Specification and Simulation of Statistical Query Algorithms for Efficiency and Noise Tolerance

Javed A. Aslam* Scott E. Decatur†

Aiken Computation Laboratory
Harvard University
Cambridge, MA 02138

Abstract

A recent innovation in computational learning theory is the statistical query (SQ) model. The advantage of specifying learning algorithms in this model is that SQ algorithms can be simulated in the PAC model, both in the absence *and* in the presence of noise. However, simulations of SQ algorithms in the PAC model have non-optimal time and sample complexities. In this paper, we introduce a new method for specifying statistical query algorithms based on a type of *relative error* and provide simulations in the noise-free and noise-tolerant PAC models which yield efficient algorithms. Requests for estimates of statistics in this new model take the form: "Return an estimate of the statistic within a $1 \pm \mu$ factor, or return '$\perp$', promising that the statistic is less than θ." In addition to showing that this is a very natural language for specifying learning algorithms, we also show that this new specification is polynomially equivalent to standard SQ, and thus, known learnability and hardness results for statistical query learning are preserved.

We then give highly efficient PAC simulations of relative error SQ algorithms. We show that the learning algorithms obtained by simulating efficient relative error SQ algorithms in both the absence of noise and in the presence of malicious noise have roughly optimal sample complexity. We also show that the simulation of efficient relative error SQ algorithms in the presence of classification noise yield learning algorithms at least as efficient as those obtained through standard methods, and in some cases improved, roughly optimal results are achieved. The sample complexities for all of these simulations are based on the d_ν metric which is a type of relative error metric useful for quantities which are small or even zero. We show that uniform convergence with respect to the d_ν metric yields "uniform convergence" with respect to (μ, θ) accuracy.

Finally, while we show that many *specific* learning algorithms can be written as highly efficient relative error SQ algorithms, we also show, in fact, that *all* SQ algorithms can be written efficiently by proving general upper bounds on the complexity of (μ, θ) queries as a function of the accuracy parameter ε. As a consequence of this result, we give general upper bounds on the complexity of learning algorithms achieved through the use of relative error SQ algorithms and the simulations described above.

*Supported by DARPA Air Force Contract F49620-92-J-0466. Part of this research was conducted while the author was at MIT and supported by DARPA Contract N00014-87-K-825 and by NSF Grant CCR-89-14428. Email: jaa@das.harvard.edu

†Supported by an NDSEG Fellowship and by NSF Grant CCR-92-00884. Email: sed@das.harvard.edu

To appear in *Proceedings of the Eighth Annual ACM Conference on Computational Learning Theory*. ACM Press, July 1995.

1 Introduction

In this paper, we focus on the development of efficient, fault-tolerant algorithms for learning in the "Probably Approximately Correct" (PAC) model [17]. An algorithm for "PAC learning" a class of functions (concepts) uses examples drawn from an oracle (the environment) in order to approximate a hidden target function selected from the class. The examples are labelled according to the hidden function. Although the goal of the learner is to output an approximation (hypothesis) with error at most $\varepsilon > 0$, it is allowed probability $\delta > 0$ of failing to meet this criteria. Two important complexity measures of algorithms in this setting are their *time complexity* and *sample complexity* (number of examples drawn from the oracle). The sample complexity is often crucial due to the scarcity of training data in many situations.

In addition to developing learning algorithms for the standard PAC setting described above, for many applications it is important to develop learning algorithms which are robust in that they tolerate errors in the training data. Two formalizations of learning with faulty data are the variants of the PAC model with *classification noise* and *malicious errors*. In these models, the classification of examples or the entire examples themselves, respectively, may be corrupted, yet the goal of the learner remains to approximate the underlying target function with respect to noise-free examples.

A recently developed tool for creating efficient, noise-tolerant, learning algorithms is the statistical query (SQ) model [12]. In this model, instead of using labelled examples, the algorithm asks for the estimates of the val-

ues of statistics defined over the distribution of labelled examples. This model may be viewed as a restriction on the way an algorithm uses the PAC example oracle since the example oracle could be used to simulate these statistical queries. However, this restriction has been found to be quite mild in that most every class of concepts which is learnable in the PAC model is also learnable by statistical queries [12]. The most important use of the SQ model stems from the property that statistical queries can actually be simulated with the use of *noisy* example oracles. Specifically, an SQ algorithm can be simulated in the PAC model in the presence of classification noise, malicious errors, attribute noise and even hybrid models combining these different noises [12, 5, 7, 6].

A key parameter in the complexity of the PAC algorithm generated by the simulation of SQ algorithms is the tolerance of the SQ algorithm, τ_*, which quantifies the largest additive error that the SQ algorithm can tolerate when receiving an answer to its most sensitive query. The limitation of the additive model is evident in even the standard, *noise-free* simulation of additive error statistical query algorithms [12] which uses $\Omega(1/\tau_*^2)$ examples. Since $1/\tau_* = \Omega(1/\varepsilon)$ for all SQ algorithms [12], this simulation effectively uses $\Omega(1/\varepsilon^2)$ examples. This is clearly suboptimal when compared to the basically tight general upper and lower bounds on the noise-free sample complexity whose dependence on ε is $\tilde{\Theta}(1/\varepsilon)$ [4, 8].[1]

Thus, while there is an incentive for developing algorithms in the statistical query model due to the noise tolerance gained, there is also a disincentive towards doing so due to the inefficiency of the simulations: Algorithm designers must choose between writing algorithms in the SQ model to achieve noise-tolerance and writing algorithms in the PAC models (noise-free and noise-tolerant) in order to achieve efficiency.

This $\Omega(1/\tau_*^2) = \Omega(1/\varepsilon^2)$ sample complexity results from the worst case assumption that large probabilities may need to be estimated with small additive error. Either the nature of statistical query learning is such that learning sometimes requires the estimation of large probabilities with small additive error, or it is always sufficient to estimate each probability with an additive error comparable to the probability. If the former were the case, then the present model and simulations would be the best that one could hope for. We show that the latter is true, and that a model in which queries are specified with a type of *relative error* is a natural and powerful tool.

We accomplish the dual goal of efficiency and noise-tolerance by allowing a richer language for specification of SQ algorithms and by providing nearly optimal simulations of this richer language in the PAC model and its noise variants. Specifically, we propose a new way of specifying statistical query learning algorithms via *relative error statistical queries* which effectively accomplishes the goal of producing *efficient* algorithms in both the absence and presence of noise. In this model, the statistical queries are asked with relative error (μ, θ), denoting that the algorithm requests the value of a statistic within a relative factor of $1 \pm \mu$ but does not require an answer if the value is below θ. We argue that relative error SQ learning is a very natural model for specifying algorithms, and we use one such example throughout the paper, a simple relative error SQ algorithm for learning monotone conjunctions, in order to demonstrate the ease of specification and power of this new model.

Before giving PAC simulations for this new model, we first demonstrate that polynomial learnability in this model is equivalent to polynomial learnability in the additive error SQ model. Thus, known learnability and hardness results for additive error SQ learning are preserved in relative error SQ learning. This includes the results of Kearns [12] showing that almost all classes known to be PAC learnable are learnable by additive error statistical queries, the hardness result of Kearns [12] for learning parity functions, and the general hardness results of Blum *et al.* [3] based on Fourier analysis.

The advantages of the new model are then demonstrated by the simulations of relative error SQ algorithms in the noise-free PAC model, the malicious error PAC model and the classification noise PAC model. In each case, we determine the complexity of the PAC algorithm as a function of the complexity of the SQ algorithm. In order to prove sample complexity bounds for these simulations, we make use of the d_ν metric [15, 11]. Haussler gives sample complexity bounds sufficient for uniform convergence, with respect to the d_ν metric, of probabilities and their estimates based on a sample. We relate this uniform convergence to uniform convergence with respect to (μ, θ) accuracy, from which we then determine sample complexities sufficient for relative error SQ simulations.

