5

Learning in the Presence of Noise

In order to obtain a clean and simple starting point for a theoretical study of learning, many unrealistic assumptions were made in defining the PAC model. One of the most unjustified of these assumptions is that learning algorithms have access to a noise-free oracle for examples of the target concept. In reality, we need learning algorithms with at least some tolerance for the occasional mislabeled example.

In this chapter we investigate a generalization of the PAC model in which the examples received by the learning algorithm are corrupted with *classification noise*. This is *random* and essentially "white" noise affecting only the label of each example. (Learning in the presence of this type of noise implies learning in some slightly more realistic models, and more adversarial error models have also been examined in the literature; see the Bibliographic Notes at the end of the chapter.) In this setting we will see that much of the theory developed so far is preserved even in the presence of such noise. For instance, all of the classes we have shown to be efficiently PAC learnable remain so even with a classification noise rate approaching the information-theoretic barrier of 1/2.

To show this, we will actually introduce another new model, called *learning from statistical queries*. This model is a specialization of the PAC model in which we restrict the learning algorithm to form its hypothesis solely on the basis of estimates of probabilities. We will then give a theorem stating that any class efficiently learnable from statistical queries can be efficiently learned in the presence of classification noise. While we show that conjunctions of literals can be efficiently learned from statistical queries (and thus in the presence of classification noise), we leave it to the reader (in the exercises) to verify that all of the other efficient PAC learning algorithms we have given have efficient statistical query analogues.

5.1 The Classification Noise Model

In the classification noise model, a PAC learning algorithm will now be given access to a modified and noisy oracle for examples, denoted $EX^{\eta}_{CN}(c, \mathcal{D})$. Here $c \in \mathcal{C}$ and $\mathcal{D}$ are the target concept and distribution, and $0 \leq \eta < 1/2$ is a new parameter called the **classification noise rate**. This new oracle behaves in the following manner: as with $EX(c, \mathcal{D})$, a random input $x \in X$ is drawn according to the distribution $\mathcal{D}$. Then with probability $1 - \eta$, the labeled example $\langle x, c(x) \rangle$ is returned to the learning algorithm, but with probability η, the (incorrectly) labeled example $\langle x, \neg c(x) \rangle$ is returned, where $\neg c(x)$ is the complement of the binary value $c(x)$. Despite the classification noise in the examples received, the goal of the learner remains that of finding a good approximation h to the target concept c with respect to the distribution $\mathcal{D}$. Thus, on inputs ϵ and δ and given access to $EX^{\eta}_{CN}(c, \mathcal{D})$, the learning algorithm is said to succeed if with probability at least $1 - \delta$ it outputs a hypothesis h satisfying $error(h) \equiv \mathbf{Pr}_{x \in \mathcal{D}}[c(x) \neq h(x)] \leq \epsilon$.

Although the criterion for success remains unchanged in the noisy model, we do need to modify the definition of efficient learning. Note that if we allow the noise rate η to equal $1/2$, then PAC learning becomes impossible in any amount of computation time, because every label seen by the algorithm is the outcome of an unbiased coin flip, and conveys no information about the target concept. Similarly, as the noise rate *approaches* $1/2$, the labels provided by the noisy oracle are providing less and less information about the target concept. Thus we see there is a need to allow the learning algorithm more oracle calls and more computation time as the noise rate approaches 1/2.

We also need to specify what knowledge the learning algorithm has, if any, about the value of the noise rate η. For simplicity we will assume that the learning algorithm is provided with an upper bound $1/2 > \eta_0 \geq \eta$ on the noise rate. (This assumption can in fact be removed; see Exercise 5.4.) The new notion of efficiency can then be formalized by allowing the learning algorithm's running time to depend on the quantify $1/(1-2\eta_0)$, which increases as the upper bound η_0 approaches 1/2. (Making rigorous the informal arguments used here to argue that this dependence is needed is the topic of Exercise 5.5.)

Definition 13 *(PAC Learning in the Presence of Classification Noise) Let C be a concept class and let $\mathcal{H}$ be a representation class over X. We say that C is* **PAC learnable using $\mathcal{H}$ in the presence of classification noise** *if there exists an algorithm L with the following property: for any concept $c \in C$, any distribution $\mathcal{D}$ on X, any $0 \leq \eta < 1/2$, and any $0 < \epsilon < 1$, $0 < \delta < 1$, and η_0 (where $\eta \leq \eta_0 < 1/2$), if L is given access to $EX^{\eta}_{CN}(c, \mathcal{D})$ and inputs ϵ, δ and η_0, then with probability at least $1-\delta$, L outputs a hypothesis concept $h \in \mathcal{H}$ satisfying $error(h) \leq \epsilon$. This probability is taken over the randomization in the calls to $EX^{\eta}_{CN}(c, \mathcal{D})$, and any internal randomization of L.*

If L runs in time polynomial in n, $1/\epsilon$, $1/\delta$ and $1/(1-2\eta_0)$ we say that C is **efficiently PAC learnable using $\mathcal{H}$ in the presence of classification noise**.

