In *Proceedings of the Ninth International Conference on Machine Learning* (1992). Aberdeen: Morgan Kaufmann.

Induction of One-Level Decision Trees

Wayne Iba*
AI Research Branch
Mail Stop 269–2
NASA Ames Research Center
Moffett Field, CA 94035

Pat Langley
AI Research Branch
Mail Stop 269–2
NASA Ames Research Center
Moffett Field, CA 94035

Abstract

In recent years, researchers have made considerable progress on the worst-case analysis of inductive learning tasks, but for theoretical results to have impact on practice, they must deal with the average case. In this paper we present an average-case analysis of a simple algorithm that induces one-level decision trees for concepts defined by a single relevant attribute. Given knowledge about the number of training instances, the number of irrelevant attributes, the amount of class and attribute noise, and the class and attribute distributions, we derive the expected classification accuracy over the entire instance space. We then examine the predictions of this analysis for different settings of these domain parameters, comparing them to experimental results to check our reasoning.

1 INTRODUCTION

In recent years, machine learning has made considerable progress in both the theoretical analysis of learning tasks (e.g., Kearns, Li, Pitt, & Valiant, 1987; Haussler, 1990) and in the experimental evaluation of specific algorithms (Kibler & Langley, 1988). However, most theoretical work has remained disconnected from practical algorithms, and the worst-case predictions of the PAC learning framework have been strikingly different from results obtained in experiments.

Recently, a few researchers have presented average-case formulations of particular algorithms. Pazzani and Sarrett (1991) analyzed a simple conjunctive learning method, whereas Hirschberg and Pazzani (1991) studied an algorithm for inducing k-CNF concepts. In these two studies, the authors used analyses of the algorithms' behavior under various conditions, along with information about the domain, to predict average-case learning curves. They also compared these predictions to the methods' actual behavior on the same domain.

We believe that this work constitutes an excellent example of analytical evaluation, and we hope that others will follow its approach. However, the results to date have been drawn from analyses of algorithms that make little contact with ones that are used in the practice of machine learning. In this paper, we follow a similar path with respect to a more relevant algorithm, first reporting our theoretical treatment and then its predictions to experimental results.

Our long-term goal is an average-case analysis of methods for decision-tree induction, but here we focus on a simpler algorithm that constructs one-level decision trees, or 'decision stumps'. Although these may seem trivial at first glance, they force one to address issues that arise in the induction of full decision trees, and we anticipate that many of the lessons learned will transfer to an average-case analysis of this more general problem. In addition, previous work in theoretical psychology has addressed the learning of "single attribute discriminations" in humans (Levine, 1966). Also, Holte (1991) reports experimental results suggesting that, in many domains, decision stumps are nearly as accurate as full decision trees. Thus, despite its simplicity, the algorithm has some potential as a practical induction method. Now let us turn to our analysis of the algorithm.

2 ANALYSIS OF THE ONE-LEVEL ALGORITHM

Consider a simple algorithm ONE-LEVEL that induces a one-level 'decision tree' from a set of preclassified training instances. In this scheme, one selects a single attribute for predicting class membership. To simplify matters, we consider only Boolean concepts where instances are represented as a set of Boolean attributes; in particular, we focus on concepts consisting of a single relevant attribute A°, and q irrelevant attributes

*Also affiliated with RECOM Technologies

$A_1, \ldots, A_q$. For the purposes of our analysis, we define an attribute A_i to be *irrelevant* if the logical description of the target concept (in this case A°) does not contain A_i. We also assume that the irrelevant attributes are independently drawn from the same probability distribution $P(A_i)$.

For each attribute A, ONE-LEVEL computes a score measuring how well A separates the classes. Since the concept and attributes are Boolean, we can count the number of times, over a training set of size n, an attribute and the concept have the same values (both true or both false) and the number of times they have different values (one true and the other false). If we loosely refer to these counts as $|A \equiv C|$ and $|A \not\equiv C|$, respectively, the score is computed by the expression

$$score(A) = \frac{max(|A \equiv C|,\ |A \not\equiv C|)}{n} \quad ,$$

and has the range $1/2 \leq score(A) \leq 1$. Mingers (1989) presents an excellent review of measures that have been used in inducing decision trees, including Quinlan's (1986) original *information gain* metric. Although our function is considerably simpler than these measures, it gives the same ordering on attributes in domains that involve only Boolean attributes, and it is more amenable to analysis. As in more complex algorithms, ONE-LEVEL prefers the attribute with the best score. In case of ties, the algorithm randomly selects one of the best-scoring attributes.

