, , 1–30 ()

On the Optimality of the Simple Bayesian Classifier under Zero-One Loss

PEDRO DOMINGOS pedrod@ics.uci.edu
MICHAEL PAZZANI pazzani@ics.uci.edu
Department of Information and Computer Science, University of California, Irvine, CA 92697

Editor: Gregory Provan

Abstract. The simple Bayesian classifier is known to be optimal when attributes are independent given the class, but the question of whether other sufficient conditions for its optimality exist has so far not been explored. Empirical results showing that it performs surprisingly well in many domains containing clear attribute dependences suggest that the answer to this question may be positive. This article shows that, although the Bayesian classifier's probability estimates are only optimal under quadratic loss if the independence assumption holds, the classifier itself can be optimal under zero-one loss (misclassification rate) even when this assumption is violated by a wide margin. The region of quadratic-loss optimality of the Bayesian classifier is in fact a second-order infinitesimal fraction of the region of zero-one optimality. This implies that the Bayesian classifier has a much greater range of applicability than previously thought. For example, in this article it is shown to be optimal for learning conjunctions and disjunctions, even though they violate the independence assumption. Further, studies in artificial domains show that it will often outperform more powerful classifiers for common training set sizes and numbers of attributes, even if its bias is *a priori* much less appropriate to the domain. This article's results also imply that detecting attribute dependence is not necessarily the best way to extend the Bayesian classifier, and this is also verified empirically.

Keywords: Simple Bayesian classifier, naive Bayesian classifier, zero-one loss, optimal classification, induction with attribute dependences

1. Introduction

In classification learning problems, the learner is given a set of training examples and the corresponding class labels, and outputs a classifier. The classifier takes an unlabeled example and assigns it to a class. Many classifiers can be viewed as computing a set of *discriminant functions* of the example, one for each class, and assigning the example to the class whose function is maximum (Duda & Hart, 1973). If E is the example, and $f_i(E)$ is the discriminant function corresponding to the ith class, the chosen class C_k is the one for which[1]

$$f_k(E) > f_i(E) \quad \forall\, i \neq k. \tag{1}$$

Suppose an example is a vector of a attributes, as is typically the case in classification applications. Let v_{jk} be the value of attribute A_j in the example, $P(X)$ denote the probability of X, and $P(Y|X)$ denote the conditional probability of Y given X. Then one possible set of discriminant functions is

$$f_i(E) = P(C_i) \prod_{j=1}^{a} P(A_j = v_{jk}|C_i). \tag{2}$$

The classifier obtained by using this set of discriminant functions, and estimating the relevant probabilities from the training set, is often called the *naive Bayesian classifier*. This is because, if the "naive" assumption is made that the attributes are independent given the class, this classifier can easily be shown to be optimal, in the sense of minimizing the misclassification rate or *zero-one loss*, by a direct application of Bayes' theorem, as follows. If $P(C_i|E)$ is the probability that example E is of class C_i, zero-one loss is minimized if, and only if, E is assigned to the class C_k for which $P(C_k|E)$ is maximum (Duda & Hart, 1973). In other words, using $P(C_i|E)$ as the discriminant functions $f_i(E)$ is the optimal classification procedure. By Bayes' theorem,

$$P(C_i|E) = \frac{P(C_i)\,P(E|C_i)}{P(E)}. \tag{3}$$

$P(E)$ can be ignored, since it is the same for all classes, and does not affect the relative values of their probabilities. If the attributes are independent given the class, $P(E|C_i)$ can be decomposed into the product $P(A_1 = v_{1k}|C_i) \ldots P(A_a = v_{ak}|C_i)$, leading to $P(C_i|E) = f_i(E)$, as defined in Equation 2, Q.E.D.

In practice, attributes are seldom independent given the class, which is why this assumption is "naive." However, the question arises of whether the Bayesian classifier, as defined by Equations 1 and 2, can be optimal even when the assumption of attribute independence does not hold, and therefore $P(C_i|E) \neq f_i(E)$. In these situations, the Bayesian classifier can no longer be said to compute class probabilities given the example, but the discriminant functions defined by Equation 2 may still minimize misclassification error. The question of whether these situations exist has practical relevance, since the Bayesian classifier has many desirable properties (simplicity, low time and memory requirements, etc.), and thus may well be the classifier of choice for such situations (i.e., it will be chosen over other classifiers that are also optimal, but differ in other respects). However, even though the Bayesian classifier has been known for several decades, to our knowledge this question has so far not been explored; the tacit assumption has always been that the Bayesian classifier will not be optimal when attribute independence does not hold.

In spite of this restrictive view of its applicability, in recent years there has been a gradual recognition among machine learning researchers that the Bayesian classifier can perform quite well in a wide variety of domains, including many where clear attribute dependences exist. Evidence of the Bayesian classifier's surprising practical value has also led to attempts to extend it by increasing its tolerance of attribute independence in various ways, but the success of these attempts has been uneven. This is described in more detail in the next section.

This article derives the most general conditions for the Bayesian classifier's optimality, giving a positive answer to the question of whether it can still be optimal when attributes are not independent given the class. A corollary of these results is that the Bayesian classifier's true region of optimal performance is in fact far greater than that implied by the attribute independence assumption, and that its range of applicability is thus much broader than previously thought. This tolerance of attribute dependence also helps to explain why extending the Bayesian classifier by attempting to reduce it will not necessarily lead to significant performance improvements.

The remainder of the article elaborates on these ideas. Section 2 reviews previous empirical results on the Bayesian classifier in the machine learning literature, and recent attempts to extend it. Section 3 describes an empirical study showing that the Bayesian classifier outperforms several more sophisticated approaches on a large number of data sets, and that this is not due to the absence of attribute dependences in those data sets. Section 4 presents a simple example that illustrates some of the key points to be made subsequently. Section 5 derives necessary and sufficient conditions for the local optimality of the Bayesian classifier (i.e., its optimality for any given example), and computes how often these conditions will be satisfied. Section 6 generalizes the previous results to a necessary and sufficient condition for the Bayesian classifier's global optimality (i.e., its optimality for any given data set). It also shows that the Bayesian classifier has some fundamental limitations, but is optimal for learning conjunctions and disjunctions. Section 7 formulates some hypotheses as to when the Bayesian classifier is likely to outperform more flexible ones, even if it is not optimal, and reports empirical tests of these hypotheses. Section 8 verifies empirically that attempting to reduce attribute dependence is not necessarily the best approach to improving the Bayesian classifier's accuracy. The paper concludes with discussion and directions for future work.

2. The simple Bayesian classifier in machine learning

Due to its perceived limitations, the simple Bayesian classifier has traditionally not been a focus of research in machine learning.[2] However, it has sometimes been used as a "straw man" against which to compare more sophisticated algorithms. Clark and Niblett (1989) compared it with two rule learners and a decision-tree learner, and found that it did surprisingly well. Cestnik (1990) reached similar conclusions. Kononenko (1990) reported that, in addition, at least one class of users (doctors) finds the Bayesian classifier's representation quite intuitive and easy to understand, something which is often a significant concern in machine learning. Langley, Iba, and Thompson (1992) compared the Bayesian classifier with a decision tree learner, and found it was more accurate in four of the five data sets used. Pazzani, Muramatsu, and Billsus (1996) compared several learners on a suite of information filtering tasks, and found that the Bayesian classifier was the most accurate one overall.

John and Langley (1995) showed that the Bayesian classifier's performance can be much improved if the traditional treatment of numeric attributes, which assumes Gaussian distributions, is replaced by kernel density estimation. This showed that the Bayesian classifier's limited performance in many domains was not in fact intrinsic to it, but due to the additional use of unwarranted Gaussian assumptions. Dougherty, Kohavi, and Sahami (1995) reached similar conclusions by instead discretizing numeric attributes, and found the Bayesian classifier with discretization slightly outperformed a decision-tree learner in 16 data sets, on average.

Although the reasons for the Bayesian classifier's good performance were not clearly understood, these results were evidence that it might constitute a good starting point for further development. Accordingly, several authors attempted to extend it by addressing its main perceived limitation—its inability to deal with attribute dependences.

Langley and Sage (1994) argued that, when two attributes are correlated, it might be better to delete one attribute than to assume the two are conditionally independent. They found that an algorithm for feature subset selection (forward sequential selection) improved accuracy on some data sets, but had little or no effect in others. In a related approach, Kubat, Flotzinger, and Pfurtscheller (1993) found that using a decision-tree learner to select features for use in the Bayesian classifier gave good results in the domain of EEG signal classification.

Kononenko (1991) proposed successively joining dependent attribute values, using a statistical test to judge whether two attribute values are significantly dependent. Experimental results with this method were not encouraging. On two domains, the modified Bayesian classifier had the same accuracy as the simple Bayesian classifier, and on the other two domains tested, the modified version was one percent more accurate, but it is not clear whether this difference was statistically significant. Pazzani (1996) proposed joining attributes instead of attribute values. Rather than using a statistical test, as in Kononenko (1991), Pazzani's algorithm used cross-validation to estimate the accuracy of a classifier with each possible join, and made the single change that most improved accuracy. This process was repeated until no change resulted in an improvement. This approach substantially improved the accuracy of the Bayesian classifier on several artificial and natural data sets, with the largest improvements in accuracy occurring in data sets where the Bayesian classifier is substantially less accurate than decision-tree learners.

The simple Bayesian classifier is limited in expressiveness in that it can only create linear frontiers (Duda & Hart, 1973). Therefore, even with many training examples and no noise, it does not approach 100% accuracy on some problems. Langley (1993) proposed the use of "recursive Bayesian classifiers" to address this limitation. In his approach, the instance space is recursively divided into subregions by a hierarchical clustering process, and a Bayesian classifier is induced for each region. Although the algorithm worked on an artificial problem, it did not provide a significant benefit on any natural data sets. In a similar vein, Kohavi (1996) formed decision trees with Bayesian classifiers at the nodes, and showed that it tended to outperform either approach alone, especially on large data sets.

Friedman, Geiger, and Goldszmidt (1997) compared the simple Bayesian classifier with Bayesian networks, a much more powerful representation that has the Bayesian classifier as a special case, and found that the latter approach tended to produce no improvements, and sometimes led to large reductions in accuracy. This led them to attempt a much more limited extension, allowing each attribute to depend on at most one other attribute (in addition to the class). This conservative approach achieved the best overall results. Sahami (1996) proposed a related scheme, and, in a similar spirit, Singh and Provan (1995, 1996) obtained good results by forming Bayesian networks using only a subset of the attributes.

