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

Javed A. Aslam*
Department of Computer Science
Dartmouth College
Hanover, NH 03755

Scott E. Decatur†
Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139

*Portions of this work were performed while the author was at Harvard University and supported by Air Force Contract F49620-92-J-0466 and while the author was at MIT and supported by DARPA Contract N00014-87-K-825 and by NSF Grant CCR-89-14428. Author's current net address: `jaa@cs.dartmouth.edu`

†This work was performed while the author was at Harvard University and supported by an NDSEG Doctoral Fellowship and by NSF Grant CCR-92-00884. Author's current net address: `sed@dimacs.rutgers.edu`

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 more 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.

1 Introduction

In this paper, we focus on the development of efficient, fault-tolerant algorithms for learning in the *probably approximately correct* (PAC) model [19]. 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) which has 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 [14]. In this model, instead of using labelled examples, the algorithm asks for the estimates of the values 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 [14]. The most important use of the SQ model stems from the property that statistical queries can also 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 [14, 6, 8, 7].

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 [14] which uses $\Omega(1/\tau_*^2)$ examples. Since $1/\tau_* = \Omega(1/\varepsilon)$ for all SQ algorithms [14], 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 $\tilde{\Theta}(1/\varepsilon)$ [5, 9].[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 a single algorithm in the SQ model to achieve both noise-free and noise-tolerant learning and writing multiple algorithms in the various noise-free and noise-tolerant PAC models 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. In this paper, we effectively show that the latter is true.

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 [14] showing that almost all classes known to be PAC learnable are learnable by additive error statistical queries, the hardness result of Kearns [14] for learning parity functions, and the general

[1] For asymptotically growing functions g, $g > 1$, we define $\tilde{O}(g)$ to mean $O(g \log^c g)$ for some constant $c \geq 0$. For asymptotically shrinking functions g, $0 < g < 1$, we define $\tilde{O}(g)$ to mean $O(g \log^c(1/g))$ for some constant $c \geq 0$. We define $\tilde{\Omega}$ similarly for constants $c \leq 0$. Finally, we define $\tilde{\Theta}$ to mean both $\tilde{O}$ *and* $\tilde{\Omega}$. This asymptotic notation, read "soft-O," "soft-Omega," and "soft-Theta," is convenient for expressing bounds while ignoring lower order factors. Note that it is somewhat different than the standard soft-order notation.

hardness results of Blum, *et al.* [4] 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 [17, 13]. 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 $\tilde{\Theta}(\frac{n^2}{\varepsilon^2} \log \frac{1}{\delta})$ examples and tolerate at most $\Theta(\varepsilon/n)$ malicious error, our simulations of the relative error SQ algorithm for conjunctions use $\tilde{\Theta}(\frac{n}{\varepsilon} \log \frac{1}{\delta})$ 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 result for learning this class in the malicious error model.

In the case of classification noise, the simulation for conjunctions has the same sample complexity as the standard simulation, $\tilde{\Theta}(\frac{n^2}{\varepsilon^2(1-2\eta_b)^2} \log \frac{1}{\delta})$, 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 $\tilde{\Theta}(\frac{n}{\varepsilon(1-2\eta_b)^2} \log \frac{1}{\delta})$ improving on the $\tilde{\Theta}(\frac{n^2}{\varepsilon^2(1-2\eta_b)^2} \log \frac{1}{\delta})$ 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 known simulations of standard additive error SQ algorithms (which use $\tilde{\Omega}(1/\varepsilon^2)$ examples), our simulation of relative error SQ algorithms requires only $\tilde{O}(1/\varepsilon)$ examples in the absence of noise. Furthermore, our simulation uses $\tilde{O}(1/\varepsilon)$ examples in the presense of malicious errors, while retaining the $\tilde{\Omega}(\varepsilon)$ error tolerance present in the known simulation. Note that a linear dependence on $1/\varepsilon$ is optimal for both sample complexities [5, 9], and a linear dependence on ε is optimal for the error tolerance in the malicious error model [15].

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 $\mathbb{R}^n$. 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 $\langle x, \ell \rangle$ where the example $x \in X$ is drawn randomly and independently according to the unknown distribution D, and the label $\ell = f(x)$. We often refer to a multiset 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 $\frac{1}{1-2\eta_b}$. (Note that $\frac{1}{1-2\eta_b}$ is linear in the inverse of the difference between $1/2$ and η_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 $\ell \in \{0, 1\}$, and returns $\langle x, \ell\rangle$. 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 $\langle x, \ell\rangle$ has variable $x_4 = 0$ and $\ell = 1$?"), 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.*, $\chi : X \times \{0, 1\} \rightarrow \{0, 1\}$) corresponding to an indicator function for those labelled examples about which a statistic is 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 $size(f)$. We let τ_* be a lower bound on the additive error of every query made by an SQ algorithm, and we say that an SQ learning algorithm uses *query space* $\mathcal{Q}$ if it only makes queries of the form $[\chi, \tau]$ where $\chi \in \mathcal{Q}$.

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 described in the introduction, we modify the standard model of statistical query learning to allow for estimates to be requested with relative error. We replace the additive error SQ oracle with a relative error SQ 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 error 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: One can take any query χ to the additive error oracle which requires additive error τ and simulate it by calling the relative error oracle with relative error τ and threshold τ. If $\hat{P}_\chi = \bot$, then return 0; otherwise, return $\hat{P}_\chi$. Note that if $\hat{P}_\chi = \bot$, then $P_\chi < \tau$ which implies that 0 is a sufficiently accurate estimate. Conversely, if $\hat{P}_\chi \neq \bot$, then $\hat{P}_\chi$ must be within a multiplicative $1 \pm \tau$ factor of P_χ or, equivalently, within a $\pm\tau P_\chi$ additive factor of P_χ. Since $\tau P_\chi \leq \tau$, $\hat{P}_\chi$ is a sufficiently accurate estimate.

Similarly, one can take any query to the relative error oracle which requires relative error μ and threshold θ and simulate it by calling the additive error oracle with additive error $\mu\theta/3$. If $\hat{P}_\chi < \theta(1-\mu/3)$, then return $\bot$; otherwise, return $\hat{P}_\chi$. Note that if $\hat{P}_\chi < \theta(1-\mu/3)$, then $P_\chi < \theta$ which implies that $\bot$ is a valid response. Conversely, if $\hat{P}_\chi \geq \theta(1-\mu/3)$, then we must ensure that $\hat{P}_\chi$ is within a $1 \pm \mu$ multiplicative factor of P_χ. This is tantamount to showing that $\mu\theta/3 \leq \mu P_\chi$, which we demonstrate as follows.

$$
\begin{array}{rrcl}
 & \hat{P}_\chi & \geq & \theta(1-\mu/3) \\
\Rightarrow & P_\chi & \geq & \theta(1-2\mu/3) \\
\Rightarrow & \mu P_\chi & \geq & \mu\theta(1-2\mu/3) \\
 & & \geq & \mu\theta(1-2/3) \\
 & & = & \mu\theta/3
\end{array}
$$

In each direction, the simulation uses polynomially bounded parameters if and only if the original algorithm uses polynomially bounded parameters. $\Box$

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: $\chi_i(x, \ell) \doteq [x_i = 0 \wedge \ell = 1]$. 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 includes 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 $1/\tau_*^2$ [14], 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 $1/(\mu_*^2\theta_*) = \Theta(n/\varepsilon)$.

The relative error SQ algorithms for many commonly studied problems, including many covering algorithms, learning k-decision lists and learning axis parallel rectangles over $\mathbb{R}^n$, 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 [17, 13]:

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

Haussler ([13]: 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 $m \geq \frac{8}{\alpha^2\nu}(2d \ln \frac{8e}{\alpha\nu} + \ln \frac{8}{\delta})$ 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 the 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).

Theorem 3 *If $d_{\theta/4}(\hat{P}_\chi, P_\chi) \leq \mu/4$, then*

(1) $\hat{P}_\chi < \theta/2$ implies that $P_\chi < \theta$, and

(2) $\hat{P}_\chi \geq \theta/2$ implies that $(1-\mu)P_\chi \leq \hat{P}_\chi \leq (1+\mu)P_\chi$.