Before proceeding further, let us convince ourselves with some concrete examples that learning in this apparently more difficult model really does require some new ideas. Recall that one of the first PAC learning algorithms we gave in Chapter 1 was for the class of boolean conjunctions of literals. The algorithm initializes the hypothesis to be the conjunction of all $2n$ literals over $x_1, \ldots, x_n$, and deletes any literal that appears negated in a positive example of the target conjunction (the negative examples received are ignored). The problem with using this same algorithm in the classification noise setting is obvious and fatal. With the noisy oracle, the algorithm may actually be given a negative example of the target conjunction as a positively labeled example, resulting in unwarranted and costly deletions of literals. For instance, suppose that the target conjunction c contains at least one unnegated literal, say x_1. Then the vector of all 0's is a negative example of the target. However, if this single vector has significant weight under $\mathcal{D}$, say weight γ, then there is probability $\gamma\eta$ that the learning algorithm will receive the vector of all 0's as a *negatively* labeled example from $EX^{\eta}_{CN}(c, \mathcal{D})$, causing the deletion of all unnegated literals from the hypothesis.

Similarly, consider our algorithm from Chapter 1 for PAC learning axis-aligned rectangles in the real plane. This algorithm takes a sufficiently large sample of random examples of the target rectangle, and chooses as its hypothesis the most specific (smallest area) rectangle that includes all of the positive examples but none of the negative examples. But such a rectangle may not even exist for a sample from the noisy oracle $EX^{\eta}_{CN}(c, \mathcal{D})$.

5.2 An Algorithm for Learning Conjunctions from Statistics

Intuitively, the problem with our conjunctions learning algorithm in the classification noise setting is that the algorithm will make drastic and irreversible changes to the hypothesis on the basis of a single example. In the noisy setting, where every individual example received from $EX^{\eta}_{CN}(c, \mathcal{D})$ is suspect since its label could be the result of an error, it seems natural to seek algorithms that instead form their hypotheses based on the properties of large samples, or that learn from *statistics*.

As an example, consider the following rather different algorithm for

PAC learning boolean conjunctions (still in the original noise-free setting). For each literal z over the boolean input variables $x_1, \ldots, x_n$, denote by $p_0(z)$ the probability that z is set to 0 in a random instance drawn according to the distribution $\mathcal{D}$. If $p_0(z)$ is extremely small, then we can intuitively "ignore" z, since it is almost always set to 1 (satisfied) with respect to $\mathcal{D}$. We define $p_{01}(z)$ to be the probability that a random instance from $\mathcal{D}$ fails to satisfy z, but does satisfy (that is, is a positive example of) the target conjunction c. Note that for any literal appearing in c, $p_{01}(z) = 0$. If $p_{01}(z)$ is large, then we would like to avoid including z in our hypothesis conjunction, since there is a reasonable chance of drawing a positive example of c in which z is 0. We say that z is ***significant*** if $p_0(z) \geq \epsilon/8n$ and ***harmful*** if $p_{01}(z) \geq \epsilon/8n$. Note that since we always have $p_{01}(z) \leq p_0(z)$, any harmful literal is also significant.

We now argue that if h is the conjunction of all the significant literals that are not harmful, then h has error less than ϵ with respect to c and $\mathcal{D}$. First we consider $\mathbf{Pr}_{a \in \mathcal{D}}[c(a) = 0 \wedge h(a) = 1]$. Note that the event $c(a) = 0 \wedge h(a) = 1$ occurs only when there is some literal z appearing in c that does not appear in h, and z is set to 0 in a. Since h contains all the significant literals that are not harmful, and c contains no harmful literals, any such literal z must not be significant. Then we have that $\mathbf{Pr}_{a \in \mathcal{D}}[c(a) = 0 \wedge h(a) = 1]$ is at most the probability that some insignificant literal is 0 in a, which by the union bound is at most $2n(\epsilon/8n) = \epsilon/4$. To bound $\mathbf{Pr}_{a \in \mathcal{D}}[c(a) = 1 \wedge h(a) = 0]$, we simply observe that the event $c(a) = 1 \wedge h(a) = 0$ occurs only when there is some literal z not appearing in c but appearing in h, and z is set to 0 in a. Since h contains no harmful literals, we have that $\mathbf{Pr}_{a \in \mathcal{D}}[c(a) = 1 \wedge h(a) = 0]$ is bounded by the probability that some harmful literal is set to 0 in a but $c(a) = 1$, which by the union bound is at most $2n(\epsilon/8n) = \epsilon/4$. Thus $\mathit{error}(h) \leq \epsilon/4 + \epsilon/4 = \epsilon/2$.

