Tractable Average-Case Analysis of Naive Bayesian Classifiers

Pat Langley (langley@isle.org)
Stephanie Sage (sage@isle.org)
Institute for the Study of Learning and Expertise
2164 Staunton Court, Palo Alto, CA 94306 USA

Abstract

In this paper we present an average-case analysis of the naive Bayesian classifier, a simple induction algorithm that performs well in many domains. Our analysis assumes a monotone 'M of N' target concept and training data that consists of independent Boolean attributes. The analysis supposes a known target concept and distribution of instances, but includes parameters for the number of training cases, the number of irrelevant, relevant, and necessary attributes, the probability of each attribute, and the amount of class noise. Our approach differs from previous average-case analyses by introducing approximations to achieve computational tractability. This lets us explore the behavioral implications of the model for larger training and attribute sets than the earlier exact analyses, and experimental studies show that the analysis makes very accurate predictions despite its use of approximations. In closing, we suggest promising directions for future research on the average-case analysis of induction.

Keywords: average-case analysis, naive Bayesian classifier, probabilistic induction

Contact author: Pat Langley, langley@isle.org, (650) 494-3884

Electronic version: PostScript

1. Introduction and Motivation

Most theoretical analyses of machine learning focus on worst-case results, including those in the 'probably approximately correct' framework (Haussler, 1990). Although this lets analysts obtain quite general, distribution-free results, it also means their predictions of learning rate are much slower than those observed in practice. As a result, the link between theory and experiment in machine learning has become tenuous.

An alternative approach involves the *average-case* analysis of specific induction algorithms on domains with known characteristics. Pazzani and Sarrett (1992) report early results of this sort for a conjunctive learning method, and similar studies have been done for decision stumps (Iba & Langley, 1992), the naive Bayesian classifier (Langley, Iba, & Thompson, 1992), 1-nearest neighbor (Langley & Iba, 1992), and k-nearest neighbor (Okamoto & Nobuhiro, 1997). Each analysis produced predictions about the effect of domain characteristics, averaged over different training sets, that fit experimental data very closely.

However, this theoretical accuracy came with a price. For even simple methods like naive Bayes and nearest neighbor, the calculations needed to predict behavior could take drastically longer than actually running experiments with synthetic data, even when the latter averaged over many training sets. The difficulty resulted from the analyses' reliance on the exact calculation of probabilities for all possible combinations of events. For many induction methods, the number of such events grows exponentially with the number of attributes and size of the training set. This meant that theoretical predictions were only possible for small domains and early parts of the learning curve.

In this paper, we present a more tractable approach to the average-case analysis of induction algorithms. We demonstrate the framework with a new treatment of the naive Bayesian classifier, both because of the growing interest with this simple yet powerful method (e.g., Domingos & Pazzani, 1997) and because our earlier analysis was especially problematic in computational terms. The new analysis uses many of the same techniques as the previous one, but it introduces approximations based on the normal distribution that let us calculate means and variances for the sums and differences of quantities, rather

than about the probabilities of their explicit combinations.[1] The result is a tractable average-case analysis of naive Bayes that, as experimental studies reveal, remains accurate despite its use of approximations.

2. Review of the Naive Bayesian Classifier

Although it has a long history in pattern recognition (Duda & Hart, 1973), the naive Bayesian classifier first appeared in the machine learning literature as a straw man against which to compare more sophisticated methods (e.g., Cestnik, Konenenko, & Bratko, 1987). Only gradually did researchers become aware of its potential, but now it is widely recognized as a viable and robust approach to supervised induction.

Before beginning the analysis, we should first review the manner in which naive Bayes operates. To classify a new instance I, which is simply a conjunction of attribute values, the method calculates the probability of each class given the instance using

$$P(C_i|I) = \frac{P(C_i)P(I|C_i)}{P(I)} = \frac{P(C_i)\prod_{atts} P(v_j|C_i)}{\sum_{classes} P(C_k)\prod_{atts} P(v_j|C_k)} \ ,$$

where $P(C_i)$ is the probability of class C_i and $P(v_j|C_i)$ is the conditional probability for each attribute value v_j given the class C_i. The product of conditional probabilities comes from the assumption that attributes are independent given the class, which greatly simplifies the computation of the class scores and eases the induction process. After calculating $P(C_i|I)$ for each class, the algorithm assigns the instance to the class with the highest overall score or probability.