This theorem is an immediate consequence of two lemmas which we state here and prove in the appendix. Our 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 1 $(\forall r, s \geq 0)(\forall \theta > 0)(\forall \mu, 0 \leq \mu \leq 1)$ *If $d_{\theta/4}(r,s) \leq \mu/4$, then*

1. *If $r < \theta/2$, then $s < \theta$.*

2. *If $r \geq \theta/2$, then $s \geq \theta/4$.*

Our 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 2 $(\forall r, s \geq 0)(\forall \nu > 0)(\forall \mu, 0 \leq \mu \leq 1)$ *If $s \geq \nu$, then $d_\nu(r,s) \leq \mu/4$ implies that $(1-\mu)s \leq r \leq (1+\mu)s$.*

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

Corollary 1 *Let $\mathcal{Q}$ be a set of queries, and let μ_* and θ_* be constants. With probability at least $1-\delta$, a single sample from a noise-free example oracle can be used 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 $m = \frac{64}{\mu_*^2 \theta_*} \ln \frac{2|\mathcal{Q}|}{\delta}$ is sufficient, while if $\mathcal{Q}$ has finite VC-dimension q, then a sample of size $m = \frac{512}{\mu_*^2 \theta_*}(2q \ln \frac{128e}{\mu_* \theta_*} + \ln \frac{8}{\delta})$ is sufficient.*

Proof: By Theorem 3, in order to correctly answer the request $[\chi, \mu_*, \theta_*]$, it is sufficient to ensure that $d_{\theta_*/4}(\hat{P}_\chi, P_\chi) \leq \mu_*/4$. By setting $\nu = \theta_*/4$ and $\alpha = \mu_*/4$ in Theorem 2, we find that the request $[\chi, \mu_*, \theta_*]$ for every $\chi \in \mathcal{Q}$ can be correctly ascertained using a single sample whose size is as given above. $\square$

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 directly to simulate every query by computing the fraction of examples in the sample for which χ is true. 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 $O(\frac{1}{\mu_*^2\theta_*}\log\frac{|\mathcal{Q}|}{\delta})$ when $\mathcal{Q}$ is finite or $O(\frac{q}{\mu_*^2\theta_*}\log\frac{1}{\mu_*\theta_*}+\frac{1}{\mu_*^2\theta_*}\log\frac{1}{\delta})$ 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 $O(\frac{n}{\varepsilon}\log\frac{n}{\delta})$, an n/ε factor better than that obtainable via the standard simulation of additive error SQ algorithms. We note that the sample size required by additive error SQ algorithms could also be improved by either accuracy boosting[2] or Occam techniques,[3] and the sample complexities obtained would be 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 [6] 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.

[2]Set ε to a constant and use the additive error SQ algorithm as a "weak learner" in a boosting scheme.

[3]Draw a single sample as dictated by Occam's Razor [5], set ε to the inverse of the sample size, and simulate the additive error SQ algorithm using a uniform distribution over the sample.

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 up to $\beta = \Theta(\mu_*\theta_*)$. The sample complexity required is $O(\frac{1}{\mu_*^2\theta_*}\log\frac{|\mathcal{Q}|}{\delta})$ when $\mathcal{Q}$ is finite or $O(\frac{q}{\mu_*^2\theta_*}\log\frac{1}{\mu_*\theta_*} + \frac{1}{\mu_*^2\theta_*}\log\frac{1}{\delta})$ 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 in the standard model. Here, all m labelled examples are first drawn in a noise-free manner. Then, m coins are flipped, each with probability $\beta \doteq \mu_*\theta_*/24$ 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 sufficiently large to ensure, with high probability, that $\hat{\beta} \leq 2\beta$ and that for all $\chi \in \mathcal{Q}$, $d_{\theta_*/8}(\hat{P}_\chi, P_\chi) \leq \mu_*/8$. Applying standard Chernoff bounds [1], the former condition can be guaranteed with probability at least $1-\delta/2$ using a sample of size $\frac{72}{\mu_*\theta_*}\ln\frac{2}{\delta}$. Applying Theorem 2, the latter condition can be guaranteed with probability at least $1-\delta/2$ using a sample of size $\frac{512}{\mu_*^2\theta_*}\ln\frac{4|\mathcal{Q}|}{\delta}$ if $\mathcal{Q}$ is finite or $\frac{4096}{\mu_*^2\theta_*}\left(2q\ln\frac{512e}{\mu_*\theta_*} + \ln\frac{16}{\delta}\right)$ if $\mathcal{Q}$ has finite VC-dimension q. Thus, both conditions can be guaranteed with probability at least $1-\delta$ using the latter sample complexities.[4] Finally, note that the condition $\hat{\beta} \leq 2\beta$ implies that for all χ, $|\hat{P}_\chi - \hat{P}_\chi^\beta| \leq \hat{\beta} \leq \mu_*\theta_*/12$.

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 $\hat{P}_\chi^\beta < \theta_*/3$ implies that $\hat{P}_\chi < \theta_*/3 + \mu_*\theta_*/12 < \theta_*/2$. Since $d_{\theta_*/8}(\hat{P}_\chi, P_\chi) \leq \mu_*/8$ implies that $d_{\theta_*/4}(\hat{P}_\chi, P_\chi) \leq \mu_*/4$, by Theorem 3 we have $P_\chi < \theta_*$. In this case, $\perp$ is returned.

Otherwise, $\hat{P}_\chi^\beta \geq \theta_*/3$, 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 $\hat{P}_\chi^\beta \geq \theta_*/3$ and $\hat{\beta} \leq \theta_*/12$, we know that $\hat{P}_\chi \geq \theta_*/4$. Since $d_{\theta_*/8}(\hat{P}_\chi, P_\chi) \leq \mu_*/8$ and $\hat{P}_\chi \geq \theta_*/4$, by Theorem 3 we have $P_\chi(1-\mu_*/2) \leq \hat{P}_\chi \leq P_\chi(1+\mu_*/2)$. Therefore, $P_\chi \geq \frac{\theta_*/4}{1+\mu_*/2} \geq \theta_*/6$. We may now bound the additive difference between P_χ and $\hat{P}_\chi^\beta$ as follows:

$$\begin{aligned} |P_\chi - \hat{P}_\chi^\beta| &\leq |P_\chi - \hat{P}_\chi| + |\hat{P}_\chi - \hat{P}_\chi^\beta| \\ &\leq P_\chi \cdot \mu_*/2 + \mu_*\theta_*/12 \end{aligned}$$

[4] Note that we have not attempted to optimize the constants in these bounds.

Note that for this difference to be at most $\mu_* P_\chi$, we must have

$$\begin{array}{rrcl} & P_\chi \cdot \mu_*/2 + \mu_*\theta_*/12 & \leq & \mu_* P_\chi \\ \Leftrightarrow & P_\chi & \geq & \theta_*/6 \end{array}$$

which has been shown. □

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 $O(\frac{n}{\varepsilon}\log\frac{n}{\delta})$, an n/ε factor better than that obtainable via standard additive error statistical queries [6]. 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 for learning conjunctions in the presence of malicious errors when there are many irrelevant attributes. By duality, identical results also hold for learning disjunctions. We give an algorithm for learning conjunctions which tolerates a malicious error rate *independent* of the number of irrelevant attributes, thus depending only on the number of *relevant* attributes and the desired accuracy. The algorithm is based on a relative error SQ algorithm and therefore has a sample complexity whose dependence on ε roughly matches the general lower bound for noise-free PAC learning. The relative error SQ algorithm is obtained by appropriate modifications to the additive error SQ algorithm for this problem [14]. In the relative error SQ algorithm obtained, μ_* is a constant, $\theta_* = \Omega(\frac{\varepsilon}{k\log(1/\varepsilon)})$ and $\log|\mathcal{Q}| = O(k\log n\log(1/\varepsilon))$. These parameters, combined with Theorem 5, yield the following theorem.

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

Note that this malicious error tolerance is the best known for this problem (identical to results from additive error SQ [6]), 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 error SQ).