As an example, we show how efficiently the relative error SQ algorithm for monotone conjunctions is simulated. While the previous noise-free and malicious error simulations of the additive error SQ algorithm for conjunctions use $\tilde{\Theta}\left(\frac{n^2}{\varepsilon^2}\log\frac{1}{\delta}\right)$ examples and tolerate at most $\Theta(\varepsilon/n)$ malicious error, our simulations of the relative error SQ algorithm for conjunctions use $\tilde{\Theta}\left(\frac{n}{\varepsilon}\log\frac{1}{\delta}\right)$ examples while still tolerating up to $\Theta(\varepsilon/n)$ malicious error. We further show improvements for learning the class of conjunctions with few relevant variables, obtaining the best known results for learning this class in the malicious error model.

In the case of classification noise, the simulation for

[1] In this paper, the $\tilde{O}$, $\tilde{\Omega}$ and $\tilde{\Theta}$ notations will refer to asymptotic bounds ignoring lower order, typically logarithmic, factors.

conjunctions uses the same complexity as the standard simulation, $\tilde{\Theta}\left(\frac{n^2}{\varepsilon^2(1-2\eta_b)^2}\log\frac{1}{\delta}\right)$, where η_b is an upper bound on the classification noise rate. Yet we also give a relative error SQ algorithm for a different class, symmetric functions, whose simulation actually uses a roughly optimal sample complexity of $\tilde{\Theta}\left(\frac{n}{\varepsilon(1-2\eta_b)^2}\log\frac{1}{\delta}\right)$ improving on the $\tilde{\Theta}\left(\frac{n^2}{\varepsilon^2(1-2\eta_b)^2}\log\frac{1}{\delta}\right)$ sample size required by an additive error SQ simulation. Thus, we show for the first time that a non-trivial class can be PAC-learned in polynomial time and in the presence of classification noise using a sample size whose dependence on ε is $o(1/\varepsilon^2)$.

Although we demonstrate the complexities of relative error SQ simulations for specific classes, we are also interested in the complexity of these simulations for any *arbitrary* class. We therefore examine the complexity of relative error SQ algorithms in general. Since the complexity of the PAC simulations depend on the parameters μ and θ, we focus on these measures. We first note that large classes of algorithms, including those for axis parallel rectangles and various covering algorithms, work using a constant μ, and θ only linear in ε. Such conditions on (μ, θ) yield very efficient PAC algorithms. Furthermore, we show general upper bounds on the complexity of learning *any* class by relative error SQ algorithms. These general upper bounds state that if a class is polynomially learnable with any efficiency by statistical queries (additive or relative) then it is learnable by a relative error SQ algorithm using μ independent of ε and $\theta = \tilde{\Omega}(\varepsilon)$. Thus, compared to the ε dependence of standard simulations of any standard additive error SQ algorithm (which use $\tilde{\Omega}(1/\varepsilon^2)$ examples), our simulations of relative error SQ algorithms require only $\tilde{O}(1/\varepsilon)$ examples in the absence of noise, and $\tilde{O}(1/\varepsilon)$ examples in the presense of malicious errors with identical, $\tilde{\Omega}(\varepsilon)$, error tolerance as in the old simulations. Note that a linear dependence on $1/\varepsilon$ is optimal for both sample complexities, and a linear dependence on ε is optimal for the error tolerance in the malicious error model.

2 Background

Before presenting the new model, we give formal definitions of the other learning models used throughout this paper. We begin by defining the example-based PAC learning model as well as the classification noise and malicious error variants. We then define the standard additive error statistical query model.

2.1 Example-Based PAC Learning

In an instance of PAC learning, a learner is given the task of determining a close approximation of an unknown $\{0,1\}$-valued target function from labelled examples of that function. The unknown target function f is assumed to be an element of a known function class $\mathcal{F}$ defined over an example space X. The example space X is typically either the Boolean hypercube $\{0,1\}^n$ or n-dimensional Euclidean space $\Re^n$. We use the parameter n to denote the common length of examples $x \in X$.

We assume that the examples are distributed according to some unknown probability distribution D on X. The learner is given access to an example oracle $EX(f, D)$ as its source of data. A call to $EX(f, D)$ returns a labelled example $\langle x, l\rangle$ where the example $x \in X$ is drawn randomly and independently according to the unknown distribution D, and the label $l = f(x)$. We often refer to a sequence of labelled examples drawn from an example oracle as a *sample*.

A learning algorithm draws a sample from $EX(f, D)$ and eventually outputs an hypothesis h from some hypothesis class $\mathcal{H}$ defined over X. For any hypothesis h, the *error rate* of h is defined to be the distribution weight of those examples in X where h and f differ. By using the notation $\Pr_D[P(x)]$ to denote the probability of drawing an example in X according to D which satisfies the predicate P, we may define $error(h) = \Pr_D[h(x) \neq f(x)]$. We often think of $\mathcal{H}$ as a class of representations of functions in $\mathcal{F}$, and as such we define $size(f)$ to be the size of the smallest representation in $\mathcal{H}$ of the target function f.

The learner's goal is to output, with probability at least $1-\delta$, an hypothesis h whose error rate is at most ε, for the given *error parameter* ε and *confidence parameter* δ. A learning algorithm is said to be *polynomially efficient* if its running time is polynomial in $1/\varepsilon$, $1/\delta$, n and $size(f)$.

2.1.1 Classification Noise

In the *classification noise model*, the labelled example oracle $EX(f, D)$ is replaced by a noisy example oracle $EX^{\eta}_{\mathrm{CN}}(f, D)$. Each time this noisy example oracle is called, an example $x \in X$ is drawn according to D. The oracle then outputs $\langle x, f(x)\rangle$ with probability $1-\eta$ or $\langle x, \neg f(x)\rangle$ with probability η, randomly and independently for each example drawn. Despite the noise in the labelled examples, the learner's goal remains to output an hypothesis h which, with probability at least $1-\delta$, has error rate $error(h) = \Pr_D[h(x) \neq f(x)]$ at most ε.

While the learner does not typically know the exact value of the *noise rate* η, the learner is given an upper bound η_b on the noise rate, $0 \leq \eta \leq \eta_b < 1/2$, and the learner is said to be polynomially efficient if its running time is polynomial in the usual PAC learning parameters as well as $\frac{1}{1-2\eta_b}$.

2.1.2 Malicious Errors

In the *malicious error model*, the labelled example oracle $EX(f, D)$ is replaced by a noisy example oracle $EX^{\beta}_{\mathrm{MAL}}(f, D)$. When a labelled example is requested from this oracle, with probability $1-\beta$, an example x

is chosen according to D and $\langle x, f(x)\rangle$ is returned to the learner. With probability β, a malicious adversary selects any example x, selects a label $l \in \{0,1\}$, and returns $\langle x, l\rangle$. Again, the learner's goal is to output an hypothesis h which, with probability at least $1-\delta$, has error rate $error(h) = \Pr_D[h(x) \neq f(x)]$ at most ε.

2.2 Statistical Query Based Learning

In the SQ model, the example oracle $EX(f, D)$ from the standard PAC model is replaced by a statistics oracle $STAT(f, D)$. An SQ algorithm queries the $STAT$ oracle for the values of various statistics on the distribution of labelled examples (*e.g.* "What is the probability that a randomly chosen labelled example $\langle x, l\rangle$ has variable $x_i = 0$ and $l = 1$?"), and the $STAT$ oracle returns the requested statistics within some specified additive error. Formally, a statistical query is of the form $[\chi, \tau]$, where χ is a mapping from labelled examples to $\{0,1\}$ (*i.e.* $\chi : X \times \{0,1\} \to \{0,1\}$) corresponding to an indicator function for those labelled examples about which statistics are to be gathered, while τ is an additive error parameter. A call $[\chi, \tau]$ to $STAT(f, D)$ returns an accurate *estimate* $\hat{P}_\chi$ of $P_\chi = \Pr_D[\chi(x, f(x))]$, in that $\hat{P}_\chi$ satisfies $|\hat{P}_\chi - P_\chi| \leq \tau$.

A call to the $STAT$ oracle can be simulated, with high probability, by drawing a sufficiently large sample from the example oracle and outputting the fraction of labelled examples which satisfy χ as the estimate $\hat{P}_\chi$. Since the required sample size depends polynomially on $1/\tau$ and the simulation time additionally depends on the time required to evaluate χ, an SQ learning algorithm is said to be polynomially efficient if $1/\tau$, the time required to evaluate each χ, and the running time of the SQ algorithm are all bounded by polynomials in $1/\varepsilon$, n and $size(f)$. An SQ learning algorithm is said to use *query space* $\mathcal{Q}$ if it only makes queries of the form $[\chi, \tau]$ where $\chi \in \mathcal{Q}$, and we let τ_* be a lower bound on the additive error of every query made by an SQ algorithm.