The goal of our analysis is to predict $P(R)_n$, the probability that the induced 'decision stump' will make a correct classification on a test instance after n training instances. We will consider the effects of four factors: the number of irrelevant attributes; the amount of class and attribute noise; the class and attribute distributions (frequencies); and the number of training instances observed. To make our analysis tractable, we will also assume the above evaluation function for measuring the discriminating power of each attribute.

2.1 THE NUMBER OF IRRELEVANT ATTRIBUTES

We begin our analysis by examining how the number of irrelevant attributes influences the probability of selecting the relevant one. Suppose we present ONE-LEVEL with training data from a domain in which there is one relevant attribute A° and q irrelevant attributes $A_1, \ldots, A_q$.

We want to determine the probability that, over a training set of n instances, exactly i of the q irrelevant attributes will distinguish the class label as well as the relevant attribute *and* the remaining $q - i$ irrelevant attributes score worse; we use $P(\Upsilon_i)_n$ to denote the probability of this event. Let $\hat{x}$ be the observed score of the relevant attribute A° on the n training instances, and let $\hat{y}$ be the analogous score for a particular irrelevant attribute.[1] In the noise-free case, $\hat{x} = 1$ and $P(\Upsilon)$ therefore depends on the likelihood of an irrelevant attribute perfectly partitioning the training set into positive and negative instances. Given q irrelevant attributes, there are many ways in which exactly i of the q irrelevant attributes will score as well as the relevant attribute A°, and in which each of the remaining $q - i$ attributes scores worse than A°. If we assume that all irrelevant terms follow a *product* distribution (i.e., they are sampled from the same probability distribution), we can compute the probability of this event as

$$P(\Upsilon_i)_n = \binom{q}{i} P(\hat{y} = \hat{x})^i P(\hat{y} < \hat{x})^{q-i} \quad , \qquad (1)$$

where $P(\hat{y} < \hat{x}) = 1 - P(\hat{y} = \hat{x})$ for the noise-free condition. This expression is analogous to the binomial distribution obeyed by a sequence of flips with a biased coin.

Recall that, given two or more attributes with equal scores, the ONE-LEVEL algorithm selects one of these at random and uses this feature in classifying test instances. With this strategy, the probability that the single relevant attribute A° will be selected after exactly n training instances is

$$\mathcal{S}(A^\circ)_n = \sum_{i=0}^{q} \frac{1}{i+1} P(\Upsilon_i)_n \quad . \qquad (2)$$

This expression incorporates the case in which the relevant term wins outright ($i = 0$) and the situation in which it ties with one or more irrelevant terms but is selected anyway ($i > 0$).

2.2 NOISE AND FREQUENCY

An issue central to the above analysis was the fact that $\hat{x}$, the score of the relevant attribute, was always equal to 1. This will not be the case in the presence of noise. Because ONE-LEVEL uses the evaluation function *score* to select the attribute on which to base its predictions, we would like to know the expected $score(A_i)$ for a given attribute. For this we must calculate $P(A_i \equiv C)$, the expected probability that attribute A_i has the same value as the class label C. For an irrelevant attribute A_i, this probability is

$$P(A_i \equiv C) = P(C)P(A_i) + P(\bar{C})P(\bar{A}_i) \quad ,$$

where $P(C)$ and $P(A_i)$ are the probabilities of a positive instance and a positive value for an irrelevant attribute, respectively, and where $P(\bar{C}) = 1 - P(C)$ and $P(\bar{A}_i) = 1 - P(A_i)$. However, since the probability for the relevant attribute A° is *not* independent of the class label, we must handle it separately; if there is no noise in the training data, we have $P(A^\circ \equiv C) = 1$.

[1] In this analysis, most values of interest are dependent on the number of training instances, n, but we will omit the subscript for $\hat{x}$ and $\hat{y}$ in order to reduce clutter.

Noise in the training instances modifies the expected scores for both relevant and irrelevant attributes. Let z be the level of class noise – the probability that the actual value of the class attribute will be replaced with the opposite value. Similarly, let w be the level of attribute noise – the probability that the actual value of a particular attribute (relevant or irrelevant) will be replaced with its opposite.[2] We use $P(B)$ to denote the probability of some event B before noise has been added and $P'(B)$ to denote the probability after noise has been inserted.