Proof: We construct a PAC algorithm which tolerates this malicious error by simulating an efficient relative error SQ algorithm for the problem. The SQ algorithm uses the set cover approach for learning conjunctions of k literals [12]. The additive error SQ version of this algorithm is a conversion from elimination and covering *examples* to elimination and

1. **Learn-k-Conjunction**:
2. $\quad V := \{x_1, \ldots, x_n, \overline{x_1}, \ldots, \overline{x_n}\}$
3. $\quad h := \mathit{TRUE}$
4. $\quad r := 2k \ln(2/\varepsilon)$
5. $\quad$ foreach $v_i \in V$ do begin
6. $\quad\quad \hat{P}_\chi := \mathit{STAT}(f, D)[\overline{v_i} = 1 \wedge \ell = 1, 1/2, \varepsilon/(2r)]$
7. $\quad\quad$ if $(\hat{P}_\chi \neq \perp)$ *and* $(\hat{P}_\chi \neq 0)$ then $V := V - \{v_i\}$
8. $\quad$ end
9. $\quad$ for $z = 1$ to r do begin
10. $\quad\quad i := \mathrm{argmax}_j\ \mathit{STAT}(f, D)[h(x) \wedge \overline{v_j} = 1 \wedge \ell = 0, 1/9, \varepsilon/(6k)]$
11. $\quad\quad V := V - \{v_i\}$
12. $\quad\quad h := h \wedge v_i$
13. $\quad$ end
14. $\quad$ return (h).

Figure 1: Algorithm for learning conjunctions of few relevant variables

covering *probabilities of examples* [14]. Making use of the additive error SQ algorithm would allow the malicious error tolerance stated in the theorem, but would do so using a sample complexity larger than that stated in the theorem by a factor of $\frac{k\log(1/\varepsilon)}{\varepsilon}$. We therefore design a relative error SQ algorithm for learning this class. A relative error SQ learning algorithm for conjunctions of size k is given in Figure 1 and we analyze its correctness below.

The target function f is a conjunction of k literals. We construct an hypothesis h which is a conjunction of $r = O(k \log \frac{1}{\varepsilon})$ literals such that the distribution weight of misclassified positive examples is at most $\varepsilon/2$ and the distribution weight of misclassified negative examples is also at most $\varepsilon/2$.

First, all literals which could contribute more than $\varepsilon/(2r)$ error on the positive examples are eliminated from consideration. This is accomplished in Steps 5–8 in the manner used by the standard SQ algorithm for conjunctions given in Section 3.2. Thus, the probability of a false negative is less than $\varepsilon/2$ since of the at most r literals which eventually compose the hypothesis, each contributes at most $\varepsilon/(2r)$ to this error.

Next, the negative examples are greedily "covered" so that the distribution weight of misclassified negative examples is no more than $\varepsilon/2$. We say that a literal covers all negative examples for which this literal is false. We know that the set of literals of f is a cover of size k for the entire space of negative examples. In Steps 9–13, we iteratively construct h by AND-ing new literals such that the distribution weight of negative examples covered by each new literal is at least a $\frac{1}{2k}$ fraction of the distribution weight of negative examples remaining to be covered.

We demonstrate the correctness of Steps 9–13 as follows. Given a partially constructed hypothesis $h_j = v_{i_1} \wedge v_{i_2} \wedge \cdots \wedge v_{i_j}$, let X_j^- be the set of negative examples not covered by h_j; in other words, $X_j^- = \{x : f(x) = 0 \wedge h_j(x) = 1\}$. Let D_j^- be the *conditional distribution* on X_j^- induced by D; in other words, for any $x \in X_j^-$, $D_j^-(x) = D(x)/D(X_j^-)$. By definition, X_0^- is the space of negative examples and D_0^- is the conditional distribution on X_0^-. We know that the target literals not yet in h_j cover the remaining examples in X_j^-, and therefore there exists a cover of X_j^- of size at most k. Thus there exists at least one literal which covers a set of negative examples in X_j^- whose distribution weight with respect to D_j^- is at least $\frac{1}{k}$.

Given h_j, for each v_i, let $\chi_{j,i}(x, \ell) = [A|B] = [v_i = 0 | \ell = 0 \wedge h_j(x) = 1]$. Note that $P_{\chi_{j,i}}$ is the distribution weight, with respect to D_j^-, of negative examples in X_j^- covered by v_i. Thus there exists a literal v_i such that $P_{\chi_{j,i}}$ is at least $\frac{1}{k}$. To find such a literal, we ask queries of the above form with relative error $\frac{1}{3}$ and threshold $\frac{2}{3k}$.[5] Since there exists a literal v_i such that $P_{\chi_{j,i}} \geq \frac{1}{k}$, we are guaranteed to find some literal $v_{i'}$ such that the estimate $\hat{P}_{\chi_{j,i'}}$ is at least $\frac{1}{k}(1 - \frac{1}{3}) = \frac{2}{3k}$. Note that if $\hat{P}_{\chi_{j,i'}} \geq \frac{2}{3k}$, then $P_{\chi_{j,i'}} \geq \frac{2}{3k}/(1 + \frac{1}{3}) = \frac{1}{2k}$. Thus, by AND-ing $v_{i'}$ with h_j, we are guaranteed to cover a set of negative examples in X_j^- whose distribution weight with respect to D_j^- is at least $\frac{1}{2k}$. Since the distribution weight, with respect to D_0^-, of uncovered negative examples is reduced by at least a $(1 - \frac{1}{2k})$ factor in each iteration, it is easy to show that this method requires no more than $r = 2k \ln \frac{2}{\varepsilon}$ iterations to cover all but a set of negative examples whose distribution weight, with respect to D_0^- (and therefore with respect to D) is at most $\varepsilon/2$.

We now show how to estimate the conditional probability query $[A|B]$ with relative error $\mu = \frac{1}{3}$ and threshold $\theta = \frac{2}{3k}$. We estimate both queries which constitute the standard expansion of the conditional probability: $[A|B] = [A \wedge B]/[B]$. Appealing to Lemma 15 (in Section 6), we first estimate $[B]$, the probability that a negative example is not covered by h, using relative error $\mu/3 = 1/9$ and threshold $\varepsilon/2$. If this estimate is $\perp$ or at most $\varepsilon/2 \cdot (1 - 1/9) = \frac{4\varepsilon}{9}$, then the weight of negative examples misclassified by h is at most $\varepsilon/2$ so we halt and output h. Otherwise we estimate $[A \wedge B]$ with relative error $\mu/3 = 1/9$ and threshold $\theta/2 \cdot \varepsilon/2 = \frac{\varepsilon}{6k}$.

This algorithm uses worst case relative error $\mu_* = 1/9$ and worst case threshold $\theta_* = \varepsilon/(4k \ln(2/\varepsilon))$. The size of the query space $\mathcal{Q}$ used by the algorithm may be bounded by noting that each query is of the form $(h \wedge v_i = 1) \wedge (\ell = b)$ where $b \in \{0, 1\}$ and $h \wedge v_i$ contains at most $r + 1$ of the possible $2n$ literals. Thus, $\log |\mathcal{Q}| = O(k \log \frac{1}{\varepsilon} \log n)$, and the theorem then follows from Theorem 5. □

Finally, we highlight a property of the SQ algorithm used in the above proof which

[5] Note that this is a query for a *conditional* probability, which must be determined by the ratio of two *unconditional* probabilities as shown below.

contributes to its efficiency, and note that many other SQ algorithms (such as those for learning decision lists and axis parallel rectangles) naturally exhibit this property as well. This property allows one to simulate these SQ algorithms in the PAC model with malicious errors with roughly optimal malicious error tolerance *and* sample complexity. The crucial property is that for every query, we need only to determine whether P_χ falls below some threshold or above some constant fraction of this threshold. This allows the relative error parameter μ to be a constant in which case we obtain very efficient malicious error tolerant algorithms.

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 examples than the standard technique. This improved hypothesis testing is achieved by generalizing a result of Laird [16].

