Machine Learning 2: 343-370, 1988

Learning From Noisy Examples

DANA ANGLUIN (ANGLUIN@YALE.EDU)

Department of Computer Science, Yale University, P.O. Box 2158, New Haven, CT 06520, U.S.A.

PHILIP LAIRD (LAIRD%PLU@IO.ARC.NASA.GOV)

NASA Ames Research Center, MS 244-17, Moffett Field, CA 94035, U.S.A.

(Received: June 1, 1987)

(Revised: November 20, 1987)

Keywords: Concept learning, learning from examples, probably approximately correct learning, noisy data, theoretical limitations

Abstract. The basic question addressed in this paper is: how can a learning algorithm cope with incorrect training examples? Specifically, how can algorithms that produce an "approximately correct" identification with "high probability" for reliable data be adapted to handle noisy data? We show that when the teacher may make independent random errors in classifying the example data, the strategy of selecting the most consistent rule for the sample is sufficient, and usually requires a feasibly small number of examples, provided noise affects less than half the examples on average. In this setting we are able to estimate the rate of noise using only the knowledge that the rate is less than one half. The basic ideas extend to other types of random noise as well. We also show that the search problem associated with this strategy is intractable in general. However, for particular classes of rules the target rule may be efficiently identified if we use techniques specific to that class. For an important class of formulas – the k-CNF formulas studied by Valiant – we present a polynomial-time algorithm that identifies concepts in this form when the rate of classification errors is less than one half.

1. Introduction

The ability to form general concepts on the basis of particular examples is an essential ingredient of intelligent behavior. If the examples may contain errors, the task of useful generalization becomes harder. In this paper we address the question of how to compensate for randomly introduced errors, or "noise", in the example data. The examples are assumed to be generated by a sampling procedure that first produces a correctly classified example; subsequently the example is subjected to a noise process before being presented to the learning algorithm. The noise affects each example independently. Our criterion for correct identification is that of "probably approximately correct identification," introduced by Valiant (1984).

The main contributions of this paper are the introduction of a simple model of noise (the Classification Noise Process), a general upper bound on the size of a sample sufficient for learning in finite domains in the presence of classification noise, and evidence that computationally feasible algorithms exist for learning in the presence of classification noise in non-trivial domains. In addition, we indicate how some of the ideas may be used in more general settings. In the remainder of this section we define the notion of "probably approximately correct identification," give an example of this process, and introduce our model of random noise in the data.

1.1 Probably approximately correct identification

Valiant (1984) has proposed a general criterion of correct identification of a concept from examples in a stochastic setting. The idea is that after randomly sampling examples and non-examples of a concept, an identification procedure should conjecture a concept that with "high probability" is "not too different" from the correct concept.

For example, suppose a customs official requires the ability to recognize smugglers on sight. Her/his goal is to formulate a yes-or-no decision rule based on visual attributes (sex, hairstyle, nervousness, etc.), assuming that the attributes are sufficient to discriminate smugglers from non-smugglers exactly. Initially the new official goes through a learning phase in which each traveler's luggage and person is checked thoroughly, and a determination is made as to whether the person is a smuggler or not. After a certain number of examples, the official formulates a classification rule.

We do not expect the rule to be perfect (e.g., it might not apply to customs traffic elsewhere in the country), but it should be nearly correct for the typical distribution of travelers at this site. There is also some chance that, because of an unusual event (e.g., a sudden temporary drop in the local value of smuggled goods), the distribution of smugglers during the training phase is abnormal, and as a result the decision rule performs poorly under normal conditions. However, the likelihood of this is small. Important issues about this procedure include the number of training examples (detailed inspections) the official must conduct in order to refine the decision rule to within a specified accuracy and the computational complexity of the learning procedure for a given class of possible rules.

The ideas illustrated by this example are made precise by the following definitions. Let $L_1, L_2, \ldots$ be a countable family of subsets of a countable universe U, and let D be an unknown probability distribution on the elements of U. The task is to identify an unknown one of these sets, L_*, given access only to a sampling oracle $EX(\,)$. Each call to $EX(\,)$ randomly selects an element x from the universe U according to the distribution D and returns $\langle x, + \rangle$ if $x \in L_*$, and returns $\langle x, - \rangle$ otherwise.

Let us relate this to our smuggling example. U is the class of all possible travelers, as described by the attributes we have chosen to observe. The class of smugglers constitutes a subset of these people. In this case, the sampling oracle is realized by the arrival of travelers at the customs inspection site, with positive examples being those who are smugglers, and negative ones being those who are not. The identification procedure makes a number of calls to $EX(\ )$ and then conjectures one of the sets, L_h. The success of the identification is measured by two parameters, ϵ and δ, which are given as inputs to the identification procedure.

The parameter ϵ (the *tolerance*) is a bound on the "difference" between the conjectured set L_h and the unknown set L_*. Define

$$d(S,T) = \sum_{x \in S \triangle T} \Pr_D(x),$$

where S and T are any subsets of U, $S \triangle T$ is the symmetric difference[1] of S and T, and $\Pr_D$ denotes probability with respect to the distribution D. Thus, $d(S,T)$ is precisely the probability that in one call to $EX(\ )$ we will draw an element that is in one but not the other of the two sets.

The parameter δ is a *confidence* parameter. Because the calls to $EX(\ )$ are random experiments, there is always the possibility of getting a wildly unrepresentative sample and drawing a ridiculous conclusion. The parameter δ is a bound on the likelihood of such an event.

In terms of our example, it may be acceptable for the customs official to identify smugglers at least 80% of the time; in this case the tolerance is $\epsilon = 0.2$. Since training time is expensive (and quite irksome to the travelers), the officials want to be 98% sure that a single training period will result in an adequate rule; given this goal, they should choose δ to be 0.02.

The identification procedure is said to do *probably approximately correct identification* of L_* if and only if

$$\Pr[d(L_*, L_h) \geq \epsilon] \leq \delta,$$

where the probability is taken over all possible runs of the procedure. We abbreviate "probably approximately correct identification" as *pac*-identification. Less formally, the requirement is that the difference between the correct rule L_* and the conjectured rule L_h be small (less than ϵ) with high probability (greater than $1 - \delta$).

[1] $S \triangle T = (S - T) \cup (T - S)$, the set of elements in S or in T but not both.

1.2 An example: Finite classes

Let $\mathcal{L} = \{L_1, \ldots, L_N\}$ be any finite set of N rules. A simple algorithm[2] that *pac*-identifies $\mathcal{L}$ requests $m = (1/\epsilon)\ln(N/\delta)$ examples and then outputs any rule that agrees with all these examples. Since some rule L_* in $\mathcal{L}$ is correct, it is always possible to find such a rule. We can show that any rule agreeing with m or more randomly chosen examples has error greater than ϵ only with probability less than δ.

Consider a rule L with error $d(L, L_*) \geq \epsilon$. This means the likelihood that a random example agrees with L is less than $(1-\epsilon)$; hence for m such examples, the likelihood that L agrees with all of them is less than $(1-\epsilon)^m$. We can bound $(1-\epsilon)^m$ by $e^{-\epsilon m}$. Substituting the value of m above, this in turn is bounded by δ/N. Finally, there are at most $N-1$ such rules with unacceptably large error; summing the probabilities that any one of them agrees with all m examples, we have a probability less than δ. The requirements for *pac*-identification are therefore satisfied.

Consider now what happens to this procedure when some of the examples may be incorrect: there may no longer be any rule in the class $\mathcal{L}$ that is consistent with all the examples. In the worst case this could happen even if a single example is erroneous. Thus this algorithm is unsuitable even for very low rates of noise in the training data.

1.3 Related research

Many of the algorithms in the literature suffer similarly from a critical dependency on complete correctness in the training data, but there are noteworthy exceptions.

A variety of heuristic techniques have been devised to handle particular types of rules under special noise conditions. Recent examples include Schlimmer and Granger (1986) and Wilkins and Buchanan (1986). Also, Quinlan (1986) performed an experimental study of the effects of noise on learning classification rules. By independently varying the rates of noise affecting each attribute and also by allowing random misclassification (errors in the sign), he was able to quantify the impact of the noise with respect to the importance of the attribute in the target rule. Generally speaking, classification errors were found to be more significant than attribute noise.