3 Statistical Queries with Relative Error Estimates

In this section, we formally define the model of learning from relative error statistical queries and relate learnability in this new model to learnability in the standard model of additive error statistical queries. We then give some examples which show that relative error SQ is a very natural language for specifying learning algorithms.

3.1 The Relative Error SQ Model

Given the motivation above, we modify the standard model of statistical query learning to allow for estimates to be requested with relative error. We replace the additive error $STAT(f, D)$ oracle with a relative error *Rel-STAT*(f, D) oracle which accepts a query χ, a relative error parameter μ, and a threshold parameter θ. The value $P_\chi = \Pr_D[\chi(x, f(x))]$ is defined as before. If P_χ is less than the threshold θ, then the oracle may return the symbol $\perp$. If the oracle does not return $\perp$, then it must return an estimate $\hat{P}_\chi$ such that

$$P_\chi(1-\mu) \leq \hat{P}_\chi \leq P_\chi(1+\mu).$$

Note that the oracle may chose to return an accurate estimate even if $P_\chi < \theta$. A class is said to be learnable by relative error statistical queries if it satisfies the same conditions of additive error statistical query learning except we instead require that $1/\mu$ and $1/\theta$ are polynomially bounded. Let μ_* and θ_* be lower bounds on the relative error and threshold of every query made by an SQ algorithm. Given this definition of relative error statistical query learning, we have the following desirable equivalence which preserves learnability and hardness results from the additive SQ model.

Theorem 1 *$\mathcal{F}$ is polynomially learnable by additive error statistical queries if and only if $\mathcal{F}$ is polynomially learnable by relative error statistical queries.*

Proof: Simulate each additive query $[\chi, \tau]$ by a relative query $[\chi, \tau, \tau]$. (If $\hat{P}_\chi = \perp$, then return 0; else, return $\hat{P}_\chi$.) Similarly, simulate each relative query $[\chi, \mu, \theta]$ by an additive query $[\chi, \mu\theta/3]$. (If $\hat{P}_\chi < \theta(1-\mu/3)$, then return $\perp$; else, return $\hat{P}_\chi$.) □

3.2 A Natural Example of Relative Error SQ

We next examine a learning problem which has both a simple additive error SQ algorithm and a simple relative error SQ algorithm. We consider the problem of learning a monotone conjunction of Boolean variables in which the learning algorithm must determine which subset of the variables $\{x_1, \ldots, x_n\}$ are contained in the unknown target conjunction f.

Both algorithms attempt to construct an hypothesis h which contains all the variables in the target function f, and thus correctly classifies all negative examples. These algorithms further attempt to guarantee that for each variable x_i in h, the distribution weight of examples which satisfy "$x_i = 0$ and $f(x) = 1$" is at most ε/n. Therefore, the distribution weight of positive examples which h misclassifies is at most ε, and such an hypothesis has error rate at most ε.

Consider the query: $\chi_i(x, l) \doteq [(x_i = 0) \wedge (l = 1)]$. P_{χ_i} is simply the probability that x_i is false and $f(x)$ is true. Accurate knowledge of the value of P_{χ_i} is sufficient for finding the type of hypothesis described above, since if variable x_i is in f, then $P_{\chi_i} = 0$, while if a variable x_i not in f is included in h, then the error due to this inclusion is at most P_{χ_i}.

Using this strategy, an additive error SQ algorithm submits each query χ_i with additive error $\varepsilon/2n$ and in-

cludes all variables for which the estimate $\hat{P}_{\chi_i} \leq \varepsilon/2n$. If $\hat{P}_{\chi_i} > \varepsilon/2n$, then $P_{\chi_i} > 0$ so it is correct to not include variable x_i. If $\hat{P}_{\chi_i} \leq \varepsilon/2n$ then $P_{\chi_i} \leq \varepsilon/n$. Including variables of this type will collectively incur an error of at most ε on the positive examples, and therefore this SQ algorithm satisfies the learning criteria.

Note that the SQ oracle is constrained, by the specification of the algorithm, to return an estimate of P_{χ_i} with additive error at most $\varepsilon/2n$, even if the value of the query is quite large. A simple relative error SQ algorithm can be constructed which avoids this pitfall. The relative error SQ algorithm submits each query χ_i with relative error $1/2$ and threshold ε/n. All variables are included in the hypothesis for which the estimate $\hat{P}_{\chi_i} = 0$ or $\perp$. This algorithm achieves the same goals as the additive error SQ algorithm described above, yet does so much more efficiently. Specifically, the sample complexity of the standard, noise-free PAC simulation of additive error SQ algorithms depends linearly on $1/\tau_*^2$ [12], while in Section 5, we show that the sample complexity of a noise-free PAC simulation of relative error SQ algorithms depends linearly on $1/\mu_*^2\theta_*$. Note that in the above algorithms for learning conjunctions, $1/\tau_*^2 = \Theta(n^2/\varepsilon^2)$ while $1/\mu_*^2\theta_* = \Theta(n/\varepsilon)$.

As alluded to in the introduction, the relative error SQ algorithms for many commonly studied problems, including many covering algorithms, learning k-decision lists and learning axis parallel rectangles over $\Re^n$, are similar to the algorithm for conjunctions in that they use a *constant* μ_* and a θ_* which depends only *linearly* on ε. These relative error SQ algorithms yield very efficient PAC algorithms. One may observe that this property on (μ, θ) is achieved by noting that these algorithms simply require answers to the following type of question, "Is P_χ less than θ, or is it greater than a $1-\mu$ factor of θ?" where μ is a constant such as $1/2$, and θ depends linearly on ε. We show in Section 6 that *no* learning problem requires μ_* to depend on ε and that the ε-dependence of θ_* need only be $\tilde{\Omega}(\varepsilon)$.

4 Relative Error (μ, θ) Uniform Convergence

In order to prove sample complexity bounds for the simulations given in Section 5, we first derive the number of examples required to answer (μ, θ) queries. Although simple cases could be analyzed using Chernoff bounds on the tail of a binomial distribution, we shall desire accurate estimates of a possibly infinite set of probabilities through the use of a single sample of data. We therefore look to take advantage of uniform convergence results based on the Vapnik-Chervonenkis dimension of the class of queries to be estimated.

Consider the d_ν metric defined over the non-negative reals as follows [15, 11]:

$$d_\nu(r, s) = \frac{|r - s|}{\nu + r + s}$$

Haussler ([11]: Definition 3 and Theorems 1 and 7) effectively proves the following theorem on the sample size sufficient to ensure uniform convergence, with respect to the d_ν metric, of empirical estimates to true probabilities.

Theorem 2 (Haussler) *Let $\mathcal{G}$ be a set of $\{0,1\}$-valued functions and for a given $g \in \mathcal{G}$, let $E(g)$ be the true probability of drawing an example x such that $g(x) = 1$, and let $\hat{E}(g)$ be the fraction of examples x in a random sample of size m such that $g(x) = 1$. Then*

$$\Pr[\exists g \in \mathcal{G} \ : \ d_\nu(\hat{E}(g), E(g)) > \alpha] \leq \delta$$

for $m \geq \frac{8}{\alpha^2\nu}\big(2d \ln \frac{8e}{\alpha\nu} + \ln \frac{8}{\delta}\big)$ if $\mathcal{G}$ has finite VC-dimension d or $m \geq \frac{1}{\alpha^2\nu} \ln \frac{2|\mathcal{G}|}{\delta}$ if $\mathcal{G}$ is finite. The probability is over the random draw of the sample.

Our goal is to draw a single sample sufficient to ensure that the d_ν metric converges uniformly (*i.e.* for every query to be estimated) and to use this sample to determine (μ, θ) estimates for any query. In order to do this, we prove the following theorem which relates bounded d_ν metric to relative error estimation in the (μ, θ) sense. Specifically, it states that if we bound the d_ν distance between the true and empirical values of a query, then the empirical value either implies that the true value is below the threshold θ or the empirical value itself is a sufficiently accurate estimate (*i.e.* within a $1 \pm \mu$ factor). The proof of Theorem 3 appears in the appendix.