The above analysis immediately suggests an efficient algorithm for PAC learning conjunctions (in our original noise-free model). The probabilities $p_0(z)$ for each literal z can be estimated using $EX(c, \mathcal{D})$ by drawing a sufficiently large set of examples and computing the fraction of inputs on which z is set to 0. Similarly, the probabilities $p_{01}(z)$ can be estimated by drawing a sufficiently large set of examples and computing the fraction on which z is set to 0 and the label is 1. Note that while we cannot exactly determine which literals are harmful and which are significant (since we can only estimate the $p_0(z)$ and $p_{01}(z)$), we have left enough room to maneuver in the preceding analysis that accurate estimates are sufficient. For instance, it can be verified using Chernoff bounds (see the Appendix in Chapter 9) that if our algorithm takes a sufficiently large (but still only polynomial in n, $1/\epsilon$ and $1/\delta$) sample for its estimates, and chooses as its hypothesis h the conjunction of all literals z such that the resulting estimate $\hat{p}_0(z)$ for $p_0(z)$ satifies $\hat{p}_0(z) \geq \epsilon/8n$, but the estimate $\hat{p}_{01}(z)$ for $p_{01}(z)$ satifies $\hat{p}_{01}(z) \leq \epsilon/2n$, and the sample size is sufficient to make our estimates $\hat{p}_0(z)$ and $\hat{p}_{01}(z)$ within an additive error of $\epsilon/8n$ of their true values, then with probability $1-\delta$, h will satisfy $error(h) \leq \epsilon$.

A nice property of this new algorithm is that it forms its hypothesis solely on the basis of estimates of a small number of probabilities (namely, the $p_0(z)$ and $p_{01}(z)$). Of course, at this point all we have shown is another efficient algorithm for PAC learning conjunctions. The feeling that this algorithm is somehow more robust to classification noise than our original algorithm is nothing more than an intuition. We now generalize and formalize the notion of PAC learning solely on the basis of probability estimates. This is most easily done by introducing yet another model of learning. We then proceed to verify our intuition by showing that efficient learning in the new model automatically implies efficient PAC learning in the presence of classification noise.

5.3 The Statistical Query Learning Model

Our new learning model can be viewed as placing a ***restriction*** on the way in which a PAC learning algorithm can use the random examples it receives from the oracle $EX(c,\mathcal{D})$. Let C be a concept class over X. In the statistical query model, if $c \in C$ is the target concept and $\mathcal{D}$ is the target distribution, then we replace the usual PAC oracle $EX(c,\mathcal{D})$ with an oracle $STAT(c,\mathcal{D})$ that accepts **statistical queries** of the form (χ,τ). Here χ is a mapping $\chi : X \times \{0,1\} \rightarrow \{0,1\}$ and $0 < \tau \leq 1$. We think of χ as a function that maps a labeled example $\langle x, c(x)\rangle$ of the target concept to 0 or 1, indicating either the presence or absence of some property in $\langle x, c(x)\rangle$. For instance, in our new algorithm for PAC learning conjunctions we took a large random sample, and for each $\langle a, c(a)\rangle$ in the sample we computed the predicate $\chi_z(a, c(a))$ that is 1 if and only if the literal z is 0 in a but $c(a) = 0$. This predicate corresponds to the probability $p_{01}(z)$, that is, $p_{01}(z) = \mathbf{Pr}_{a\in\mathcal{D}}[\chi_z(a, c(a)) = 1]$.

In general, for a fixed target concept $c \in C$ and distribution $\mathcal{D}$, let us define

$$P_\chi = \mathbf{Pr}_{x\in\mathcal{D}}[\chi(x, c(x)) = 1].$$

We interpret a statistical query (χ,τ) as a request for the value P_χ. However, on input (χ,τ) the oracle $STAT(c,\mathcal{D})$ will not return exactly P_χ, but only an approximation. More precisely, the output of $STAT(c,\mathcal{D})$ on input query (χ,τ) is allowed to be *any* value $\hat{P}_\chi$ satisfying $P_\chi - \tau \leq \hat{P}_\chi \leq P_\chi + \tau$. Thus, the output of $STAT(c,\mathcal{D})$ is simply any estimate of P_χ that is accurate within additive error τ. We assume that each query to $STAT(c,\mathcal{D})$ takes unit time.

We call τ the **tolerance** of the statistical query, and the choice of both χ and τ are left to the learning algorithm (modulo some important restrictions discussed momentarily). For instance, in our conjunctions example, recall that by the analysis of the last section it suffices to estimate the probabilities $p_{01}(z) = P_{\chi_z}$ to within tolerance $\tau = \epsilon/8n$.

At this point, it should be clear that given access to the oracle $EX(c,\mathcal{D})$, it is a simple matter to simulate the behavior of the oracle $STAT(c,\mathcal{D})$ on a query (χ,τ) with probability at least $1-\delta$. We just draw from $EX(c,\mathcal{D})$ a sufficient number of random labeled examples $\langle x, c(x)\rangle$, and use the fraction of the examples for which $\chi(x, c(x)) = 1$ as our estimate $\hat{P}_\chi$ of

P_χ. Now by Chernoff bounds, the number of calls to $EX(c,\mathcal{D})$ required will be polynomial in $1/\tau$ and $\log(1/\delta)$, and the time required will be polynomial in the time required to evaluate χ, and in $1/\tau$ and $\log(1/\delta)$. To ensure that efficient algorithms for learning using $STAT(c,\mathcal{D})$ can be efficiently simulated using $EX(c,\mathcal{D})$, we must place some natural restrictions on τ (namely, that it is an inverse polynomial in the learning problem parameters) and on χ (namely, that it can be evaluated in polynomial time). Thus we require that algorithms only ask $STAT(c,\mathcal{D})$ for estimates of sufficiently "simple" probabilities, with sufficiently coarse tolerance. This is done in the following definition, which formalizes the model of learning from statistical queries. The intuition that algorithms with access to $STAT(c,\mathcal{D})$ can be efficiently simulated given access to $EX(c,\mathcal{D})$ is then formalized in greater detail as Theorem 5.1 below.