For probabilistic identification, fewer results are available. Vapnik (1982), studying the statistical problem of choosing a rule that best accounts for empirical data, defines a model incorporating random variations in the classification of examples, and presents a statistical algorithm for finding

[2]Blumer, Ehrenfeucht, Haussler, and Warmuth (1986) present a more general version of this algorithm.

the most successful classification rule for an unknown population of data. Vapnik was not concerned with identifying the rule being presented, but despite the different nature of his objectives and his model of noise, his approach is similar to the one we describe below.

Valiant (1984) gives an algorithm for *pac*-identifying an important subclass of Boolean formulas, and elsewhere (Valiant, 1985) he modifies the algorithm to handle a certain amount of error in the examples. If n and k are positive integers, $CNF(n,k)$ denotes the class of all propositional formulas in conjunctive normal form over the variables $x_1, x_2, \ldots, x_n$ with at most k literals per clause. For example, $(x_1 \vee x_2) \wedge (\sim x_3 \vee x_4 \vee x_5)$ is in $CNF(5,3)$ but not $CNF(5,2)$.

For fixed n, the universe U is the set of all truth assignments a mapping each variable $x_1, x_2, \ldots, x_n$ to the set $\{0,1\}$. A formula ϕ in $CNF(n,k)$ is interpreted as representing the set of all assignments a from U that satisfy ϕ, i.e., such that $a(\phi) = 1$. A sampling oracle $EX(\ )$ returns assignments (represented as vectors of length n of 0's and 1's) marked either $+$ or $-$ according to whether they satisfy the unknown formula ϕ_*.

Valiant (1984) gives an identification procedure V that takes n, k, ϵ, and δ as input. V has access to a sampling oracle $EX^+(\ )$ for positive examples of an unknown formula ϕ_*, runs in time polynomial in n^k, $1/\epsilon$, and $\log 1/\delta$, and does *pac*-identification of ϕ_*, for any ϕ_* from $CNF(n,k)$.

The procedure V calculates from n, k, ϵ, and δ a number, m, of samples to draw, makes m calls to $EX^+(\ )$, and then outputs the conjunction of all clauses over $x_1, x_2, \ldots, x_n$ with at most k literals per clause that are satisfied by every positive example, i.e., by every assignment a such that some call to $EX^+(\ )$ returned the value $\langle a, + \rangle$.

Valiant (1985) considers how this algorithm (and its dual for $DNF(n,k)$ with $EX^-(\ )$) can be extended to handle a small rate of errors in the examples – errors possibly chosen in the most damaging way by an adversary. For each example, a biased coin is flipped, and if it comes up heads (with probability $1-\eta$), an example a is drawn and correctly classified as before. However, if it comes up tails (with probability η), an adversary is allowed to choose the example and classify it (perhaps incorrectly). This is called the *malicious error model*, since the algorithm must be guaranteed to work correctly for the worst possible set of choices by the adversary.

Valiant's result shows that for a very low rate of error $\eta << \epsilon$, his algorithm can be modified to achieve *pac*-identification. He suggests that only low error rates in general can be permitted if successful identification is to be possible. Kearns and Li (1987) show that this is the case for the malicious error model. In particular, for a very wide class of hypothesis spaces, if the rate of errors, η, is greater than or equal to the desired accu-

racy, ϵ, then no learning algorithm can be successful at *pac*-identification. Their results show even more stringent bounds on the error rate in the case where only positive or only negative examples are used by the learning algorithm. Our results show that a much larger rate of errors can be overcome for other, more predictable, models of errors in the data.

2. Learning despite classification noise

In this section we first define a simple model of random noise, and then consider how to modify the algorithm of Section 1.2 to accommodate errors of this type. We then consider in general the computational complexity of the solution. In the next section we show that the class $CNF(n, k)$ can be *pac*-identified efficiently despite classification noise.

2.1 A simple noise model

We introduce a model of random errors, or "noise," in the sampling oracle $EX(\ )$, called the *Classification Noise Process*. We assume that the sampling oracle is able to draw elements from the relevant distribution D without error, but that the process of determining and reporting whether the example is positive or negative is subject to independent random mistakes with some unknown probability $\eta < 1/2$. Thus the experiment performed by $EX(\ )$ involves drawing a random element x from U according to the distribution D, and then flipping a coin that comes up heads with probability $1 - \eta$. If the coin comes up heads, one reports x with the correct sign, otherwise, one reports x with the reverse of the correct sign. To indicate that the oracle is subject to errors of this type, we will denote it by $EX_\eta(\ )$. $EX_0(\ )$ is the sampling oracle with no errors of reporting.

We can interpret the Classification Noise Process using the example of the customs official learning to recognize smugglers. Every so often, a smuggler's stash is overlooked, or an ordinary traveler is mistakely nabbed because someone has hidden contraband in his or her luggage. With some probability η, such a false identification occurs independently for each traveler.

Why do we restrict η to be less than $1/2$? Clearly, when $\eta = 1/2$, the errors in the reporting process destroy all possible information about membership in the unknown set L_*, and no identification procedure could be expected to work. When $\eta > 1/2$, there is information about L_*, but it is equally information about the complement of L_* with the smaller error $1 - \eta$. While in principle we might be able to recognize this situation in domains that are not closed under complement with respect to U, we have chosen not to pursue this possibility.

If η is very close to $1/2$, how could an identification procedure be expected to work? For purposes of exposition, we assume that there is some information about η available as input to the identification procedure, namely an upper bound η_b such that $\eta \leq \eta_b < 1/2$. (Later, we show that this assumption is unnecessary.) Just as an "efficient" identification procedure is permitted in the absence of noise to run in time polynomial in $1/\epsilon$ and $1/\delta$, in the presence of noise we will permit the polynomial to have $1/(1-2\eta_b)$ as one of its arguments. This quantity is inversely proportional to how close η_b is to $1/2$, so the closer the upper bound on the error rate is to $1/2$, the longer the identification procedure will be permitted to run.

How general is this model of noise? It seems appropriate to a setting in which there is an observable, reliable mechanism selecting examples, and a separate, noisy one classifying them. However, there are many situations for which this is not a reasonable assumption. For example, if correct examples are being transmitted over a noisy line (say, with independent noise in each bit), then not only is the sign of the example subject to errors, but a given example x may be changed into another one x'. In this case, the examples x' reported by the sampling oracle may come from a different distribution D'. Even if our results were applicable in this situation, the "difference" of the hypothesis from the correct set would be measured with respect to the observed distribution D' instead of the true distribution D, which is not necessarily what is wanted.

Note the difference between the classification noise model and that treated by Valiant (1985). In the earlier study, the errors could be maliciously rather than randomly chosen. Valiant's results for $CNF(n,k)$ hold only for a small rate of noise, and indeed we shall see that this model can tolerate only a small rate of noise for any domain. However, the basic ideas behind the analysis of classification noise are applicable to other types of noise, and they can be used to derive estimates of the amount of tolerable noise and the number of examples required.

2.2 How many noisy examples are enough?

Forgetting for a moment the question of computational feasibility, how can we be sure that there is enough information in a certain number of samples drawn from a noisy oracle to determine the unknown set L_* to within ϵ error with probability at least $1-\delta$? We consider the simple case of a finite set of hypotheses, say, $L_1, L_2, \ldots, L_N$. For the noise-free case, the result described in Section 1.2 can be summarized as follows.

Theorem 1 (Blumer et al., 1986) If L_i is *any* hypothesis that agrees with at least

$$m = \frac{1}{\epsilon} \ln\left(\frac{N}{\delta}\right)$$

samples drawn from the $EX_0(\ )$ oracle, then

$$\Pr[d(L_i, L_*) \geq \epsilon] \leq \delta.$$

This result is simple but significant. It says that there is enough information in a feasibly small number of examples to *pac*-identify any finite domain. In this approach, the examples serve as a probabilistic filter to screen out unacceptably bad hypotheses.