We begin by 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 Boolean operators for AND, exclusive-OR and equivalence, respectively. For any query χ, we define the following four queries:

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

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. Also note that $P_\chi = P_{\chi_1} + P_{\chi_2}$. Finally, for any query χ, let P_χ^η be the probability that $\chi = 1$ with respect to examples drawn from the *noisy* example oracle $EX_{\mathrm{CN}}^\eta(f, D)$. Given the above definitions, we have the following.

Lemma 3 $P_\chi = P^\eta_{\chi_2} + \dfrac{P^\eta_{\chi_1} - \eta P^\eta_{\chi_\oplus}}{1-2\eta}$.

Proof: We first note that $P_{\chi_2} = P^\eta_{\chi_2}$ since $\chi_2 = \chi \wedge \chi_\equiv$ and $\chi_\equiv$ is only true when the query χ does not depend on the label. Similarly, we note that $P_{\chi_\oplus} = P^\eta_{\chi_\oplus}$ since $\chi_\oplus$ is independent of the labels given to examples. Finally, we denote the conditional probability that a labelled example satisfies query A given that it satisfies query B, both in the absence and presence of noise, by $P_{A|B}$ and $P^\eta_{A|B}$, respectively.

Now consider the probability $P^\eta_{\chi_1}$; we have the following:

$$\begin{aligned}
P^\eta_{\chi_1} &= P^\eta_{\chi\wedge\chi_\oplus} \\
&= P^\eta_{\chi|\chi_\oplus} \cdot P^\eta_{\chi_\oplus} \\
&= \left[(1-\eta)\cdot P_{\chi|\chi_\oplus} + \eta \cdot P_{\neg\chi|\chi_\oplus}\right] \cdot P^\eta_{\chi_\oplus} \\
&= \left[(1-\eta)\cdot P_{\chi|\chi_\oplus} + \eta \cdot (1 - P_{\chi|\chi_\oplus})\right] \cdot P^\eta_{\chi_\oplus} \\
&= \left[\eta + (1-2\eta)\cdot P_{\chi|\chi_\oplus}\right] \cdot P^\eta_{\chi_\oplus} \\
&= \eta \cdot P^\eta_{\chi_\oplus} + (1-2\eta)\cdot P_{\chi|\chi_\oplus} P^\eta_{\chi_\oplus} \\
&= \eta \cdot P^\eta_{\chi_\oplus} + (1-2\eta)\cdot P_{\chi|\chi_\oplus} P_{\chi_\oplus} \\
&= \eta \cdot P^\eta_{\chi_\oplus} + (1-2\eta)\cdot P_{\chi\wedge\chi_\oplus} \\
&= \eta \cdot P^\eta_{\chi_\oplus} + (1-2\eta)\cdot P_{\chi_1}.
\end{aligned}$$

Solving for P_{χ_1}, we obtain $P_{\chi_1} = (P^\eta_{\chi_1} - \eta P^\eta_{\chi_\oplus})/(1-2\eta)$. We therefore have the following:

$$\begin{aligned}
P_\chi &= P_{\chi_2} + P_{\chi_1} \\
&= P^\eta_{\chi_2} + \frac{P^\eta_{\chi_1} - \eta P^\eta_{\chi_\oplus}}{1-2\eta}.
\end{aligned}$$

□

In order to compute an estimate of P_χ with (μ, θ) error, we determine below how accurately the various quantities on the right-hand side of the above equation must be estimated. 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 $\rho = 1$; otherwise, $\rho = \theta / P_{\chi_\oplus}$. Thus, ρ is the ratio between θ and $P_{\chi_\oplus}$ for queries with a "significant" dependence on the labels of examples, and $\rho \in [\theta, 1]$.

For notational convenience we say that a probability p is estimated with $\langle \alpha, \nu \rangle$ error if the estimate $\hat{p}$ satisfies $d_\nu(p, \hat{p}) \leq \alpha$. We first state a lemma which allows an $\langle \alpha, \nu \rangle$ error approximation to be decomposed over a sum of two quantities. The proof of this lemma appears in the appendix.

Lemma 4 *If $d_\nu(a, \hat{a}) \leq \alpha$ and $d_\nu(b, \hat{b}) \leq \alpha$, then $d_{2\nu}(a + b, \hat{a} + \hat{b}) \leq \alpha$.*

We next state three lemmas which allow an additive approximation to be decomposed into either the product, ratio, or sum of two quantities. The proofs of these lemmas appear in the appendix.

Lemma 5 *If $0 \leq a, b, c, \tau \leq 1$ and $a = bc$, then to obtain an estimate of a within additive error τ, it is sufficient to obtain estimates of b and c within additive error $\tau/3$.*

Lemma 6 *If $0 \leq a, b, c, \tau \leq 1$ and $a = b/c$, then to obtain an estimate of a within additive error τ, it is sufficient to obtain estimates of b and c within additive error $c\tau/3$.*

Lemma 7 *If $0 \leq a, b, c, \tau \leq 1$ and $a = b+c$, then to obtain an estimate of a within additive error τ, it is sufficient to obtain estimates of b and c within additive error $\tau/2$.*

We next state three lemmas which allow conversion between different types of estimation measures. The proofs of these lemmas also appear in the appendix.

Lemma 8 *If $|a - \hat{a}| \leq \tau = \alpha\nu$, then $d_\nu(a, \hat{a}) \leq \alpha$.*

Lemma 9 *For all $\alpha \in [0, 1]$, if $a \leq \nu$ and $d_\nu(a, \hat{a}) \leq \alpha/4$, then $|a - \hat{a}| \leq \alpha\nu$.*

Lemma 10 *If $\phi = \min\{1/2, \tau/(6a)\}$ and $d_a(a, \hat{a}) \leq \phi$, then $|a - \hat{a}| \leq \tau$.*

We now state the main sensitivity analysis lemma. Since our sample complexity depends on $1/(\alpha^2\nu)$, we will lower bound the quantity $\alpha^2\nu$ when necessary. Recall that we use the following notation to specify d_ν, relative/threshold and additive convergence, respectively: $\langle \cdot, \cdot \rangle$, $(\cdot, \cdot)$, and $[\cdot]$.

Lemma 11 *In order to determine P_χ with (μ, θ) error, it is sufficient to obtain an estimate of η with $[c_1\mu\theta(1-2\eta)]$ error 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 ${\alpha_i}^2 \nu_i \geq c_2\mu^2\theta\rho(1-2\eta_b)^2$, and c_1 and c_2 are constants.*

Proof: In each step of the decomposition, the theorem or lemma used to derive the implication is noted over the arrow. A probability followed by a colon and then a precision measure denotes an estimate of the probability with the specified precision. The final probability/precision pairs which are to be guessed or estimated by sampling are boxed.

$$
\begin{array}{rcl}
P_\chi : (\mu, \theta) & \stackrel{\text{T3}}{\Leftarrow} & P_\chi : \langle \mu/4, \theta/4 \rangle \\
& \stackrel{\text{L4}}{\Leftarrow} & P_{\chi_1} : \langle \mu/4, \theta/8 \rangle \ \text{ and } \ P_{\chi_2} : \langle \mu/4, \theta/8 \rangle \\
& \stackrel{\text{L3}}{\Leftarrow} & \dfrac{P^\eta_{\chi_1} - \eta P^\eta_{\chi_\oplus}}{1-2\eta} : \langle \mu/4, \theta/8 \rangle \ \text{ and } \ \boxed{P^\eta_{\chi_2} : \langle \mu/4, \theta/8 \rangle} \\
& \stackrel{\text{L8}}{\Leftarrow} & \dfrac{P^\eta_{\chi_1} - \eta P^\eta_{\chi_\oplus}}{1-2\eta} : [\mu\theta/32] \\
& \stackrel{\text{L6}}{\Leftarrow} & (P^\eta_{\chi_1} - \eta P^\eta_{\chi_\oplus}) : [\mu\theta(1-2\eta)/96] \ \text{ and} \\
& & (1-2\eta) : [\mu\theta(1-2\eta)/96] \\
& \stackrel{\text{L7}}{\Leftarrow} & \boxed{P^\eta_{\chi_1} : [\mu\theta(1-2\eta)/192]} \ \text{ and} \\
& & \eta P^\eta_{\chi_\oplus} : [\mu\theta(1-2\eta)/192] \ \text{ and} \\
& & \eta : [\mu\theta(1-2\eta)/192] \\
& \stackrel{\text{L5}}{\Leftarrow} & \boxed{P^\eta_{\chi_\oplus} : [\mu\theta(1-2\eta)/576]} \ \text{ and} \\
& & \boxed{\eta : [\mu\theta(1-2\eta)/576]}
\end{array}
$$

In order to convert the estimate precision for $P^\eta_{\chi_1}$ and $P^\eta_{\chi_\oplus}$ from additive to d_ν, we apply Lemmas 9 and 10 as well as the definition of ρ.

