Machine Learning, 58, 5–24, 2005
2005 Springer Science + Business Media, Inc. Manufactured in The Netherlands.

Not So Naive Bayes: Aggregating One-Dependence Estimators

GEOFFREY I. WEBB geoff.webb@infotec.monash.edu
JANICE R. BOUGHTON
ZHIHAI WANG
School of Computer Science and Software Engineering, Monash University, Vic. 3800, Australia

Editor: Tom Fawcett

Abstract. Of numerous proposals to improve the accuracy of naive Bayes by weakening its attribute independence assumption, both LBR and Super-Parent TAN have demonstrated remarkable error performance. However, both techniques obtain this outcome at a considerable computational cost. We present a new approach to weakening the attribute independence assumption by averaging all of a constrained class of classifiers. In extensive experiments this technique delivers comparable prediction accuracy to LBR and Super-Parent TAN with substantially improved computational efficiency at test time relative to the former and at training time relative to the latter. The new algorithm is shown to have low variance and is suited to incremental learning.

Keywords: naive Bayes, semi-naive Bayes, attribute independence assumption, probabilistic prediction

1. Introduction

Due to its simplicity, efficiency and efficacy, naive Bayes (NB) is widely deployed for classification learning. It delivers optimal classification subject only to the accuracy of the estimation of the base conditional probabilities on which it relies and to the constraints of its simplifying attribute independence assumption. Notwithstanding the fact that some violations of the attribute independence assumption do not matter (Domingos & Pazzani, 1996), it is clear that many do, and there is an increasing body of work developing techniques to retain NB's desirable simplicity and efficiency while alleviating the problems of the attribute independence assumption (Friedman, Geiger, & Goldszmidt, 1997; Keogh & Pazzani, 1999; Kohavi, 1996; Kononenko, 1991; Langley, 1993; Langley & Sage, 1994; Pazzani, 1996; Sahami, 1996; Singh & Provan, 1996; Webb & Pazzani, 1998; Webb, 2001; Xie et al., 2002; Zheng & Webb, 2000; Zheng, Webb, & Ting, 1999).

Of these techniques, two have demonstrated remarkable accuracy. Lazy Bayesian Rules (LBR) (Zheng & Webb, 2000) has demonstrated accuracy comparable to boosting decision trees (Zheng, Webb, & Ting, 1999) and Super-Parent TAN (SP-TAN) (Keogh & Pazzani, 1999), a variant of Tree Augmented Naive Bayes (TAN) (Friedman, Geiger, & Goldszmidt, 1997), has demonstrated accuracy comparable to LBR (Wang & Webb, 2002). However, these two techniques have high computational overheads, SP-TAN having high computational complexity at training time and LBR having high computational complexity at classification time. This reduces their usefulness as an alternative to NB.

This paper first introduces NB, LBR, TAN and SP-TAN. An analysis of the sources of strength of each algorithm together with the determinants of their computational profiles leads to the development of AODE, a new efficient technique that utilizes a weaker attribute independence assumption than NB, thereby improving prediction accuracy without undue computational overheads. We present an experimental comparison of performance on selected UCI data sets together with a bias-variance analysis. AODE demonstrates comparable error to LBR and SP-TAN coupled with a computational profile that avoids the high training cost of SP-TAN and the high classification cost of LBR.

2. Naive Bayes

We wish to predict from a training sample of classified objects the class of an example $\mathbf{x} = \langle x_1, \ldots, x_n \rangle$, where x_i is the value of the ith attribute. We can minimize error by selecting $\text{argmax}_y\ P(y \mid \mathbf{x})$, where $y \in c_1, \ldots c_k$ are the k classes. To this end we seek an estimate $\hat{P}(y \mid \mathbf{x})$ of $P(y \mid \mathbf{x})$ and perform classification by selecting $\text{argmax}_y\ \hat{P}(y \mid \mathbf{x})$.