Note that because of the dependence of the sample size on $\log N$, large increases in the number of rules in the class $\mathcal{L}$ cause only much smaller growth in the size of the sample required. For the same reason we can significantly decrease the confidence limit δ with only a small increase in the sample size. To use this approach in a practical setting, we need to consider the computational complexity of searching for a hypothesis that is consistent with the samples drawn. For some domains this is known to be a hard problem (Blumer et al., 1986).

In the presence of noise this approach may fail because there is no guarantee that any of the hypotheses will be consistent with all the examples. However, if we replace the goal of consistency with that of minimizing the number of disagreements with the examples, and permit the number of samples to depend on the upper bound η_b on the error rate, we get an analogous result, given by Theorem 2 below.[3]

This theorem is most usefully interpreted as a simple, general result giving an upper bound on the size of a sample sufficient for *pac*-identification in finite domains in the presence of classification noise. Minimizing the number of disagreements with the examples can be a computationally difficult problem (see Theorem 4 for evidence of this), so this approach generally does not yield an efficient algorithm. More sophisticated approaches are possible in specific domains, as we show in Section 3.

Let $\sigma = \langle x_1, s_1\rangle, \langle x_2, s_2\rangle, \ldots, \langle x_m, s_m\rangle$ denote a sequence of samples drawn from an $EX_\eta(\ )$ oracle, where each x_i is in the universe U and each s_i is either $+$ or $-$. If L_i is any possible hypothesis, let $F(L_i, \sigma)$ denote the number of indices j for which L_i disagrees with $\langle x_j, s_j\rangle$, that is, $s_j = +$ and x_j is not in L_i or $s_j = -$ and x_j is in L_i.

[3]Shackelford and Volper (1987) discovered this theorem independently.

Theorem 2 If we draw a sequence σ of

$$m \geq \frac{2}{\epsilon^2(1-2\eta_b)^2} \ln\left(\frac{2N}{\delta}\right) \tag{1}$$

samples from an $EX_\eta(\ )$ oracle and find any hypothesis L_i that minimizes $F(L_i, \sigma)$, then

$$\Pr[d(L_i, L_*) \geq \epsilon] \leq \delta.$$

PROOF: We analyze the expected rate of disagreement between any hypothesis L_i and sample sequences produced by the oracle $EX_\eta(\ )$ with unknown set L_*. Let

$$d_i = d(L_i, L_*).$$

The probability that an example produced by $EX_\eta(\ )$ disagrees with L_i is the probability that an example is drawn from $L_i \triangle L_*$ and reported correctly (which is just $d_i(1-\eta)$) plus the probability that an example is drawn from the complement of $L_i \triangle L_*$ and reported incorrectly (which is just $(1-d_i)\eta$.) Let p_i denote the probability that an example from $EX_\eta(\ )$ disagrees with L_i; then we have

$$p_i = d_i(1-\eta) + (1-d_i)\eta.$$

In the case that the hypothesis L_i is equal to L_*, we have $p_i = \eta$, since disagreements will only arise as the result of reporting errors. The expression for p_i may be rewritten as

$$p_i = \eta + d_i(1-2\eta).$$

Since $\eta < 1/2$, this shows that any hypothesis L_i has an expected rate of disagreement of at least η. In particular, if we define a hypothesis L_i to be ϵ-*bad* if and only if $d_i \geq \epsilon$, then for any ϵ-bad hypothesis L_i we have

$$p_i \geq \eta + \epsilon(1-2\eta).$$

Thus we have a separation of at least $\epsilon(1-2\eta)$ between the disagreement rates of correct and ϵ-bad hypotheses. By our assumptions, η is not known, but an upper bound $\eta_b < 1/2$ is known, so we have a known lower bound on the separation, $\epsilon(1-2\eta_b)$.

The problem is reduced to guaranteeing that the number m of samples drawn from $EX_\eta(\ )$ is sufficient to guarantee that no ϵ-bad hypothesis has a lower observed rate of disagreement with the samples than L_*, with probability greater than $1-\delta$.

At this point we must introduce some notation. Let p be a number between 0 and 1, and suppose that we have a coin C_p whose probability of

coming up heads on each toss is p. Let r be a number between 0 and 1, and let m be a non-negative integer. Then $GE(p, m, r)$ will denote the probability of getting at least rm heads in a sequence of m independent flips of the coin C_p. (Formally, this is the probability of getting at least rm successes in m independent Bernoulli trials with probability p.) Analogously $LE(p, m, r)$ will denote the probability of at most rm successes in m independent Bernoulli trials with probability p. We present lemmas bounding these quantities in the Appendix.

Let $s = \epsilon(1 - 2\eta_b)$, and let σ denote a sequence of m examples drawn from the noisy sampling oracle $EX_\eta(\ )$. In order for some ϵ-bad hypothesis L_i to minimize $F(L_i, \sigma)$, either

$$F(L_*, \sigma)/m \geq \eta + s/2$$

or

$$F(L_i, \sigma)/m \leq \eta + s/2$$

for some ϵ-bad hypothesis L_i, or both. Applying Lemma 10 in the Appendix,

$$\begin{aligned} \Pr[F(L_*, \sigma)/m \geq \eta + s/2] &= GE(\eta, m, \eta + s/2) \\ &\leq \delta/2N \\ &\leq \delta/2, \end{aligned}$$

and if L_i is ϵ-bad then

$$\begin{aligned} \Pr[F(L_i, \sigma)/m \leq \eta + s/2] &\leq LE(\eta + s, m, \eta + s/2) \\ &\leq \delta/2N. \end{aligned}$$

Thus the probability that any ϵ-bad hypothesis L_i has $F(L_i, \sigma)/m \leq \eta + s/2$ is at most $\delta/2$, since there are at most $N - 1$ ϵ-bad hypotheses. Putting these two inequalities together, the probability that some ϵ-bad hypothesis minimizes $F(L_i, \sigma)$ is at most δ. ■

Note that the bound m on the number of examples is polynomial in $\log N$, $1/\epsilon$, $\log(1/\delta)$, and $1/(1 - 2\eta_b)$. Thus the noise has increased the number of examples we must obtain, but not to an infeasible number. Laird (1987) has calculated a better upper bound than that in Equation (1), as well as a lower bound; in particular, m depends only on ϵ^{-1}, and not on ϵ^{-2}.

Theorems 1 and 2 both depend on the fact that the set $\mathcal{L}$ of rules is finite, but suppose $\mathcal{L}$ is a countable or even continuous class? We will not discuss this case in detail, but Blumer et al. (1986) have shown that whether or not an infinite class can be identified by means of a finite set of reliable examples depends on a property of the class known as the *Vapnik-Chervonenkis dimension.* Only classes with finite dimension d can be so

identified. Using their result, together with the ideas of Theorem 2, one can also show (Laird, 1987) that classes with finite d are precisely those which can be so identified when the examples are afflicted by classification noise. The finite situation above is a special case, with $d \leq \log N$.

Vapnik (1982), whose work we have already mentioned, suggests a similar statistical approach when the examples are independently subjected to random classifications that may depend on the specific example. That is, instead of a uniform error rate η for all examples x drawn from D, the probability $\eta(x)$ of a classification error may depend upon the particular example x. Let L_0 be the hypothesis with the smallest expected rate of disagreement with the oracle $EX_\eta(\ )$; Vapnik shows that the sampling technique of Theorem 2 can be used to find (with probability greater than $1-\delta$) a hypothesis L such that $d(L, L_0) \leq \epsilon$.