By Lemma 9, if $P^\eta_{\chi_\oplus} < \theta$ then $\langle \mu(1-2\eta)/2304, \theta \rangle$ is sufficient to ensure $[\mu\theta(1-2\eta)/576]$. Note that $\alpha^2\nu = \mu^2\theta(1-2\eta)^2/(2304)^2$ in this case. Conversely, if $P^\eta_{\chi_\oplus} \geq \theta$, then in order to apply Lemma 10 we must consider the value of

$$
\begin{aligned}
\phi &= \min\{1/2, (\mu\theta(1-2\eta)/576)/(6 \cdot P^\eta_{\chi_\oplus})\} \\
&= \min\{1/2, (\mu\theta(1-2\eta))/(3456 \cdot P^\eta_{\chi_\oplus})\}.
\end{aligned}
$$

Note that since $P^\eta_{\chi_\oplus} \geq \theta$, the latter quantity is less than 1/2. By Lemma 10, we then have that $\langle \mu\theta(1-2\eta)/(3456 \cdot P^\eta_{\chi_\oplus}), P^\eta_{\chi_\oplus}\rangle$ is sufficient to ensure $[\mu\theta(1-2\eta)/576]$. Noting again that $P^\eta_{\chi_\oplus} \geq \theta$, we have

$$\begin{aligned} \alpha^2\nu &= \mu^2\theta^2(1-2\eta)^2/(P^\eta_{\chi_\oplus} \cdot (3456)^2) \\ &= \mu^2\theta\rho(1-2\eta)^2/(3456)^2 \end{aligned}$$

in this case. Therefore, $\alpha^2\nu \geq \mu^2\theta\rho(1-2\eta)^2/(3456)^2$ in either case.

Considering $P^\eta_{\chi_1}$, we first note that $\eta P^\eta_{\chi_\oplus} \leq P^\eta_{\chi_1} \leq (1-\eta)P^\eta_{\chi_\oplus}$. By Lemma 9, if $P^\eta_{\chi_1} < \theta(1-\eta)$ then $\langle \mu(1-2\eta)/(768 \cdot (1-\eta)), \theta(1-\eta)\rangle$ is sufficient to ensure $[\mu\theta(1-2\eta)/192]$. Note that $\alpha^2\nu = \mu^2\theta(1-2\eta)^2/((768)^2(1-\eta))$ in this case. Conversely, if $P^\eta_{\chi_1} \geq \theta(1-\eta)$ then $P^\eta_{\chi_\oplus} \geq \theta$. In order to apply Lemma 10 we must consider the value of

$$\begin{aligned} \phi &= \min\{1/2, (\mu\theta(1-2\eta)/192)/(6 \cdot P^\eta_{\chi_1})\} \\ &= \min\{1/2, (\mu\theta(1-2\eta))/(1152 \cdot P^\eta_{\chi_1})\}. \end{aligned}$$

As before, the latter quantity is less than 1/2. By Lemma 10, we then have that $\langle \mu\theta(1-2\eta)/(1152 \cdot P^\eta_{\chi_1}), P^\eta_{\chi_1}\rangle$ is sufficient to ensure $[\mu\theta(1-2\eta)/192]$. Noting again that $P^\eta_{\chi_1} \leq (1-\eta)P^\eta_{\chi_\oplus}$ and $P^\eta_{\chi_\oplus} \geq \theta$, we have

$$\begin{aligned} \alpha^2\nu &= \mu^2\theta^2(1-2\eta)^2/(P^\eta_{\chi_1} \cdot (1152)^2) \\ &\geq \mu^2\theta^2(1-2\eta)^2/((1-\eta)P^\eta_{\chi_\oplus} \cdot (1152)^2) \\ &= \mu^2\theta\rho(1-2\eta)^2/((1-\eta) \cdot (1152)^2) \end{aligned}$$

in this case. Therefore, $\alpha^2\nu \geq \mu^2\theta\rho(1-2\eta)^2/((1-\eta)\cdot(1152)^2) \geq \mu^2\theta\rho(1-2\eta)^2/(1152)^2$ in either case.

Thus, it is sufficient to estimate η with $[\mu\theta(1-2\eta)/576]$ error and to estimate $P^\eta_{\chi_2}$, $P^\eta_{\chi_1}$ and $P^\eta_{\chi_\oplus}$ with $\langle \alpha, \nu\rangle$ error such that, in each case, $\alpha^2\nu \geq \mu^2\theta\rho(1-2\eta)^2/(3456)^2$. Note that we have not attempted to optimize the constants in this bound. □

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.[6] The following lemma states how many noise rate guesses are sufficient.

Lemma 12 *For all $\gamma, \eta_b < 1/2$, there exists a sequence of η-guesses $\{\eta_0, \eta_1, \ldots, \eta_i\}$ where $i = O(\frac{1}{\gamma}\log\frac{1}{1-2\eta_b})$ such that for all $\eta \in [0, \eta_b]$, there exists an η_j which satisfies $|\eta - \eta_j| \leq \gamma(1-2\eta)$.*

Proof: 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 $\eta - \eta_{j-1} \leq \gamma(1-2\eta)$. Solving for η, we find that η_{j-1} is a valid estimate for all $\eta \in [\eta_{j-1}, \frac{\eta_{j-1}+\gamma}{1+2\gamma}]$. Now consider an $\eta_j \geq \eta > \frac{\eta_{j-1}+\gamma}{1+2\gamma}$. The value η_j is a valid estimate for all $\eta \leq \eta_j$ which satisfy $\eta_j - \eta \leq \gamma(1-2\eta)$. Solving for η, we find that η_j is a valid estimate for all $\eta \in [\frac{\eta_j-\gamma}{1-2\gamma}, \eta_j]$. 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

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

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

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

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

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

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

Given that

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

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 have

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

as an upper bound on the number of guesses required. $\square$

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 in order to apply Theorem 2 directly, would require the use of the individual worst-case α_* and ν_*. Yet, the quantity $\alpha_*^2 \nu_*$ is smaller than our lower bound, for all queries, on $\alpha^2 \nu$ shown in Lemma 11. However, we may use this improved lower bound by modifying the result of Theorem 2 to ensure uniform convergence for all $\langle \alpha_i, \nu_i \rangle$ satisfying ${\alpha_i}^2 \nu_i \geq z$.

Lemma 13 *Let $\mathcal{G}$ be a set of $\{0, 1\}$-valued functions and for a given $g \in \mathcal{G}$, let $E(g)$ be the 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 $\alpha^2 \nu \geq z$*

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

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

Proof: 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

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

for $0 \le i < \lceil \lg 1/z \rceil$.