Theorem 3 *If $d_{\theta/4}(P_\chi, \hat{P}_\chi) \leq \mu/4$, then*
(1) $\hat{P}_\chi < \theta/2$ implies that $P_\chi < \theta$, and
(2) $\hat{P}_\chi \geq \theta/2$ implies that $(1-\mu)P_\chi \leq \hat{P}_\chi \leq (1+\mu)P_\chi$.

Combining Theorem 3 with Theorem 2, we have the following corollary:

Corollary 1 *Let $\mathcal{Q}$ be a set of queries. With probability at least $1-\delta$, we can use a single sample from a noise-free example oracle in order to correctly answer the request $[\chi, \mu, \theta]$ for every $\chi \in \mathcal{Q}$. If $\mathcal{Q}$ is finite, then a sample of size $m = O\big(\frac{1}{\mu^2\theta} \log \frac{|\mathcal{Q}|}{\delta}\big)$ is sufficient, while if $\mathcal{Q}$ has finite VC-dimension q, then a sample of size $m = O\big(\frac{q}{\mu^2\theta} \log \frac{1}{\mu\theta} + \frac{1}{\mu^2\theta} \log \frac{1}{\delta}\big)$ is sufficient.*

5 Simulating Relative Error SQ Algorithms

In this section, we give the complexity of simulating relative error SQ algorithms in the PAC model, both in the absence and presence of noise.

5.1 Simulation in the Absence of Noise

The simulation of relative error SQ algorithms in the noise-free PAC model simply draws one sample and uses it to simulate every query. The sample complexity of this simulation follows directly from Corollary 1.

Theorem 4 *If $\mathcal{F}$ is learnable by a statistical query algorithm which makes queries from query space $\mathcal{Q}$ with worst case relative error μ_* and worst case threshold θ_*, then $\mathcal{F}$ is PAC learnable with sample complexity $O\big(\frac{1}{\mu_*^2\theta_*}\log\frac{|\mathcal{Q}|}{\delta}\big)$ when $\mathcal{Q}$ is finite or $O\big(\frac{q}{\mu_*^2\theta_*}\log\frac{1}{\mu_*\theta_*}+\frac{1}{\mu_*^2\theta_*}\log\frac{1}{\delta}\big)$ when $\mathcal{Q}$ has finite VC-dimension q.*

Note that this implies that monotone conjunctions can be learned via relative error statistical queries using a sample of size $O\big(\frac{n}{\varepsilon}\log\frac{n}{\delta}\big)$, an n/ε factor better than that obtainable via the standard simulation of additive error SQ algorithms. We note that the simulations of additive error SQ algorithms could also be improved by either accuracy boosting or Occam techniques, and the sample complexities obtained are roughly the same as given here.

5.2 Simulation in the Presence of Malicious Errors

We next consider the simulation of relative error SQ algorithms in the presence of malicious errors. Decatur [5] showed that an SQ algorithm can be simulated in the presence of malicious errors with a maximum allowable error rate which depends on τ_*, the smallest additive error required by the SQ algorithm. In Theorem 5, we show that an SQ algorithm can be simulated in the presence of malicious errors with a maximum allowable error rate and sample complexity which depend on μ_* and θ_*, the minimum *relative error* and *threshold* required by the SQ algorithm.

The key to this malicious-error tolerant simulation is to draw a large enough sample to ensure that for each query, the combined error in an estimate due to both the adversary and the statistical fluctuation on error-free examples is less than the accuracy required for the query.

Theorem 5 *If $\mathcal{F}$ is learnable by a statistical query algorithm which uses query space $\mathcal{Q}$ with worst case relative error μ_* and worst case threshold θ_*, then $\mathcal{F}$ is PAC learnable in the presence of malicious errors with error rate $\beta \le \mu_*\theta_*/32$. The sample complexity required is $O\big(\frac{1}{\mu_*^2\theta_*}\log\frac{|\mathcal{Q}|}{\delta}\big)$ when $\mathcal{Q}$ is finite or $O\big(\frac{q}{\mu_*^2\theta_*}\log\frac{1}{\mu_*\theta_*}+\frac{1}{\mu_*^2\theta_*}\log\frac{1}{\delta}\big)$ when $\mathcal{Q}$ has finite VC-dimension q.*

Proof: We show this result for a model of malicious errors in which the adversary is at least as powerful as the standard model. Here, all m labelled examples are first drawn in a noise-free manner. Then, m coins are flipped, each with probability β of HEADS, and we let $\hat{\beta}$ be the number of occurrences of HEADS divided by m. The adversary may then corrupt *any* $\hat{\beta}m$ of the labelled examples.

We define $\hat{P}_\chi$ to be the fraction of examples in the uncorrupted sample which satisfy χ, and $\hat{P}_\chi^\beta$ to be the fraction of examples in the corrupted sample which satisfy χ. The sample size m is chosen to be sufficient to ensure, with high probability, that for all $\chi \in \mathcal{Q}$, $d_{\theta/16}(P_\chi, \hat{P}_\chi) \le \mu/16$ and that $\hat{\beta} \le 2\beta$. Note that this latter condition implies that for all χ, $|\hat{P}_\chi - \hat{P}_\chi^\beta| \le \hat{\beta} \le \mu\theta/16$.

The simulation of a given query $[\chi, \mu, \theta]$ first computes $\hat{P}_\chi^\beta$, the fraction of examples in the (corrupted) sample which satisfy χ. Note that $\hat{P}_\chi^\beta < \theta/4$ implies that $\hat{P}_\chi < \theta/2$, and since $d_{\theta/16}(P_\chi, \hat{P}_\chi) \le \mu/16$ implies that $d_{\theta/4}(P_\chi, \hat{P}_\chi) \le \mu/4$, by Theorem 3 we have $P_\chi < \theta$. In this case, $\perp$ is returned.

Otherwise, $\hat{P}_\chi^\beta \ge \theta/4$, in which case $\hat{P}_\chi^\beta$ is returned. We must therefore ensure that this estimate is within a $1 \pm \mu$ factor of P_χ, or equivalently within an additive $\pm\mu P_\chi$ of P_χ. Since $\hat{P}_\chi^\beta \ge \theta/4$ and $\hat{\beta} \le \theta/8$, we know that $\hat{P}_\chi \ge \theta/8$. By the uniform convergence ensured on the noise-free sample, we have $P_\chi(1-\mu/4) \le \hat{P}_\chi \le P_\chi(1+\mu/4)$ and therefore, $P_\chi \ge \theta/(8+2\mu)$. The additive difference between $\hat{P}_\chi^\beta$ and P_χ is:

$$|\hat{P}_\chi^\beta - P_\chi| \le |\hat{P}_\chi^\beta - \hat{P}_\chi| + |\hat{P}_\chi - P_\chi| \le \hat{\beta} + P_\chi \cdot \mu/4.$$

Since $\hat{\beta} \le \mu\theta/16$ and $P_\chi \ge \theta/10$, this difference is less than μP_χ. □

Note that this implies that monotone conjunctions can be learned in the presence of malicious errors via relative error statistical queries using a sample of size $O\big(\frac{n}{\varepsilon}\log\frac{n}{\delta}\big)$, an n/ε factor better than that obtainable via standard additive error statistical queries [5]. The maximum allowable malicious error rate is $\Omega(\varepsilon/n)$, identical to that achieved by the additive error simulation and optimal with respect to ε.