In summary, the Bayesian classifier has repeatedly performed better than expected in empirical trials, but attempts to build on this success by relaxing the independence assumption have had mixed results. Both these observations seem to conflict with the current theoretical understanding of the Bayesian classifier. This article seeks to resolve this apparent contradiction.

3. Empirical evidence

Whenever theoretical expectations and empirical observations disagree, either could be at fault. On the empirical side, two potential sources of error can be readily identified. The results of previous authors could be a fluke, due to unusual characteristics of the data sets used (especially since, in several cases, the number of data sets used was relatively small). Alternatively, these data sets might contain no significant attribute dependences, and in this case the Bayesian classifier would indeed be expected to perform well. In order to test both these hypotheses, we conducted an empirical study on 28 data sets, comparing the Bayesian classifier with other learners, and measuring the degree of attribute dependence in the data sets. The learners used were state-of-the art representatives of three major approaches to classification learning: decision tree induction (C4.5 release 8, Quinlan, 1993), instance-based learning (PEBLS 2.1, Cost & Salzberg, 1993) and rule induction (CN2 version 6.1, Clark & Boswell, 1991). A simple Bayesian classifier was implemented for these experiments. Three main issues arise here: how to handle numeric attributes, zero counts, and missing values. We deal with each in turn.

- *Numeric attributes* were discretized into ten equal-length intervals (or one per observed value, whichever was least). Although Dougherty et al. (1995) found this approach to be slightly less accurate than a more informed one, it has the advantage of simplicity, and is sufficient for verifying that the Bayesian classifier performs as well as, or better than, other learners. A version incorporating the conventional assumption of Gaussian distributions was also implemented, for purposes of comparison with the discretized one.

- *Zero counts* are obtained when a given class and attribute value never occur together in the training set, and can be problematic because the resulting zero probabilities will wipe out the information in all the other probabilities $P(A_j = v_{jk}|C_i)$ when they are multiplied according to Equation 2. A principled solution to this problem is to incorporate a small-sample correction into all probabilities, such as the Laplace correction (Niblett, 1987). If n_{ijk} is the number of times class C_i and value v_{jk} of attribute A_j occur together, and n_i is the total number of times class C_i occurs in the training set, the uncorrected estimate of $P(A_j = v_{jk}|C_i)$ is n_{ijk}/n_i, and the Laplace-corrected estimate is $P(A_j = v_{jk}|C_i) = (n_{ijk} + f)/(n_i + fn_j)$, where n_j is the number of values of attribute A_j (e.g., 2 for a Boolean attribute), and f is a multiplicative factor. Following Kohavi, Becker, and Sommerfield (1997), the Laplace correction was used with $f = 1/n$, where n is the number of examples in the training set.

- *Missing values* were ignored, both when computing counts for the probability estimates and when classifying a test example. This ensures the Bayesian classifier does not inadvertently have access to more information than the other algorithms, and if anything biases the results against it.

Twenty-eight data sets from the UCI repository (Merz, Murphy & Aha,1997) were used in the study. Twenty runs were conducted for each data set, randomly selecting $\frac{2}{3}$ of the data for training and the remainder for testing. Table 1 shows the average accuracies obtained. As a baseline, the default accuracies obtained by guessing the most frequent class are also shown. Confidence levels for the observed differences in accuracy between the (discretized) Bayesian classifier and the other algorithms, according to a one-tailed paired t test, are also reported.[3]

The results are summarized in Table 2. The first line shows the number of domains in which the Bayesian classifier was more accurate than the corresponding classifier, versus the number in which it was less. For example, the Bayesian classifier was more accurate than C4.5 in 19 domains, and less in 9. The second line considers only those domains where the accuracy difference was significant at the 5% level, using a one-tailed paired t test. For example, the Bayesian classifier was significantly more accurate than C4.5 in 12 data sets. According to both these measures, the Bayesian classifier wins out over each of the other approaches. The third line shows the confidence levels obtained by applying a binomial sign test to the results in the first line, and results in high confidence that the Bayesian classifier is more accurate than C4.5 and CN2, if this sample of data sets is assumed to be representative. The fourth line shows the confidence levels obtained by applying the more sensitive Wilcoxon test (DeGroot, 1986) to the 28 average accuracy differences obtained, and results in high confidence that the Bayesian classifier is more accurate than each of the other learners. The fifth line shows the average accuracy across all data sets, and again the Bayesian classifier performs the best. The last line shows the average rank of each algorithm, computed for each domain by assigning rank 1 to the most accurate algorithm, rank 2 to the second best, and so on. The Bayesian classifier is the best-ranked of all algorithms, indicating that when it does not win it still tends to be one of the best.

The comparative results of the discretized and Gaussian versions also confirm the advantage of discretization, although on this larger ensemble of data sets the difference is less pronounced than that found by Dougherty et al. (1995), and the Gaussian version also does quite well compared to the non-Bayesian learners.

Table 1. Classification accuracies and sample standard deviations, averaged over 20 random training/test splits. "Bayes" is the Bayesian classifier with discretization and "Gauss" is the Bayesian classifier with Gaussian distributions. Superscripts denote confidence levels for the difference in accuracy between the Bayesian classifier and the corresponding algorithm, using a one-tailed paired t test: 1 is 99.5%, 2 is 99%, 3 is 97.5%, 4 is 95%, 5 is 90%, and 6 is below 90%.

Data Set	**Bayes**	**Gauss**	**C4.5**	**PEBLS**	**CN2**	**Def.**
Audiology	73.0±6.1	73.0±6.1^{6}	72.5±5.8^{6}	75.8±5.4^{3}	71.0±5.1^{5}	21.3
Annealing	95.3±1.2	84.3±3.8^{1}	90.5±2.2^{1}	98.8±0.8^{1}	81.2±5.4^{1}	76.4
Breast cancer	71.6±4.7	71.3±4.3^{6}	70.1±6.8^{5}	65.6±4.7^{1}	67.9±7.1^{1}	67.6
Credit	84.5±1.8	78.9±2.5^{1}	85.9±2.1^{3}	82.2±1.9^{1}	82.0±2.2^{1}	57.4
Chess endgames	88.0±1.4	88.0±1.4^{6}	99.2±0.1^{1}	96.9±0.7^{1}	98.1±1.0^{1}	52.0
Diabetes	74.5±2.4	75.2±2.1^{6}	73.5±3.4^{5}	71.1±2.4^{1}	73.8±2.7^{6}	66.0
Echocardiogram	69.1±5.4	73.4±4.9^{1}	64.7±6.3^{1}	61.7±6.4^{1}	68.2±7.2^{6}	67.8
Glass	61.9±6.2	50.6±8.2^{1}	63.9±8.7^{6}	62.0±7.4^{6}	63.8±5.5^{6}	31.7
Heart disease	81.9±3.4	84.1±2.8^{1}	77.5±4.3^{1}	78.9±4.0^{1}	79.7±2.9^{3}	55.0
Hepatitis	85.3±3.7	85.2±4.0^{6}	79.2±4.3^{1}	79.0±5.1^{1}	80.3±4.2^{1}	78.1
Horse colic	80.7±3.7	79.3±3.7^{1}	85.1±3.8^{1}	75.7±5.0^{1}	82.5±4.2^{2}	63.6
Hypothyroid	97.5±0.3	97.9±0.4^{1}	99.1±0.2^{1}	95.9±0.7^{1}	98.8±0.4^{1}	95.3
Iris	93.2±3.5	93.9±1.9^{6}	92.6±2.7^{6}	93.5±3.0^{6}	93.3±3.6^{6}	26.5
Labor	91.3±4.9	88.7±10.6^{6}	78.1±7.9^{1}	89.7±5.0^{6}	82.1±6.9^{1}	65.0
Lung cancer	46.8±13.3	46.8±13.3^{6}	40.9±16.3^{5}	42.3±17.3^{6}	38.6±13.5^{3}	26.8
Liver disease	63.0±3.3	54.8±5.5^{1}	65.9±4.4^{1}	61.3±4.3^{6}	65.0±3.8^{3}	58.1
LED	62.9±6.5	62.9±6.5^{6}	61.2±8.4^{6}	55.3±6.1^{1}	58.6±8.1^{2}	8.0
Lymphography	81.6±5.9	81.1±4.8^{6}	75.0±4.2^{1}	82.9±5.6^{6}	78.8±4.9^{3}	57.3
Post-operative	64.7±6.8	67.2±5.0^{3}	70.0±5.2^{1}	59.2±8.0^{2}	60.8±8.2^{4}	71.2
Promoters	87.9±7.0	87.9±7.0^{6}	74.3±7.8^{1}	91.7±5.9^{3}	75.9±8.8^{1}	43.1
Primary tumor	44.2±5.5	44.2±5.5^{6}	35.9±5.8^{1}	30.9±4.7^{1}	39.8±5.2^{1}	24.6
Solar flare	68.5±3.0	68.2±3.7^{6}	70.6±2.9^{1}	67.6±3.5^{6}	70.4±3.0^{2}	25.2
Sonar	69.4±7.6	63.0±8.3^{1}	69.1±7.4^{6}	73.8±7.4^{1}	66.2±7.5^{5}	50.8
Soybean	100.0±0.0	100.0±0.0^{6}	95.0±9.0^{3}	100.0±0.0^{6}	96.9±5.9^{3}	30.0
Splice junctions	95.4±0.6	95.4±0.6^{6}	93.4±0.8^{1}	94.3±0.5^{1}	81.5±5.5^{1}	52.4
Voting records	91.2±1.7	91.2±1.7^{6}	96.3±1.3^{1}	94.9±1.2^{1}	95.8±1.6^{1}	60.5
Wine	96.4±2.2	97.8±1.2^{3}	92.4±5.6^{1}	97.2±1.8^{6}	90.8±4.7^{1}	36.4
Zoology	94.4±4.1	94.1±3.8^{6}	89.6±4.7^{1}	94.6±4.3^{6}	90.6±5.0^{1}	39.4

Table 2. Summary of accuracy results.