Although the above formulation of naive Bayes is the traditional one, we can express the score for each class in another form that is more tractable for analytical purposes. The basic idea is that, if we are concerned only with predictive accuracy, we can invoke any monotonic transformation that does not affect the *ordering* on class scores. One such transformation involves removing

1. Golea and Marchand (1993) report an average-case analysis of perceptron learning that also incorporates normal approximations, but their work includes other ideas from statistical mechanics that make it inaccessible to the machine learning community.

the denominator, which is the same for each class, and another involves taking the logarithm of the numerator. Together, these produce a new score

$$\mathcal{S}_C = logP(C) + \sum_{atts} logP(v_j|C) \ .$$

In fact, this form is often used in practice, since it is efficient to calculate and reduces round-off errors due to small fractions. The new score $\mathcal{S}_C$ is no longer a probability, but is quite sufficient if one only wants to predict the most probable class.

We should also discuss the estimation of probabilities in this expression. The typical implementation of naive Bayes stores an overall count n, one count k_C for each class, and one count c_j for each attribute value given the class. During training, the induction algorithm increments the overall count on each training instance, increments the class count for each example of that class, and increments the class-specific count for each attribute value that appears in the instance. During performance, the classifier uses these counts to estimate probabilities, giving

$$\mathcal{S}_C = log\left(\frac{k_C}{n}\right) + \sum_{atts} log\left(\frac{c_j}{k_C}\right) .$$

However, if the algorithm has not encountered a particular class or an attribute value for some class during learning, a slight problem arises because a count is zero. A common response is to initialize counts to 1 for each value, which we will also assume in our analysis. Moreover, since we assume Boolean attributes, we will set the initial count for each class to 2, and since we assume two classes, we will use 4 as the initial total count. This results in prior probabilities of $\frac{1}{2}$ for both classes and for each value given a class.

These decisions about how to initialize counts require a slight revision of the score for each class. Moreover, our assumption that attributes are Boolean lets us store only one count c_j for each attribute, for when it is present in an instance (i.e., has value TRUE). When the attribute is absent from an instance (i.e., has value FALSE), we can simply use $k_C - c_j$ as the count. We can reflect this in the score by separating the contribution for features that are present

and absent, giving

$$\mathcal{S}_C = log\left(\frac{k_C + 2}{n + 4}\right) + \sum_{atts} log\left(\frac{c_j + 1}{k_C + 2}\right) + \sum_{\overline{atts}} log\left(\frac{k_C - c_j + 1}{k_C + 2}\right) \ .$$

We can also remove the denominator $n + 4$ in the first term, since this is the same for both classes. We will use this modified form of the class score in our average-case analysis of naive Bayes.

3. Analysis of Classification Accuracy

We will assume that the target concept C is a monotone function of r relevant Boolean attributes that returns TRUE if q or more of these r attributes are present and returns FALSE otherwise. Thus, for a '3 of 5' concept, any instance that has three, four, or five features present is a positive example of the concept. We further suppose that there are i irrelevant Boolean attributes that play no role in the target concept, giving $r + i$ total Boolean features, each of which occurs with probability $P(A)$. We also assume that attributes are independent of each other. For now, we also assume that both training and test data are noise free, though we will return to this issue later.

Given this information, we want to predict the probability $A(n)$ of classifying a test case correctly after naive Bayes has seen n training instances. We can partition this accuracy measure into two components, one dealing with positive test cases and the other with negative test cases. Both terms involve summing over different types of test cases that differ in the number j out of i irrelevant attributes that are present and in the number s out of r relevant attributes that are present. This gives us

$$A(n) = \sum_{j=0}^{i} \sum_{s=q}^{r} T_{j,s} \mathcal{A}^{+}_{j,s}(n) + \sum_{j=0}^{i} \sum_{s=0}^{q-1} T_{j,s} \mathcal{A}^{-}_{j,s}(n) \ ,$$

where $T_{j,s}$ is the probability of encountering a test case with exactly j out of i irrelevant attributes and exactly s out of r relevant attributes, $\mathcal{A}^{+}_{j,s}(n)$ is the probability of correctly classifying such a test case, and $\mathcal{A}^{-}_{j,s}(n)$ is the analogous term for a negative case.