Definition 14 *(The Statistical Query Model) Let C be a concept class and let $\mathcal{H}$ be a representation class over X. We say that C* **is efficiently learnable from statistical queries using** *$\mathcal{H}$ if there exists a learning algorithm L and polynomials $p(\cdot,\cdot,\cdot)$, $q(\cdot,\cdot,\cdot)$ and $r(\cdot,\cdot,\cdot)$ with the following property: for any $c \in C$, for any distribution $\mathcal{D}$ over X, and for any $0 < \epsilon < 1/2$, if L is given access to $STAT(c,\mathcal{D})$ and input ϵ, then*

- *For every query (χ,τ) made by L, the predicate χ can be evaluated in time $q(1/\epsilon, n, size(c))$, and $1/\tau$ is bounded by $r(1/\epsilon, n, size(c))$.*
- *L will halt in time bounded by $p(1/\epsilon, n, size(c))$.*
- *L will output a hypothesis $h \in \mathcal{H}$ that satisfies $error(h) \leq \epsilon$.*

Notice that the confidence parameter δ has disappeared from this definition. Recall that this parameter guarded against the small but nonzero probability that an extremely unrepresentative sample is drawn from $EX(c,\mathcal{D})$ in the PAC learning model. Since $EX(c,\mathcal{D})$ has now been replaced by the oracle $STAT(c,\mathcal{D})$, whose behavior is completely determined modulo the query tolerance τ, there is no need for δ. Of course, we could allow a certain failure probability for the case of randomized learning algorithms, but choose not to for the sake of simplicity, since we will only examine deterministic algorithms.

The following theorem verifies that we have defined the statistical query model in a way that ensures efficient simulation in the PAC model. Its proof is the subject of Exercise 5.6. Thus, we have found a model that specializes the PAC model in a way that allows learning algorithms to estimate probabilities, but to do nothing else.

Theorem 5.1 *Let C be a concept class and $\mathcal{H}$ be a representation class over X. Then if C is efficiently learnable from statistical queries using $\mathcal{H}$, C is efficiently PAC learnable using $\mathcal{H}$.*

In the following section we will show a much more interesting and useful result: any class that is efficiently learnable from statistical queries is in fact efficiently PAC learnable even in the presence of classification noise. Before this, however, we pause to note that by the analysis of Section 5.2, we already have our first positive result in the statistical query model:

Theorem 5.2 *The representation class of conjunctions of literals is efficiently learnable from statistical queries.*

5.4 Simulating Statistical Queries in the Presence of Noise

Let us fix the target concept $c \in C$ and the distribution $\mathcal{D}$, and suppose we are given a statistical query (χ, τ). We now give an efficient method for obtaining an accurate estimate of

$$P_\chi = \mathbf{Pr}_{x\in\mathcal{D}}[\chi(x, c(x)) = 1]$$

given access only to the noisy examples oracle $EX_{CN}^{\eta}(c, \mathcal{D})$. We will then show how this method can be used to efficiently simulate any statistical query learning algorithm in the presence of classification noise.

5.4.1 A Nice Decomposition of P_χ

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

Having defined the regions X_1 and X_2, we can now define the *induced distributions* on these regions. Thus, we let $p_1 = \mathbf{Pr}_{x \in \mathcal{D}}[x \in X_1]$ and $p_2 = \mathbf{Pr}_{x \in \mathcal{D}}[x \in X_2]$ (note that $p_1 + p_2 = 1$), and we define $\mathcal{D}_1$ over X_1 by letting

$$\mathbf{Pr}_{x \in \mathcal{D}_1}[x \in S] = \frac{\mathbf{Pr}_{x \in \mathcal{D}}[x \in S]}{p_1}$$

for any subset $S \subseteq X_1$. Thus, $\mathcal{D}_1$ is just $\mathcal{D}$ restricted to X_1. Similarly, we define $\mathcal{D}_2$ over X_2 by letting

$$\mathbf{Pr}_{x \in \mathcal{D}_2}[x \in S] = \frac{\mathbf{Pr}_{x \in \mathcal{D}}[x \in S]}{p_2}$$

for any subset $S \subseteq X_2$.