Another application of Theorem 5 is learning conjunctions of at most k relevant variables. By appropriately modifying the additive error SQ algorithm for this problem [12], one can obtain a relative error SQ algorithm where μ_* is a constant, $\theta_* = \Omega\big(\frac{\varepsilon}{k\log(1/\varepsilon)}\big)$ and $\log|\mathcal{Q}| = O\big(k\log n\log(1/\varepsilon)\big)$. In the full paper, we give the details of this modification. We then have the following result:

Theorem 6 *The class of conjunctions of size k over n variables is PAC learnable with malicious errors. The maximum allowable malicious error rate is $\Omega\big(\frac{\varepsilon}{k\log(1/\varepsilon)}\big)$, and the sample complexity required is $O\big(\frac{k^2}{\varepsilon}\log^2\frac{1}{\varepsilon}\log n + \frac{k}{\varepsilon}\log\frac{1}{\varepsilon}\log\frac{1}{\delta}\big)$.*

Note that this malicious error tolerance is the best

known for this problem (identical to results from additive SQ), while the sample complexity is the best known for achieving this error tolerance (a $\frac{k}{\varepsilon}\log\frac{1}{\varepsilon}$ factor better than results from additive SQ).

5.3 Simulation in the Presence of Classification Noise

For SQ simulations in the classification noise model, we improve the standard techniques in multiple ways. We first introduce a new decomposition of P_χ into quantities that can be estimated directly using the classification noise example oracle. We then describe a scheme for "guessing" the noise rate which is significantly more efficient than the standard technique. We then describe our strategy for estimating a single query, which uses no more examples than the standard estimation, and for some classes of queries, uses significantly fewer examples. We characterize the number of examples required as a function of parameters of the query. Finally, we give a technique for hypothesis testing (required to determine which noise rate guess was correct) which uses fewer example than the standard technique. This improved hypothesis testing is achieved by generalizing a result of Laird [14].

We begin be giving a new decomposition of P_χ into quantities that may be "guessed" or estimated using the classification noise oracle. Let $\wedge$, $\oplus$ and $\equiv$ be the standard binary operators for AND, exclusive-OR and equivalence, respectively. For any query χ, we define the following four queries:

$$\begin{aligned}
\chi_\oplus(x,l) &\doteq (\chi(x,0)\oplus\chi(x,1))\\
\chi_\equiv(x,l) &\doteq (\chi(x,0)\equiv\chi(x,1))\\
\chi_1(x,l) &\doteq (\chi(x,l)\wedge\chi_\oplus(x,l))\\
\chi_2(x,l) &\doteq (\chi(x,l)\wedge\chi_\equiv(x,l))
\end{aligned}$$

Note that $P_{\chi_\oplus}$ and $P_{\chi_\equiv}$ are the probabilities that χ does or does not depend on the label, respectively, while P_{χ_1} and P_{χ_2} are the probabilities that $\chi=1$ in those respective regions. Note that $P_\chi = P_{\chi_1}+P_{\chi_2}$. Finally, for any query χ, let P^η_χ be the probability that $\chi=1$ with respect to examples drawn from the *noisy* example oracle $EX^\eta_{\mathrm{CN}}(f,D)$. We then have the following decomposition of P_χ, whose derivation is given in the full paper:

Lemma 1 $P_\chi = P^\eta_{\chi_2} + \dfrac{P^\eta_{\chi_1} - \eta P^\eta_{\chi_\oplus}}{1-2\eta}$

In order to determine an estimate of P_χ with (μ,θ) error, we must determine how accurately the various quantities on the right-hand side of the above equation must be estimated. This is accomplished by a *sensativity analysis*, the result of which is given below. Best results are achieved if we parameterize the dependence of P_χ on the labels of examples. For any query $[\chi,\mu,\theta]$, define ρ as follows: If $P_{\chi_\oplus}<\theta$, then $\rho=1$; else $\rho=\theta/P_{\chi_\oplus}$. Thus, ρ is the ratio between θ and $P_{\chi_\oplus}$ for queries with a "significant" dependence on the labels of examples, and note that $\rho\in[\theta,1]$. Finally, for notational convenience we say that a probability p is estimated with $\langle\alpha,\nu\rangle$ error if the estimate $\hat{p}$ satisfies $d_\nu(p,\hat{p})\le\alpha$.

Lemma 2 *In order to determine P_χ with (μ,θ) error, it is sufficient to obtain an estimate of η within additive error $c_1\mu\theta(1-2\eta)$ and to obtain estimates of $P^\eta_{\chi_2}$, $P^\eta_{\chi_1}$ and $P^\eta_{\chi_\oplus}$ with $\langle\alpha_1,\nu_1\rangle$, $\langle\alpha_2,\nu_2\rangle$ and $\langle\alpha_3,\nu_3\rangle$ error, respectively, where, for all i, $\alpha_i,\nu_i\in(0,1)$ and ${\alpha_i}^2\nu_i \ge c_2\mu^2\theta\rho(1-2\eta_b)^2$, and c_1 and c_2 are constants.*

The proof of Lemma 2 appears in the full paper. In order to make use of this decomposition, we run the SQ algorithm multiple times, each with a different guess for the unknown noise rate η. A specific run of the algorithm uses this guess and computes accurate estimates of $P^\eta_{\chi_2}$, $P^\eta_{\chi_1}$ and $P^\eta_{\chi_\oplus}$ for each χ requested. On any run in which the noise rate guess is sufficiently accurate, by the correctness of the SQ algorithm, the hypothesis output is ε-good. By making a set of noise rate guesses such that at least one is sufficiently accurate, we obtain a set of hypotheses, at least one of which is ε-good.[2] The following lemma, whose proof is given in the appendix, states how many noise rate guesses are sufficient.

Lemma 3 *For all $\gamma,\eta_b<1/2$, there exists a sequence of η-guesses $\{\eta_0,\eta_1,\ldots,\eta_i\}$ where $i=O\big(\frac{1}{\gamma}\log\frac{1}{1-2\eta_b}\big)$ such that for all $\eta\in[0,\eta_b]$, there exists an η_j which satisfies $|\eta-\eta_j|\le\gamma(1-2\eta)$.*

The proof of Lemma 3 is given in the appendix. We next show how many examples are required to estimate the values of $P^\eta_{\chi_2}$, $P^\eta_{\chi_1}$ and $P^\eta_{\chi_\oplus}$ within their required tolerances for every query χ. Note that we cannot apply Theorem 2 directly since we only have a lower bound on the product $\alpha^2\nu$ from Lemma 2 above. However, it is possible to modify the result of Theorem 2 to ensure uniform convergence *for all* $\langle\alpha_i,\nu_i\rangle$ satisfying ${\alpha_i}^2\nu_i\ge z$ as shown below.

Lemma 4 *Let $\mathcal{G}$ be a set of $\{0,1\}$-valued functions and for a given $g\in\mathcal{G}$, let $E(g)$ be the true probability of drawing an example x such that $g(x)=1$, and let $\hat{E}(g)$ be the fraction of examples x in a random sample of size m such that $g(x)=1$. Then for any $z\in(0,1)$ and for all α,ν such that $\alpha^2\nu\ge z$*

$$\Pr[\exists g\in\mathcal{G} : d_\nu(\hat{E}(g),E(g))>\alpha]\le\delta$$

for $m\ge\frac{16}{z}\big(2d\ln\frac{16e}{z}+\ln\frac{8\lceil\lg(1/z)\rceil}{\delta}\big)$ if $\mathcal{G}$ has finite VC-dimension d or $m\ge\frac{2}{z}\ln\frac{2|\mathcal{G}|\lceil\lg(1/z)\rceil}{\delta}$ if $\mathcal{G}$ is finite. The probability is over the random draw of the sample.

The proof of Lemma 4 is given in the appendix. Now, let ρ_* be a lower bound on ρ for every query

[2] We shall actually desire an $\varepsilon/2$-good hypothesis and therefore run the SQ algorithm with accuracy parameter $\varepsilon/2$.