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

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

Thus, $\alpha_i \le \alpha$ and $\nu_i < \nu$, and so $\langle \alpha, \nu \rangle$ convergence is implied by $\langle \alpha_i, \nu_i \rangle$ convergence. Since $\nu < 1$, we need only ensure uniform convergence for $\langle \alpha_i, \nu_i \rangle$ for $0 \le i < \lceil \lg 1/z \rceil$. To ensure uniform convergence for all of these $\lceil \lg 1/z \rceil$ pairs with probability at least $1 - \delta$, we allocate $\delta / \lceil \lg 1/z \rceil$ probability of failure to the uniform convergence with respect to each pair. The final sample complexities are obtained from Theorem 2 by noting that $1/{\alpha_i}^2 \nu_i = 2/z$ for all i. □

Now, let ρ_* be a lower bound on ρ for every query $[\chi, \mu, \theta]$, $\rho_* \in [\theta_*, 1]$, and note that the VC-dimension and size of our new query space (composed of queries of the form χ_1, χ_2, and $\chi_\oplus$) are only increased by constant factors.[7] We then have the following:

Lemma 14 *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*

$$m = O\left(\tfrac{1}{\mu_*^2 \theta_* \rho_* (1-2\eta_b)^2} \log\left(\tfrac{|\mathcal{Q}|}{\delta} \log \tfrac{1}{\mu_* \theta_* \rho_* (1-2\eta_b)}\right)\right)$$

is sufficient, while if $\mathcal{Q}$ has finite VC-dimension q, then a sample of size

$$m = O\left(\tfrac{q}{\mu_*^2 \theta_* \rho_* (1-2\eta_b)^2} \log \tfrac{1}{\mu_* \theta_* \rho_* (1-2\eta_b)} + \tfrac{1}{\mu_*^2 \theta_* \rho_* (1-2\eta_b)^2} \log \tfrac{1}{\delta}\right)$$

is sufficient.

[7] It is clear that the size of the query space is only increased by a constant factor since we create and use exactly three queries corresponding to each query χ: $\chi_\oplus$, χ_1 and χ_2. To see that the VC-dimension is also only increased by a constant factor, we note that each new query created is a constant size Boolean function of the original query and/or "restrictions" of the original query.

If the probabilities are estimated accurately and some η-guess is sufficiently close to η, then some run of the SQ algorithm should have produced a "good" hypothesis. We are then 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 Theorem 8 below. Theorem 8 is based on Theorem 7, a generalization of a theorem due to Laird [16]. Implicit in Laird's Theorem 5.32, there are two important parameters which determine the number of labelled examples sufficient to perform hypothesis testing between two hypotheses using any type of (possibly noisy) example oracle $EX^*(f, D)$. These parameters are (1) t — the probability of drawing a labelled example on which the two hypotheses disagree; and (2) α — the conditional probability, given that the hypotheses disagree with each other, that the hypothesis with larger true error agrees with the label.

Theorem 7 *Let h_1 and h_2 be hypotheses with error rates $\varepsilon_1 < \varepsilon_2$, respectively, with respect to $EX(f, D)$. Let t be the probability of drawing a labelled example $\langle x, \ell \rangle$ from $EX^*(f, D)$ such that $h_1(x) \neq h_2(x)$. Let α be the conditional probability of drawing a labelled example $\langle x, \ell \rangle$ from $EX^*(f, D)$ such that $h_2(x) = \ell$ given that $h_1(x) \neq h_2(x)$. Then with probability at least $1 - \delta$, h_1 agrees with more labelled examples than h_2 does on a sample of size*

$$m = O\left(\frac{\log(1/\delta)}{t(1-2\alpha)^2}\right)$$

drawn randomly from $EX^*(f, D)$.

Proof: Laird [16] (Theorem 5.32) compares a perfect hypothesis with one of error ε, does so specifically for a classification noise oracle, and in doing so shows that for the value m stated in Theorem 7, the following inequality holds:

$$\sum_{k=0}^{m}\left[\binom{m}{k} t^k (1-t)^{m-k} \sum_{i=0}^{\lfloor k/2 \rfloor} \binom{k}{i} (1-\alpha)^i \alpha^{k-i}\right] \leq \delta.$$

We then observe that the left side of the above inequality is the exact probability that h_2 agrees with at least as many examples as h_1 does for the conditions and definitions of Theorem 7. $\square$

Theorem 8 *Let $\mathcal{H}$ be a set of hypotheses, at least one of which has error rate at most ε_1. Let ε_2 be any error rate strictly larger than ε_1. Then the hypothesis in $\mathcal{H}$ which has the fewest disagreements on a labelled sample of size*

$$O\left(\frac{\varepsilon_1+\varepsilon_2}{(\varepsilon_2-\varepsilon_1)^2(1-2\eta_b)^2}\log\frac{|\mathcal{H}|}{\delta}\right)$$

drawn from $EX_{\mathrm{CN}}^{\eta}(f,D)$ will, with probability at least $1-\delta$, have true error rate at most ε_2.

Proof: It is enough to show that the hypothesis with error rate at most ε_1 performs better than any of the at most $|\mathcal{H}|-1$ hypotheses with error rate more than ε_2. Without loss of generality, consider hypothesis h_1 with error rate ε_1 and h_2 with error rate ε_2. We first calculate t and α for $EX_{\mathrm{CN}}^{\eta}(f,D)$ and these hypotheses. Let d be the probability of drawing an example x according to D on which $h_1(x)=h_2(x)\neq f(x)$. Let t_1 be the probability of drawing a labelled example $\langle x,\ell\rangle$ from $EX_{\mathrm{CN}}^{\eta}(f,D)$ in which $h_1(x)\neq h_2(x)=\ell$. Similarly, let t_2 be the probability of drawing a labelled example $\langle x,\ell\rangle$ from $EX_{\mathrm{CN}}^{\eta}(f,D)$ in which $h_2(x)\neq h_1(x)=\ell$. Probabilities t_1, t_2, and d are with respect to the draw of x or $\langle x,\ell\rangle$ and any randomization in h_1 or h_2. We then have

$$\begin{aligned} t_1 &= (\varepsilon_1-d)(1-\eta)+(\varepsilon_2-d)\eta \\ t_2 &= (\varepsilon_2-d)(1-\eta)+(\varepsilon_1-d)\eta \end{aligned}$$

and

$$t=t_1+t_2=\varepsilon_1+\varepsilon_2-2d.$$

The value of α is simply t_1/t, and therefore $t^{-1}(1-2\alpha)^{-2}=t(t-2t_1)^{-2}$. The sample complexity in this theorem then follows from Theorem 7 by allocating $\delta/(|\mathcal{H}|-1)$ probability of failure to each comparison. $\square$

Note that in our case, $\varepsilon_1=\varepsilon/2$ and $\varepsilon_2=\varepsilon$, and thus $(\varepsilon_1+\varepsilon_2)/(\varepsilon_2-\varepsilon_1)^2=\Theta(1/\varepsilon)$, while $|\mathcal{H}|$ corresponds to the number of noise rate guesses which is $O(\frac{1}{\mu_*\theta_*}\log\frac{1}{1-2\eta_b})$. Thus, the sample size required for testing is

$$m_2=O\left(\frac{1}{\varepsilon(1-2\eta_b)^2}\log\Big(\frac{1}{\delta\mu_*\theta_*}\log\frac{1}{1-2\eta_b}\Big)\right). \tag{1}$$

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 [14]), we obtain the following:

Theorem 9 *The total sample complexity required to simulate an SQ algorithm in the presence of classification noise is*

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

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

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

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

When Theorem 9 is 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 $\tilde{O}(\frac{n^2}{\varepsilon^2(1-2\eta_b)^2}\log\frac{1}{\delta})$. Note that this is the same sample complexity that can be obtained via a simulation of the appropriate additive error SQ learning algorithm. Our relative error SQ learning algorithm for conjunctions does not give an improved simulation, and it is an open question as to whether an algorithm exists for conjunctions using a number of examples roughly linear in $1/\varepsilon$. However, as we next describe, some natural relative error SQ algorithms can be simulated with this linear dependence.

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 $\tilde{O}(\frac{n}{\varepsilon(1-2\eta_b)^2}\log\frac{1}{\delta})$ examples, while the simulation of the standard additive error SQ algorithm uses $\tilde{O}(\frac{n^2}{\varepsilon^2(1-2\eta_b)^2}\log\frac{1}{\delta})$ examples. Note that our sample complexity roughly matches the lower bound of $\Omega(\frac{n}{\varepsilon(1-2\eta)^2}\log\frac{1}{\delta})$ [3]. This is the first non-trivial (*i.e.*, super-polynomial sized) 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 given in Figure 2.