Thus, to determine the expected score for an irrelevant attribute A_i in a noisy domain, we must compute

$$P'(A_i \equiv C) = P'(C)P'(A_i) + P'(\bar{C})P'(\bar{A}_i) \quad .$$

Using our definitions of class and attribute noise, we can express the post-noise probability of C as

$$\begin{aligned} P'(C) &= (1-z)P(C) + z(P(\bar{C})) \\ &= P(C)[1-2z] + z \end{aligned}$$

and the post-noise probability of A_i as

$$\begin{aligned} P'(A_i) &= (1-w)P(A_i) + w(P(\bar{A}_i)) \\ &= P(A_i)[1-2w] + w \quad . \end{aligned}$$

Note that these expressions include both the case in which the attribute was actually true and noise has not corrupted this value, and the case in which it was actually false and noise has replaced it with true as the observed value.

In contrast, we know that, for the noise-free case, we have $P(A^\circ \equiv C) = 1$. Thus, the relevant attribute A° can have the same value as C in the presence of class noise z and attribute noise w only if neither or both of A° and C are corrupted by noise. In the presence of noise, we have

$$P'(A^\circ \equiv C) = (1-w)(1-z) + wz \quad .$$

Note that this probability is independent of the class frequency and depends only on the noise levels.

2.3 THE NUMBER OF TRAINING INSTANCES

Our goal in this endeavor was to predict the *estimated* score for a particular attribute, and we now nearly have the tools to accomplish this. Let us define the term $Eqv(A, n, m)$ as the probability that a given attribute A will have the same value as the class label on exactly m instances in a training set of size n. This probability is simply

$$Eqv(A, n, m) = P'(A \equiv C)^m P'(A \not\equiv C)^{n-m} \quad .$$

Recall that $\hat{x}$ denotes the estimated score for the relevant attribute based on a sample of n instances, and that $\hat{y}$ indicates the estimated score for irrelevant attribute A_i on the same instances. We can express the probability distributions for $\hat{x}$ and $\hat{y}$ using the binomial and the terms developed above, which gives

$$P(\hat{x} = \frac{k}{n}) = \binom{n}{k}[Eqv(A^\circ, n, k) + Eqv(A^\circ, n, n-k)]$$

and

$$P(\hat{y} = \frac{k}{n}) = \binom{n}{k}[Eqv(A_i, n, k) + Eqv(A_i, n, n-k)] \quad .$$

Note that these expressions give different values for different numbers of training instances.[3]

Now we are ready to generalize equation (1) from Section 2.1, which calculates the probability $P(\Upsilon_i)$, that exactly i irrelevant attributes will score the same as the relevant attribute A° and that the remaining $q-i$ irrelevant attributes will score worse than A°. In the noise-free case, there was only one possible score for A°, but now we must consider all possible scores for A°. Furthermore, with the presence of noise there is the possibility that an irrelevant attribute may actually score *better* than the relevant attribute; the following equations do not include the likelihood of this occurrence. For each possible score $\hat{x}$ for the relevant attribute, we must consider the probability that i irrelevant attributes score $\hat{y} = \hat{x}$ and that the remaining ones score $\hat{y} < \hat{x}$. This expands to

$$P'(\Upsilon_i)_n = \sum_{k=\lceil \frac{n}{2} \rceil}^{n} P(\hat{x} = \frac{k}{n}) \binom{q}{i} P(\hat{y} = \hat{x})^i P(\hat{y} < \hat{x})^{q-i} \quad ,$$

where

$$P(\hat{y} < \frac{m}{n}) = \sum_{j=\lceil \frac{n}{2} \rceil}^{m-1} P(\hat{y} = \frac{j}{n}) \quad .$$

By substituting $P'(\Upsilon_i)_n$ for $P(\Upsilon_i)_n$ in equation (2), we obtain a means for predicting the correctness of the ONE-LEVEL algorithm for different levels of class and attribute noise, for different numbers of irrelevant attributes, and for different numbers of training instances.