With this procedure, the intent is to discover the best classification rule for describing the sample data from a noisy source EX_η. By contrast, the intent of the above identification procedure is to discover L_*, the classification rule underlying the data. These need not be the same; indeed, the rule L obtained with Vapnik's procedure may not satisfy the condition $d(L, L_*) \leq \epsilon$, even when the mean rate of classification errors over the examples is less than one half. For example, let $U = \{a, b\}$ and $L_* = \{a, b\}$, with examples distributed as follows: $\Pr_D[a] = 0.1, \Pr_D[b] = 0.9$, and $\eta(a) = 0.6, \eta(b) = 0.48$. Here the probability that L_* disagrees with a random example is $\Pr_D[a]\eta(a) + \Pr_D[b]\eta(b) = 0.492$, whereas the hypothesis $L_0 = \{b\}$ disagrees at a rate of $\Pr_D[a](1-\eta(a)) + \Pr_D[b]\eta(b) = 0.472$. Note that L_0 fails on average less often than the correct set L_*, even though the mean rate of noise is 0.492 (less than half). So for this noise model, the procedure may propose an ϵ-bad rule with unacceptably high probability.

2.3 Determining η_b

So far we have assumed that the identification procedure is told an upper bound $\eta_b < 1/2$ on the noise rate η. We now show that this assumption is unnecessary.

Continuing with the example of the customs official learning to spot smugglers, whereas before the person was (somehow) told that at most 5% (say) of the customs inspections will yield an incorrect classification, no such information is now provided. The only assumption is that fewer than half the examples are wrong on average. Surprisingly, the official can estimate an upper bound (less than 1/2) on the rate of noise with a feasible number of examples.

We have seen that the rate at which a rule L disagrees with the examples is at least η, and that for the target rule L_* this rate is precisely η. Hence we

Table 1. The algorithm E.

Let $\mathcal{L} = \{L_1, L_2, \ldots, L_N\}$.

1. Initialize: $\hat{\eta}_b \leftarrow 1/4$ and $r \leftarrow 1$.
2. (Round r) Repeat until the halt condition is fulfilled:
   - 2.1 Request $m_r(N, \delta)$ examples. (The value of m_r is given in the text.)
   - 2.2 For each rule $L_i \in \mathcal{L}$, test L_i against all the examples and determine $\hat{p}_i = F_i/m_r$, the proportion of examples in disagreement with L_i. Let $\hat{p}_{\min}$ be the minimum such value.
   - 2.3 If $\hat{p}_{\min} < \hat{\eta}_b - 2^{-(r+2)}$, then halt and output $\hat{\eta}_b$.
   - 2.4 Else,
     - 2.41 $r \leftarrow r + 1$.
     - 2.42 $\hat{\eta}_b \leftarrow \frac{1}{2} - 2^{-(r+1)}$.

can use the minimum rate of disagreement over all the rules as an estimator for the noise η. Once again this does not yield a feasible algorithm in general, since a direct implementation entails minimizing the number of disagreements over the whole hypothesis space. However, in Section 3 we see that the basic method can be adapted to be computationally feasible in a specific situation.

We describe a procedure that outputs a value η_b such that with probability at least $1 - \delta$, η_b is between η and $1/2$. Given this value, we can use Theorem 2 to find an acceptable hypothesis with probability at least $1 - \delta$. The probability that either of these procedures fails is then less than 2δ.

Our algorithms require that η_b be an upper bound for η and also be less than $1/2$. One idea is to take enough samples so that the empirical rate of disagreement for each hypothesis is "very close" to its average. However, we have no way in advance of knowing how close η is to $1/2$, and η_b must squeeze in between them. Thus it seems that we must use an iterative search procedure that successively reduces the gap assumed to exist between η and $1/2$.

We begin by guessing that η is less than $1/4$, and take $\eta_b = 1/4$. If that value fails a certain test, we increase the guess to $3/8$, then $7/16$, etc., each time halving the distance between the previous guess and $1/2$. For the test, we draw some examples and estimate the failure probability of each of the rules in $\mathcal{L}$. The smallest empirical failure rate $\hat{p}_i = F(L_i, \sigma)/m$ is compared to the current value of η_b. If $\hat{p}_i < \eta_b$, we halt and output η_b as our bound. Otherwise we increase η_b and repeat. The size of the sample drawn is increased at each iteration.

Table 1 presents a specific algorithm E that implements this strategy. This leads to the following theorem:

Theorem 3 Let

$$m_r(N,\delta) = 2^{2r+3} \cdot \ln\left(\frac{N2^{r+2}}{\delta}\right).$$

Then with probability greater than $1-\delta$, algorithm E halts on or before round $r' = 1 + \lceil \log_2(1-2\eta)^{-1} \rceil$ and outputs an estimate $\hat{\eta}_b$ such that $\eta < \hat{\eta}_b < 1/2$.

PROOF: The value of m_r has been chosen so that, in m_r examples,

$$GE(p, m_r, p + 2^{-(r+2)}) \leq \frac{1}{2} \cdot \frac{\delta/2}{N2^r}$$

and

$$LE(p, m_r, p - 2^{-(r+2)}) \leq \frac{1}{2} \cdot \frac{\delta/2}{N2^r}.$$

(Apply Lemma 9 with $2^{-(r+2)}$ in place of s and $\delta/(N2^{r+2})$ in place of δ.) Thus if a rule L_i is expected to disagree with a fraction p_i of the examples, the probability that $|\hat{p}_i - p_i| \geq 2^{-(r+2)}$ is at most $(\delta/2)/(N2^r)$. After summing over N rules and over all possible rounds, we find that the probability that in any round r the empirical value $\hat{p}_i$ for some rule differs from its expected value p_i by as much as $2^{-(r+2)}$ is at most $\delta/2$. We claim that, with probability greater than $1-\delta/2$,

- the algorithm halts on or before round $r' = 1 + \lceil \log_2(1-2\eta)^{-1} \rceil$.
- when it halts, $\hat{\eta}_b > \eta$.

In round r', $\eta \leq \frac{1}{2} - 1/2^{r'}$, and $m_{r'}$ is sufficient to ensure that $\hat{p}_{\min} \leq \eta + 2^{-(r'+2)}$, with a probability of more than $1-\delta/2$. However,

$$\begin{aligned}
\hat{\eta}_b - 2^{-(r'+2)} &= \left(\frac{1}{2} - \frac{1}{2^{r'+1}}\right) - \frac{1}{2^{r'+2}} \\
&> \left(\frac{1}{2} - \frac{1}{2^{r'}}\right) - \frac{1}{2^{r'+2}} \\
&\geq \eta + 2^{-(r'+2)} \\
&\geq \hat{p}_{\min}.
\end{aligned}$$

with probability $> 1-\delta/2$. Thus the algorithm will halt at or before round r' with this probability.

Suppose the algorithm halts in round r. By choice of m_r, $\hat{p}_{\min} \geq \eta - 2^{-(r+2)}$ with probability more than $1 - \delta/2$. The fact that it stops implies that $\hat{p}_{\min} < \hat{\eta}_b - 2^{-(r+2)}$. Thus

$$\eta - \frac{1}{2^{r+2}} < \hat{\eta}_b - \frac{1}{2^{r+2}},$$

and hence $\eta < \hat{\eta}_b$ with probability $> 1 - \delta/2$.

Finally, the algorithm fails only if at least one of the above two conditions fails. Since each occurs with probability at most $\delta/2$, failure occurs with probability at most δ. ■

Note that with probability zero, the algorithm could fail to halt. Thus strictly speaking, it is not a finite procedure. Assuming it halts in round $r_0 = 1 + \lceil \log_2(1-2\eta)^{-1} \rceil$, the total number of examples required is $\mathcal{O}((1-2\eta)^{-2} \cdot \ln[N/(1-2\eta)\delta])$. Thus asymptotically the process of determining η_b increases the sample size only slightly. Also, we can accelerate the convergence by allowing $\hat{\eta}_b$ in each round to be the larger of the value obtained in step 2.42 and $\hat{p}_{\min}$.

2.4. How hard is minimizing disagreements?

The approach suggested by Theorem 2 is to draw a feasibly small sample from $EX_\eta(\,)$ and then find a hypothesis that minimizes disagreements with the sample. We now show that this direct approach may be computationally infeasible even in very simple domains. Note that this result concerns only this approach, and should not be confused with the stronger results of Kearns et al. (1987), which establish that some learning problems in the Valiant model may be computationally intractable, no matter what approach is taken.