Measure	**Bayes**	**Gauss**	**C4.5**	**PEBLS**	**CN2**
No. wins	-	12-7	19-9	16-11	20-8
No. signif. wins	-	6-5	12-8	12-6	16-6
Sign test	-	75.0	96.0	75.0	98.0
Wilcoxon test	-	70.0	96.0	94.0	99.8
Average	79.1	77.8	77.2	77.6	76.2
Rank	2.43	2.75	3.14	3.21	3.46

In summary, the present large-scale study confirms previous authors' observations on smaller ensembles of data sets; in fact, the current results are even more favorable to the Bayesian classifier. However, this does not by itself disprove the notion that the Bayesian classifier will only do well when attributes are independent given the class (or nearly so). As pointed out above, the Bayesian classifier's good performance could simply be due to the absence of significant attribute dependences in the data. To investigate this, we need to measure the degree of attribute dependence in the data in some way. Measuring high-order dependencies is difficult, because the relevant probabilities are apt to be very small, and not reliably represented in the data. However, a first and feasible approach consists in measuring pairwise dependencies (i.e., dependencies between pairs of attributes given the class). Given attributes A_m and A_n and the class variable C, a possible measure of the degree of pairwise dependence between A_m and A_n given C (Wan & Wong, 1989; Kononenko, 1991) is

$$D(A_m, A_n|C) = H(A_m|C) + H(A_n|C) - H(A_m A_n|C), \tag{4}$$

where $A_m A_n$ represents the Cartesian product of attributes A_m and A_n (i.e., a derived attribute with one possible value corresponding to each combination of values of A_m and A_n), and for all classes i and attribute values k,

$$H(A_j|C) = \sum_i P(C_i) \sum_k -P(C_i \wedge A_j = v_{jk}) \log_2 P(C_i \wedge A_j = v_{jk}). \tag{5}$$

The $D(A_m, A_n|C)$ measure is zero when A_m and A_n are completely independent given C, and increases with their degree of dependence, with the maximum occurring when the class and one attribute completely determine the other.[4]

D was computed for all classes and attribute pairs in each data set, using uniform discretization as before, ignoring missing values, and excluding pairings of an attribute with itself. The results appear in Table 3.[5] For comparison purposes, the first column shows the Bayesian classifier's rank in each domain (i.e., 1 if it was the most accurate algorithm, 2 if it was the second most accurate, etc., ignoring the Gaussian version). The second column shows the maximum value of D observed in the data set. The third column shows the percentage of all attributes that exhibited a degree of dependence with some other attribute of at least 0.2. The fourth column shows the average D for all attribute pairs in the data set.

This table leads to two important observations. One is that the Bayesian classifier achieves higher accuracy than more sophisticated approaches in many domains where there is substantial attribute dependence, and therefore the reason for its good comparative performance is not that there are no attribute dependences in the data. The other is that the correlation between the average degree of attribute dependence and the difference in accuracy between the Bayesian classifier and other algorithms is very small ($R^2 = 0.04$ for C4.5, 0.0004 for PEBLS, and 0.002 for CN2), and therefore attribute dependence is not a good predictor of the Bayesian classifier's differential performance vs. approaches that can take it into account. Given this empirical evidence, it is clear that a new theoretical understanding of the Bayesian classifier is needed. We now turn to this.

Table 3. Empirical measures of attribute dependence.

Data Set	**Rank**	**Max.** D	**%** $D > 0.2$	**Avg.** D
Breast cancer	1	0.548	66.7	0.093
Credit	2	0.790	46.7	0.063
Chess endgames	4	0.383	25.0	0.015
Diabetes	1	0.483	62.5	0.146
Echocardiogram	1	0.853	85.7	0.450
Glass	4	0.836	100.0	0.363
Heart disease	1	0.388	53.8	0.085
Hepatitis	1	0.899	57.9	0.103
Horse colic	3	2.780	100.0	0.286
Hypothyroid	3	2.777	60.0	0.095
Iris	3	0.731	100.0	0.469
Labor	1	1.514	100.0	0.474
Lung cancer	1	1.226	98.2	0.165
Liver disease	3	0.513	100.0	0.243
LED	1	0.060	0.0	0.025
Lymphography	2	0.410	55.6	0.076
Post-operative	2	0.181	0.0	0.065
Promoters	2	0.394	98.2	0.149
Solar flare	3	0.216	16.7	0.041
Sonar	2	1.471	100.0	0.491
Soybean	1	0.726	31.4	0.016
Splice junctions	1	0.084	0.0	0.017
Voting records	4	0.316	25.0	0.052
Wine	2	0.733	100.0	0.459
Zoology	2	0.150	0.0	0.021

4. An example of optimality without independence

Consider a Boolean concept, described by three attributes A, B and C. Assume that the two classes, denoted by $+$ and $-$, are equiprobable ($P(+) = P(-) = \frac{1}{2}$). Given an example E, let $P(A|+)$ be a shorthand for $P(A = a_E|+)$, a_E being the value of attribute A in the instance, and similarly for the other attributes. Let A and C be independent, and let $A = B$ (i.e., A and B are completely dependent). Therefore B should be ignored, and the optimal classification procedure for a test instance is to assign it to class $+$ if $P(A|+)P(C|+) - P(A|-)P(C|-) > 0$, to class $-$ if the inequality has the opposite sign, and to an arbitrary class if the two sides are equal. On the other hand, the Bayesian classifier will take B into account as if it was independent from A, and this will be equivalent to counting A twice. Thus, the Bayesian classifier will assign the instance to class $+$ if $P(A|+)^2P(C|+) - P(A|-)^2P(C|-) > 0$, and to $-$ otherwise.

Applying Bayes' theorem, $P(A|+)$ can be reexpressed as $P(A)P(+|A)/P(+)$, and similarly for the other probabilities. Since $P(+) = P(-)$, after canceling like terms this leads to the equivalent expressions $P(+|A)P(+|C) - P(-|A)P(-|C) > 0$ for the optimal decision, and $P(+|A)^2P(+|C) - P(-|A)^2P(-|C) > 0$ for the Bayesian classifier. Let $P(+|A) = p$ and $P(+|C) = q$. Then class $+$ should be selected

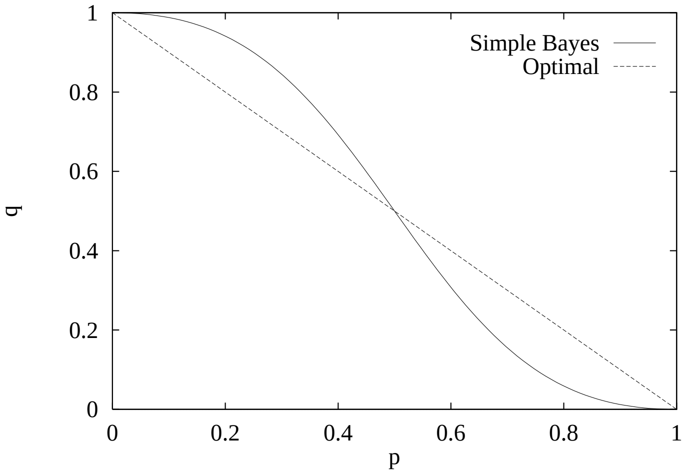

Figure 1. Decision boundaries for the Bayesian classifier and the optimal classifier.

when $pq - (1-p)\,(1-q) > 0$, which is equivalent to $q > 1-p$. With the Bayesian classifier, it will be selected when $p^2q - (1-p)^2(1-q) > 0$, which is equivalent to $q > \frac{(1-p)^2}{p^2+(1-p)^2}$. The two curves are shown in Figure 1. The remarkable fact is that, even though the independence assumption is decisively violated because $B = A$, the Bayesian classifier disagrees with the optimal procedure only in the two narrow regions that are above one of the curves and below the other; everywhere else it performs the correct classification. Thus, for all problems where (p, q) does not fall in those two small regions, the Bayesian classifier is effectively optimal. By contrast, according to the independence assumption it should be optimal only when the two expressions are identical, i.e. at the three isolated points where the curves cross: $(0,\,1)$, $(\frac{1}{2},\,\frac{1}{2})$ and $(1,\,0)$. This shows that the Bayesian classifier's range of applicability may in fact be much broader than previously thought. In the next section we examine the general case and formalize this result.

5. Local optimality

We begin with some necessary definitions.

DEFINITION 1 *Let $C(E)$ be the actual class of example E, and let $C_X(E)$ be the class assigned to it by classifier X. Then the* zero-one loss *of X on E, denoted $L_X(E)$, is defined as*

$$L_X(E) = \begin{cases} 0 & \textit{if } \; C_X(E) = C(E) \\ 1 & \textit{otherwise.} \end{cases} \tag{6}$$

Zero-one loss is an appropriate measure of performance when the task is classification, and it is the most frequently used one. It simply assigns a cost (loss) of one to the failure to guess the correct class. In some situations, different types of misclassification have different costs associated with them, and the use of a full cost matrix, specifying a loss value for each $(C(E), C_X(E))$ pair, will then be appropriate. (For example, in medical diagnosis the cost of diagnosing an ill patient as healthy is generally different from that of diagnosing a healthy patient as ill.)

In practice, it often occurs that examples with exactly the same attribute values have different classes. This reflects the fact that those attributes do not contain all the information necessary to uniquely determine the class. In general, then, an example E will not be associated with a single class, but rather with a vector of class probabilities $P(C_i|E)$, where the ith component represents the fraction of times that E appears with class C_i. The zero-one loss or *misclassification rate* of X on E is then more generally defined as

$$L_X(E) = 1 - P(C_X|E), \tag{7}$$

where $C_X(E)$, the class assigned by X to E, is abbreviated to C_X for simplicity. $P(C_X|E)$ is the accuracy of X on E. This definition reduces to Equation 6 when one class has probability 1 given E.