Let us first consider the probability of encountering a given type of test case. Because each attribute occurs independently with probability $P(A)$, the number of irrelevant attributes j that appear in a test case follows a binomial distribution with probability of success $P(A)$, as does the number of relevant attributes s that it contains. Taken together, this gives

$$T_{j,s} = \binom{i}{j} P(A)^j [1 - P(A)]^{i-j} \binom{r}{s} P(A)^s [1 - P(A)]^{r-s} \ .$$

Intuitively, this holds because the two probabilities are independent, with the former occurring in $\binom{i}{j}$ different ways and the latter occurring in $\binom{r}{s}$ ways.

Now we can focus on the predictive accuracy $\mathcal{A}^+_{j,s}(n)$ for a positive test case in which j irrelevants and s relevants are present. For this, we must average over the various concept descriptions that naive Bayes will acquire from different training sets. We can decompose the accuracies after n training cases into weighted sums of accuracies $A^+_{j,s}(k)$ and $A^-_{j,s}(k)$ for different numbers of positive instances, giving

$$\mathcal{A}^+_{j,s}(n) = \sum_{k=0}^{n} P(c_+ = k) A^+_{j,s}(k) \qquad \text{and} \qquad \mathcal{A}^-_{j,s}(n) = \sum_{k=0}^{n} P(c_+ = k) A^-_{j,s}(k) \ ,$$

where $P(c_+ = k)$ is the probability that exactly k out of n training instances are positive. Since each training case is independent of the others, the number k follows a binomial distribution, so that we have

$$P(c_+ = k) = \binom{n}{k} P(C)^k [1 - P(C)]^{n-k} \ ,$$

where $P(C)$ is the probability that any given instance will be positive.

Let us turn to the accuracy on a positive test cast with j irrelevant attributes and s relevant attributes present, given that naive Bayes has observed k out of n positive training cases. Recall that for each test case, the Bayesian classifier produces a score for the positive class, which we will call $\mathcal{S}$, and a score for the negative class, which we will call $\bar{\mathcal{S}}$. For a positive test case, the expected accuracy is precisely the probability that $\mathcal{S} > \bar{\mathcal{S}}$. We can restate this relation by defining the difference between these two scores $d = \bar{\mathcal{S}} - \mathcal{S}$, which lets us compute the accuracy $A^+_{j,s}(k)$ as the probability $P(d \leq 0)$.

4. Analysis of Class Scores

As explained earlier, our previous average-case analysis of naive Bayes was computationally intractable because it used the binomial distribution to compute the exact probability for every possible combination of counts for classes and attributes. However, one of the key results in statistics states that, given a reasonable number of samples, one can approximate a binomial distribution with mean μ and variance σ^2 using a normal distribution with the same parameters. Another key idea is that a linear combination of normally distributed variables will also follow a normal distribution. And given a normally distributed variable, one can convert its values into the standard normal distribution $\Phi(x)$, which has a mean of zero and variance of one, and for which one can easily compute the area for any interval.

We will take this latter approach to compute $P(d \leq 0)$, assuming that d follows a normal distribution by reasoning we will explain shortly. This assumption lets us convert d's values into the standard normal form $\Phi(x)$ by subtracting its mean and dividing by its variance, which gives

$$P\left(\frac{d - \mu_d}{\sqrt{\sigma_d^2}} \leq x\right) = \Phi(x) \ .$$

If we let x be $-\mu_d/\sqrt{\sigma_d^2}$, then we have

$$A_{j,s}^{+}(k) = P(d \leq 0) = \Phi\left(\frac{-\mu_d}{\sqrt{\sigma_d^2}}\right) \ ,$$

which we can compute for different values of μ_d and σ_d^2 by approximating the area under the standard normal distribution for a given number of positive training cases k. For each negative test case, we have the same scores s and $\bar{s}$, but the expected accuracy on such instances equals the probability that $\bar{\mathcal{S}} > \mathcal{S}$; this gives us $A_{j,s}^{-}(k) = P(d > 0) = 1 - P(d \leq 0) = 1 - \Phi\left(-\mu_d/\sqrt{\sigma_d^2}\right)$, which we can again calculate from the standard normal distribution.