$[\chi, \mu, \theta]$, $\rho_* \in [\theta_*, 1]$, and note that the size and VC-dimension of our new query space (composed of queries of the form χ_1, χ_2, and $\chi_\oplus$) are only increased by constant factors. We then have the following:

Lemma 5 *Let $\mathcal{Q}$ be a set of queries. With probability at least $1-\delta$, we can use a single sample from a classification noise oracle in order to properly estimate $P^\eta_{\chi_2}$, $P^\eta_{\chi_1}$ and $P^\eta_{\chi_\oplus}$ for every $\chi \in \mathcal{Q}$. If $\mathcal{Q}$ is finite, then a sample of size $m = O\big(\frac{1}{\mu_*^2\theta_*\rho_*(1-2\eta_b)^2}\log\big(\frac{|\mathcal{Q}|}{\delta}\log\frac{1}{\mu_*\theta_*\rho_*(1-2\eta_b)}\big)\big)$ is sufficient, while if $\mathcal{Q}$ has finite VC-dimension q, then a sample of size $m = O\big(\frac{q}{\mu_*^2\theta_*\rho_*(1-2\eta_b)^2}\log\frac{1}{\mu_*\theta_*\rho_*(1-2\eta_b)} + \frac{1}{\mu_*^2\theta_*\rho_*(1-2\eta_b)^2}\log\frac{1}{\delta}\big)$ is sufficient.*

If our quantities are estimated sufficiently accurately and some η-guess was good, then some run of the SQ algorithm should have produced a "good" hypothesis. We are now left with the task of finding this good hypothesis among those produced. Assume that we have run our SQ algorithms with an accuracy parameter $\varepsilon' = \varepsilon/2$. Then we can find some ε-good hypothesis through the use of the following generalization of a theorem due to Laird [14]. We give the proof of this lemma in the full paper.

Lemma 6 *Let $\mathcal{H}$ be a set of hypotheses at least one of which has true error rate at most e_1, and let e_2 be any threshold strictly larger than e_1. Then the hypothesis in $\mathcal{H}$ which minimizes disagreements with respect to a sample of size $O\big(\frac{e_1+e_2}{(e_2-e_1)^2(1-2\eta_b)^2}\log\frac{|\mathcal{H}|}{\delta}\big)$ drawn from $EX^\eta_{\mathrm{CN}}(f, D)$ will, with probability at least $1-\delta$, have true error rate at most e_2.*

Note that in our case, $e_1 = \varepsilon/2$ and $e_2 = \varepsilon$, and thus $(e_1+e_2)/(e_2-e_1)^2 = \Theta(1/\varepsilon)$, while $|\mathcal{H}|$ corresponds to the number of noise rate guesses which is $O\big(\frac{1}{\mu_*\theta_*}\log\frac{1}{1-2\eta_b}\big)$.

Combining the sample complexities for estimating queries and hypothesis testing, and noting that $\mu_*\theta_* = O(\varepsilon)$ (by combining the proof of Theorem 1 with a theorem of Kearns [12]), we obtain the following:

Theorem 7 *The total sample complexity required to simulate an SQ algorithm in the presence of classification noise is*

$$O\big(\tfrac{\log(1/\mu_*\theta_*)}{\varepsilon(1-2\eta_b)^2} + \tfrac{\log(|\mathcal{Q}|\log(1/\mu_*\theta_*\rho_*(1-2\eta_b))/\delta)}{\mu_*^2\theta_*\rho_*(1-2\eta_b)^2}\big) = \tilde{O}\big(\tfrac{\log(|\mathcal{Q}|/\delta)}{\mu_*^2\theta_*\rho_*(1-2\eta_b)^2}\big)$$

when $\mathcal{Q}$ is finite, or

$$O\big(\tfrac{q\log(1/\mu_*\theta_*\rho_*(1-2\eta_b))+\log(1/\delta)}{\mu_*^2\theta_*\rho_*(1-2\eta_b)^2}\big) = \tilde{O}\big(\tfrac{q+\log(1/\delta)}{\mu_*^2\theta_*\rho_*(1-2\eta_b)^2}\big)$$

when $\mathcal{Q}$ has finite VC-dimension q.

When applied to the algorithm for learning monotone conjunctions described in Section 3.2, we find that monotone conjunctions can be learned in the presence of classification noise via relative error statistical queries using a sample of size $\tilde{O}\big(\frac{n^2}{\varepsilon^2(1-2\eta_b)^2}\log\frac{1}{\delta}\big)$. Note that this is roughly the same sample complexity that can be obtained via a simulation of the appropriate additive error SQ learning algorithm. Essentially, our relative error SQ learning algorithm for conjunctions does not take advantage of the improved simulation, and it is an open question as to whether such a relative error SQ learning algorithm for conjunctions exists. However, some natural relative error SQ algorithms can be simulated highly efficiently as we describe below.

The class $\mathcal{S}$ of symmetric functions over the domain $\{0,1\}^n$ is the set of all Boolean functions f for which $H(x) = H(y)$ implies $f(x) = f(y)$, where $H(x)$ represents that number of components which are 1 in the Boolean vector x. Note that there are 2^{n+1} functions in $\mathcal{S}$ and that the VC-dimension of $\mathcal{S}$ is $n+1$.

This class has a very simple relative error SQ algorithm with $|Q| = O(n)$, $\mu_* = \Omega(1)$, $\theta_* = \Omega(\varepsilon/n)$ and $\rho_* = \Omega(1)$. Therefore, the classification noise simulation of this algorithm uses only $\tilde{O}\big(\frac{n}{\varepsilon(1-2\eta_b)^2}\log\frac{1}{\delta}\big)$ examples, while the simulation of the analogous additive error SQ algorithm uses $\tilde{O}\big(\frac{n^2}{\varepsilon^2(1-2\eta_b)^2}\log\frac{1}{\delta}\big)$ examples. Note that our sample complexity roughly matches the lower bound of $\Omega\big(\frac{n}{\varepsilon(1-2\eta)^2}\log\frac{1}{\delta}\big)$ [2]. This is the first non-trivial class to be shown learnable in the presence of classification noise using a sample complexity whose dependence on ε is $o(1/\varepsilon^2)$. The relative error SQ algorithm achieving this bound is as follows (for appropriate constants c_i):

1. $\forall i \in \{0, \ldots, n\}$, let
   $p_i = STAT(f, D)[H(x) = i, c_1, c_2\varepsilon/n]$.
2. $\forall i \in \{0, \ldots, n\}$,
   if $p_i \geq c_3\varepsilon/n$ then let
   $q_i = STAT(f, D)[(l = 1) \wedge (H(x) = i), c_4, c_5 p_i]$
   else let
   $q_i = 0$.
3. Output an hypothesis which classifies x as 1 if and only if $q_{H(x)} \geq p_{H(x)}/2$.

Note that the only queries which depend on the label, and therefore determine ρ, are those in Step 2 in which θ is within a constant factor of $P_{\chi_\oplus}$, implying that $\rho = \Omega(1)$.

6 General Bounds on the Complexity of Relative Error Statistical Query Learning

In this section we show general upper bounds on the complexity of relative error statistical query learning. We do so by applying boosting techniques [9, 10, 16] and specifically, these techniques as applied in the statistical

query model [1]. The proof of Theorem 8 is given in the appendix.

Theorem 8 *If a concept class $\mathcal{F}$ is SQ learnable by algorithm A, then $\mathcal{F}$ is SQ learnable with $O\big(N_0 \log^2(1/\varepsilon)\big)$ queries each with threshold $\Omega\big(\mu_0\theta_0\varepsilon/\log(1/\varepsilon)\big)$ and relative error $\Omega(\mu_0)$. Here N_0, μ_0 and θ_0 are the number of queries, worst case relative error, and worst case threshold, respectively, of algorithm A run with a constant accuracy parameter. Note that N_0, μ_0 and θ_0 are independent of ε.*

Combining the above general upper bounds for relative error SQ with the noise-free and noise-tolerant PAC simulations given in Section 5, we have the following corollaries stating general upper bounds on the sample complexity of the PAC simulations of relative error SQ algorithms.

Corollary 2 *If $\mathcal{F}$ is SQ learnable, then $\mathcal{F}$ is PAC learnable with a sample complexity whose dependence on ε is $\tilde{O}(1/\varepsilon)$.*

Corollary 3 *If $\mathcal{F}$ is SQ learnable, then $\mathcal{F}$ is PAC learnable in the presence of malicious errors. The dependence on ε of the maximum allowable error rate is $\tilde{\Omega}(\varepsilon)$, while the dependence on ε of the required sample complexity is $\tilde{O}(1/\varepsilon)$.*

Corollary 4 *If $\mathcal{F}$ is SQ learnable, then $\mathcal{F}$ is PAC learnable in the presence of classification noise. The dependence on ε and η_b of the required sample complexity is $\tilde{O}\big(\frac{1}{\varepsilon^2(1-2\eta_b)^2}\big)$.*

Corollary 2 implies that any class learnable by statistical queries can be learned by simulating relative error statistical queries using a sample size whose dependence on ε is roughly optimal, whereas the use of additive error statistical queries yielded an additional factor of $1/\varepsilon$.