DEFINITION 2 *The* Bayes rate *for an example is the lowest zero-one loss achievable by any classifier on that example (Duda & Hart, 1973).*

DEFINITION 3 *A classifier is* locally optimal *for a given example iff its zero-one loss on that example is equal to the Bayes rate.*

DEFINITION 4 *A classifier is* globally optimal *for a given sample (data set) iff it is locally optimal for every example in that sample. A classifier is globally optimal for a given problem (domain) iff it is globally optimal for all possible samples of that problem (i.e., for all data sets extracted from that domain).*

The use of zero-one loss for classification tasks should be contrasted with that of *squared error loss* for probability estimation. This is defined as

$$SE_X(E) = [P(C|E) - P_X(C|E)]^2, \tag{8}$$

where X is the estimating procedure and C is the variable whose probability (or probability density) we seek to estimate. If there is uncertainty associated with $P(C|E)$, the squared error loss is defined as the expected value of the above expression. The main point of this article, shown in this section, can now be stated as follows. When the independence assumption is violated, Equation 2 will in general be suboptimal as a probability estimating procedure under the squared error loss function, but combined with Equation 1 it can nevertheless still be optimal as a classification procedure under the zero-one loss function. This result is a direct consequence of the differing properties of these two loss measures: Equation 2 yields minimal squared-error estimates of the class probabilities only when the estimates are equal to the true values (i.e., when the independence assumption holds); but, with Equation 1, it can yield minimal zero-one loss even when the class probability estimates diverge widely from the true values, as long as the class with highest estimated probability, $C_X(E)$, is the class with highest true probability.

For instance, suppose there are two classes + and −, and let $P(+|E) = 0.51$ and $P(-|E) = 0.49$ be the true class probabilities given example E. The optimal classification decision is then to assign E to class + (i.e., to set $C_X(E) = +$). Suppose also that Equation 2 gives the estimates $\hat{P}(+|E) = f_+(E) = 0.99$ and $\hat{P}(-|E) = f_-(E) = 0.01$. The independence assumption is violated by a wide margin, and the squared-error loss is large, but the Bayesian classifier still makes the optimal classification decision, minimizing the zero-one loss.

Consider the two-class case in general. Let the classes be + and − as before, $p = P(+|E)$, $r = P(+) \prod_{j=1}^{a} P(A_j = v_{jk}|+)$, and $s = P(-) \prod_{j=1}^{a} P(A_j = v_{jk}|-)$ (refer to Equation 2). We will now derive a necessary and sufficient condition for the local optimality of the Bayesian classifier, and show that the volume of the Bayesian classifier's region of optimality in the space of valid values of (p, r, s) is half of this space's total volume.

THEOREM 1 *The Bayesian classifier is locally optimal under zero-one loss for an example E iff* $(p \geq \frac{1}{2} \wedge r \geq s) \vee (p \leq \frac{1}{2} \wedge r \leq s)$ *for E.*

Proof: The Bayesian classifier is optimal when its zero-one loss is the minimum possible. When $p = P(+|E) > \frac{1}{2}$, the minimum loss is $1 - p$, and is obtained by assigning E to class +. The Bayesian classifier assigns E to class + when $f_+(E) > f_-(E)$ according to Equation 2, i.e., when $r > s$. Thus if $p > \frac{1}{2} \wedge r > s$ the Bayesian classifier is optimal. Conversely, when $p = P(+|E) < \frac{1}{2}$, the minimum zero-one loss is p, and is obtained by assigning E to class −, which the Bayesian classifier does when $r < s$. Thus the Bayesian classifier is optimal when $p < \frac{1}{2} \wedge r < s$. When $p = \frac{1}{2}$, either decision is optimal, so the inequalities can be generalized as shown. ∎

Note that this is not an asymptotic result: it is valid even when the probability estimates used to compute r and s are obtained from finite samples.

COROLLARY 1 *The Bayesian classifier is locally optimal under zero-one loss in half the volume of the space of possible values of* (p, r, s).

Proof: Since p is a probability, and r and s are products of probabilities, (p, r, s) only takes values in the unit cube $[0, 1]^3$. The region of this cube satisfying the condition in Theorem 1 is shown shaded in Figure 2; it can easily be seen to occupy half of the total volume of the cube. However, not all (r, s) pairs correspond to valid probability combinations. Since p is unconstrained, the projection of the space U of

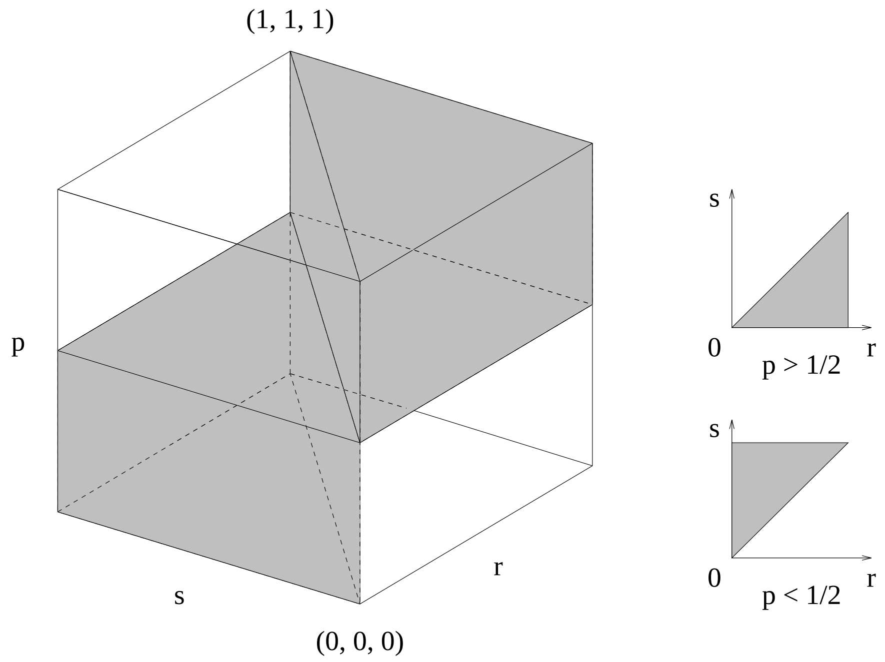

Figure 2. Region of optimality of the simple Bayesian classifier.

valid probability combinations on all planes $p = p_0$ is the same. By Theorem 1, the region of optimality on planes below $p_0 = \frac{1}{2}$ becomes the region of nonoptimality on planes above $p_0 = \frac{1}{2}$, and vice versa (i.e., the optimal region for projections below $p_0 = \frac{1}{2}$ is the photographic negative of the optimal region for projections above). Thus, if S is the area of U's projection and S_O is the area of the optimal region for $p_0 < \frac{1}{2}$, the area of the optimal region for $p_0 > \frac{1}{2}$ is $S - S_O$, and the total volume of the region of optimality is $\frac{1}{2}S_O + \frac{1}{2}(S - S_O) = \frac{1}{2}S$. (Also, since if (r, s) corresponds to a valid probability combination then so does (s, r), the region of optimality is symmetric about $s = r$, and therefore $S_O = \frac{1}{2}S$ both above and below $p_0 = \frac{1}{2}$.) ■

In contrast, under squared error loss, Equation 2 is optimal as a set of probability estimates $P(C_i|E)$ only when the independence assumption holds, i.e., on the line where the planes $r = p$ and $s = 1 - p$ intersect. Thus the region of optimality of Equation 2 under squared-error loss is a second-order infinitesimal fraction of its region of optimality under zero-one loss. The Bayesian classifier is effectively an optimal predictor of the most likely class for a broad range of conditions in which the independence assumption is violated. Previous notions of the Bayesian classifier's limitations can now be seen as resulting from incorrectly applying intuitions based on squared-error loss to the Bayesian classifier's performance under zero-one loss.

6. Global optimality

The extension of Theorem 1 to global optimality is immediate. Let p, r and s for example E be indexed as p_E, r_E and s_E.

THEOREM 2 *The Bayesian classifier is globally optimal under zero-one loss for a sample (data set)* Σ *iff* $\forall\, E \in \Sigma\; (p_E \geq \frac{1}{2} \wedge r_E \geq s_E) \vee (p_E \leq \frac{1}{2} \wedge r_E \leq s_E)$.

Proof: By Definition 4 and Theorem 1. ■

However, verifying this condition directly on a test sample will in general not be possible, since it involves finding the true class probabilities for all examples in the sample. Further, verifying it for a given domain (i.e, for all possible samples extracted from that domain) will in general involve a computation of size proportional to the number of possible examples, which is exponential in the number of attributes, and therefore computationally infeasible. Thus the remainder of this section is dedicated to investigating more concrete conditions for the global optimality of the Bayesian classifier, some necessary and some sufficient. A zero-one loss function is assumed throughout.

6.1. Necessary conditions

Let a be the number of attributes, as before, let c be the number of classes, let v be the maximum number of values per attribute, and let d be the number of different numbers representable on the machine implementing the Bayesian classifier. For example, if numbers are represented using 16 bits, $d = 2^{16} = 65536$.

THEOREM 3 *The Bayesian classifier cannot be globally optimal for more than* $d^{c(av+1)}$ *different problems.*

Proof: Since the Bayesian classifier's state is composed of $c(av+1)$ probabilities, and each probability can only have d different values, the Bayesian classifier can only be in at most $d^{c(av+1)}$ states, and thus it cannot distinguish between more than this number of concepts. ■

Even though $d^{c(av+1)}$ can be very large, this is a significant restriction because many concept classes have size doubly exponential in a (e.g., arbitrary DNF formulas in Boolean domains), and due to the extremely rapid growth of this function the Bayesian classifier's capacity will be exceeded even for commonly-occurring values of a. On the other hand, this restriction is compatible with concept classes whose size grows only exponentially with a (e.g., conjunctions).

This result reflects the Bayesian classifier's limited capacity for information storage, and should be contrasted with the case of classifiers (like instance-based, rule and decision tree learners) whose memory size can be proportional to the sample size. It also shows that the condition in Theorem 2 is satisfied by an exponentially decreasing fraction of all possible domains as a increases. This is consistent with the fact that local optimality must be verified for every possible combination of attribute values if the Bayesian classifier is to be globally optimal for a domain (Definition 4), and the probability of this decreases exponentially with a, starting at 100% for $a = 1$. However, a similar statement is true for other learners; it simply reflects the fact that it is very difficult to optimally learn a very wide class of concepts. The information storage capacity of the Bayesian classifier is $O(a)$. If e is the training set size, learners that can memorize all the individual examples (or the equivalent) have a storage capacity of $O(ea)$, and therefore can in principle converge to optimal when $e \to \infty$. However, for any finite e there is a value of a after which the fraction of problems on which those learners can be optimal also starts to decrease exponentially with a.