2.4 PREDICTIVE ACCURACY OF THE INDUCED TREE

If we hope to determine the predictive accuracy of decision stumps generated by the ONE-LEVEL algorithm,

[2] Our treatment of noise owes much to discussions with Michael Pazzani, who made a number of helpful suggestions. Quinlan (1986) uses an alternate definition in which the noise level equals the probability that a value will be replaced with one selected randomly from the set of possible values (including the original). Pazzani (personal communication, 1991) has revised our analysis to handle this formulation of noise.

[3] The equations as given here hold only for numbers $k > \frac{n}{2}$. When $k = \frac{n}{2}$ (i.e., the lowest possible score an attribute may have), only one $Eqv(A, n, k)$ term should be included in the expression.

we need more than the probability that it will select the relevant attribute. We also need to understand the accuracy that results when this occurs and when it does not.

Whether the attribute A that ONE-LEVEL selects is relevant or irrelevant, there are two possible ways that A can split the decision stump. In one case, the presence of the feature A indicates class membership (i.e., the presence of C); in the other case, the absence of A is associated with class membership. If one has selected the relevant attribute by associating A° with C and $\bar{A}^\circ$ with $\bar{C}$, which we denote with a subscript "+", the probability of correct classification $R_+(A^\circ) = 1$, provided one assumes that test cases are free of noise. Conversely, if one has selected A° with its absence predicting C, then the probability of correct classification $R_-(A^\circ) = 0$. To compute $R(A^\circ)$, the overall probability of correct classification for A°, we must multiply the probability of selecting A° in both associations by their respective accuracies. For the relevant attribute, this gives the expression

$$R(A^\circ) = (1) \sum_{k=\lceil \frac{n}{2} \rceil}^{n} \binom{n}{k} Eqv(A^\circ, n, k) \quad ,$$

since the term for $R_-(A^\circ)$ cancels to zero.

If instead one has selected the irrelevant attribute A_i and associated its presence with C, the probability of correct prediction is the probability that A_i and C are both either present or absent in the test instance, or

$$R_+(A_i) = P(C)P(A_i) + P(\bar{C})P(\bar{A}_i) \quad .$$

In contrast, if one has selected A_i and associated its absence with C, the probability of correct prediction is

$$R_-(A_i) = P(C)P(\bar{A}_i) + P(\bar{C})P(A_i) \quad .$$

Note that these are simply the noise-free probabilities that an irrelevant attribute and the class label will have the same value in any given instance. To compute the overall probability of correct classification when one has selected A_i, we must multiply these two terms by the probability of selecting A_i with the respective associations, which gives

$$\begin{aligned} R(A_i) \;=\; & R_+(A_i) \sum_{k=\lceil \frac{n}{2} \rceil}^{n} \binom{n}{k} Eqv(A_i, n, k) \\ & + R_-(A_i) \sum_{k=\lceil \frac{n}{2} \rceil}^{n} \binom{n}{k} Eqv(A_i, n, n-k) \quad . \end{aligned}$$

Finally, we can compute the overall probability of correctly classifying a given test case after n training instances, whether ONE-LEVEL has selected the relevant attribute or some irrelevant attribute. Using terms from the above analyses, we have

$$P(R)_n = R(A^\circ)\mathcal{S}(A^\circ)_n + R(A_i)[1 - \mathcal{S}(A^\circ)_n] \quad .$$

This expression describes the probability of correct prediction on a test instance using the 'decision stump' constructed by the ONE-LEVEL algorithm. From this equation, one can predict the effect on accuracy of the number of training instances, the amount of class and attribute noise, the class and attribute frequencies, and the number of irrelevant attributes. Thus, we have accomplished our original goal.

3 BEHAVIOR OF THE ONE-LEVEL ALGORITHM

Developing equations that relate domain characteristics to an algorithm's behavior is only the first step toward understanding. We are also interested in the practical implications of these equations for the algorithm, and in whether the behavior predicted by the equations corresponds to the algorithm's actual behavior. In this section we graphically depict the effects of the factors we considered in the analysis, including the number of training instances, irrelevant attributes, noise, and frequency.

3.1 THE EFFECTS OF TRAINING INSTANCES

The independent variable most frequently manipulated in machine learning papers is the number of training instances. A performance measure such as accuracy, when plotted as a function of this variable, produces a *learning curve*.[4] The primary characteristic of interest in learning curves is whether performance improves with the number of training instances.