In Corollary 3, these bounds are within logarithmic factors of both the $O(\varepsilon)$ maximum allowable malicious error rate [13] and the $\Omega(1/\varepsilon)$ lower bound on the sample complexity for *noise-free* PAC learning [8]. We also note that in this malicious-error tolerant PAC simulation, the sample, time, space and hypothesis size complexities are asymptotically identical to the corresponding complexities in our noise-free PAC simulation.

For the case of classification noise, Corollary 4 is identical to the corresponding result obtained by additive error statistical queries. Thus, although an improvement with respect to ε is not made for *every* learning problem, the improvements made by relative error SQ for specific problems do not come at the expense of losses in general upper bounds.

References

[1] Javed Aslam and Scott Decatur. General bounds on statistical query learning and PAC learning with noise via hypothesis boosting. In *Proceedings of the* 34^{th} *Annual Symposium on Foundations of Computer Science*, pages 282–291, November 1993. To appear in *Information and Computation*.

[2] Javed Aslam and Scott Decatur. A general lower bound on the sample complexity of PAC learning in the presence of classification noise. In preparation, 1994.

[3] Avrim Blum, Merrick Furst, Jeffery Jackson, Michael Kearns, Yishay Mansour, and Steven Rudich. Weakly learning DNF and characterizing statistical query learning using Fourier analysis. In *Proceedings of the* 26^{th} *Annual ACM Symposium on the Theory of Computing*, May 1994.

[4] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36(4):929–865, 1989.

[5] Scott Decatur. Statistical queries and faulty PAC oracles. In *Proceedings of the Sixth Annual ACM Workshop on Computational Learning Theory*, pages 262–268. ACM Press, July 1993.

[6] Scott Decatur. Learning in hybrid noise environments using statistical queries. To appear in *Proccedings of the Fifth International Workshop on Artificial Intelligence and Statistics*, January 1995.

[7] Scott Decatur and Rosario Gennaro. On learning from noisy and incomplete examples. In *Proceedings of the Eigth Annual ACM Workshop on Computational Learning Theory*. ACM Press, July 1995.

[8] Andrzej Ehrenfeucht, David Haussler, Michael Kearns, and Leslie Valiant. A general lower bound on the number of examples needed for learning. *Information and Computation*, 82(3):247–251, September 1989.

[9] Yoav Freund. Boosting a weak learning algorithm by majority. In *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pages 202–216. Morgan Kaufmann, 1990.

[10] Yoav Freund. An improved boosting algorithm and its implications on learning complexity. In *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, pages 391–398. ACM Press, 1992.

[11] David Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100:78–150, 1992.

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

[13] Michael Kearns and Ming Li. Learning in the presence of malicious errors. In *Proceedings of the* 20^{th} *Annual ACM Symposium on Theory of Computing*, Chicago, Illinois, May 1988.

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

[15] D. Pollard. Rates of uniform almost-sure convergence for empirical processes indexed by unbounded classes of functions. Manuscript, 1986.

[16] Robert Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–226, 1990.

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

Appendix

Proof of Theorem 3

We first note that $(\forall r,s\geq 0)(\forall \nu>0)(\forall \alpha, 0\leq\alpha<1)$, if $d_\nu(r,s)\leq\alpha$, then

$$\begin{aligned} r &\leq \tfrac{1+\alpha}{1-\alpha}s+\tfrac{\alpha}{1-\alpha}\nu && (1)\\ r &\geq \tfrac{1-\alpha}{1+\alpha}s-\tfrac{\alpha}{1+\alpha}\nu && (2)\\ s &\leq \tfrac{1+\alpha}{1-\alpha}r+\tfrac{\alpha}{1-\alpha}\nu && (3)\\ s &\geq \tfrac{1-\alpha}{1+\alpha}r-\tfrac{\alpha}{1+\alpha}\nu. && (4)\end{aligned}$$

These four bounds follow from the constraint: $|r-s|\leq\alpha(\nu+r+s)$ implied by $d_\nu(r,s)\leq\alpha$. These bounds are then used to prove the following two lemmas which imply the theorem. The first lemma essentially shows that if $d_\nu(r,s)$ is sufficiently "small," then if r is "small" then s is "small" and if r is "large" then s is "large."

Lemma 7 $(\forall r,s\geq 0)(\forall\theta>0)$ *If* $d_{\theta/4}(r,s)\leq 1/4$, *then*
(1) If $r<\theta/2$, *then* $s<\theta$.
(2) If $r\geq\theta/2$, *then* $s\geq\theta/4$.

This lemma follows from Equations 3 and 4. The second lemma essentially shows that if s is sufficiently "large" and if $d_\nu(r,s)$ is sufficiently "small," then r is a "good" approximation of s.

Lemma 8 $(\forall r,s\geq 0)(\forall\nu>0)(\forall\mu,0\leq\mu\leq 1)$ *If* $s\geq\nu$, *then* $d_\nu(r,s)\leq\mu/4$ *implies that* $(1-\mu)s\leq r\leq(1+\mu)s$.

This lemma follows from Equations 1 and 2.

Proof of Lemma 3

The sequence is constructed as follows. Let $\eta_0=0$ and consider how to determine η_j from η_{j-1}. The value η_{j-1} is a valid estimate for all $\eta\geq\eta_{j-1}$ which satisfy $\eta-\gamma(1-2\eta)\leq\eta_{j-1}$. Solving for η, we find that η_{j-1} is a valid estimate for all $\eta\in\left[\eta_{j-1},\frac{\eta_{j-1}+\gamma}{1+2\gamma}\right]$. Consider an $\eta_j>\frac{\eta_{j-1}+\gamma}{1+2\gamma}$. The value η_j is a valid estimate for all $\eta\leq\eta_j$ which satisfy $\eta+\gamma(1-2\eta)\geq\eta_j$. Solving for η, we find that η_j is a valid estimate for all $\eta\in\left[\frac{\eta_j-\gamma}{1-2\gamma},\eta_j\right]$. To ensure that either η_{j-1} or η_j is a valid estimate for any $\eta\in[\eta_{j-1},\eta_j]$, we set

$$\tfrac{\eta_{j-1}+\gamma}{1+2\gamma}=\tfrac{\eta_j-\gamma}{1-2\gamma}.$$

Solving for η_j in terms of η_{j-1}, we obtain

$$\eta_j=\tfrac{1-2\gamma}{1+2\gamma}\eta_{j-1}+\tfrac{2\gamma}{1+2\gamma}.$$

Substituting $\gamma'=2\gamma/(1+2\gamma)$, we obtain the following recurrence:

$$\eta_j=(1-2\gamma')\eta_{j-1}+\gamma'$$

Note that if $\gamma<1/2$, then $\gamma'<1/2$ as well.

By constructing η-guesses using this recurrence, we ensure that for all $\eta\in[0,\eta_i]$, at least one of $\{\eta_0,\ldots,\eta_i\}$ is a valid estimate. Solving this recurrence, we find that

$$\eta_i=\gamma'\sum_{j=0}^{i-1}(1-2\gamma')^j+\eta_0(1-2\gamma')^i.$$

Since $\eta_0=0$ and we are only concerned with $\eta\leq\eta_b$, we may bound the number of guesses required by finding the smallest i which satisfies

$$\gamma'\sum_{j=0}^{i-1}(1-2\gamma')^j\geq\eta_b.$$

Given that

$$\gamma'\sum_{j=0}^{i-1}(1-2\gamma')^j=\gamma'\tfrac{1-(1-2\gamma')^i}{1-(1-2\gamma')}=\tfrac{1-(1-2\gamma')^i}{2}$$

we need $(1-2\gamma')^i\leq 1-2\eta_b$. Solving for i, we find that any $i\geq\ln\frac{1}{1-2\eta_b}/\ln\frac{1}{1-2\gamma'}$ is sufficient. Using the fact that $1/x>1/\ln\frac{1}{1-x}$ for all $x\in(0,1)$, we find that

$$i=\tfrac{1}{2\gamma'}\ln\tfrac{1}{1-2\eta_b}=\tfrac{1+2\gamma}{4\gamma}\ln\tfrac{1}{1-2\eta_b}$$

is an upper bound on the number of guesses required.