However, we have yet to characterize μ_d and σ_d^2, the relevant terms for positive test cases, using more primitive expressions. We can assume that the

variable d follows a normal distribution if we can express it as the difference of two other normally distributed variables, say the negative score $\bar{\mathcal{S}}$ and the positive score $\mathcal{S}$. We can also rewrite its mean and variance using those for its components, which gives $\mu_d = \mu_{\bar{s}} - \mu_s$ and $\sigma_d^2 = \sigma_{\bar{\mathcal{S}}}^2 + \sigma_{\mathcal{S}}^2$.

We can treat the scores $\mathcal{S}$ and $\bar{\mathcal{S}}$ as normally distributed provided we assume they are both sums of variables that are themselves normally distributed. Recall that earlier we expressed the score $\mathcal{S}$ for each class as the sum of logarithms for observed counts, storing only one count c for each Boolean attribute to reflect the number of times it has been present in instances of that class. We can compute the other count, for the number of times it was absent, as $k - c$. We can also decompose the score into different terms for irrelevant and relevant attributes, giving

$$\mathcal{S} = (1 - i - r)log(k + 2) + \sum_{w=0}^{j} log(c_w + 1) + \sum_{x=j+1}^{i} log(k - c_x + 1) + \sum_{y=0}^{s} log(c_y + 1) + \sum_{z=s+1}^{r} log(k - c_z + 1) \ ,$$

where the first term represents influence from the count for the class, the second from the counts for the j irrelevant attributes present in the test case, the third from the counts for the $i - j$ irrelevant attributes that are absent, and the last two terms from the analogous counts for the s relevant attributes that are present and the $r - s$ ones that are absent. The added constants 1 and 2 comes from our earlier decisions about how to initialize various counts.

The counts for each attribute follow a binomial distribution, which means we can approximate them with a normal distribution for large numbers of training cases. However, the score uses not the counts themselves but rather a logarithmic transformation of the counts, which clearly does *not* follow a binomial. Nevertheless, since each attribute is independent, the central limit theorem tells us that we can use a normal curve to approximate their distribution. Moreover, we can reexpress the mean and variance for this distribution using the means and variances of the score $\mathcal{S}$'s components, giving us

$$\mu_{\mathcal{S}} = (1 - i - r)log(k + 2) + j \cdot \mu_{\star} + (i - j) \cdot \mu_{\bar{\star}} + s \cdot \mu_{\circ} + (r - s) \cdot \mu_{\bar{\circ}}$$

and

$$\sigma_{\mathcal{S}}^2 = j \cdot \sigma_{\star}^2 + (i - j) \cdot \sigma_{\bar{\star}}^2 + s \cdot \sigma_{\circ}^2 + (r - s) \cdot \sigma_{\bar{\circ}}^2 \ ,$$

where $\mu_\star$ is the mean of the log counts for each irrelevant attribute present in the test case, $\mu_{\bar{\star}}$ is the mean for each irrelevant attribute absent from the test case, $\mu_\circ$ and $\mu_{\bar{\circ}}$ are the analogous terms for relevant attributes, and the σ terms specify the variances for the various situations.

We must still calculate the means and variances for the log counts of irrelevant and relevant attributes. We do not know any closed-form expressions for these quantities, but we can compute them from their definitions for a given number of positive training cases k. In particular, for any variable x we have $\mu_x = \sum xP(x)$, and we know that the probability of each possible log value is the same as the probability for its corresponding count. Thus, we have

$$\mu_\star = \sum_{m=0}^{k} log(m+1) \binom{k}{m} P(A_\star|C)^m [1 - P(A_\star|C)]^{k-m} ,$$

where $P(A_\star|C)$ is the probability that an irrelevant attribute will appear in a positive instance. The mean for irrelevant attributes not present in the test case is

$$\mu_{\bar{\star}} = \sum_{m=0}^{k} log(k-m+1) \binom{k}{m} P(A_\star|C)^m [1 - P(A_\star|C)]^{k-m} ,$$

since the two differ only in their contributions to the overall score and not in their probabilities. The analogous expressions for relevant attributes, $\mu_\circ$ and $\mu_{\bar{\circ}}$, are identical except that they replace $P(A_\star|C)$ with $P(A_\circ|C)$.