We will consider the domain of products of positive literals. Let n be a positive integer. Let $PP(n)$ denote the set of all products of a subset of the literals $x_1, x_2, \ldots, x_n$. There are 2^n such products; the empty product is interpreted as equivalent to "true." Each product π in $PP(n)$ is interpreted as denoting the set of truth-value assignments that satisfy it. $PP(n)$ is a subset of the formulas in $CNF(n, 1)$.

A sample sequence σ will consist of a finite sequence of ordered pairs of the form $\langle a_j, s_j \rangle$, where a_j is a truth-value assignment to the variables $x_1, x_2, \ldots, x_n$ and s_j is either $+$ or $-$. If $\pi \in PP(n)$ and σ is a sample sequence, then $F(\pi, \sigma)$ is the number of pairs $\langle a_j, s_j \rangle$ in σ such that $s_j = +$ and $a_j(\pi) = 0$ or $s_j = -$ and $a_j(\pi) = 1$. That is, $F(\pi, \sigma)$ is the number of disagreements between π and the sample sequence σ.

Theorem 4 Given positive integers n and c and a sample sequence σ, the problem of determining whether there is an element $\pi \in PP(n)$ such that $F(\pi, \sigma) \leq c$ is NP-complete.

PROOF: The proof is a polynomial-time reduction of the vertex cover problem to the specified problem. The vertex cover problem is specified by an undirected graph G of n vertices and a positive integer $c \leq n$, and the question is whether there exists a set C of at most c vertices of G such that every edge of G is incident to at least one vertex in C. (Such a set C is called a *vertex cover.*) The vertex cover problem is NP-complete.

Let a vertex cover problem, $\langle G, c \rangle$, be given. Suppose the vertices of G are $v_1, v_2, \ldots, v_n$. There will be n variables: $x_1, x_2, \ldots, x_n$. For each vertex v_i, define a truth assignment a_i that maps x_i to 0 and every other x_j to 1. For each edge $e = \{v_i, v_j\}$, define a truth assignment b_e that maps x_i and x_j to 0 and every other x_k to 1. The sample sequence σ consists of one copy of $\langle a_i, + \rangle$ for each vertex v_i and $n + 1$ copies of $\langle b_e, - \rangle$ for each edge e in G. Then we claim that G has a vertex cover of at most c vertices if and only if there is an element π of $PP(n)$ such that $F(\pi, \sigma) \leq c$.

Suppose G has a vertex cover C of at most c vertices. Let π denote the product of those x_i such that v_i is in C. How many examples from σ disagree with π? For each vertex v_i, the assignment a_i assigns 0 to π if and only if $v_i \in C$. Thus, π disagrees with at most c positive examples from σ. For each edge $e = \{v_i, v_j\}$, the set C contains at least one of v_i or v_j, so the product π contains at least one of x_i or x_j. Since the assignment b_e is 0 on both x_i and x_j, it must be 0 on π. Thus, π agrees with all the negative examples in σ. Hence $F(\pi, \sigma) \leq c$, as claimed.

Now suppose that there exists some $\pi \in PP(n)$ such that $F(\pi, \sigma) \leq c$. Since $c \leq n$, this means that π must agree with all the negative examples in σ, since each one is repeated $n + 1$ times. Hence π can only disagree with positive examples in σ, and at most c of them. Thus π must contain at most c literals x_i. Define the set C to be all those vertices v_i such that x_i appears in the product π. Then C contains at most c vertices; it remains to see that it is a vertex cover. If $e = \{v_i, v_j\}$ is any edge in G then the assignment b_e must assign 0 to π, since π agrees with all the negative examples. But b_e assigns 0 to π if and only if π contains at least one of x_i or x_j. Thus C contains at least one of v_i or v_j, so C is a vertex cover of G.

The computation of n, c, and σ from $\langle G, c \rangle$ can clearly be carried out in polynomial time. ■

This result indicates that even for a very simple domain the approach of directly trying to minimize the number of disagreements with the sample may not be computationally feasible. In the next section, we show that a somewhat more sophisticated approach does permit efficient *pac*-identification of k-CNF formulas from noisy samples.

3. Efficient *pac*-identification of k-CNF formulas in the presence of noise

We now describe a procedure V' that does *pac*-identification of k-CNF formulas in polynomial time. The main idea is that instead of searching for the formula in $CNF(n,k)$ with the fewest disagreements, one tests the clauses individually and includes those that disagree least often on positive examples. Since there are exponentially fewer clauses than formulas, the procedure is much more efficient. Note that this method does *not* solve an NP-hard problem: the resulting k-CNF formula may not be the best in terms of minimizing error on the examples. But it will (with high probability) have error less than ϵ.

The inputs to the procedure are n, k, ϵ, δ, η_b, and a noisy oracle $EX_\eta(\,)$ for an unknown formula ϕ_* from $CNF(n,k)$, using an unknown distribution D to sample truth-assignments. The accuracy and confidence parameters ϵ and δ must be between 0 and 1. And again, for expository purposes we assume that a bound η_b on the rate of noise is provided such that $0 \leq \eta \leq \eta_b < 1/2$.

Once n and k are fixed, there is a set $\mathcal{C}$ of all possible clauses over the variables $x_1, \ldots, x_n$ with at most k literals per clause. Let M denote the cardinality of $\mathcal{C}$. It is easy to show that M is at most $(2n+1)^k$.

Let ϕ_* be the target formula. Without loss of generality we may assume that ϕ_* is maximally consistent – i.e., it includes every clause C with at most k literals such that C is logically implied by ϕ_*.

3.1 Motivation for the procedure V'

Once D is fixed we define two probabilities for each clause C from $\mathcal{C}$:

$$\begin{aligned} p_0(C) &= \Pr[a(C) = 0] \\ p_1(C) &= \Pr[a(C) = 1]. \end{aligned}$$

If ϕ_* is also fixed, we may subdivide these probabilities into four cases, p_{rs}, for $r = 0, 1$ and $s = 0, 1$ as follows:

$$p_{rs}(C) = \Pr[a(C) = r \text{ and } a(\phi_*) = s].$$

Note that $p_0(C) = p_{00}(C) + p_{01}(C)$.

We use these probabilities to classify each clause as follows. A clause C is defined to be *important* if and only if

$$p_0(C) \geq Q_I,$$

where

$$Q_I = \epsilon/16M^2.$$

A clause C is defined to be *harmful* if and only if

$$p_{01}(C) \geq Q_H,$$

where

$$Q_H = \epsilon/2M.$$

Note that $Q_H \geq Q_I$, so every harmful clause is important. Note also that no clause contained in ϕ_* can be harmful.

The intuition is that a non-important clause is almost always assigned the value 1 by assignments chosen according to D, so it may be included or not in the final hypothesis without significantly affecting the outcome. On the other hand, a harmful clause is one for which a significant fraction of the assignments chosen from D make the clause 0 but the correct hypothesis 1. If a harmful clause is included in the final hypothesis, it will cause a nontrivial probability of disagreement between the final hypothesis and the correct hypothesis. Thus, the strategy of the procedure V' is to attempt to include in the final hypothesis all the important clauses contained in ϕ_* and no harmful clauses. Our first lemma shows that if V' succeeds in this attempt, then the final hypothesis is indeed an ϵ-approximation of ϕ_*.

Lemma 1 Let D and ϕ_* be fixed. Let ϕ be any product of clauses from $\mathcal{C}$ that contains every important clause in ϕ_* and contains no harmful clauses. Then $d(\phi, \phi_*) < \epsilon$.

