specific learning as well.

10 Open Problems

In addition to the long-standing problems of finding efficient distribution-free noise-tolerant learning algorithms for the classes of perceptrons and parity concepts (or proving that none exist), several equivalences between the models studied here are open. For instance, is efficient learning with noise equivalent to efficient learning from statistical queries? Even stronger, is any class efficiently learnable in the Valiant model also efficiently learnable with noise? Note that any counterexamples to such equivalences should not depend on syntactic hypothesis restrictions, but should be representation independent [16].

Acknowledgements

Thanks to Umesh Vazirani for the early conversations from which this research grew, to Rob Schapire for many insightful comments and his help with the proof of Theorem 5, and to Jay Aslam, Avrim Blum, Jin-yi Cai, Dick Lipton, Yishay Mansour, Ron Rivest, Madhu Sudan, and Andy Yao for enlightening discussions.

References

[1] Dana Angluin and Philip Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1988.

[2] Javed A. Aslam and Scott E. Decatur. General bounds on statistical query learning and pac learning with noise via hypothesis boosting. In *34th Annual Symposium on Foundations of Computer Science*, pages 282–291, 1993.

[3] Javed A. Aslam and Scott E. Decatur. Specification and simulation of statistical query algorithms for efficiency and noise tolerance. In *Proceedings of the Eighth Annual Workshop on Computational Learning Theory*, pages 437–446, 1995.

[4] Eric B. Baum and Yuh-Dauh Lyuu. The transition to perfect generalization in perceptrons. *Neural Computation*, 3:386–401, 1991.

[5] Avrim Blum, Alan Frieze, Ravi Kannan, and Sampath Vempala. A polynomial-time algorithm for learning noisy linear threshold functions. In *37th Annual Symposium on Foundations of Computer Science*, pages 330–338, 1996.

[6] Avrim Blum, Merrick Furst, Jeff Jackson, Michael Kearns, Yishay Mansour, and Steven Rudich. Weakly learning DNF and characterizing statistical query learning using fourier analysis. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on the Theory of Computing*, 1994.

[7] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the Association for Computing Machinery*, 36(4):929–965, October 1989.

[8] Edith Cohen. Learning noisy perceptrons by a perceptron in polynomial time. In *38th Annual Symposium on Foundations of Computer Science*, 1997.

[9] A. Ehrenfeucht, D. Haussler, M. Kearns, and L. Valiant. A general lower bound on the number of examples needed for learning. In *First Workshop on Computatinal Learning Theory*, pages 139–154, Cambridge, Mass. August 1988. Morgan Kaufmann.

[10] Paul Fischer and Hans Ulrich Simon. On learning ring-sum expansions. *SIAM J. Computing*, 21(1):181–192, 1992.

[11] Merrick L. Furst, Jeffrey C. Jackson, and Sean W. Smith. Improved learning of AC^0 functions. In *Proceedings of the Fourth Annual Workshop on Computational Learning Theory*, pages 317–325, August 1991.

[12] E. Gardner and B. Derrida. Three unfinished works on the optimal storage capacity of networks. *J. Phys. A: Math. Gen.*, 22:1983–1994, 1989.

[13] Thomas Hancock and Yishay Mansour. Learning monotone $k\mu$ DNF formulas on product distributions. In *Proceedings of the Fourth Annual Workshop on Computational Learning Theory*, pages 179–183, August 1991.

[14] David Haussler. Quantifying inductive bias: AI learning algorithms and Valiant's learning framework. *Artificial Intelligence*, 36:177–221, 1988.

[15] David Helmbold, Robert Sloan, and Manfred K. Warmuth. Learning integer lattices. *SIAM Journal on Computing*, 21(2):240–266, 1992.

[16] Michael Kearns. *The Computational Complexity of Machine Learning*. The MIT Press, 1990.

[17] Michael Kearns and Ming Li. Learning in the presence of malicious errors. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, pages 267–280, May 1988. To appear, *SIAM Journal on Computing*.

[18] Michael Kearns, Ming Li, Leonard Pitt, and Leslie Valiant. On the learnability of Boolean formulae. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, pages 285–295, May 1987.

[19] Michael Kearns and Leonard Pitt. A polynomial-time algorithm for learning k-variable pattern languages from examples. In *Proceedings of the Second Annual Workshop on Computational Learning Theory*, pages 57–71, July 1989.

[20] Michael J. Kearns and Robert E. Schapire. Efficient distribution-free learning of probabilistic concepts. In *31st Annual Symposium on Foundations of Computer Science*, pages 382–391, October 1990. To appear, *Journal of Computer and System Sciences*.

[21] Michael J. Kearns and Umesh V. Vazirani. *An Introduction to Computational Learning Theory*. The MIT Press, 1994.

[22] Philip D. Laird. *Learning from Good and Bad Data*. Kluwer international series in engineering and computer science. Kluwer Academic Publishers, Boston, 1988.

[23] Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, Fourier transform, and learnability. In *30th Annual Symposium on Foundations of Computer Science*, pages 574–579, October 1989.

[24] Marvin Minsky and Seymour Papert. *Perceptrons: An Introduction to Computational Geometry (Expanded Edition)*. The MIT Press, 1988.

[25] Leonard Pitt and Leslie G. Valiant. Computational limitations on learning from examples. *Journal of the Association for Computing Machinery*, 35(4):965–984, October 1988.

[26] Ronald L. Rivest. Learning decision lists. *Machine Learning*, 2(3):229–246, 1987.

[27] Yasubumi Sakakibara. *Algorithmic Learning of Formal Languages and Decision Trees*. PhD thesis, Tokyo Institute of Technology, October 1991. Research Report IIAS-RR-91-22E, International Institute for Advanced Study of Social Information Science, Fujitsu Laboratories, Ltd.

[28] Robert E. Schapire. Learning probabilistic read-once formulas on product distributions. In *Proceedings of the Fourth Annual Workshop on Computational Learning Theory*, August 1991. To appear, *Machine Learning*.

[29] Robert Elias Schapire. *The Design and Analysis of Efficient Learning Algorithms*. The MIT Press, 1992.

[30] H.S. Seung, H. Sompolinsky, and N. Tishby. Statistical mechanics of learning from examples. *Physical Review A*, 45(8):6056–6091, April 1992.

[31] Robert H. Sloan. Types of noise in data for concept learning. In *Proceedings of the 1988 Workshop on Computational Learning Theory*, pages 91–96, August 1988.

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

[33] L. G. Valiant. Learning disjunctions of conjunctions. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, pages 560–566, August 1985.

[34] V. N. Vapnik and A. Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its applications*, XVI(2):264–280, 1971.