Similar reasoning lets us determine the variances for the log counts. For any variable x, we have $\sigma_x^2 = [\sum x^2 P(x)] - [\mu_x]^2$. Again, since we know the probability of each possible log value from the probability for its corresponding count, we have

$$\sigma_\star^2 = \left(\sum_{m=0}^{k} log(m+1)^2 \binom{k}{m} P(A_\star|C)^m [1 - P(A_\star|C)]^{k-m} \right) - \mu_\star^2$$

as the variance for an irrelevant attribute that is present in a positive instance and

$$\sigma_{\bar{\star}}^2 = \left(\sum_{m=0}^{k} log(k-m+1)^2 \binom{k}{m} P(A_\star|C)^m [1 - P(A_\star|C)]^{k-m} \right) - \mu_{\bar{\star}}^2$$

as the variance for an irrelevant feature that is not present in a positive instance. Again, the variances for relevant attributes, $\sigma_\circ$ and $\sigma_{\bar{\circ}}$, are the same except that they replace $P(A_\star|C)$ with $P(A_\circ|C)$.

For the negative class, we can decompose the mean $\mu_{\bar{s}}$ and variance $\sigma^2_{\bar{s}}$ in the same manner. We will not give details here, but note only that the means and variances for irrelevant and relevant attributes are identical, except that they replace k with $n-k$, $P(A_\star|C)$ with $P(A_\star|\bar{C})$, and $P(A_\circ|C)$ with $P(A_\circ|\bar{C})$.

5. Analysis of Component Terms

Our remaining unknown terms include the conditional probability of an attribute given a class and the probabilities of the classes themselves, $P(C)$ and $P(\bar{C})$. Recall that we know the distribution of instances for the domain, which is determined by the given probability $P(A)$ that each attribute will occur in an arbitrary instance. Since we assume these attributes are independent, we know that the number of attributes present in an instance follows a binomial distribution with probability of success $P(A)$. However, we are concerned here with q of r concepts, so we care only about those instances with q or more relevant attributes present out of the r possible. Thus, we have

$$P(C) = \sum_{u=q}^{r} \binom{r}{u} P(A)^u [1 - P(A)]^{r-u}$$

as the probability of observing a positive instance, and since there exist only two classes, we know that $P(\bar{C}) = 1 - P(C)$.

Because the naive Bayesian classifier stores counts for each predictive attribute, we must also calculate the probability $P(A_\circ|C)$ that a relevant attribute will occur in an arbitrary positive instance. By reasoning we will not detail here, one can show that

$$P(A_\circ|C) = \frac{P(A_\circ \wedge C)}{P(C)} = \frac{1}{P(C)} \sum_{u=q}^{r} \binom{r-1}{u-1} P(A)^u [1 - P(A)]^{r-u} \ .$$

Similarly, we can specify the probability $P(A_\circ|\bar{C})$ that a relevant attribute will

occur in an arbitrary negative instance using an analogous expression:

$$P(A_\circ|\bar{C}) = \frac{P(A_\circ \wedge \bar{C})}{P(\bar{C})} = \frac{1}{P(\bar{C})} \sum_{u=1}^{q-1} \binom{r-1}{u-1} P(A)^u [1 - P(A)]^{r-u} \ ,$$

which sums over all values that are less than q and that conditionalizes over the probability of a negative instance.

Finally, we must express the probabilities that an irrelevant attribute $A_\star$ will appear in an arbitrary positive and in a negative instance. However, the definition of an irrelevant attribute is that its values are independent of the class. This means that $P(A_\star|C) = P(A_\star|\bar{C}) = P(A_\star)$, so that we can use each irrelevant attribute's probability $P(A)$ directly in our various calculations.