Let a *nominal* attribute be defined as one whose domain is finite and unordered, a *feature* be defined as an attribute with a given value (i.e., $A_j = v_{jk}$ is a feature), and a set of classes be *discriminable* by a set of functions $f_i(E)$ if every possible example E can be optimally classified by applying Equation 1 with this set of functions. Then the following result is an immediate extension to the general nominal case of a well-known one for Boolean attributes (Duda & Hart, 1973).

THEOREM 4 *When all attributes are nominal, the Bayesian classifier is not globally optimal for classes that are not discriminable by linear functions of the corresponding features.*

Proof: Define one Boolean attribute b_{jk} for each feature, i.e., $b_{jk} = 1$ if $A_j = v_{jk}$ and 0 otherwise, where v_{jk} is the kth value of attribute A_j. Then, by taking the logarithm of Equation 2, the Bayesian classifier is equivalent to a linear machine (Duda & Hart, 1973) whose discriminant function for class C_i is $\log P(C_i) + \sum_{j,k} \log P(A_j = v_{jk}|C_i)\, b_{jk}$ (i.e., the weight of each Boolean feature is the log-probability of the corresponding attribute value given the class). ∎

This is not a sufficient condition, because the Bayesian classifier cannot learn some linearly separable concepts. For example, it fails for some m-of-n concepts, even though they are linearly separable. An m-of-n concept is a Boolean concept that is true if m or more out of the n attributes defining the example space are true. For example, if examples are described by three attributes A_0, A_1 and A_2, the concept 2-of-3 is true if A_0 and A_1 are true, or A_0 and A_2 are true, or A_1 and A_2 are true, or all three are true.[6]

THEOREM 5 *The Bayesian classifier is not globally optimal for* m*-of-*n *concepts.*

Proof: This follows directly from the definition of global optimality, and the fact that there exist m-of-n concepts for which the Bayesian classifier makes errors, even when the examples are noise-free (i.e., an example always has the same class) and the Bayes rate is therefore zero (e.g., 3-of-7, Kohavi, 1995). ∎

Let $P(A|C)$ represent the probability that an arbitrary attribute A is true given that the concept C is true, let a bar represent negation, and let all examples be equally probable. In general, if the Bayesian classifier is trained with all 2^n examples of an m-of-n concept, and a test example has exactly j true-valued attributes, then the Bayesian classifier will make a false positive error if $\mathit{Diff}(m, n, j)$ is positive and $j < m$, and it will make a false negative error if $\mathit{Diff}(m, n, j)$ is negative and $j \geq m$, where

$$\begin{aligned}
\mathit{Diff}(m,n,j) &= P(C)\,P(A|C)^j\,[1-P(A|C)]^{n-j} \\
&\quad -P(\overline{C})\,P(A|\overline{C})^j\,[1-P(A|\overline{C})]^{n-j} \\
P(C) &= \frac{\sum_{i=m}^{n}\binom{n}{i}}{2^n} \\
P(\overline{C}) &= \frac{\sum_{i=0}^{m-1}\binom{n}{i}}{2^n} \\
P(A|C) &= \frac{\sum_{i=m-1}^{n-1}\binom{n-1}{i}}{\sum_{i=m}^{n}\binom{n}{i}} \\
P(A|\overline{C}) &= \frac{\sum_{i=0}^{m-2}\binom{n-1}{i}}{\sum_{i=0}^{m-1}\binom{n}{i}}.
\end{aligned}$$

For example, $\mathit{Diff}(8, 25, j)$ is positive for all $j \geq 6$. Therefore, the Bayesian classifier makes false positive errors for all examples that have 6 or 7 attributes that are true. Similarly, $\mathit{Diff}(17, 25, j)$ is negative for all $j \geq 19$ and the Bayesian classifier makes false negative errors when there are 17 and 18 attributes that are true. However, a simple modification of the Bayesian classifier will allow it to perfectly discriminate all positive examples from negatives: adding a constant to the discriminant function for the concept, or subtracting the same constant from the discriminant function for its negation (Equation 1). We have implemented an extension to the Bayesian classifier for two-class problems that finds the value of the constant that maximizes predictive accuracy on the training data. In preliminary experiments, we have observed that this extension achieves 100% accuracy on all m-of-n concepts when trained on all 2^n examples, for n less than 18. Furthermore, we have tested this extension on the mushroom data set from the UCI repository with 800 examples, and found that the average accuracy on 64 trials significantly increased from 93.9% without this extension to 96.2% with this extension (with 99.9% confidence using a one-tailed paired t test).

Since in nominal domains the basic Bayesian classifier cannot learn some linearly separable concepts, in these domains its range of optimality is a subset of the perceptron's, or of a linear machine's (Duda & Hart, 1973). This leads to the following result.

Let the Vapnik-Chervonenkis dimension, or *VC dimension* for short, be defined as in (Haussler, 1988).

COROLLARY 2 *In domains composed of* a *nominal attributes, the VC dimension of the simple Bayesian classifier is* O(a).

Proof: This result follows immediately from Theorem 4 and the fact that, given a attributes, the VC dimension of linear discriminant functions is $O(a)$ (Haussler, 1988). ■

Thus, in nominal domains, the PAC-learning guarantees that apply to linear machines apply also to the Bayesian classifier. In particular, given a classification problem for which the Bayesian classifier is optimal, the number of examples required for it to learn the required discrimination to within error ϵ with probability $1 - \delta$ is linear in the number of attributes a.

In numeric domains, the Bayesian classifier is not restricted to linearly separable problems; for example, if classes are normally distributed, nonlinear boundaries and multiple disconnected regions can arise, and the Bayesian classifier is able to identify them (see Duda & Hart, 1973).

6.2. Sufficient conditions

In this section we establish the Bayesian classifier's optimality for some common concept classes.

THEOREM 6 *The Bayesian classifier is globally optimal if, for all classes* C_i *and examples* $E = (v_1, v_2, \ldots, v_a)$, $P(E|C_i) = \prod_{j=1}^{a} P(A_j = v_j|C_i)$.

This result was demonstrated in Section 1, and is restated here for completeness. The crucial point is that this condition is sufficient, but not necessary.

THEOREM 7 *The Bayesian classifier is globally optimal for learning conjunctions of literals.*

Proof: Suppose there are n literals L_j in the conjunction. A literal may be a Boolean attribute or its negation. In addition, there may be $a - n$ irrelevant attributes; they simply cause each row in the truth table to become 2^{a-n} rows with the same values for the class and all relevant attributes, each of those rows corresponding to a possible combination of the irrelevant attributes. For simplicity, they will be ignored from here on (i.e., $n = a$ will be assumed without loss of generality). Recall that, in the truth table for conjunction, the class C is 0 (false) for all but $L_0 = L_1 = \cdots = L_n = 1$ (true). Thus, using a bar to denote negation, $P(C) = \frac{1}{2^n}$, $P(\overline{C}) = \frac{2^n-1}{2^n}$, $P(L_j|C) = 1$, $P(\overline{L_j}|C) = 0$, $P(\overline{L_j}|\overline{C}) = \frac{2^{n-1}}{2^n-1}$ (the number of times the literal is 0 in the truth table, divided by the number of times the class is 0), and $P(L_j|\overline{C}) = \frac{2^{n-1}-1}{2^n-1}$ (the number of times the literal is 1 minus the one time it corresponds to C, divided by the number of times the class is 0). Let E be an arbitrary example, and let m of the conjunction's literals be true in E. For simplicity, the factor $1/P(E)$ will be omitted from all probabilities. Then we have

$$P(C|E) \;=\; P(C)\,P^m(L_j|C)\,P^{n-m}(\overline{L_j}|C) \;=\; \begin{cases} \frac{1}{2^n} & \text{if } m = n \\ 0 & \text{otherwise} \end{cases}$$

and

$$\begin{aligned} P(\overline{C}|E) \;&=\; P(\overline{C})\,P^m(L_j|\overline{C})\,P^{n-m}(\overline{L_j}|\overline{C}) \\ &=\; \frac{2^n-1}{2^n}\left(\frac{2^{n-1}-1}{2^n-1}\right)^m\left(\frac{2^{n-1}}{2^n-1}\right)^{n-m}. \end{aligned}$$

Notice that $\frac{2^{n-1}-1}{2^n-1} < \frac{1}{2}$ for all n. Thus, for $m = n$, $P(\overline{C}|E) = P(\overline{C})\left(\frac{2^{n-1}-1}{2^n-1}\right)^n < P(\overline{C})(\frac{1}{2})^n < \frac{1}{2^n} = P(C|E)$, and class 1 wins. For all $m < n$, $P(C|E) = 0$ and $P(\overline{C}|E) > 0$, and thus class 0 wins. Therefore the Bayesian classifier always makes the correct decision, i.e., it is globally optimal. ∎

Conjunctive concepts satisfy the independence assumption for class 1, but not for class 0. (For example, if $C = A_0 \wedge A_1$, $P(A_1|\overline{C}) = \frac{1}{3} \neq P(A_1|\overline{C}, A_0) = 0$, by inspection of the truth table.) Thus conjunctions are an example of a class of concepts where the Bayesian classifier is in fact optimal, but would not be if it required attribute independence.

This analysis assumes that the whole truth table is known, and that all examples are equally likely. What will happen if either of these restrictions is removed? Consider first the case where examples are not distributed uniformly. For $m < n$, the Bayesian classifier always produces the correct class, given a sufficient sample. For $m = n$, the result will, in general, depend on the distribution. The more interesting and practical case occurs when $P(C) > \frac{1}{2^n}$, and in this case one can easily verify that the Bayesian classifier continues to give the correct answers (and, in fact, is now more robust with respect to sample fluctuations). It will fail if $P(C) < \frac{1}{2^n}$, but this is a very artificial situation: in practice, examples of such a conjunction would never appear in the data, or they would appear so infrequently that learning the conjunction would be of little or no relevance to the accuracy.