For each i, the first loop queries the estimate of p_i, the probability of drawing instances with i number of 1's. Only those i which occur rarely enough ($< \frac{\varepsilon}{n+1}$) are ignored in the

1. **Learn-Symmetric-Functions**(n, ε):
2. foreach $i \in \{0, \ldots, n\}$ do
3. $\hat{p}_i := STAT(f, D)[H(x) = i, 1/2, \frac{\varepsilon}{n+1}]$
4. foreach $i \in \{0, \ldots, n\}$ do
5. if $\hat{p}_i \geq \frac{\varepsilon}{2(n+1)}$ then
6. $\hat{q}_i := STAT(f, D)[H(x) = i \wedge \ell = 1, 1/2, \hat{p}_i \cdot \frac{2}{3}]$
7. else
8. $\hat{q}_i := 0$
9. return h which classifies x as 1 if and only if $\hat{q}_{H(x)} > 0$

Figure 2: Relative error SQ Algorithm for Learning Symmetric Functions

second round of queries and given default predictions of 0. The only error of the hypothesis will be due to these rare examples. The remaining i's are each tested to see if they are labelled 1 or 0. The value q_i will equal 0 if the label of these instances is 0 or p_i if the label of these instances is 1. Note that in Line 6, $0 < \hat{p}_i \cdot \frac{2}{3} \leq p_i$ and q_i is either 0 or p_i. Thus $\hat{q}_i = 0$ or $\perp$ if the label is 0 while $\hat{q}_i > 0$ and not $\perp$ if the label is 1. The only queries which depend on the label, and therefore determine ρ, are those in the second loop 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 accuracy boosting techniques [10, 11, 18] and specifically, these techniques as applied in the statistical query model [2].

Theorem 10 *If a concept class $\mathcal{F}$ is SQ learnable by an algorithm A using hypothesis class $\mathcal{H}$, then $\mathcal{F}$ is SQ learnable with $O(N_0 \log^2(1/\varepsilon))$ queries each with threshold $\Omega(\mu_0 \theta_0 \varepsilon / \log(1/\varepsilon))$ and relative error $\Omega(\mu_0)$. The complexity of the requisite query space, $\mathcal{Q}'$, is*

$$\log|\mathcal{Q}'| = O(\log|\mathcal{Q}_0| + \log\tfrac{1}{\varepsilon}\log|\mathcal{H}_0|)$$

when $\mathcal{Q}_0$ and $\mathcal{H}_0$ are finite, or

$$VC(\mathcal{Q}') = O(VC(\mathcal{Q}_0) + VC(\mathcal{H}_0) \cdot \log\tfrac{1}{\varepsilon}\log\log\tfrac{1}{\varepsilon})$$

when $\mathcal{Q}_0$ and $\mathcal{H}_0$ have finite VC-dimension. Here N_0, μ_0, θ_0, $\mathcal{Q}_0$ and $\mathcal{H}_0$ are the number of queries, worst case relative error, worst case threshold, query space and hypothesis class, respectively, of algorithm A run with a constant *(e.g. 1/4) accuracy parameter. Note that N_0, μ_0, θ_0, $\mathcal{Q}_0$ and $\mathcal{H}_0$ are independent of ε.*

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

Lemma 15 *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 follows that $\hat{b}/\hat{c}$ is within a $1 \pm \mu$ factor of a. □

Lemma 16 *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 $s_B < \mu\theta/3$. 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. □

Aslam and Decatur [2] show that given an SQ learning algorithm A, one can construct a very efficient SQ algorithm by combining the output of $O(\log(1/\varepsilon))$ 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:[8]

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

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 $\Phi = \Omega(\varepsilon / \log(1/\varepsilon))$, then the query need not be estimated. Using Lemmas 15 and 16, it is easy to show that a (μ, θ) 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 $O(N_0 \log^2(1/\varepsilon))$. $\square$

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 $\tilde{O}(\frac{1}{\varepsilon^2(1-2\eta_b)^2})$.*

[8]It is further shown that the complexity of this new query space is as given in the statement of Theorem 10.

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 [15] and the $\Omega(1/\varepsilon)$ lower bound on the sample complexity for *noise-free* PAC learning [9]. 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.

7 Conclusions

We have defined the relative error statistical query model and shown that "efficient" relative error statistical query algorithms can be simulated in the PAC model very efficiently. By the boosting results of the previous section, we have further shown that such efficient relative error statistical query algorithms are guaranteed to exist (so long as an SQ algorithm exists).

A primary advantage of the relative error statistical query model is that it allows one to create a *single* (*Rel-SQ*) algorithm and be guaranteed highly efficient learning algorithms in the absence of noise, in the presence of malicious errors, and in the presence of classification noise. Our simulations in the absence of noise and in the presence of malicious errors are roughly optimal. In the presence of classification noise, our simulation is no worse than similar simulations for the additive error statistical query model, and under certain conditions ($\rho = \Omega(1)$), our simulation can be roughly optimal. We have shown a non-trivial learning problem for which a simulation in the presence of classification noise yields a roughly optimal sample complexity. An interesting problem left open by this work would be to characterize those problems for which it is possible to learn in the presence of classification noise with a $o(1/\varepsilon^2)$ sample complexity. A characterization of those problems for which $\rho = \Omega(1)$ would be one such solution.

A Appendix

In this appendix, we give proofs for a number of technical lemmas used throughout the paper.

A.1 Proof of Lemmas 1 and 2

We begin by noting the following properties about the d_ν metric:

Claim 1 $(\forall r, s \geq 0)(\forall \nu > 0)(\forall \alpha, 0 \leq \alpha < 1)$ *If* $d_\nu(r,s) \leq \alpha$, *then*

$$\begin{aligned} r &\leq \frac{1+\alpha}{1-\alpha}s + \frac{\alpha}{1-\alpha}\nu &(2)\\ r &\geq \frac{1-\alpha}{1+\alpha}s - \frac{\alpha}{1+\alpha}\nu &(3)\\ s &\leq \frac{1+\alpha}{1-\alpha}r + \frac{\alpha}{1-\alpha}\nu &(4)\\ s &\geq \frac{1-\alpha}{1+\alpha}r - \frac{\alpha}{1+\alpha}\nu &(5)\end{aligned}$$

Proof: If $d_\nu(r,s) \leq \alpha$, then $|r-s| \leq \alpha(\nu + r + s)$. This implies the following:

$$\begin{aligned} & r \leq s + \alpha(\nu + r + s)\\ \Leftrightarrow\quad & r - \alpha r \leq s + \alpha s + \alpha\nu\\ \Leftrightarrow\quad & r \leq \frac{1+\alpha}{1-\alpha}s + \frac{\alpha}{1-\alpha}\nu \end{aligned}$$

$$\begin{aligned} & r \geq s - \alpha(\nu + r + s)\\ \Leftrightarrow\quad & r + \alpha r \geq s - \alpha s - \alpha\nu\\ \Leftrightarrow\quad & r \geq \frac{1-\alpha}{1+\alpha}s - \frac{\alpha}{1+\alpha}\nu \end{aligned}$$

Equations 4 and 5 follow immediately since the d_ν metric is symmetric in its arguments. □

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

1. *If* $r < \theta/2$, *then* $s < \theta$.