Proof of Lemma 4

We first note the fact that for all $\alpha'\leq\alpha$ and $\nu'\leq\nu$, $d_{\nu'}(r,s)\leq\alpha'$ implies that $d_\nu(r,s)\leq\alpha$. Our strategy is as follows. To ensure uniform convergence for all possible $\langle\alpha,\nu\rangle$ subject to the constraint that $\alpha,\nu\in(0,1)$ and $\alpha^2\nu\geq z$, we construct a finite set of $\langle\alpha_i,\nu_i\rangle$ pairs such that for any $\langle\alpha,\nu\rangle$ there exists an appropriate $\langle\alpha_i,\nu_i\rangle$ pair where $\alpha_i\leq\alpha$ and $\nu_i\leq\nu$. We then draw a sample of sufficient size to ensure uniform convergence for each of these $\langle\alpha_i,\nu_i\rangle$ pairs.

Our sequence of pairs is constructed as follows:

$$\langle\alpha_i,\nu_i\rangle=\langle 2^{-(i+1)/2},z2^i\rangle$$

Consider any $\langle\alpha,\nu\rangle$ subject to the constraints $\alpha,\nu\in(0,1)$ and $\alpha^2\nu\geq z$. Since $\alpha<1$ and $\alpha^2\nu\geq z$, we have $\nu>z=\nu_0$. Let ν_i be the largest element of our constructed sequence such that $\nu_i<\nu$; thus, $\nu_i<\nu\leq\nu_{i+1}$. Since $\alpha^2\geq z/\nu$ and $\nu\leq\nu_{i+1}$, we have

$$\alpha^2\geq z/\nu_{i+1}=z/(z2^{i+1})=2^{-(i+1)}={\alpha_i}^2.$$

Thus, $\alpha_i\leq\alpha$ and $\nu_i<\nu$, and so $\langle\alpha,\nu\rangle$ is "covered" by $\langle\alpha_i,\nu_i\rangle$. Since $\nu<1$, we need only ensure uniform convergence for $\langle\alpha_i,\nu_i\rangle$ for $0\leq i<\lceil\lg 1/z\rceil$. To ensure uniform convergence for each of these $\lceil\lg 1/z\rceil$ pairs, with probability at least $1-\delta$, we need only draw a sample whose size depends on $\delta'=\delta/\lceil\lg 1/z\rceil$. The final sample complexity is obtained by noting that $1/{\alpha_i}^2\nu_i=2/z$ for all i.

Proof of Theorem 8

We begin by proving some useful lemmas which allow us to decompose relative estimates of ratios and sums.

Lemma 9 *Let $a = b/c$ where $0 \leq a, b, c \leq 1$. If an estimate of a is desired with (μ, θ) error provided that $c \geq \Phi$, then it is sufficient to estimate c with $(\mu/3, \Phi)$ error and b with $(\mu/3, \theta\Phi/2)$ error.*

Proof: If the estimate $\hat{c}$ is $\perp$ or less than $\Phi(1-\mu/3)$, then $c < \Phi$. Therefore an estimate for a is not required, and we may halt. Otherwise $\hat{c} \geq \Phi(1-\mu/3)$, and therefore $c \geq \Phi(1-\mu/3)/(1+\mu/3) \geq \Phi/2$ since $\mu \leq 1$. If the estimate $\hat{b}$ is $\perp$, then $b < \theta\Phi/2$. Therefore $a = b/c < \theta$, so we may answer $\hat{a} = \perp$. Otherwise, $\hat{b}$ and $\hat{c}$ are estimates of b and c, each within a $1 \pm \mu/3$ factor. It is easy to show that $\hat{b}/\hat{c}$ is within a $1 \pm \mu$ factor of a. □

Lemma 10 *Let $s = \sum p_i z_i$ where the $\{p_i\}$ are known, $0 \leq s, p_i, z_i \leq 1$ and $\sum_i p_i \leq 1$. If an estimate of s is desired with (μ, θ) error, then it is sufficient to estimate each z_i with $(\mu/3, \mu\theta/3)$ error.*

Proof: Let $B = \{i : \text{estimate of } z_i \text{ is } \perp\}$, $E = \{i : \text{estimate of } z_i \text{ is } \hat{z}_i\}$, $s_B = \sum_B p_i z_i$ and $s_E = \sum_E p_i z_i$. Note that $s_B < \theta\mu/3$. Let $\hat{s}_E = \sum_E p_i \hat{z}_i$. If $\hat{s}_E < \theta(1-\mu/3)^2$ then we return $\perp$, otherwise we return $\hat{s}_E$.

If $\hat{s}_E < \theta(1-\mu/3)^2$, then $s_E < \theta(1-\mu/3)$. But in this case $s = s_E + s_B < \theta(1-\mu/3) + \theta\mu/3 = \theta$, so we are correct in returning $\perp$.

Otherwise we return $\hat{s}_E$ which is at least $\theta(1-\mu/3)^2$. If $B = \emptyset$, then it is easy to see that $\hat{s}_E$ is within a $1 \pm \mu/3$ (and therefore $1 \pm \mu$) factor of s. Otherwise, we are implicitly setting $\hat{z}_i = 0$ for each $i \in B$, and therefore it is enough to show that $\hat{s}_E \geq s(1-\mu)$.

Since $\hat{s}_E \geq \theta(1-\mu/3)^2$, we have $s_E \geq \theta(1-\mu/3)^2/(1+\mu/3)$. Using the fact that for all $\mu \leq 1$, $(1-\mu/3)/(1+\mu/3) \geq 1/2$, we have $s_E \geq \theta(1-\mu/3)/2$. If $\hat{s}_E \geq (\theta\mu/3 + s_E)(1-\mu)$, then $\hat{s}_E \geq s(1-\mu)$ since $s_B < \theta\mu/3$ and $s = s_B + s_E$. But since $\hat{s}_E \geq s_E(1-\mu/3)$, this condition holds when $s_E(1-\mu/3) \geq (\theta\mu/3 + s_E)(1-\mu)$. Solving for s_E, this final condition holds when $s_E \geq \theta(1-\mu/3)/2$ which we have shown to be true whenever an estimate is returned. □

Now, Aslam and Decatur [1] show that given an SQ learning algorithm A, one can construct a very efficient SQ algorithm by combining the output of $O\big(\log(1/\varepsilon)\big)$ runs of A. Each run of A is made with respect to a different distribution and uses accuracy parameter $\varepsilon = 1/4$. Each run makes at most N_0 queries, each with relative error no smaller than μ_0 and threshold no smaller than θ_0. In run $i+1$, the algorithm makes queries of the form $STAT(f, D_{i+1})[\chi(x, f(x))]$ where D_{i+1} is a distribution based on D. Since the learning algorithm only has access to a statistics oracle for D, it is shown that a query with respect to D_{i+1} may be written in terms of new queries with respect to D as follows:

$$STAT(f, D_{i+1})[\chi(x, f(x))] = \frac{\sum_{j=0}^{w} \lambda_j^w \cdot STAT(f, D)[\chi(x, f(x)) \wedge \chi_j^w(x, f(x))]}{\sum_{j=0}^{w} \lambda_j^w \cdot STAT(f, D)[\chi_j^w(x, f(x))]}$$

In the above equation $w \leq i$, the values $\lambda_j^w \in [0, 1]$ are known, and $\sum_j \lambda_j^w \leq 1$. It is also the case that if the denominator of the right-hand side of the above equation is less than $\Phi = \Omega\big(\varepsilon/\log(1/\varepsilon)\big)$, then the query need not be estimated. Using Lemmas 9 and 10, it is easy to show that a query to $STAT(f, D_{i+1})$ can be estimated by making queries to $STAT(f, D)$ with relative error at least $\mu_0/9$ and threshold at least $\mu_0\theta_0\Phi/18$. Since a query with respect to D_{i+1} requires $O(i)$ queries with respect to D, the total number of queries made is $O\big(N_0 \log^2(1/\varepsilon)\big)$.