At first sight, the Bayesian classifier can also fail if the probabilities $P(L_j|\overline{C})$ are such that the product of all n such probabilities is greater than $\frac{1}{2^n}$ (or, more precisely, greater than $P(C|E)/P(\overline{C})$). $P(L_j|\overline{C})$ can be increased by increasing the frequency with which L_j is 1 but the class is not (i.e., at least one of the other literals in the conjunction is 0). However, doing this necessarily decreases $P(C)$, leading to the artificial situation just described. Further, because increasing $P(L_j|\overline{C})$ also decreases $P(L_k|\overline{C})$ for the L_k that are 0 when L_j is 1 and the class is 1, it can be shown that the product can never be greater than $\frac{1}{2^n}$. Thus, a very small $P(C)$ is effectively the only situation where the Bayesian classifier will not be optimal. In short, although distributional assumptions cannot be entirely removed, they can be relaxed to exclude only the more pathological cases.

The Bayesian classifier's average-case behavior for insufficient samples (i.e., samples not including all possible examples) was analyzed by Langley et al. (1992), who plotted sample cases and found the rate of convergence to 100% accuracy to be quite rapid.[7] Comparing Langley et al.'s results with Pazzani and Sarrett's (1990) average-case formulas for the classical wholist algorithm for learning conjunctions shows that the latter converges faster, which is not surprising, considering that it was specifically designed for this concept class. On the other hand, as Langley et al. (1992) point out, the Bayesian classifier has the advantage of noise tolerance.

THEOREM 8 *The Bayesian classifier is globally optimal for learning disjunctions of literals.*

Proof: Similar to that for Theorem 7, letting m be the number of the disjunction's literals that are false in E. ■

Conversely, disjunctions satisfy the independence assumption for class 0 but not for class 1, and are another example of the Bayesian classifier's optimality even when the independence assumption is violated.

As corollaries, the Bayesian classifier is also optimal for negated conjunctions and negated disjunctions, as well as for the identity and negation functions, with any number of irrelevant attributes.

7. When will the Bayesian classifier outperform other learners?

The previous sections showed that the Bayesian classifier is, in fact, optimal under a far broader range of conditions than previously thought. However, even when it is not optimal, the Bayesian classifier may still perform better than classifiers with greater representational power, such as C4.5, PEBLS and CN2, with which it was empirically compared in Section 3. Thus, a question of practical significance arises: is it possible to identify conditions under which the Bayesian classifier can be expected to do well, compared to these other classifiers? The current state of knowledge in the field does not permit a complete and rigorous answer to this question, but some elements can be gleaned from the results in this article, and from the literature.

It is well known that squared error loss can be decomposed into three additive components (Friedman, 1996): the intrinsic error due to noise in the sample, the statistical *bias* (systematic component of the approximation error, or error for an infinite sample) and the *variance* (component of the error due to the approximation's sensitivity to the sample, or error due to the sample's finite size). A trade-off exists between bias and variance, and knowledge of it can often help in understanding the relative behavior of estimation algorithms: those with greater representational power, and thus greater ability to respond to the sample, tend to have lower bias, but also higher variance.

Recently, several authors (Kong & Dietterich, 1995; Kohavi & Wolpert, 1996; Tibshirani, 1996; Breiman, 1996; Friedman, 1996) have proposed similar bias-variance decompositions for zero-one loss functions. In particular, Friedman (1996) has shown, using normal approximations to the class probabilities, that the bias-variance interaction now takes a very different form. Zero-one loss can be highly insensitive to squared-error bias in the classifier's probability estimates, as Theorem 1 implies,[8] but, crucially, will in general still be sensitive to estimation variance. Thus, as long as Theorem 1's preconditions hold for most examples, a classifier with high bias and low variance will tend to produce lower zero-one loss than one with low bias and high variance, because only the variance's effect will be felt. In this way, the Bayesian classifier can often be a more accurate classifier than (say) C4.5, even if in the infinite-sample limit the latter would provide a better approximation. This may go a significant way towards explaining some of the results in Section 3.

This effect should be especially visible at smaller sample sizes, since variance decreases with sample size. Indeed, Kohavi (1996) has observed that the Bayesian classifier tends to outperform C4.5 on smaller data sets (hundreds to thousands of examples), and conversely for larger ones (thousands to tens of thousands). PAC-learning theory (e.g., Corollary 2) also lends support to this notion: even though it provides only distribution-independent worst-case results, these suggest that good performance on a small sample by the Bayesian classifier (or another limited-capacity classifier) should be predictive of good out-of-sample accuracy, while no similar statement can be made for classifiers with VC dimension on the order of C4.5's. Further, since the VC dimension of a classifier typically increases with the number of attributes, the Bayesian classifier should be particularly favored when, in addition to being small, the sample consists of examples described by many attributes.

These hypotheses were tested by conducting experiments in artificial domains. The independent variables were the number of examples n and the number of attributes a, and the dependent variables were the accuracies of the Bayesian classifier and C4.5. Concepts defined as Boolean functions in disjunctive normal form (i.e., sets of rules) were used. The number of literals in each disjunct (i.e., the number of conditions in each rule) was set according to a binomial distribution with mean d and variance $d(a-d)$; this is obtained by including each attribute in the disjunct with probability d/a (negated or not with equal probability). The number of disjuncts was set to $2^d - 1$, so as to produce approximately equal numbers of positive and negative examples, and positive examples were distributed evenly among the disjuncts. The number of examples n was varied between 10 and 10000, and a was varied between 16 and 64. A value of $d = 8$ was used, reflecting a bias for concepts of intermediate complexity ($d = 1$ would produce the simplest concepts, and $d = a$ the most complex ones). One hundred different domains were generated at random for each (n, a) pair. For each domain, n examples were generated for training, and 1000 for testing. Test-set accuracy was then averaged across domains. The C4.5RULES postprocessor, which converts decision trees to rules and thus better matches the target concept class, was used, and found to indeed increase accuracy, by as much as 10% for larger n. All the results reported are for C4.5RULES.

The results appear graphically in Figure 3. All accuracy differences are significant with 99.9% confidence using a one-tailed paired t test.[9] For this broad class of domains, the Bayesian classifier is indeed more accurate than C4.5 at smaller sample sizes (up to 1000, which includes many practical situations), and the crossover point increases with the number of attributes, as does the Bayesian classifier's accuracy advantage up to that point. These results are especially remarkable in light of the fact that C4.5RULES's learning bias is far more appropriate to these domains than the Bayesian classifier's, and illustrate how far variance can dominate bias as a source of error in small to medium data sets. This can be seen as follows. Since the Bayes rate is zero for these domains, the only components of the error are bias and variance. If bias is taken to be the asymptotic error (i.e., the error for an infinite sample), and variance the difference between total error for a given sample size and the bias (i.e., the "finite sample penalty"), then C4.5's bias is zero, since its accuracy asymptotes at 100%, and the Bayesian classifier has a high bias (approximately 30–35%, depending on the number of attributes). On the other hand, C4.5's variance, which approaches 50% for the smaller sample sizes, is much higher than the Bayesian classifier's, and thus the sum of bias and variance for C4.5 is greater than that for the Bayesian classifier up to the crossover point.

Other authors have verified by Monte Carlo simulation that "choosing a simple method of discrimination is often beneficial even if the underlying model assumptions are wrong" (Flury, Schmid, & Narayanan (1994) for quadratic discriminant functions; Russek, Kronmal, & Fisher (1983) for the Bayesian classifier vs. multivariate Gaussian models). In general, the amount of structure that can be induced for a domain will be limited by both the available sample and the learner's representational power. When the sample is the dominant limiting factor, a simple learner like the Bayesian classifier may be better. However, as the sample size increases, the Bayesian classifier's capacity to store information about the domain will be exhausted sooner than that of more powerful classifiers, and it may then make sense to use the latter. Of course, the Bayesian classifier may still outperform other classifiers at larger samples sizes, if its learning bias happens to be more appropriate for the domain.

The Bayesian classifier's exact degree of sensitivity to variance will depend on the difference $r - s$, for r and s (see Section 5) estimated from an infinite sample. If this difference is large, errors in r and s due to small sample size will tend to

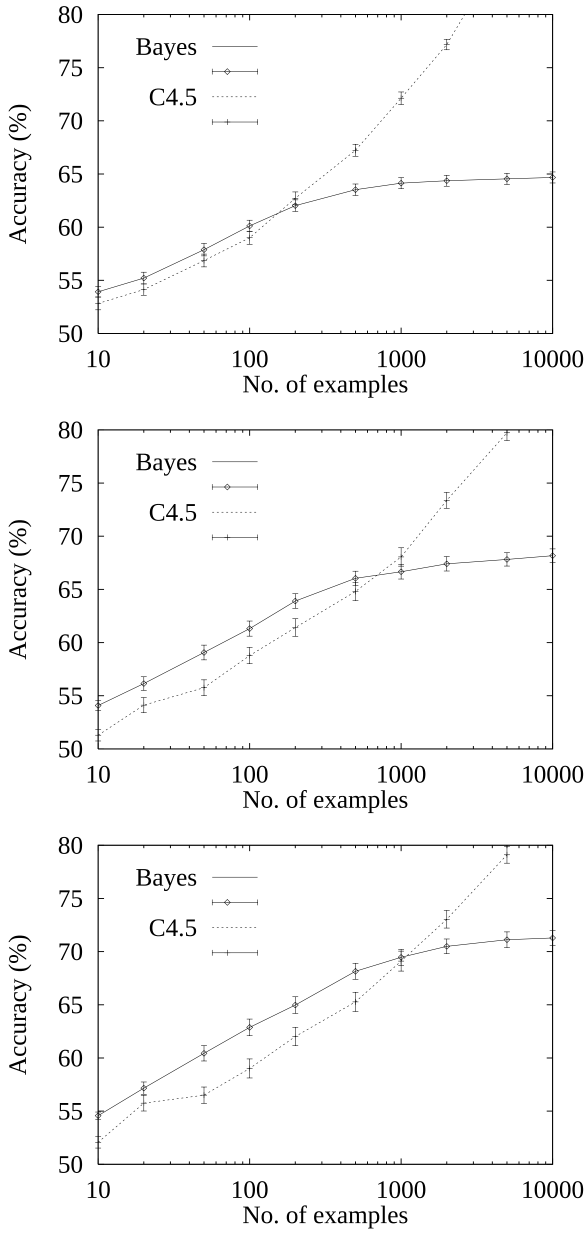

Figure 3. Accuracy of the Bayesian classifier and C4.5RULES as a function of the number of examples, given 16 attributes (upper), 32 attributes (middle), and 64 attributes (lower). Error bars have a height of two standard deviations of the sample mean. All accuracy differences are significant with 99.9% confidence using a one-tailed paired t test.

leave the sign of $r - s$ unchanged, and thus have no effect. On the other hand, if $r \simeq s$, even small errors can cause the sign to change. If p and the infinite-sample values of r and s satisfy the preconditions of Theorem 1, this will lead to classification errors. Conversely, if they do not, this will lead to a reduction in the misclassification rate, because incorrect classifications will be flipped to correct ones. Thus an increase in variance can sometimes lead to a reduction in zero-one loss. Overall, Ben-Bassat, Klove, and Weil (1980) have shown that the Bayesian classifier is quite robust with respect to errors in probability estimates due to small sample size; this is not surprising, since it can be attributed to the same factors that make it robust with respect to violations of the independence assumption.