PROOF: We analyze the probability of an assignment a such that $a(\phi_*) = 1$ and $a(\phi) = 0$ or vice versa. Let $\phi - \phi_*$ denote the set of clauses in ϕ but not in ϕ_*.

$$\begin{aligned} \Pr[a(\phi_*) = 1 \text{ and } a(\phi) = 0] &\leq \sum_{C \in \phi - \phi_*} p_{01}(C), \\ &< MQ_H \text{ (no element of } \phi - \phi_* \text{ is harmful)}, \\ &= \epsilon/2. \end{aligned}$$

For the other side,

$$\begin{aligned} \Pr[a(\phi) = 1 \text{ and } a(\phi_*) = 0] &\leq \sum_{C \in \phi_* - \phi} p_0(C), \\ &< MQ_I \text{ (no element of } \phi_* - \phi \text{ is important)}, \\ &< \epsilon/2. \end{aligned}$$

Thus, $$\Pr[a(\phi) \neq a(\phi_*)] < \epsilon/2 + \epsilon/2.$$ ■

The procedure V' has no direct information about whether a clause is important or harmful – it must rely on the noisy oracle $EX_\eta(\ )$ for its information about D and ϕ_*. Since the oracle $EX_\eta(\ )$ reports assignments according to the distribution D, $p_0(C)$ can be directly estimated by sampling the oracle and calculating the fraction of assignments that assign 0 to C. The procedure V' uses this to construct a set I that, with high probability, contains all the important clauses C from $\mathcal{C}$. If this is accomplished, the remaining problem is to identify all the harmful clauses in I. (Note that V' depends in an essential way upon the fact that, in this model, the distribution D is not perturbed by the presence of noise.)

However, the definition of a harmful clause refers to the values of assignments on ϕ_*, which are subject to reporting errors and cannot be estimated directly. For each clause C we define one more probability:

$$p_{0+}(C) = \Pr[\text{a sample } \langle a, s\rangle \text{ drawn from } EX_\eta(\ ) \text{ has } a(C) = 0 \text{ and } s = +]$$

This may be directly estimated using calls to $EX_\eta(\ )$. A sample $\langle a, s\rangle$ will have $a(C) = 0$ and $s = +$ if and only if either $a(C) = 0$ and $a(\phi_*) = 1$ and there was no reporting error, or $a(C) = 0$ and $a(\phi_*) = 0$ and there was a reporting error. Thus

$$\begin{aligned} p_{0+}(C) &= (1-\eta)p_{01}(C) + \eta p_{00}(C) \\ &= \eta(p_{00}(C) + p_{01}(C)) + (1-2\eta)p_{01}(C) \\ &= \eta p_0(C) + (1-2\eta)p_{01}(C). \end{aligned}$$

If $p_0(C) \neq 0$, then

$$\frac{p_{0+}(C)}{p_0(C)} = \eta + \frac{p_{01}(C)}{p_0(C)}(1-2\eta).$$

Since $\eta < 1/2$, this quantity is always greater than or equal to η and is equal to η if C is contained in ϕ_*. Since $p_0(C) \leq 1$, for all clauses C such that $p_0(C) \neq 0$,

$$\frac{p_{0+}(C)}{p_0(C)} \geq \eta + p_{01}(C)(1-2\eta). \tag{2}$$

Observe that if $C \in \phi^*$, then the ratio $p_{0+}(C)/p_0(C) = \eta$. If C is a harmful clause, then $p_{01}(C) \geq Q_H$, so

$$\frac{p_{0+}(C)}{p_0(C)} \geq \eta + Q_H(1-2\eta). \tag{3}$$

The quantity $p_{0+}(C)/p_0(C)$ is the proportion of those assignments falsifying C that are reported with a positive sign. The preceding calculation shows that there is a separation of at least

$$s = Q_H(1 - 2\eta)$$

in the expected value of this quantity between clauses that are to be retained (important clauses in ϕ_*) and clauses that are to be discarded (harmful clauses). Since η_b is an upper bound on η, the minimum separation is $s_b = Q_H(1 - 2\eta_b)$. Moreover, $p_{0+}(C)/p_0(C)$ can be estimated by sampling the oracle $EX_\eta(\ )$. (Recall that I contains clauses falsified by a nontrivial number of samples, so for elements of I this estimate will be sufficiently accurate.)

The procedure V' calculates an estimate η' of η and identifies as harmful all those clauses $C \in I$ whose estimated value of $p_{0+}(C)/p_0(C)$ is greater than $\eta' + s_b/2$. The final output is the product of all the other clauses in I. In order for this to work, V' needs a sufficiently accurate estimate η' for η. Where does this come from? If I contains any clause C in ϕ_*, then the estimate of $p_{0+}(C)/p_0(C)$ will be close to η. In this case, the minimum estimate of $p_{0+}(C)/p_0(C)$ for all clauses C in I will be close to η.

However, it may happen that no clause in I is contained in ϕ_*, and this minimum value may not be a good estimate of η. In this case, provided all the important clauses are in I, we know that ϕ_* does not contain any important clauses. This means that most assignments drawn from D assign the value 1 to ϕ_*. In this case, the observed overall rate of negative examples will be sufficiently close to η. Thus, the estimate of η is taken to be the minimum of two estimates: the estimated fraction of negative examples and the minimum estimated value of $p_{0+}(C)/p_0(C)$ over all clauses C in I.

3.2 Concise description of V'

Now let us summarize the description of V'. From n, k, ϵ, δ, and η_b, the procedure V' calculates the following:

$$\begin{aligned}
\mathcal{C} &= \{C : C \text{ is a clause over } n \text{ variables with at most } k \text{ literals}\},\\
M &= |\mathcal{C}|,\\
K &= 2^{10},\\
m &= \left\lceil \frac{KM^4}{\epsilon^3(1-2\eta_b)^2} \ln\left(\frac{6M}{\delta}\right) \right\rceil,\\
Q_H &= \epsilon/2M,\\
s_b &= Q_H(1-2\eta_b) = \epsilon(1-2\eta_b)/2M,\\
Q_I &= Q_H/8M = \epsilon/16M^2.
\end{aligned}$$

V' draws m samples from the oracle $EX_\eta(\ )$, say $\sigma = \langle a_1, s_1\rangle, \ldots, \langle a_m, s_m\rangle$, where each a_i is a truth-value assignment to the variables $x_1, \ldots, x_n$ and each s_i is either $+$ or $-$. The following quantities are defined using σ:

$$\begin{aligned} Z_- &= |\{j : s_j = -\}|, \\ Z_0(C) &= |\{j : a_j(C) = 0\}|, \\ Z_{0+}(C) &= |\{j : a_j(C) = 0 \text{ and } s_j = +\}|. \end{aligned}$$

Z_- is the overall number of negative samples, $Z_0(C)$ is the number of samples that assign 0 to the clause C, and $Z_{0+}(C)$ is the number of samples that assign 0 to C and are reported with the sign $+$. For each clause C in $\mathcal{C}$ such that $Z_0(C) \neq 0$, define

$$h(C) = Z_{0+}(C)/Z_0(C).$$

$h(C)$ is the estimated value of the quantity $p_{0+}(C)/p_0(C)$.

The procedure V' calculates one estimate of η:

$$\eta_1 = Z_-/m,$$

which is just the observed fraction of negative examples. The procedure V' then forms the set I by including all those clauses C in $\mathcal{C}$ such that

$$Z_0(C)/m \geq Q_I/2.$$

Note that I is non-empty, since if a clause consisting of a single variable is not in I, then the clause consisting of the complement of the variable is in I. V' then calculates a second estimate of η to be

$$\eta_2 = \min\{h(C) : C \in I\},$$

after which it calculates

$$\eta' = \min\{\eta_1, \eta_2\}.$$

The final output ϕ of V' is the product of all those clauses $C \in I$ such that

$$h(C) \leq \eta' + s_b/2.$$

It is clear from this description that V' runs in time polynomial in n^k, $1/\epsilon$, $\log 1/\delta$, and $1/(1 - 2\eta_b)$.

3.3 Proof of correctness of V'

In this section we show that V' achieves *pac*-identification of the formulas in $CNF(n, k)$.

Theorem 5 For every $\phi_* \in CNF(n,k)$, V' *pac*-identifies ϕ_*, that is,

$$\Pr[d(\phi,\phi_*) \geq \epsilon] \leq \delta.$$

PROOF: Consider how the algorithm could go astray:

- Some important clause might not be selected for inclusion in I.
- The estimate η' could be too large or too small.
- Some harmful clause could have an abnormally small number of failures on positive examples and thereby be included in the output expression.
- Some correct clause could have an abnormally large number of failures on positive examples and thereby be excluded from the output expression.