Our analysis of the ONE-LEVEL algorithm shows that its probability of a correct prediction increases with this factor. Later in this section, we present this effect graphically for different numbers of irrelevant attributes, noise levels, and attribute frequencies. We also show that, for noise-free test instances, the asymptotic accuracy for the ONE-LEVEL algorithm is always perfect. In each of our graphs, we include both the predicted learning curves (shown as lines) and the actual accuracies (using 95% confidence intervals) obtained by running ONE-LEVEL in the specified domains. Each interval on the curves represents an average over 500 runs on randomly generated training instances, in which the accuracy of the resulting decision stump was measurered on a single set of 100 randomly generated, noise-free test instances. These experimental results correspond quite well with the learning curves predicted by the analysis, thus providing a check on our reasoning and supporting our claims about average-case behavior.

[4] This term is typically used in describing the learning behavior of incremental methods, but one can measure analogous effects for nonincremental techniques.

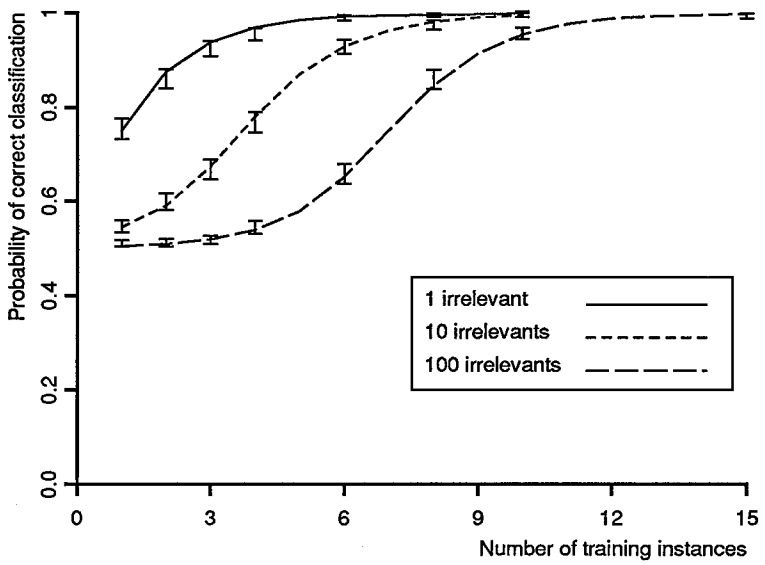

Figure 1: Three learning curves showing predicted (lines) and experimental (95% confidence intervals) results for ONE-LEVEL's accuracy as a function of the number of training instances for three different number of irrelevant attributes.

3.2 THE EFFECTS OF IRRELEVANT ATTRIBUTES

The analysis in Section 2.1 suggests that the number of irrelevant attributes in a domain will affect the ONE-LEVEL algorithm's learning curves. Specifically, the more irrelevant attributes that describe the instances, the lower the probability that the method will select the relevant attribute A° to predict the class name of the test instances. That is, as the number of irrelevant attributes q increases, so does the probability that a fixed number i of them will split the training instances as well as, or better than, the relevant attribute.

Figure 1 shows the predicted and observed learning curves for three levels of irrelevant attributes q when other domain parameters are held constant. In particular, these curves represent a noise-free domain where the class frequency is 50% and the attribute frequency is 50%.

The first result to note is the peculiar 'S' shape of the curves. Most learning curves previously reported in the literature immediately begin to improve and then level off. We believe this occurs because most inductive learning research has focused on domains with relatively few irrelevant attributes. In contrast, the 'S' shape arises from the disparity between the number of relevant and irrelevant attributes. As one increases the number of irrelevant features from one to 10 to 100, the 'S' shape becomes more and more pronounced.

As we noted above, the number of irrelevant attributes has no affect on the *level* of asymptotic accuracy; our intuition suggests that domains with more irrelevant features would require more instances to reach this asymptote. However, inspection of the curves in Figure 1 show a second interesting result – that the number of training instances required to reach a given level of accuracy increases only logarithmically with the number of irrelevant attributes. Littlestone (1988) has demonstrated a similar effect for another algorithm.

3.3 THE EFFECTS OF CLASS AND ATTRIBUTE NOISE

Like the number of irrelevant attributes, we expect that noise of various types will also have significant effects on classification accuracy. As we described in the analysis, the level of noise is simply the probability that a value will be reversed.