We can extend the framework to handle class noise by modifying the definitions of three basic terms: $P(C)$, $P(A_\circ|C)$, and $P(A_\circ|\bar{C})$. A common definition of class noise involves the corruption of class names (i.e., replacing the actual class with its opposite) with a certain probability $0 \leq z \leq 1$. As Iba and Langley (1992) have noted, the probability of the class after one has corrupted values is

$$P'(C) = (1 - z)P(C) + z(1 - P(C)) = P(C)[1 - 2z] + z \quad ,$$

since there exists a $P(C)$ probability that the class was actually present and a $1 - z$ probability it was not corrupted, as well as a $1 - P(C)$ probability that it was not present and a z probability that corruption has made it seem present.

For an irrelevant attribute $A_\star$, the probability $P(A_\star|C)$ is unaffected by class noise and remains equal to $P(A)$, since the attribute is still independent of the class. However, the situation for relevant attributes is more complicated. In particular, we must reexpress the (corrupted) conditional probability of a relevant attribute $A_\circ$ given the (corrupted) class C as

$$P'(A_\circ|C) = \frac{P'(A_\circ \wedge C)}{P'(C)} = \frac{(1 - z)P(A_\circ \wedge C) + zP(A_\circ \wedge \bar{C})}{P'(C)} \ .$$

where the terms in the numerator are the pre-noise conditional probabilities and the denominator is the noisy class probability given above. We can use

similar reasoning to express the post-noise probability of a relevant attribute given the negative class $\bar{C}$ as

$$P'(A_\circ|C) = \frac{P'(A_\circ \wedge \bar{C})}{P'(\bar{C})} = \frac{(1-z)P(A_\circ \wedge \bar{C}) + zP(A_\circ \wedge C)}{P'(\bar{C})} ,$$

where $P'(\bar{C}) = 1 - P'(C)$. We then replace the original terms for $P(C)$, $P(A_\circ|C)$, and $P(A_\circ|\bar{C})$ with their corrupted analogs in earlier parts of the analysis to determine the effect of class noise on the naive Bayesian classifier.

6. Implications for Learning Behavior

The equations in the previous section provide a formal description of naive Bayes' behavior, but their implications are not obvious. However, we can use the analysis to make average-case predictions about the algorithm's accuracy under different domain characteristics. Because the analysis introduces normal approximations to the binomial and relies on the central limit theorem, the predictions will not be perfect, but their accuracy should increase with the number of training cases and attributes. And because the analysis is tractable computationally, we can examine larger training and attribute sets than in our previous average-case treatment of the naive Bayesian classifier.

In addition to theoretical predictions, we report learning curves that summarize runs on 100 randomly generated training sets. Each curve reports the average classification accuracy over these runs on a single noise-free test set that includes all possible instances. In each case, we bound the mean accuracy with 95% confidence intervals to show the degree to which our predicted learning curves fit the observed ones. These experimental results provide an important check on both our reasoning and the ability of the analysis to predict behavior despite its simplifying approximations.

Figure 1 (a) shows the effect of irrelevant attributes on naive Bayes' rate of learning when the training data contains no noise. In this study, we used $P(A) = \frac{1}{2}$ as the probability for each attribute and a '2 of 2' target concept, while we varied both the number of training cases and the number of irrelevant attributes. As typical with learning curves, the accuracies begin low and