The series of lemmas[4] below show that the second possibility has probability at most $\delta/2$, while the others each have probability at most $\delta/6$. In all, therefore, these mishaps have probability at most δ, and by Lemma 1 the output expression will be ϵ-good with high probability. ■

Lemma 2 With high probability the set I includes all important clauses – i.e., all clauses C such that $p_0(C) \geq Q_I$.

PROOF: For an important clause C to be omitted, the value $Z_0(C)/m$ must be less than $Q_I/2$ – an amount more than $Q_I/2$ below its expected value of at least Q_I. With the sample size m, Lemma 8 can be applied to show that $LE(p_0(c), m, Q_I/2) \leq \delta/6M$. Summing this probability over M clauses completes the proof. ■

Lemma 3 Let $s = Q_H(1-2\eta)$. Then with high probability η_1 is not "too small" – i.e., $\eta_1 \geq \eta - s/4$ with high probability.

PROOF: Consider the probability p_- that an example is classified negative by the noisy oracle. Without noise this probability is $p_0(\phi^*)$. With noise, this probability becomes

$$\begin{aligned} p_- &= (1-\eta)p_0(\phi^*) + \eta(1-p_0(\phi^*)) \\ &= \eta + p_0(\phi^*)(1-2\eta) \\ &\geq \eta. \end{aligned} \tag{4}$$

By Lemma 8, $LE(\eta, m, \eta - s/4) \leq LE(\eta, m, \eta - s_b/4) \leq \delta/6$. ■

[4] In the following technical lemmas, "with high probability" means "with probability $> 1 - \delta/6$."

Lemma 4 Given that I contains all important clauses, with high probability η_2 is not "too small" – i.e., $\eta_2 \geq \eta - s/4$ with high probability.

PROOF: η_2 will be too small iff, for some clause C,

$$h(C) < \eta - s/4.$$

But by Eq. (2) the expected value of $h(C)$ is $p_{0+}(C)/p_0(C) \geq \eta$. The sample size over which the ratio is being measured is at least $mQ_I/2$ since $C \in I$. Using Lemma 8, $LE(\eta, mQ_I/2, \eta - s/4) \leq LE(\eta, mQ_I/2, \eta - s_b/4) \leq \delta/6M$. Summing this probability over all M clauses ends the proof. ■

Lemma 5 Given that I contains all important clauses, then with high probability either η_1 or η_2 is not "too large" – i.e., either $\eta_1 \leq \eta + s/4$ or $\eta_2 \leq \eta + s/4$. Thus $\eta' = \min\{\eta_1, \eta_2\} \leq \eta + s/4$.

PROOF: There are two cases.

CASE: There is a clause C in I that is also in ϕ_*. Then

$$\frac{p_{0+}(C)}{p_0(C)} = \eta$$

by Eq. (2). By Lemma 8, $LE(\eta, mQ_I/2, \eta + s_b/4) \leq \delta/6$. Thus $\eta_2 \leq \eta + s/4$ with high probability.

CASE: There is no clause C in I that is also in ϕ_*. η_1 estimates p_-, and by Eq. (4) p_- depends on $p_0(\phi^*)$. We can bound the latter as follows:

$$\begin{aligned} p_0(\phi^*) &\leq \sum_{C \in \phi^*} p_0(C), \\ &< \sum_{C \in \phi^*} Q_I, \text{ since no clause in } \phi_* \text{ is in } I \\ &\leq MQ_I, \\ &= Q_H/8. \end{aligned}$$

Thus $p_- < \eta + Q_H(1 - 2\eta)/8 = \eta + s/8$, and by Lemma 8, $GE(\eta + s/8, m, \eta + s/4) \leq GE(\eta + s/8, m, \eta + s/8 + s_b/8) \leq \delta/6$. Hence $\eta_1 \leq \eta + s/4$ with high probability. ■

Lemma 6 Given that I contains all important clauses, with probability $> 1 - \delta/2$, η' is "close to" η – i.e., $|\eta' - \eta| \leq s/4$.

PROOF: Immediate from Lemmas 3–5. ■

Lemma 7 Given that I contains all important clauses and that $|\eta' - \eta| \leq s/4$, with high probability no harmful clause will be included in the output ϕ of V'. And with high probability, no important clause will be omitted.

PROOF: A harmful clause C is included if

$$h(C) \leq \eta' + s_b/2,$$

and given that $\eta' \leq \eta + s/4$, it certainly must be the case that

$$h(C) \leq \eta + 3s/4$$

if the clause is to be included.

But for such a clause $h(C)$ has an expected value of at least $\eta + s$. To be included, it must therefore deviate from its expected value by at least $s/4$, in a sample of at least $mQ_I/2$ positive examples. And $LE(\eta+s, mQ_I/2, \eta+3s/4) \leq LE(\eta + s, mQ_I/2, \eta + 3s_b/4) \leq \delta/6M$. Summing this probability over possibly M harmful clauses yields the first result.

For a correct clause C, $p_{0+}(c)/p_0(c) = \eta$, and it will be discarded only if the empirical value $h(C)$ of this ratio exceeds η by more than $s_b/4$. Lemma 8 shows that this probability is $\leq \delta/6M$. Summing over possibly M correct clauses yields the result. ■

Taken together, these lemmas conclude the proof of Theorem 5. Note that one need not assume that V' is given an upper bound η_b. The algorithm can estimate such an upper bound efficiently, using a version of the method in Section 2.3; Laird (1987) provides details. Also note that the algorithm V' uses both positive and negative examples; Kearns and Li (1987) have shown that both kinds of examples are necessary in this setting.

4. Random noise processes

So far we have considered errors resulting from a *Classification Noise Process* (CNP):

> Independently for each example, the sign s of the example $\langle x, s\rangle$ drawn from $EX_0(\ )$ is reversed with probability η.

We have seen that the CNP preserves *pac*-identifiability, provided $\eta < 1/2$. The CNP is just one example of a *random noise process*, in which with some fixed probability η each example is independently given to the noise process for possible modification before presentation to the learning algorithm.

How do the above algorithms and ideas change when some other noise process is at work? Typically we find that the same basic idea – choosing a rule that minimizes disagreements with the data – is effective. But the amount of data required for *pac*-identification may be different, and the maximum tolerable rate of noise will vary.

To illustrate, consider the "worst case" *Adversarial Noise Process* (ANP):

> Independently for each example, the example is replaced, with probability η, by an arbitrary example, perhaps maliciously chosen.

In selecting the replacement, the ANP adversary may have knowledge of the target L_*, the entire past history of the run, and all parameters (ϵ, δ, η, η_b, etc.), but it has the chance to do so only a fraction η of the time.

For the hapless customs official still trying to identify smugglers, an adversarial band of smugglers will sometimes intentionally pass through customs without carrying contraband, or will plant contraband among the possessions of a non-smuggler, with the fiendish purpose of confusing the official during his or her training period. The question, then, is how much more difficult the learner's task becomes in such cases.

Our approach is to distinguish by sampling the correct hypothesis L_* from an ϵ-bad hypothesis L_ϵ. We require that the expected rate of disagreement with the sample for L_* be smaller than for L_ϵ. With the ANP the expected rate of disagreement for L_* is at most η, while the expected rate of disagreement for L_ϵ is at least $\epsilon(1-\eta)$. Thus, provided $\eta < \epsilon(1-\eta)$, or equivalently, $\eta < \epsilon/(1+\epsilon)$, L_* and L_ϵ will be statistically distinguishable. Kearns and Li (1987) have shown this bound is tight. (Compare $\eta < 1/2$ for the CNP.)

As another example, consider the problem of identifying $CNF(n,k)$ rules from positive examples that are subject to adversarial noise – a problem first solved by Valiant (1985). Using the ideas developed in this paper, we give a simpler analysis.