Here we focus primarily on class noise. In unreported experiments, we have observed that class noise and attribute noise have identical affects on ONE-LEVEL's learning rates. This should not be surprising, since our analysis assumes a single relevant attribute; a particular noise level in either the class label or each of the attributes has the same effect on $P(A^\circ \equiv C)$, the probability that the relevant attribute and the class label will have the same value. The change in $P(A^\circ \equiv C)$ is the primary effect of noise, and since there is only one relevant attribute, attribute noise changes this probability the same amount as class noise. Note that the conditional probability $P(C|A_i)$ remains $P(C)$ for each irrelevant attribute regardless of attribute noise. The influence of noise in these attributes is relatively minor, as we discuss shortly.

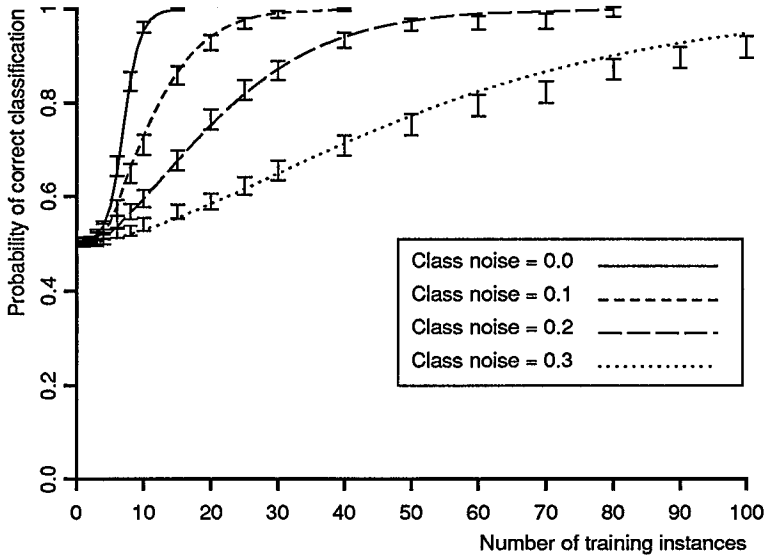

Figure 2: Predicted and experimental accuracy as a function of training set size for four levels of class noise.

Figure 2 shows the predicted and observed effects of training instances and class noise on classification accuracy and learning rate. As with irrelevant attributes, noise has no ultimate effect on the asymptotic accuracy. The algorithm converges on the perfect score for all levels of noise less than 50%.[5] Another interesting point to observe is that, unlike the number of irrelevant attributes, the noise level mainly affects the overall *rate* of improvement. That is, increasing the number of irrelevant attributes shifts the learning curves somewhat to the right, but increasing the noise level flattens or stretches the S shape. In summary, ONE-LEVEL is robust with respect to class and attribute noise, but its behavior is more seriously altered by this factor than by the number of irrelevant features.

3.4 THE EFFECTS OF CLASS AND ATTRIBUTE FREQUENCY

In our analysis of Section 2.2, we showed that the class frequency $P(C)$ and the frequency of the irrelevant attributes $P(A_i)$ directly determine $P(A_i \equiv C)$, the probability that an attribute and the class will have the same value.[6] There are two places in the general analysis where this probability is important. The first involves selecting the attribute used to split the training instances and to predict future test instances. When $P(A_i \equiv C)$ is close to either one or zero, then ONE-LEVEL is more likely to select the irrelevant attribute as the best discriminator for a given set of instances. Therefore, it will need more instances to discover the independence of the class and irrelevant attributes. Thus, skewed frequency distributions for the class and irrelevant attributes tend to increase the difficulty of selecting the relevant attribute.

The second place in which $P(A_i \equiv C)$ is important concerns predicting the class label of a test instance. Even an irrelevant attribute is reasonably good at predicting the class when the label and attribute values are usually the same (or different). That is, independent of the number of training instances, the further $P(A_i \equiv C)$ is from 0.5, the greater the probability $R(A_i)$ that a correct prediction will be made if ONE-LEVEL has selected an irrelevant attribute.

Figure 3 shows the influence of the attribute frequencies on the learning curves for the algorithm. In this case, we assumed ten irrelevant attributes, no noise, and a class frequency of 10%. These curves take into account both the greater difficulty in selecting the relevant attribute and the increased accuracy inherent in a skewed frequency distribution. Note how the curves cross each other; the skewed frequency condition starts with the better accuracy but takes longer to reach asymptote. Conversely, the balanced frequency case starts off lower but quickly discovers the relevant attribute and reaches asymptote before the other.