For convenience, let us introduce the shorthand notation $\mathbf{Pr}_{EX(c,\mathcal{D})}[\cdot]$ and $\mathbf{Pr}_{EX_{CN}^{\eta}(c,\mathcal{D})}[\cdot]$ to denote probabilities over pairs $\langle x, b \rangle \in X \times \{0,1\}$ drawn from the subscripting oracle. We will now derive an expression for $P_\chi = \mathbf{Pr}_{EX(c,\mathcal{D})}[\chi = 1]$ (we have omitted the arguments x, b to χ for brevity) involving only the quantities

$$\eta, p_1, \mathbf{Pr}_{EX_{CN}^{\eta}(c,\mathcal{D}_1)}[\chi = 1], \mathbf{Pr}_{EX_{CN}^{\eta}(c,\mathcal{D})}[(\chi = 1) \wedge (x \in X_2)].$$

Looking ahead, we will then show that an accurate guess for η can be made and verified given only the upper bound η_0, and that the latter three probabilities can in fact be estimated from the *noisy* oracle $EX^{\eta}_{CN}(c, \mathcal{D})$.

To derive the desired expression for P_χ, we may write:

$$\begin{aligned}
P_\chi &= \mathbf{Pr}_{EX(c,\mathcal{D})}[\chi = 1] \\
&= \mathbf{Pr}_{EX(c,\mathcal{D})}[(\chi = 1) \wedge (x \in X_1)] + \mathbf{Pr}_{EX(c,\mathcal{D})}[(\chi = 1) \wedge (x \in X_2)] \\
&= \mathbf{Pr}_{EX(c,\mathcal{D})}[x \in X_1]\mathbf{Pr}_{EX(c,\mathcal{D})}[\chi = 1 | x \in X_1] \\
&\quad + \mathbf{Pr}_{EX(c,\mathcal{D})}[(\chi = 1) \wedge (x \in X_2)] \\
&= p_1 \mathbf{Pr}_{EX(c,\mathcal{D}_1)}[\chi = 1] + \mathbf{Pr}_{EX^{\eta}_{CN}(c,\mathcal{D})}[(\chi = 1) \wedge (x \in X_2)] \qquad (5.1)
\end{aligned}$$

where to obtain the final equality we have used the fact that for $x \in X_2$, we may replace the correct label by a noisy label without changing the probability that $\chi = 1$.

Note that since χ is always dependent on the label in region X_1, we also have:

$$\begin{aligned}
\mathbf{Pr}_{EX^{\eta}_{CN}(c,\mathcal{D}_1)}[\chi = 1] &= (1 - \eta)\mathbf{Pr}_{EX(c,\mathcal{D}_1)}[\chi = 1] + \eta \mathbf{Pr}_{EX(c,\mathcal{D}_1)}[\chi = 0] \\
&= (1 - \eta)\mathbf{Pr}_{EX(c,\mathcal{D}_1)}[\chi = 1] \\
&\quad + \eta(1 - \mathbf{Pr}_{EX(c,\mathcal{D}_1)}[\chi = 1]) \\
&= \eta + (1 - 2\eta)\mathbf{Pr}_{EX(c,\mathcal{D}_1)}[\chi = 1].
\end{aligned}$$

Solving for $\mathbf{Pr}_{EX(c,\mathcal{D}_1)}[\chi = 1]$ and substituting into Equation 5.1, we obtain:

$$P_\chi = p_1 \frac{\mathbf{Pr}_{EX^{\eta}_{CN}(c,\mathcal{D}_1)}[\chi = 1] - \eta}{1 - 2\eta} + \mathbf{Pr}_{EX^{\eta}_{CN}(c,\mathcal{D})}[(\chi = 1) \wedge (x \in X_2)] \quad (5.2)$$

As promised, we now show that the probabilities

$$p_1, \mathbf{Pr}_{EX^{\eta}_{CN}(c,\mathcal{D}_1)}[\chi = 1], \mathbf{Pr}_{EX^{\eta}_{CN}(c,\mathcal{D})}[(\chi = 1) \wedge (x \in X_2)]$$

appearing in Equation (5.2) can in fact be estimated from the noisy oracle $EX^{\eta}_{CN}(c, \mathcal{D})$. In a later section we return to the issue of estimating the noise rate.

First, note that it is easy to estimate p_1 using only calls to $EX^{\eta}_{CN}(c,\mathcal{D})$: we simply take many noisy examples $\langle x,b\rangle$ from $EX^{\eta}_{CN}(c,\mathcal{D})$, ignore the provided label b, and test whether $\chi(x,0)\neq\chi(x,1)$. If so, then $x\in X_1$, otherwise $x\in X_2$. Thus for a large enough sample, the fraction of the x falling in X_1 will be a good estimate for p_1 by Chernoff bounds. The fact that the labels are noisy does not bother us, since membership in X_1 is a property of the input x alone.

Next, $\mathbf{Pr}_{EX^{\eta}_{CN}(c,\mathcal{D}_1)}[\chi=1]$ can be estimated from $EX^{\eta}_{CN}(c,\mathcal{D})$. Note that we do not have direct access to the subscripting oracle, since it is defined with respect to $\mathcal{D}_1$ and not $\mathcal{D}$. Instead, we simply sample pairs $\langle x,b\rangle$ returned by $EX^{\eta}_{CN}(c,\mathcal{D})$ and use only those inputs x that fall in X_1 (using the membership test $\chi(x,0)\neq\chi(x,1)$). For such x, we compute $\chi(x,b)$ (using the noisy label b given with x) and use the fraction of times $\chi(x,b)=1$ as our estimate.