8. How is the Bayesian classifier best extended?

One significant consequence of the Bayesian classifier's optimality even when strong attribute dependences are present is that detecting these is not necessarily the best way to improve performance. This section empirically tests this claim by comparing Pazzani's (1996) extension with one that differs from it solely by using the method for attribute dependence detection described in (Kononenko, 1991) and (Wan & Wong, 1989). In each case, the algorithm finds the single best pair of attributes to join by considering all possible joins. Two measures for determining the best pair were compared. Following Pazzani (1996), the first measure was estimated accuracy, as determined by leave-one-out cross validation on the training set. In the second measure, Equation 4 was used to find the attributes that had the largest violation of the conditional independence assumption.

To conduct an experiment to compare these two approaches, a method is also required to decide when to stop joining attributes. Rather than selecting an arbitrary threshold, experiments were conducted in two ways:

- Joining only a single pair of attributes using each evaluation measure (provided the change appeared beneficial to the measure).
- With the cross-validation measure, joining of attributes stopped when no further joining resulted in an improvement. With Equation 4, the optimal stopping criterion was assumed to be given by an oracle. This was implemented by selecting the threshold that performed best on the test data.

Two artificial concepts were used to compare the approaches: exclusive OR with two relevant attributes and six irrelevant attributes, and parity with six relevant attributes and six irrelevant attributes. Experiments on UCI data sets were also carried out, to determine whether the methods work on problems that occur in practice as well as in artificial concepts. In this set of experiments, a multiplicative factor of 1 was used for the Laplace correction (see Section 3), and numeric attributes were discretized into five equal intervals, instead of ten. This causes the Cartesian product of two discretized attributes to have 25 values, instead of 100, and leads to substantially more reliable probability estimates, given that the

Table 4. A comparison of two approaches to extending the Bayesian classifier.

Data Set	Training Size	Bayes	Accuracy Once	Entropy Once	Accuracy Repeated	Entropy Optimal
Exclusive OR	128	46.1	100.0	100.0	100.0	100.0
4-parity	128	42.4	43.5	44.2	50.3	51.0
Chess endgames	300	86.8	93.4 +	90.3	93.9 +	90.8
Credit	250	84.0	83.7	84.1	84.0	84.6
Diabetes	500	75.5	76.1	76.1	76.1	76.1
Glass	150	41.7	48.9 +	42.6	49.3 +	42.6
Horse colic	200	81.0	80.8	79.5	80.6	81.1
Iris	100	93.1	93.2	93.3	93.3	93.6
Mushroom	800	94.0	97.4 +	93.4	99.3 +	94.0
Voting records	300	90.4	90.4	89.9	92.0	91.2
Wine	125	98.0	97.5	97.7	97.5	98.0
Wisconsin cancer	500	97.3	96.7	96.7	97.0	96.7

training set sizes are in the hundreds. The domains and training set sizes appear in the first two columns of Table 4. The remaining columns display the accuracy of the Bayesian classifier and extensions, averaged over 24 paired trials, and found by using an independent test set consisting of all examples not in the training set.

In Table 4, *Accuracy Once* shows results for the backward stepwise joining algorithm of Pazzani (1996), forming at most one Cartesian product as determined by the highest accuracy using leave-one-out cross validation on the training set; *Entropy Once* is the same algorithm except it creates at most one Cartesian product with the two attributes that have the highest degree of dependence. In this table, a paired t test between these two algorithms is used to determine which method has the highest accuracy when making a single change to the Bayesian classifier. A "+" indicates that using one method is significantly more accurate than another. Both algorithms do well on exclusive OR. In this case the joining of the two relevant attributes is clearly distinguished from others by either method. The results indicate that estimating accuracy on the training data is significantly better on three data sets and never significantly worse than using a measure of conditional independence.

The column labeled *Accuracy Repeated* gives results for the backward sequential joining algorithm; in contrast, *Entropy Optimal* repeats joining the pair of attributes that have the highest degree of dependence, stopping when the dependences fall below the optimal threshold to maximize accuracy on the test set. Paired t tests indicate that the accuracy estimation approach is often significantly better than using entropy to determine which attributes to join, and is never significantly worse.

To further explore whether the degree of dependence is a reasonable measure for predicting which attributes to join, an additional experiment was performed on the UCI data sets in which Cartesian product attributes were beneficial: we formed every possible classifier with a single pair of joined attributes (and all remaining attributes), and measured the test-set accuracy, the accuracy estimated by leave-

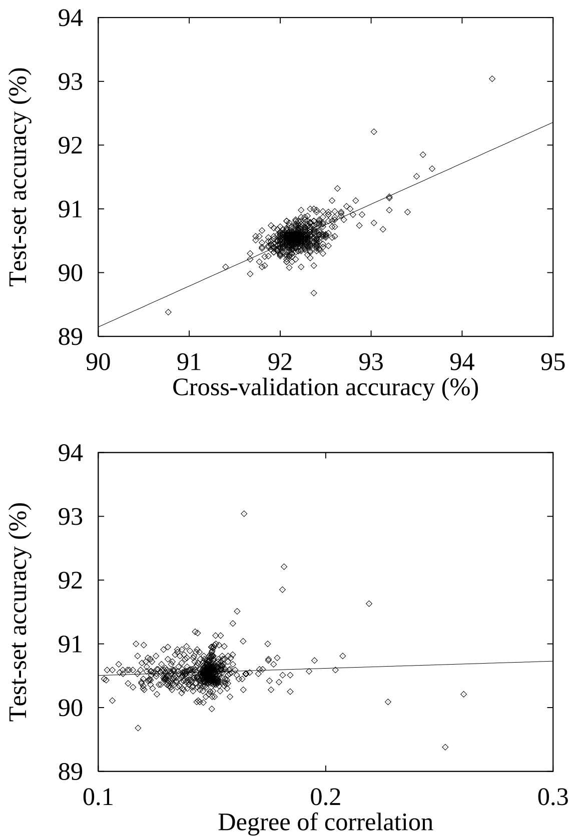

Figure 4. Upper: The relationship between accuracy on the test set and using accuracy estimation on the training set to decide which Cartesian product attribute to form, plotted for all pairs of attributes in the chess data set ($R^2 = 0.497$). Lower: The relationship between accuracy on the test set and using entropy to decide which Cartesian product attribute to form ($R^2 = 0.006$).

one-out cross validation on the training set, and the degree of dependence. Figure 4 plots the accuracy of these classifiers on the test set as a function of the other two measures (averaged over 24 trials) for the domain with the largest number of attributes: chess endgames. The graphs show that cross-validation accuracy is a better predictor of the effect of an attribute join than the degree of dependence given the class. The value of R^2 for this domain was 0.497 for cross-validation accuracy, vs. 0.006 for degree of dependence. For the voting domain, the values of R^2 were respectively 0.531 and 0.212, for the glass domain 0.242 and 0.001, and for mushroom 0.907 and 0.019.

These experiments demonstrate that joining attributes to correct for the most serious violations of the independence assumption does not necessarily yield the most accurate classifier. To illustrate the reason for this finding, we constructed examples of an artificial concept with six variables. The concept is true whenever two or more of A_1, A_5, and A_6 are true and two or more of A_2, A_3, and A_4 are true. We generated examples in which A_1 had a 50% chance of being true, an all other attributes A_i had a probability $1/i$ of having the same value as A_1. Otherwise, the value was selected randomly with a 50% chance of being true. Therefore, attributes A_1 and A_2 were the most dependent. To avoid problems of estimating probabilities from small samples, we ran each algorithm on 500 examples generated as described above and tested on a set of 500 examples generated in the same manner. We ran 24 trials of this procedure. Using this methodology, the simple Bayesian classifier was only 92.8% accurate on this problem. When using the entropy-based approach to finding a pair of attributes to join, A_1 and A_2 were always chosen, and the classifier was significantly less accurate at 90.1%. In contrast, when using cross-validation accuracy to determine which two attributes to join, A_5 and A_6 were always chosen. These are the two least dependent attributes in the data, yet thhe accuracy of the Bayesian classifier constructed in this manner was significantly higher, at 96.9%. This occurs because on this problem the representational bias of the simple Bayesian classifier presents more difficulties than the independence assumption.

The experiments in this section show that the simple Bayesian classifier can be productively extended. However, correcting the largest violation of the independence assumption does not necessarily result in the largest improvement. Rather, since under zero-one loss the Bayesian classifier can tolerate some significant violations of the independence assumption, an approach that directly estimates the effect of the possible changes on this loss measure resulted in a more substantial improvement.

9. Conclusions and future work

In this article we verified that the Bayesian classifier performs quite well in practice even when strong attribute dependences are present. We also showed that this follows at least partly from the fact that, contrary to previous assumptions, the Bayesian classifier does not require attribute independence to be optimal under zero-one loss. We then derived some necessary and some sufficient conditions for the Bayesian classifier's optimality. In particular, we showed that the Bayesian classifier is an optimal learner for conjunctive and disjunctive concepts, even though these violate the independence assumption. We hypothesized that the Bayesian classifier may often be a better classifier than more powerful alternatives when the sample size is small, even in domains where its learning model is not the most appropriate one, and verified this by means of experiments in artificial domains. We also verified that searching for attribute dependences is not necessarily the best approach to improving the Bayesian classifier's performance.