Given these insights about the effects of frequency, let us return to the results characterizing the effect of noise on predictive accuracy. As we saw above, introducing noise (of either type) has two main effects.

[5] For this noise level, we would expect the algorithm to perform at chance (50% accuracy), and for higher levels, we would expect it to converge on the opposite concept (0% accuracy).

[6] We ignore the frequency of the relevant attribute because, prior to the introduction of noise, it is identical to the class frequency.

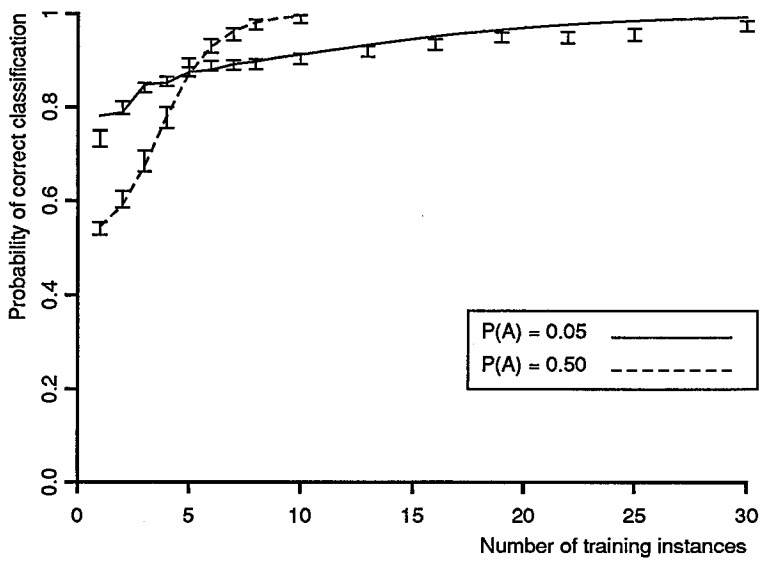

Figure 3: Learning curves showing the effects of the frequency distributions for irrelevant attributes.

First, the conditional probability $P(C|A^\circ)$ is no longer one but is reduced according to the level of noise. Second, the frequency distributions of the class (in the case of class noise) and irrelevant attributes (in the case of attribute noise) are moved closer to 50%. Earlier we showed that the first effect increased the number of training instances required to reach asymptote, and here we see that the second effect makes the selection task easier, thus reducing the number of instances to asymptote. However, our results on noise indicate that the first factor dominates the second, so that noise slows down learning overall.

In summary, the effect of frequency differs from that of irrelevant attributes and noise in that it involves an inherent tradeoff. Skewed frequencies lead to high accuracies early in training but take longer to reach asymptote, whereas balanced frequencies produce lower early accuracies but reach the asymptote with less experience. The noise level also impacts frequency, indirectly reducing the negative effects of noise but not eliminating them.

4 DISCUSSION

We are certainly not the first to carry out a theoretical analysis of inductive learning. Research within the PAC paradigm has produced a wide range of results (e.g., Kearns et al., 1987; Haussler, 1990). In many cases, these results take the form of determining, for a given class of concepts, the number of training instances required to induce a concept having accuracy $1 - \epsilon$ with probability $1 - \delta$. However, as Haussler has noted, such analyses usually predict much slower learning rates than observed in experimental studies of induction. This is not surprising, given that the PAC approach aims for worst-case bounds that are independent of the distribution of the training instances.

Recent research on average-case analyses has influenced our work to a much greater extent. In particular, Pazzani and Sarrett (1990) have reported such an analysis for an incremental conjunctive learning algorithm, whereas Hirschberg and Pazzani (1991) have presented an analogous study of inducing k-CNF concepts. As in our work, these theoretical analyses incorporated knowledge of the target concept and distributions of the attributes. They have also explored the effects of distributions and irrelevant attributes on the average-case behavior of the algorithms, comparing predicted and observed learning curves, as we have done. However, their treatments have not dealt with the effects of noise, as ours has done.