Finally, note that we can estimate $\mathbf{Pr}_{EX^{\eta}_{CN}(c,\mathcal{D})}[(\chi=1)\wedge(x\in X_2)]$ from $EX^{\eta}_{CN}(c,\mathcal{D})$ because we have a membership test for X_2, and this probability is already defined directly with repsect to the noisy oracle.

5.4.2 Solving for an Estimate of P_χ

Equation (5.2) has the desired form, being a simple algebraic expression for P_χ in terms of η and the probabilities that we have already argued can be accurately and efficiently estimated from $EX^{\eta}_{CN}(c,\mathcal{D})$. Assuming that we have "sufficiently accurate" estimates for all of the quantities on the right hand side of Equation (5.2), we can use the estimates to solve for an accurate estimate of P_χ.

Of course, in order to use this method to obtain an estimate of P_χ that is accurate within the desired additive error τ, we may need to estimate the probabilities on the right hand side of Equation (5.2) with an additive accuracy τ' that is slightly smaller than τ. For instance, for any $A,B\in[0,1]$ and $\hat{A},\hat{B}\in[0,1]$ that satisfy $A-\tau'\leq\hat{A}\leq A+\tau'$ and

$B - \tau' \leq \hat{B} \leq B + \tau'$ for some $\tau' \in [0,1]$, we have $AB - 2\tau' \leq \hat{A}\hat{B} \leq AB + 3\tau'$. Thus if we are using the product of the estimates $\hat{A}$ and $\hat{B}$ to estimate the product AB within additive error τ, then $\tau' = \tau/3$ suffices. However, Equation (5.2) is more complex than a single product, and thus we need to make τ' even smaller to prevent the accumulation of too much error when solving for P_χ. It turns out that the choice $\tau' = \tau/27$ will suffices; this comes from the fact that the right hand side of Equation (5.2) can be multiplied out to obtain a sum of three terms, with each term being a product of at most three factors. Thus if every estimated factor has additive error at most $\tau/27$, then each estimated product will have error at most $3(3\tau/27) = \tau/3$, and the estimated sum will have error at most τ, as desired. As we shall now see, however, we need to guess η with even greater accuracy.

5.4.3 Guessing and Verifying the Noise Rate

The main issue that remains unresolved is that when estimating the right hand side of Equation (5.2) to solve for P_χ, we do not know the exact value of η, but have only the upper bound η_0. This is handled by simulating the statistical query algorithm (let us denote this algorithm by L) $\lceil 1/2\Delta \rceil$ times, where $\Delta \in [0,1]$ is a quantity in our control that will be determined by the analysis. The ith time L is simulated (for $i = 0, 1, 2, \ldots, \lceil 1/2\Delta \rceil - 1$), we substitute the guess $\hat{\eta} = i\Delta$ for η whenever solving for a probability P_χ using Equation (5.2). Eventually we will choose the best of the $1/2\Delta$ hypotheses output by L on these many simulations as our final hypothesis.

Note that for some value of i, the guess $\hat{\eta} = i\Delta$ satisfies

$$\eta - \Delta \leq \hat{\eta} \leq \eta + \Delta.$$

We would now like to derive conditions on Δ that will ensure that for this i we have

$$\frac{1}{1-2\eta} - \tau_{\min} \leq \frac{1}{1-2\hat{\eta}} \leq \frac{1}{1-2\eta} + \tau_{\min}. \tag{5.3}$$

Here $\tau_{\min}$ will be a quantity smaller than any of the tolerances τ needed by L (but still an inverse polynomial in the learning problem parameters). Like the estimates for the probabilities discussed in the last section, this will ensure that on this ith run of L, our guess $1/(1-2\hat{\eta})$ for the factor $1/(1-2\eta)$ in Equation (5.2) will be sufficiently close to let us solve for P_χ within the desired τ.

Now we know

$$\frac{1}{1-2(\eta-\Delta)} \leq \frac{1}{1-2\hat{\eta}} \leq \frac{1}{1-2(\eta+\Delta)}.$$

Taking the leftmost inequality of this equation, we see that the leftmost inequality of Equation (5.3) will be satisfied if we have

$$\frac{1}{1-2\eta} - \tau_{\min} \leq \frac{1}{1-2(\eta-\Delta)}.$$

Solving for constraints on Δ gives:

$$1-2\eta+2\Delta \leq \frac{1}{\frac{1}{1-2\eta} - \tau_{\min}}$$

or

$$2\Delta \leq \frac{1}{\frac{1}{1-2\eta} - \tau_{\min}} - (1-2\eta).$$

If we set $x = 1/(1-2\eta)$ we obtain

$$2\Delta \leq \frac{1}{x-\tau_{\min}} - \frac{1}{x}$$

or, if we further define $f(x) = 1/x$,

$$2\Delta \leq f(x-\tau_{\min}) - f(x).$$

The right hand side of this inequality suggests analysis via the derivative of f. Now $f'(x) = -1/x^2$ and we may write $f(x-\tau_{\min}) \geq f(x)+c_0\tau_{\min}/x^2$ for some constant $c_0 > 0$, giving

$$\Delta \leq \frac{c_0\tau_{\min}}{2x^2} = \frac{c_0\tau_{\min}}{2}(1-2\eta)^2.$$