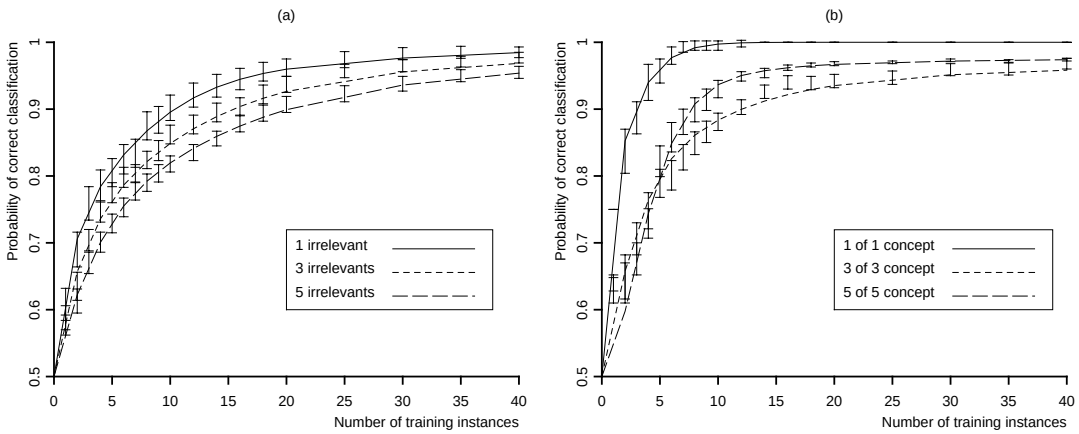

Figure 1. Theoretical and experimental learning curves for naive Bayes when (a) the domain involves a '2 of 2' target concept and varying numbers of irrelevant attributes, and (b) for a domain with one irrelevant attribute and a conjunctive target concept with varying numbers of relevant features. The error bars represent 95% confidence intervals for the experimental curves.

gradually improve with the size of the training set. The effect of irrelevants also agrees with our intuitions, in that it degrades the learning rate gracefully and does not alter asymptotic accuracy, which is 100 percent on conjunctive concepts (Domingos & Pazzani, 1997). Moreover, the predicted and observed learning curves are in close agreement, even for very small training sets where we expected divergence due to our use of approximations.

Figure 1 (b) presents the corresponding effect of relevant attributes on learning rate, again in the absence of noise. Here we used a single irrelevant attribute in all conditions, but we varied the number of training cases and the number of relevant features in a conjunctive target concept. As before, we used $P(A) = \frac{1}{2}$ for each attribute, which causes class distributions to become ever more skewed as one adds relevant features. Intuition suggests that this should make more complex concepts more difficult to master and, indeed, the '1 of 1' concept has the highest learning rate. But the analysis also predicts that, over most of their learning curves, accuracy will be higher for a conjunction of five attributes than for one with three attributes. The empirical results confirm this surprising phenomenon, matching the predictions closely enough to reflect a crossover between curves around the fifth training instance.

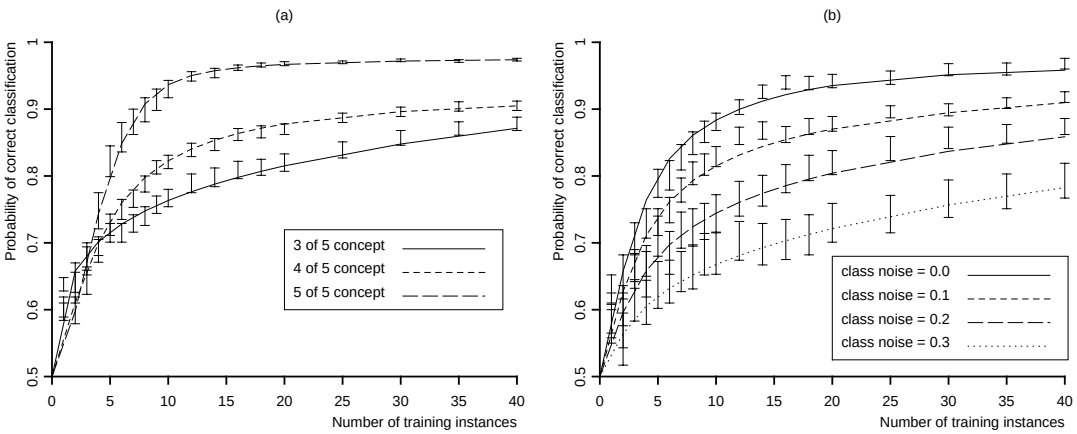

Figure 2. Theoretical and experimental learning curves for naive Bayes when the domain involves one irrelevant attribute and (a) a q of r target concept with different values for q, as well as (b) a '3 of 3' concept with varying levels of class noise.

Another interesting question concerns the effect of q, the number of necessary attributes in a q of r concept, which we could not address in our earlier study because it was limited to conjunctive concepts. Figure 2 (a) shows the learning curves for target concepts with five relevant attributes when $q = 3$, $q = 4$, and $q = 5$. Intuition suggests that concepts with the highest and lowest q values should be easiest to learn, since relevant features will be easier to identify when they are more nearly criterial. The analysis predicts this effect over most of the learning curves, and again the experimental curves match the analytic ones quite well. We have not shown curves for $q = 1$ and $q = 2$, since they are identical to those for $q = 5$ and $q = 4$, respectively.