Let M be the number of clauses. For a given target formula ϕ_* the examples oracle $EX_0(\,)$ selects satisfying assignments of ϕ_* from some distribution D^+. Before presentation, the example is subjected to an ANP that, with probability η may replace the positive example by another assignment. Clauses in ϕ_* can thus be falsified on average by at most a fraction η of the examples. By contrast, we denounce as *harmful* any clause C for which $p_{01}(C) \geq \epsilon/M$, as measured by the distribution D^+. Despite the best efforts of an adversary, a harmful clause must fail at a rate of at least $\epsilon(1-\eta)/M$. Our approach is to eliminate all harmful clauses while including all correct ones. Clearly the error in the resulting formula will then be at most ϵ.

Provided $\eta < \epsilon(1-\eta)/M$ (or, equivalently, $\eta < [(M/\epsilon)+1]^{-1}$), there is a separation of at least $s_b = \epsilon/M - \eta_b(1+\epsilon/M)$ between the rate at which a harmful clause is falsified and this rate for a correct clause. Also, harmful clauses are falsified on average by a proportion of at least $\epsilon(1-\eta_b)/M$ of the examples. Therefore we use the following algorithm, V'':

1. Obtain a sample of $m = (2/{s_b}^2)\ln(M/\delta)$ positive examples.
2. Output the conjunction of all clauses falsified by no more than $(\eta_b + s_b/2)m$ examples.

To see that this works, consider first *errors of omission* (discarding a correct clause). For this to occur, the proportion of examples falsifying a correct clause must exceed its expected value (η) by at least $s_b/2$; but m has been chosen so that (Lemma 8) the likelihood of this is at most δ/M.

Similarly, *errors of commission* (including a harmful clause) occur only when a harmful clause is falsified less than expected, by a deviation of at least $s_b/2$. The chances of this are less than δ/M. Summing the probabilities of both types of errors over at most M clauses gives a probability for error of at most δ. Thus we have shown the following theorem.

Theorem 6 There is an algorithm that runs in time polynomial in $1/\epsilon$, $\log 1/\delta$, n^k, and $1/(1-2\eta_b)$ and *pac*-identifies $CNF(n,k)$ formulas from positive examples subject to adversarial noise, provided the rate η of noise satisfies

$$\eta \leq \eta_b < \frac{1}{(M/\epsilon)+1},$$

where M is the number of clauses. ■

We direct the reader to Laird (1987) for further results on *pac*-identification with other noise processes.

5. Remarks

Summarizing, the basic idea of this paper is that algorithms for *pac*-identification can often be generalized to handle a certain amount of random noise in the data. A feasible increase in the amount of data suffices to separate acceptable rules from ones with too much error, provided the rate of noise is within certain bounds that depend on the noise process. As with *pac*-identification from noise-free data, a direct search for the best rule may not be computationally tractable for many domains of interest, but specially chosen algorithms may be found for some domains, as we illustrated for $CNF(n,k)$.

A major open question is whether there exists any domain in which *pac*-identification is computationally feasible with no noise but computationally infeasible with some "reasonable" level of noise. It would be interesting to explore the effect of noise in a situation that calls for queries as well as random sampling. For example, could Angluin's (1987) polynomial-time procedure for identifying regular sets given a sampling oracle and membership queries be modified to compensate for random errors in the sampling and query responses? Other interesting directions include models of non-random noise and problems of approximate identification when none of the rules in the space are exactly equivalent to the rule being presented – a circumstance that somewhat resembles noisy data for a correct rule.

Acknowledgements

This paper is based on material from the second author's dissertation in the Computer Science Department at Yale University. We gratefully acknowledge the support of the National Science Foundation through grant number IRI-8404226.

We also thank David Haussler for his remarks on an earlier draft and for bringing Vapnik's (1982) work to our attention, and the referee for his thoughtful criticism, which improved the paper considerably.

References

Angluin, D. (1987). Learning regular sets from queries and counterexamples. *Information and Computation*, *75*, 87–106.

Blumer, A., Ehrenfeucht, A., Haussler, D., & Warmuth, M. (1986). Classifying learnable geometric concepts with the Vapnik-Chervonenkis dimension. *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing* (pp. 273–282). Berkeley, CA: The Association for Computing Machinery.

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

Kearns, M., & Li, M. (1987). *Learning in the presence of malicious errors* (Technical Report TR-03-87). Cambridge, MA: Harvard University, Center for Research in Computing Technology.

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

Laird, P. (1987). *Learning from good data and bad.* Doctoral dissertation, Department of Computer Science, Yale University, New Haven, CT.

Quinlan, J. R. (1986). The effect of noise on concept learning. In R. S. Michalski, J. G. Carbonell, & T. M. Mitchell (Eds.), *Machine learning* (Vol. 2). Los Altos, CA: Morgan Kaufmann.

Schlimmer, J. C., & Granger, R. H. (1986). Incremental learning from noisy data. *Machine Learning*, *1*, 317–354.

Shackelford, G. G., & Volper, D. J. (1987). *Learning in the presence of noise*. Unpublished manuscript. University of California, Department of Information and Computer Science, Irvine.

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

Valiant, L. G. (1985). Learning disjunctions of conjunctions. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence* (pp. 560–566). Los Angeles, CA: Morgan Kaufmann.

Vapnik, V. N. (1982). *Estimation of dependencies based on empirical data*. New York: Springer-Verlag.

Wilkins, D. C., & Buchanan, B. G. (1986). On debugging rule sets when reasoning under uncertainty. *Proceedings of the Fifth National Conference on Artificial Intelligence* (pp. 448–454). Philadelphia, PA: Morgan Kaufmann.

Appendix: Bounding lemmas

We establish some simple tools for bounding the accuracy of estimates of Bernoulli variables. For p and r between 0 and 1 and any positive integer m, let $LE(p, m, r)$ denote the probability of at most rm successes in m independent trials of a Bernoulli variable with probability of success p, and $GE(p, m, r)$ the probability of at least rm successes. Thus,

$$GE(p, m, r) = \sum_{k=\lceil rm \rceil}^{m} \binom{m}{k} p^k (1-p)^{m-k},$$

and

$$LE(p, m, r) = \sum_{k=0}^{\lfloor rm \rfloor} \binom{m}{k} p^k (1-p)^{m-k}.$$

It is not difficult to show that for p increasing, $GE(p, m, r)$ is nondecreasing and $LE(p, m, r)$ is nonincreasing. We extend LE to have the value 0 if its third argument is less than 0, and similarly GE has the value 0 if its third argument is greater than 1.

The basic lemma we use is Hoeffding's Inequality (Hoeffding, 1963).

Lemma 8 If $0 \leq p \leq 1, 0 \leq s \leq 1$, and m is any positive integer then

$$LE(p, m, p - s) \leq e^{-2s^2 m},$$

and

$$GE(p, m, p + s) \leq e^{-2s^2 m}.$$

We apply this to obtain a simple bound on the number of samples required to assure that an estimate of p is within a distance s of the correct value with probability at least $1 - \delta$.

Lemma 9 Let $0 \leq p \leq 1$, $0 < s < 1$, and $0 < \delta < 1$. If

$$m \geq \frac{1}{2s^2} \ln\left(\frac{1}{\delta}\right)$$

then

$$LE(p, m, p - s) \leq \delta,$$

and

$$GE(p, m, p + s) \leq \delta.$$

PROOF: This follows directly from Lemma 8 by setting $e^{-2s^2 m} \leq \delta$ and solving for m. ■

Among the various bounds in this paper derived from the basic lemma is the following:

Lemma 10 Let N be a positive integer, $0 < \epsilon < 1$, $0 < \delta < 1$, and $0 \leq \eta \leq \eta_b < 1/2$. Define $s = \epsilon(1 - 2\eta_b)$ so that $0 < s < 1$. If

$$m \geq \frac{2}{\epsilon^2(1 - 2\eta_b)^2} \ln\left(\frac{2N}{\delta}\right),$$

then

$$GE(\eta, m, \eta + s/2) \leq \delta/2N,$$

and

$$LE(\eta + s, m, \eta + s/2) \leq \delta/2N.$$

PROOF: We apply Lemma 9 with $s/2$ in place of s and $\delta/2N$ in place of δ to find the indicated lower bound on m. ■