An identical analysis gives a similar bound on Δ for achieving the rightmost inequality in Equation (5.3). Thus we see that to ensure that our additive error in guessing the value of the factor $1/(1-2\eta)$ in Equation (5.2) is smaller than $\tau_{\min}$, we should make sure that the "resolution" Δ of our successive guesses for η is smaller than $c_0\tau_{\min}/(2(1-2\eta)^2)$. Since we only have the upper bound η_0, we will instead use the smaller value $\Delta = c_0\tau_{\min}/(2(1-2\eta_0)^2)$.

The preceding analysis shows that when Δ is properly chosen then on one of the simulations L our guess $\hat{\eta}$ will be sufficiently close to η, and on this run L must output a hypothesis h such that $error(h) \leq \epsilon$. We must still give some way of verifying which simulation was the good one. This is a straightforward matter. Let $h_0, \ldots, h_{\lceil 1/2\Delta \rceil - 1}$ be the hypotheses output by L on the $\lceil 1/2\Delta \rceil$ simulations. If we define $\gamma_i = \mathbf{Pr}_{EX_{CN}^{\eta}(c,\mathcal{D})}[h_i(x) \neq b]$ (this is the probability h_i disagrees with the label provided by the noisy oracle), then $\gamma_i = (1-\eta)error(h_i) + \eta(1 - error(h_i)) = \eta + (1 - 2\eta)error(h_i)$, and $\gamma_i - \gamma_j = (1-2\eta)(error(h_i) - error(h_j))$. This shows that if we estimate all of the γ_i to within an additive error of $\epsilon/(2(1-2\eta))$ (which is easily done, since γ_i is defined with respect to the noisy oracle) and choose as our final hypothesis that h_i whose associated estimate $\hat{\gamma}_i$ is smallest, then $error(h) \leq \epsilon$ with high probability. Again, having only the upper bound η_0 we can instead use the smaller additive error of $\epsilon/(1-2\eta_0)$.

5.4.4 Description of the Simulation Algorithm

We are finally ready to give a detailed outline of the overall simulation, followed by the main result of this chapter.

Algorithm Simulate-SQ$(\epsilon, \delta, \eta_0)$:

- $\tau_{\min} \leftarrow 1/(4r(1/\epsilon, n, size(c)))$, where $r(1/\epsilon, n, size(c))$ is the polynomial bound on the inverse tolerance for all queries of the statistical query algorithm L.

- $\Delta \leftarrow c_0\tau_{\min}/(2(1-2\eta_0)^2)$.
- For $i = 0$ to $\lceil 1/2\Delta \rceil - 1$:
  - $\hat{\eta} \leftarrow i\Delta$.
  - Simulate the statistical query algorithm L with accuracy parameter ϵ and using $\hat{\eta}$ as the guessed noise rate. More precisely, for every statistical query (χ, τ) made by L:
    - Randomly sample from the noisy oracle $EX^{\eta}_{CN}(c, \mathcal{D})$ to compute estimates $\hat{p}_1$ for $p_1 = \mathbf{Pr}_{EX(c,\mathcal{D})}[x \in X_1]$, $\hat{q}$ for $q = \mathbf{Pr}_{EX^{\eta}_{CN}(c,\mathcal{D}_1)}[\chi = 1]$ and $\hat{r}$ for
      $$r = \mathbf{Pr}_{EX^{\eta}_{CN}(c,\mathcal{D})}[(\chi = 1) \wedge (x \in X_2)].$$
      Here X_1, X_2 is the partition of X defined by χ. These estimates should be accurate (with high probability) within an additive error of $\tau' = \tau/27$.
    - $\hat{P}_{\chi} \leftarrow \hat{p}_1(\hat{q}-\hat{\eta})/(1-2\hat{\eta})+\hat{r}$. This is the estimated solution of Equation (5.2).
    - Return $\hat{P}_{\chi}$ to L.
  - Let h_i be the hypothesis returned by the ith simulation of L.
- For $i = 0$ to $\lceil 1/2\Delta \rceil - 1$, let $\gamma_i = \mathbf{Pr}_{EX^{\eta}_{CN}(c,\mathcal{D})}[h_i(x) \neq b]$. Randomly sample from $EX^{\eta}_{CN}(c, \mathcal{D})$ to obtain estimates $\hat{\gamma}_i$ that are accurate within additive error $\epsilon/(2(1-2\eta_0))$, and output the h_i with the smallest $\hat{\gamma}_i$.