Ideally, we would like to have a complete set of necessary and sufficient conditions for the optimality of the Bayesian classifier, efficiently verifiable on real problems. In Section 6 we began the work towards this goal. Another important area of future research concerns finding conditions under which the Bayesian classifier is not optimal, but comes very close because it makes the wrong prediction on only a small fraction of the examples. This should also shed further light on the discussion in Section 7. Much work remains to be done in the continuation of this section, further elucidating the conditions that will favor the Bayesian classifier over other classifiers. Another useful extension of the present work would be to apply a similar analysis to loss functions employing a full cost matrix (see Section 5).

In summary, the work reported here demonstrates that the Bayesian classifier has much broader applicability than previously thought. Since it also has advantages in terms of simplicity, learning speed, classification speed, storage space and incrementality, its use should perhaps be considered more often.

Acknowledgments

The first author was partly supported by PRAXIS XXI and NATO scholarships. The authors are grateful to the creators of the C4.5, PEBLS and CN2 systems, and to all those who provided the data sets used in the empirical study. Please see the documentation in the UCI Repository for detailed information.

Notes

1. If there is a tie, the class may be chosen randomly.
2. This article will not attempt to review work on the Bayesian classifier in the pattern recognition literature. Journals where this work can be found include *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *Pattern Recognition Letters*, and *Pattern Recognition*.
3. These confidence levels should be interpreted with caution, due to the t test's assumption of independently drawn samples. Thus, a 99% level for a data set means the Bayesian classifier can be expected with high confidence to outperform the corresponding algorithm on training sets drawn at random from that data set, since the accuracy results were obtained by independently drawing training sets from the data set. This is useful for cross-checking the results of this study with previous ones on the same data sets. However, no conclusions can be drawn regarding different data sets drawn at random from the same domain as the UCI data set, because with respect to the domain the training sets used here are not independent, being overlapping subsets of the same data set. See Dietterich (1996) for more on this issue.
4. For any two attributes, Equations 4 and 5 implicitly marginalize over all other attributes. In particular, they ignore that two dependent attributes could become independent given another attribute or combination of attributes.
5. The annealing, audiology, and primary tumor domains are omitted because some of the relevant entropies $H(\ldots)$ could not be computed. Due to a combination of missing values and rare classes, for these data sets there exist C_i and A_j such that $\sum_k \hat{P}(C_i \wedge A_j = v_{jk}) = 0 \neq \hat{P}(C_i)$, causing the entropy measure to become undefined.

6. More generally, some attributes may be irrelevant, i.e., an m-of-n concept may be defined using only $n < a$ attributes, where a is the total number of attributes describing the examples, and one must then specify which attributes are the n relevant ones. This article considers only the more restricted case, but the results can be trivially generalized.
7. The 100% asymptote implies optimality, but the authors did not remark on this fact.
8. Notice that Theorem 1 is valid for any classifier employing estimates r and s of the class probabilities, not just the Bayesian classifier.
9. This includes points where the error bars overlap, which is possible because the t test is paired. Also, note that these confidence levels apply to the accuracy difference in the entire domain class studied, not just a particular data set, since the training sets were drawn independently from the domain class.

References

Ben-Bassat, M., Klove, K. L., & Weil, M. H. (1980). Sensitivity analysis in Bayesian classification models: Multiplicative deviations. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 2*, 261–266.

Breiman, L. (1996). *Bias, variance and arcing classifiers* (Technical Report 460). Statistics Department, University of California at Berkeley, Berkeley, CA. ftp://ftp.stat.berkeley.edu/-users/breiman/arcall.ps.Z.

Cestnik, B. (1990). Estimating probabilities: A crucial task in machine learning. *Proceedings of the Ninth European Conference on Artificial Intelligence.* Stockholm, Sweden: Pitman.

Clark, P., & Boswell, R. (1991). Rule induction with CN2: Some recent improvements. *Proceedings of the Sixth European Working Session on Learning* (pp. 151–163). Porto, Portugal: Springer-Verlag.

Clark, P., & Niblett, T. (1989). The CN2 induction algorithm. *Machine Learning, 3*, 261–283.

Cost, S., & Salzberg, S. (1993). A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning, 10*, 57–78.

DeGroot, M. H. (1986). *Probability and statistics* (2nd ed.). Reading, MA: Addison-Wesley.

Dietterich, T. (1996). *Statistical tests for comparing supervised classification learning algorithms* (technical report). Department of Computer Science, Oregon State University, Corvallis, OR. ftp://ftp.cs.orst.edu/pub/tgd/papers/stats.ps.gz.

Dougherty, J., Kohavi, R., & Sahami, M. (1995). Supervised and unsupervised discretization of continuous features. *Proceedings of the Twelfth International Conference on Machine Learning* (pp. 194–202). Tahoe City, CA: Morgan Kaufmann.

Duda, R. O., & Hart, P. E. (1973). *Pattern classification and scene analysis.* New York, NY: Wiley.

Flury, B., Schmid, M. J., & Narayanan, A. (1994). Error rates in quadratic discrimination with constraints on the covariance matrices. *Journal of Classification, 11*, 101–120.

Friedman, J. H. (1996). *On bias, variance, 0/1 - loss, and the curse-of-dimensionality* (technical report). Department of Statistics, Stanford University, Stanford, CA. ftp://playfair.stanford.-edu/pub/friedman/kdd.ps.Z.

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning* (this volume).

Haussler, D. (1988). Quantifying inductive bias: AI learning algorithms and Valiant's learning framework. *Artificial Intelligence, 36*, 177–221.

John, G., & Langley, P. (1995). Estimating continuous distributions in Bayesian classifiers. *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence* (pp. 338–345). Montréal, Canada: Morgan Kaufmann.

Kohavi, R. (1995). *Wrappers for performance enhancement and oblivious decision graphs.* PhD thesis, Department of Computer Science, Stanford University, Stanford, CA.

Kohavi, R. (1996). Scaling up the accuracy of naive-Bayes classifiers: A decision-tree hybrid. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (pp. 202–207). Portland, OR: AAAI Press.

Kohavi, R., Becker, B., & Sommerfield, D. (1997). *Improving simple Bayes* (technical report). Data Mining and Visualization Group, Silicon Graphics Inc., Mountain View, CA. ftp://starry.-stanford.edu/pub/ronnyk/impSBC.ps.Z.

Kohavi, R., & Wolpert, D. H. (1996). Bias plus variance decomposition for zero-one loss functions. *Proceedings of the Thirteenth International Conference on Machine Learning* (pp. 275–283). Bari, Italy: Morgan Kaufmann.

Kong, E. B., & Dietterich, T. G. (1995). Error-correcting output coding corrects bias and variance. *Proceedings of the Twelfth International Conference on Machine Learning* (pp. 313–321). Tahoe City, CA: Morgan Kaufmann.

Kononenko, I. (1990). Comparison of inductive and naive Bayesian learning approaches to automatic knowledge acquisition. In B. Wielinga (Ed.), *Current Trends in Knowledge Acquisition*. Amsterdam, The Netherlands: IOS Press.

Kononenko, I. (1991). Semi-naive Bayesian classifier. *Proceedings of the Sixth European Working Session on Learning* (pp. 206–219). Porto, Portugal: Springer-Verlag.

Kubat, M., Flotzinger, D., & Pfurtscheller, G. (1993). Discovering patterns in EEG-Signals: Comparative study of a few methods. *Proceedings of the Eighth European Conference on Machine Learning* (pp. 366–371). Vienna, Austria: Springer-Verlag.

Langley, P. (1993). Induction of recursive Bayesian classifiers. *Proceedings of the Eighth European Conference on Machine Learning* (pp. 153–164). Vienna, Austria: Springer-Verlag.

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.

Langley, P., & Sage, S. (1994). Induction of selective Bayesian classifiers. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence* (pp. 399–406). Seattle, WA: Morgan Kaufmann.

Merz, C. J., Murphy, P. M., & Aha, D. W. (1997). UCI repository of machine learning databases. Department of Information and Computer Science, University of California, Irvine, CA. http:-//www.ics.uci.edu/ mlearn/MLRepository.html.

Niblett, T. (1987). Constructing decision trees in noisy domains. *Proceedings of the Second European Working Session on Learning* (pp. 67–78). Bled, Yugoslavia: Sigma.

Pazzani, M. J. (1996). Searching for dependencies in Bayesian classifiers. In D. Fisher & H.-J. Lenz (Eds.), *Learning from data: Artificial intelligence and statistics V* (pp. 239–248). New York, NY: Springer-Verlag.

Pazzani, M., Muramatsu, J., & Billsus, D. (1996). Syskill & Webert: Identifying interesting web sites. *Proceedings of the Thirteenth National Conference on Artificial Intelligence* (pp. 54–61). Portland, OR: AAAI Press.

Pazzani, M., & Sarrett, W. (1990). A framework for average case analysis of conjunctive learning algorithms. *Machine Learning, 9*, 349–372.

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. San Mateo, CA: Morgan Kaufmann.

Russek, E., Kronmal, R. A., & Fisher, L. D. (1983). The effect of assuming independence in applying Bayes' theorem to risk estimation and classification in diagnosis. *Computers and Biomedical Research, 16*, 537–552.

Sahami, M. (1996). Learning limited dependence Bayesian classifiers. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (pp. 335–338). Portland, OR: AAAI Press.

Singh, M., & Provan, G. M. (1995). A comparison of induction algorithms for selective and non-selective Bayesian classifiers. *Proceedings of the Twelfth International Conference on Machine Learning* (pp. 497–505). Tahoe City, CA: Morgan Kaufmann.

Singh, M., & Provan, G. M. (1996). Efficient learning of selective Bayesian network classifiers. *Proceedings of the Thirteenth International Conference on Machine Learning* (pp. 453–461). Bari, Italy: Morgan Kaufmann.

Tibshirani, R. (1996). *Bias, variance and prediction error for classification rules* (technical report). Department of Preventive Medicine and Biostatistics, University of Toronto, Toronto, Ontario. http://utstat.toronto.edu/reports/tibs/biasvar.ps.

Wan, S. J., & Wong, S. K. M. (1989). A measure for concept dissimilarity and its applications in machine learning. *Proceedings of the International Conference on Computing and Information* (pp. 267–273). Toronto, Ontario: North-Holland.

Received Date
Accepted Date
Final Manuscript Date