Another difference from earlier work, including the two average-case studies mentioned above, is our focus on algorithms used by members of the machine learning community who are interested in experimentation and applications. Some of the most popular methods of this sort induce decision trees (Quinlan, 1986) from preclassified training instances, but an analysis of the general case is beyond the scope of this paper. Interestingly, Holte (1991) has recently reported experiments with an algorithm that induces one-level decision trees, obtaining results nearly as accurate as full decision-tree algorithms on many of the data sets commonly used in studies of supervised learning. These results suggest that, despite its simplicity, an algorithm for constructing decision stumps and its average-case analysis can reveal interesting characteristics of the familiar data sets. Also, Holte argues that there are

advantages to studying the behavior of simpler algorithms before turning to complex ones.

The present work provides some important steps in this direction, but it makes a number of assumptions that should be remedied in the future. First, it posits that only one of many attributes is relevant to predicting the class, but some domains may contain redundant features, any one of which can predict the class equally well. This should increase the rate of ONE-LEVEL's learning, but only careful analysis will reveal the exact effect. Second, our treatment assumes that the target concept actually contains only one attribute, whereas it may involve a conjunction or disjunction of many attributes. Clearly, the current algorithm could never achieve perfect accuracy in such a domain, but future work should examine the details of its behavior under such conditions. In addition, we have assumed that the distributions of the irrelevant attributes are independent. We can carry out experiments to determine the degree to which nonindependence causes ONE-LEVEL's behavior to diverge from that predicted by our equations, as Pazzani (personal communication, 1991) has done. Finally, future research should explore behavioral interactions among the various domain characteristics, rather than focusing on individual aspects like noise and irrelevant attributes.

In the longer term, we intend to extend our average-case analysis to handle the induction of full decision trees. We believe that many of the expressions we have derived carry over directly to the more general case. These equations should apply recursively to each level of the decision tree, computing the probability that one has selected a relevant attribute and determining the distribution of training instances that are sorted to the next level. We must also deal with possible overfitting of the training data, since the trees may include more features than the target concept. This presents the greater challenge for extending our analysis.

Despite the simplicity of the ONE-LEVEL algorithm, we also believe that many of its basic behaviors will carry over to complete decision-tree algorithms. Thus, we anticipate that average-case analysis of such methods will predict that asymptotic accuracy is perfect, that irrelevant attributes affect the instances to asymptote only logarithmically, and that class noise affects this measure more significantly. On the other hand, the presence of multiple relevant attributes suggests that attribute and class noise will not be symmetrical, and Quinlan (1986) has presented some experimental evidence to this effect. Undoubtedly, other behavioral differences will also emerge, but we feel that our experience with the induction of decision stumps will stand us in good stead when we address more complex algorithms.

Acknowledgements

The authors wish to thank Michael Pazzani for discussions that helped refine many of the ideas in this paper. We also thank Jeff Schlimmer, Peter Andreae, Phil Laird, and two anonymous reviewers for their helpful comments on the paper.

References

Haussler, D. (1990). Probably approximately correct learning. *Proceedings of the Eighth National Conference on Artificial Intelligence* (pp. 1101–1108). Boston, MA: AAAI Press.

Hirschberg, D. S., & Pazzani, M. J. (1991). *Average-case analysis of a k-CNF learning algorithm* (Technical Report 91-50). Irvine: University of California, Department of Information & Computer Science.

Holte, R. C. (1991). *Very simple classification rules perform well on most data sets* (Technical Report). Ottawa, Canada: University of Ottawa, Computer Science Department.

Kearns, M., Li, M., Pitt, L., & Valiant, L. G. (1987). Recent results on Boolean concept learning. *Proceedings of the Fourth International Workshop on Machine Learning* (pp. 337–352). Irvine, CA: Morgan Kaufmann.

Kibler, D., & Langley, P. (1988). Machine learning as an experimental science. *Proceedings of the Third European Working Session on Learning* (pp. 81–92). Glasgow: Pittman.

Littlestone, N. (1988). Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning, 2*, 285-318.

Levine, M. (1966). Hypothesis behavior by humans during discrimination learning. *Journal of Experimental Psychology, 71*, 331-338.

Mingers, J. (1989). An empirical comparison of selection measures for decision-tree induction. *Machine Learning, 3*, 319–342.

Pazzani, M. J., & Sarrett, W. (1990). Average-case analysis of conjunctive learning algorithms. *Proceedings of the Seventh International Conference on Machine Learning* (pp. 339–347). Austin, TX: Morgan Kaufmann.

Quinlan, J. R. (1986a). Induction of decision trees. *Machine Learning, 1*, 81–106.