The only details missing from our analysis of this simulation is its dependence on the confidence parameter δ, and of course, a precise bound on the number of examples from $EX^{\eta}_{CN}(c, \mathcal{D})$ required by the simulation. The handling of δ is the standard one used in Section 4.3.6 when proving the equivalence of weak and strong learning. Namely, in any execution of **Simulate-SQ** there are many places in which we need to randomly sample to accurately estimate some probability, and there is always some small probability that we fail to get an accurate estimate. If N is the number of such estimates, we can simply allocate probability of failure δ/N to each and apply the union bound to bound our total probability of failure, and we can always use the running time of L as a crude bound on N. Finally, although we have been careful to argue that for every estimate we can tolerate an additive error that is polynomial in ϵ, $\tau_{\min}$ and $(1 - 2\eta_0)$ (and thus that a polynomial sample suffices by Chernoff bounds), we leave it to the reader (Exercise 5.7) to give precise bounds, and to in fact improve the simulation sample bounds in certain natural cases by drawing a *single* initial sample from $EX^{\eta}_{CN}(c, \mathcal{D})$ from which all probabilities can be estimated throughout the simulation.

The statement of our main result follows.

Theorem 5.3 *Let C be a concept class and let $\mathcal{H}$ be a representation class over X. Then if C is efficiently learnable from statistical queries using $\mathcal{H}$, C is efficiently PAC learnable using $\mathcal{H}$ in the presence of classification noise.*

From Theorems 5.2 and 5.3, we have:

Corollary 5.4 *The representation class of conjunctions of literals is efficiently PAC learnable in the presence of classification noise.*

We leave it to the reader in the exercises to verify that the other classes for which we have provided PAC learning algorithms also have statistical query algorithms, and thus are learnable in the presence of classification noise.

5.5 Exercises

5.1. Show that the representation class of decision lists is efficiently learnable from statistical queries.

5.2. Show that there is a statistical query model analogue to the efficient algorithm given in Section 2.3 for learning conjunctions with few relevant literals. Show that this statistical query algorithm can be efficiently simulated in the classification noise model using a number of calls to $EX^{\eta}_{CN}(c, \mathcal{D})$ whose dependence on the number of literals *size*(c) is polynomial, but whose dependence on the total number of variables n is only logarithmic.

5.3. Consider the variant of the statistical query model in which the learning algorithm, in addition to the oracle $STAT(c, \mathcal{D})$, is also given access to *unlabeled* random draws from the target distribution $\mathcal{D}$. Argue that Theorem 5.3 still holds for this variant, then show that the concept class of axis-aligned rectangles in $\Re^n$ can be efficiently learned in this variant (and thus is efficiently PAC learnable in the presence of classification noise).

5.4. Show that if there is an efficient algorithm for PAC learning in the presence of classification noise by an algorithm that is given a noise rate upper bound η_0 $(1/2 > \eta_0 \geq \eta \geq 0)$ and whose running time depends polynomially on $1/(1 - 2\eta_0)$, then there is an an efficient algorithm that is given no information about the noise rate and whose running time depends polynomially on $1/(1 - 2\eta)$.

5.5. Give the weakest conditions you can on a concept class C that imply that any algorithm for PAC learning C in the presence of classification noise must have a sample complexity that depends at least linearly on $1/(1 - 2\eta)$.

5.6. Prove Theorem 5.1.

5.7. Give the best sample size bounds you can for the simulation of a statistical query algorithm in the presence of classification noise given in Section 5.4.4. Now suppose further that the statistical query algorithm always chooses its queries χ from some restricted class $\mathcal{Q}$ of functions from $X \times \{0,1\}$ to $\{0,1\}$. Give a modified simulation with improved sample size bounds that depend on $\log |\mathcal{Q}|$ (in the case of finite $\mathcal{Q}$) and

$VCD(\mathcal{Q})$.

5.6 Bibliographic Notes

The classification noise variant of the PAC model was introduced by Angluin and Laird [10], who proved that boolean conjunctions are efficiently PAC learnable in the presence of classification noise. Their paper also contains several useful and general results on learning with noise, as does the book of Laird [63]. Prior to the introduction of the statistical query model, algorithms for PAC learning with classification noise were given by Sakakibara [82] and Kearns and Schapire [61, 85], who examine a model of learning *probabilistic concepts*, in which the noise rate can be regarded as dependent on the instance.

The statistical query model and the theorems given for it in this chapter are due to Kearns [56], who also establishes that the statistical query model is strictly weaker than the PAC model, and gives lower bounds on the number of statistical queries that must be made in terms of the VC dimension. The paper also examines some apparently less benign noise models in which the statistical query results given here still hold. Exercises 5.1, 5.2, 5.3, 5.6 and 5.7 are also from the paper of Kearns. The relationship between the statistical query model and other models of robust learning is examined by Decatur [28], and Decatur and Aslam [12] establish the equivalence of weak and strong learning in the statistical query model. A recent paper has given a complete characterization of the number of queries required for learning in the statistical query model (Blum et al. [18]).

In addition to the classification noise model, several other variants of the PAC model have been introduced to model errors in the data. These include PAC learning in the presence of *malicious errors* (Valiant [93]; Kearns and Li [57]),and a model of errors in which there is noise in the inputs but not in the labels (Shackelford and Volper [87]); Goldman and

Sloan [41]; Sloan [88]). The book of Laird [63] contains a nice overview of several error models. Littlestone examines a model of errors in on-line learning [67].