From the definition of conditional probability we have

$$\begin{aligned} P(y \mid \mathbf{x}) &= P(y, \mathbf{x})/P(\mathbf{x}) && (1) \\ &\propto P(y, \mathbf{x}). && (2) \end{aligned}$$

Hence, $\text{argmax}_y\ P(y \mid \mathbf{x}) = \text{argmax}_y\ P(y, \mathbf{x})$, and the latter is often calculated in practice rather than the former. Further, as $P(\mathbf{x}) = \sum_{i=1}^{k} P(c_i, \mathbf{x})$, we can always estimate Eq. (1) from estimates of Eq. (2) for each class using

$$P(y, \mathbf{x})/P(\mathbf{x}) \approx \hat{P}(y, \mathbf{x}) \bigg/ \sum_{i=1}^{k} \hat{P}(c_i, \mathbf{x}). \tag{3}$$

In consequence, in the remainder of this paper we consider only the problem of estimating Eq. (2).

Assuming the training sample is a representative sample of the joint distribution from which it is drawn, the frequency with which any event occurs in the sample will be a reasonable approximation of the probability of that event. In practice a minor adjustment is made to the observed frequency, such as the Laplace estimate, in order to allow for possible sampling error. However, if the number of attributes, n, is large, for most $\mathbf{x}$, $P(y, \mathbf{x})$ is likely to be extremely small, and hence for any y, $(y, \mathbf{x})$ is unlikely to occur in the sample. In consequence, an estimate of $P(y, \mathbf{x})$ from the sample frequencies will be uninformative. One way around this problem is to estimate $P(y, \mathbf{x})$ by a function from other probability estimates that we can derive with greater confidence from the sample frequencies.

By application of the product rule we have the following.

$$P(y, \mathbf{x}) = P(y)P(\mathbf{x} \mid y) \tag{4}$$

If the number of classes, k, is small, it should be possible to obtain a sufficiently accurate estimate of $P(y)$ from the sample frequencies. However, we still have the problem that $\mathbf{x}$

may not occur in the training data and hence $P(\mathbf{x} \mid y)$ cannot be directly estimated from the sample. NB circumvents this problem by assuming that the attributes are independent given the class. From this assumption it follows that

$$P(\mathbf{x} \mid y) = \prod_{i=1}^{n} P(x_i \mid y). \tag{5}$$

Hence NB classifies by selecting

$$\underset{y}{\operatorname{argmax}} \left(\hat{P}(y) \prod_{i=1}^{n} \hat{P}(x_i \mid y) \right), \tag{6}$$

where $\hat{P}(y)$ and $\hat{P}(x_i \mid y)$ are estimates of the respective probabilities derived from the frequency of their respective arguments in the training sample, with possible corrections such as the Laplace estimate.

At training time NB need only compile a table of class probability estimates and a table of conditional attribute-value probability estimates. The former is one-dimensional, indexed by class and the latter two-dimensional, indexed by class and attribute-value. The resulting space complexity is $O(knv)$, where v is the average number of values per attribute. To calculate the estimates requires a simple scan through the data, an operation of time complexity $O(tn)$, where t is the number of training examples. At classification time, to classify a single example has time complexity $O(kn)$ using the tables formed at training time with space complexity $O(knv)$.

3. LBR and TAN

Equation (6) is simple to calculate leading to efficient classification. However, violations of the attribute independence assumption Eq. (5) can lead to undesirably high error. Of the many approaches to obviating this problem cited in the introduction, two have demonstrated very low error: LBR (Zheng & Webb, 2000) and SP-TAN (Keogh & Pazzani, 1999). Both rely on weaker attribute independence assumptions than NB.