The presence of noise is another complicating factor that makes induction difficult. Figure 2 (b) illustrates the effect of class noise on the learning rate for naive Bayes, given a '3 of 3' target concept and one irrelevant feature. To support more direct correspondence among different noise levels, we omitted noise from the test set, which normalizes accuracies and eases comparison. Both the theoretical and experimental curves match the intuition that learning rate should slow with class noise, but here the predictions do not match the observations quite as well as before, showing there can be a cost to introducing approximations for the sake of efficiency.

7. Concluding Remarks

In this paper, we have presented an improved average-case analysis of the naive Bayesian classifier, an induction algorithm that has become increasingly popular in recent years. The treatment generalized our earlier analysis by extending it to arbitrary 'M of N' concepts. More important, the new analysis was much more tractable, in computational terms, than its predecessor, which let us make predictions about naive Bayes' behavior on larger training sets and more attributes. The price of this computational efficiency was reliance on normal approximations to the binomial, which meant that our ability to fit empirical results became an empirical question.

To explore the implications of the analysis, we plotted the predicted behavior of the algorithm as a function of the number of training instances, the amount of noise, and the number of irrelevant, relevant, and necessary attributes, finding graceful degradation as these parameters varied. As a check on our analysis, we ran the algorithm on synthetic training and test sets with the same characteristics. In general, we found excellent fits between the theoretical predictions and observed behavior, which lends support to our use of approximations to make the analysis tractable.

One issue we have not addressed is the sensitivity of our analysis to its underlying assumptions. We know that naive Bayes often behaves well even when the predictive attributes are not completely independent given the class (Domingos & Pazzani, 1997), but this does not mean that the analysis will accurately predict its learning curve under such conditions. In future work, we should run experiments with domains that violate the independence assumption to varying degrees, as Pazzani and Sarrett (1992) did in their analysis of conjunctive learning. They reported good fits to observed learning curves despite dependencies, and we anticipate similar results for naive Bayes. We also hope to extend our new analytic techniques to other induction algorithms, including nearest neighbor and methods that learn determinations.

A final direction for future research involves using average-case analysis to better understand the behavior of naive Bayes and other algorithms in natural domains. For any natural data set, we can estimate $P(C)$, $P(A)$, and $P(A|C)$,

and we know the number of training cases and attributes. We do not know the number of relevant attributes, the noise level, or the exact target concept, but experimental learning curves provide constraints that we can use to select among alternative settings for parameters from the average-case model. This would require extending the analysis to handle non-Boolean attributes and a broader range of target concepts, but the result would be a much stronger connection between the theoretical and empirical branches of machine learning, bringing the field closer to becoming a true science of the artificial.

References

Cestnik, G., Konenenko, I, & Bratko, I. (1987). Assistant-86: A knowledge-elicitation tool for sophisticated users. In I. Bratko & N. Lavrac (Eds.) *Progress in machine learning.* Sigma Press.

Clark, P., & Niblett, T. (1989). The CN2 induction algorithm. *Machine Learning*, *3*, 261–284.

Domingos, P., & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, *29*,103–130.

Golea, M., & Marchand, M. (1993). On learning perceptrons with binary weights. *Neural Computation*, *5*, 767–782.

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

Iba, W., & Langley, P. (1992). Induction of one-level decision trees. *Proceedings of the Ninth International Conference on Machine Learning* (pp. 233–240). Aberdeen, Scotland: Morgan Kaufmann.

Langley, P., & Iba, W. (1993). Average-case analysis of a nearest neighbor algorithm. *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence* (pp. 889–894). Chambery, France.

Langley, P., Iba, W., & Thompson, K. (1992). An analysis of Bayesian classifiers. *Proceedings of the Tenth National Conference on Artificial Intelligence* (pp. 223–228). San Jose, CA: AAAI Press.

Okamoto, S., & Yugami, N. (1997). An average-case analysis of the k-nearest neighbor classifier for noisy domains. *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence* (pp. 238–243). Yokohama, Japan.

Pazzani, M., & Sarrett, W. (1992). A framework for average-case analysis of conjunctive learning algorithms. *Machine Learning*, *9*, 349-372.