2. If $r \geq \theta/2$, then $s \geq \theta/4$.

Proof: Assume that $r < \theta/2$. From Equation 4, we have

$$\begin{aligned} s &\leq \frac{1+\mu/4}{1-\mu/4} r + \frac{\mu/4}{1-\mu/4}\theta/4 \\ &< \frac{1+\mu/4}{1-\mu/4}\theta/2 + \frac{\mu/4}{1-\mu/4}\theta/4 \\ &\leq \frac{1+1/4}{1-1/4}\theta/2 + \frac{1/4}{1-1/4}\theta/4 \\ &= \theta \cdot 11/12 \\ &< \theta. \end{aligned}$$

Now assume that $r \geq \theta/2$. From Equation 5, we have

$$\begin{aligned} s &\geq \frac{1-\mu/4}{1+\mu/4} r - \frac{\mu/4}{1+\mu/4}\theta/4 \\ &\geq \frac{1-\mu/4}{1+\mu/4}\theta/2 - \frac{\mu/4}{1+\mu/4}\theta/4 \\ &\geq \frac{1-1/4}{1+1/4}\theta/2 - \frac{1/4}{1+1/4}\theta/4 \\ &= \theta/4. \end{aligned}$$

$\square$

Lemma 2 $(\forall r, s \geq 0)(\forall \nu > 0)(\forall \mu, 0 \leq \mu \leq 1)$ *If $s \geq \nu$, then $d_\nu(r,s) \leq \mu/4$ implies that $(1-\mu)s \leq r \leq (1+\mu)s$.*

Proof: From Equation 2, we have

$$\begin{aligned} r &\leq \frac{1+\mu/4}{1-\mu/4} s + \frac{\mu/4}{1-\mu/4}\nu \\ &\leq \frac{1+\mu/4}{1-\mu/4} s + \frac{\mu/4}{1-\mu/4} s \\ &= \frac{1+\mu/2}{1-\mu/4} s \\ &= \left(1 + \frac{3\mu/4}{1-\mu/4}\right) s \end{aligned}$$

$$\begin{aligned} &\leq \left(1+\frac{3\mu/4}{1-1/4}\right)s \\ &= (1+\mu)s. \end{aligned}$$

From Equation 3, we have

$$\begin{aligned} r &\geq \frac{1-\mu/4}{1+\mu/4}s - \frac{\mu/4}{1+\mu/4}\nu \\ &\geq \frac{1-\mu/4}{1+\mu/4}s - \frac{\mu/4}{1+\mu/4}s \\ &= \frac{1-\mu/2}{1+\mu/4}s \\ &= \left(1-\frac{3\mu/4}{1+\mu/4}\right)s \\ &\geq (1-3\mu/4)s \\ &\geq (1-\mu)s. \end{aligned}$$

□

A.2 Proof of Lemmas 4 through 10

Lemma 4 *If $d_\nu(a,\hat{a}) \leq \alpha$ and $d_\nu(b,\hat{b}) \leq \alpha$, then $d_{2\nu}(a+b,\hat{a}+\hat{b}) \leq \alpha$.*

Proof: From the conditions $d_\nu(a,\hat{a}) \leq \alpha$ and $d_\nu(b,\hat{b}) \leq \alpha$, we have

$$\begin{aligned} |a-\hat{a}| &\leq \alpha(\nu+a+\hat{a}) \\ |b-\hat{b}| &\leq \alpha(\nu+b+\hat{b}). \end{aligned}$$

Using the triangle inequality and the above inequalities, we have

$$\begin{aligned} |(a+b)-(\hat{a}+\hat{b})| &= |(a-\hat{a})+(b-\hat{b})| \\ &\leq |a-\hat{a}|+|b-\hat{b}| \\ &\leq \alpha(2\nu+(a+b)+(\hat{a}+\hat{b})) \end{aligned}$$

which proves the lemma. □

Lemma 5 *If $0 \leq a, b, c, \tau \leq 1$ and $a = bc$, then to obtain an estimate of a within additive error τ, it is sufficient to obtain estimates of b and c within additive error $\tau/3$.*

Lemma 6 *If $0 \leq a, b, c, \tau \leq 1$ and $a = b/c$, then to obtain an estimate of a within additive error τ, it is sufficient to obtain estimates of b and c within additive error $c\tau/3$.*

Lemma 7 *If $0 \leq a, b, c, \tau \leq 1$ and $a = b + c$, then to obtain an estimate of a within additive error τ, it is sufficient to obtain estimates of b and c within additive error $\tau/2$.*

Proof: For Lemma 6, we must show that $(b + c\tau/3)/(c - c\tau/3) \leq a + \tau$ and $(b - c\tau/3)/(c + c\tau/3) \geq a - \tau$. These two inequalities may be verified as follows:

$$\begin{aligned}
\frac{b + c\tau/3}{c - c\tau/3} &= \frac{a + \tau/3}{1 - \tau/3} \\
&= (a + \tau/3)\left(1 + \frac{\tau/3}{1 - \tau/3}\right) \\
&\leq (a + \tau/3)\left(1 + \frac{\tau/3}{1 - 1/3}\right) \\
&= (a + \tau/3)(1 + \tau/2) \\
&= a + a\tau/2 + \tau/3 + \tau^2/6 \\
&\leq a + \tau,
\end{aligned}$$

$$\begin{aligned}
\frac{b - c\tau/3}{c + c\tau/3} &= \frac{a - \tau/3}{1 + \tau/3} \\
&= (a - \tau/3)\left(1 - \frac{\tau/3}{1 + \tau/3}\right) \\
&\geq (a - \tau/3)(1 - \tau/3) \\
&= a - a\tau/3 - \tau/3 + \tau^2/9 \\
&\geq a - \tau.
\end{aligned}$$

Lemmas 5 and 7 are proven similarly. □

Lemma 8 *If $|a - \hat{a}| \leq \tau = \alpha\nu$, then $d_\nu(a, \hat{a}) \leq \alpha$.*

Proof: From the condition $|a - \hat{a}| \leq \alpha\nu$, we have $|a - \hat{a}| \leq \alpha(\nu + a + \hat{a})$ which implies the lemma. □

Lemma 9 *For all $\alpha \in [0,1]$, if $a \leq \nu$ and $d_\nu(a, \hat{a}) \leq \alpha/4$, then $|a - \hat{a}| \leq \alpha\nu$.*

Proof: From the condition $d_\nu(a, \hat{a}) \leq \alpha/4$ we have $|a - \hat{a}| \leq \alpha/4 \cdot (\nu + a + \hat{a})$. Given that $a \leq \nu$, we have $|a - \hat{a}| \leq \alpha/4 \cdot (\nu + \nu + 2\nu) = \alpha\nu$, provided that $\hat{a} \leq 2\nu$. In order to show that $\hat{a} \leq 2\nu$, we note that $|a - \hat{a}| \leq \alpha/4 \cdot (\nu + a + \hat{a})$ yields

$$\hat{a} \leq a + \alpha/4 \cdot (\nu + a + \hat{a}).$$

Solving for $\hat{a}$, we obtain

$$\hat{a} \leq \frac{a + \alpha/4 \cdot (\nu + a)}{1 - \alpha/4}.$$

The right-hand side of the above inequality is maximized when $\alpha = 1$ and $a = \nu$. We therefore have

$$\hat{a} \leq \frac{\nu + 1/4 \cdot (2\nu)}{1 - 1/4} = \frac{3/2 \cdot \nu}{3/4} = 2\nu.$$

□

Lemma 10 *If $\phi = \min\{1/2, \tau/(6a)\}$ and $d_a(a, \hat{a}) \leq \phi$, then $|a - \hat{a}| \leq \tau$.*

Proof: Since $\phi \leq 1/2$, the condition $d_a(a, \hat{a}) \leq \phi$ implies that $|a - \hat{a}| \leq 1/2 \cdot (a + a + \hat{a})$ which yields

$$\hat{a} \leq a + 1/2 \cdot (a + a + \hat{a}).$$

Solving for $\hat{a}$, we obtain $\hat{a} \leq 4a$. Since we also have $\phi \leq \tau/(6a)$, the condition $d_a(a, \hat{a}) \leq \phi$ also implies that $|a - \hat{a}| \leq \tau/(6a) \cdot (a + a + \hat{a})$. But since $\hat{a} \leq 4a$, this implies the lemma. □

References

[1] Dana Angluin and Leslie G. Valiant. Fast probabilistic algorithms for Hamiltonian circuits and matchings. *Journal of Computer and System Sciences*, 18(2):155–193, April 1979.

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

[3] Javed Aslam and Scott Decatur. On the sample complexity of noise-tolerant learning. *Information Processing Letters*, 57:189–195, 1996.

[4] 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.

[5] 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.

[6] 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.

[7] Scott Decatur. Learning in hybrid noise environments using statistical queries. In D. Fisher and H. J. Lenz, editors, *Learning from Data: Artificial Intelligence and Statistics V*. Springer Verlag, 1996. Preliminary version appeared in *Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics*, pages 175–185, January 1995.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

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

[14] 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.

[15] Michael Kearns and Ming Li. Learning in the presence of malicious errors. *SIAM Journal on Computing*, 22(4):807–837, 1993.

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

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

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

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