LBR uses lazy learning. For each $\mathbf{x} = \langle x_i, \ldots, x_n \rangle$ to be classified, a set W of the attribute values is selected. Independence is assumed among the remaining attributes given W and y. Hence, $\mathbf{x}$ can be classified by selecting

$$\underset{y}{\operatorname{argmax}} \left(\hat{P}(y \mid W) \prod_{i=1}^{n} \hat{P}(x_i \mid y, W) \right). \tag{7}$$

Thus, every attribute depends both on the class and the attributes chosen for inclusion in W. W is selected by a simple heuristic wrapper approach that seeks to minimize error on the training sample.

At training time, LBR simply stores the training data, an operation of time and space complexity $O(tn)$. At classification time, however, LBR must select the attributes for inclusion in W, an operation of time complexity $O(tkn^3)$. In practice, the cumulative computation is reasonable when few examples are to be classified for each training set. When large numbers of examples are to be classified, the computational burden becomes prohibitive.

In contrast to LBR, TAN and SP-TAN allow every attribute x_i to depend upon the class and at most one other attribute, $p(x_i)$, called the parent of x_i. Hence, $\mathbf{x}$ is classified by selecting

$$\operatorname*{argmax}_{y} \left(\hat{P}(y) \prod_{i=1}^{n} \hat{P}(x_i \mid y, p(x_i)) \right). \tag{8}$$

The parent function $p(\cdot)$ is developed at training time. TAN (Friedman, Geiger, & Goldszmidt, 1997) uses conditional mutual information to select the parent function. SP-TAN (Keogh & Pazzani, 1999) uses a simple heuristic wrapper approach that seeks to minimize error on the training sample. At training time both TAN and SP-TAN generate a three-dimensional table of probability estimates for each attribute-value, conditioned by each other attribute-value and each class, space complexity $O(k(nv)^2)$. SP-TAN must also store the training data, with additional space complexity $O(tn)$. The time complexity of forming the three dimensional probability table required by both TAN and SP-TAN is $O(tn^2)$ as an entry must be updated for every training case and every combination of two attribute-values for that case. To create the parent function TAN must first calculate the conditional mutual information, requiring consideration for each pair of attributes, every pairwise combination of their respective values in conjunction with each class value $O(kn^2v^2)$. A maximal spanning tree is then generated, time complexity $O(n^2 \log n)$. The time complexity of forming the parent function for SP-TAN is $O(tkn^3)$, as the selection of a single parent is order $O(tkn^2)$ and parent selection is performed repeatedly, potentially being repeated until every attribute has a parent. At classification time both TAN and SP-TAN need only store the probability tables, space complexity $O(knv^2)$. This compression over the table required at training time is achieved by storing probability estimates for each attribute-value conditioned by the parent selected for that attribute, and the class. The time complexity of classifying a single example is $O(kn)$.

4. Averaged One-Dependence Estimators

LBR and SP-TAN appear to offer competitive error to boosting decision trees (Zheng, Webb, & Ting, 1999; Wang & Webb, 2002). However, except in the case of applying LBR to classify small numbers of examples for each training set, this is achieved at considerable computational cost. In the current research we seek techniques that weaken NB's attribute independence assumption, achieving the error performance of LBR and SP-TAN, without their computational burden.

Analysis of LBR and SP-TAN reveals that the computational burden can be attributed mainly to two factors:

- model selection: W for LBR, and $p(\cdot)$ for SP-TAN, and
- probability estimation: generated on the fly for LBR, and via the three-dimensional conditional probability table for SP-TAN.

Considering first the issue of probability estimation, it is clearly desirable to be able to pre-compute all required base probability estimates at training time, as does SP-TAN. Sahami (1996) introduces the notion of x-dependence estimators, whereby the probability of each attribute value is conditioned by the class and at most x other attributes. In general, the probability estimates required for an x-dependence estimator can be stored in an $(x+2)$-dimensional table, indexed by the target attribute-value, the class value, and the values of the x other attributes by which the target is conditioned. To maintain efficiency it appears desirable to restrict ourselves to 1-dependence classifiers.

This leaves the issue of model selection. One way to minimize the computation required for model selection is to perform no model selection, as does NB.

In addition to the desire to minimize computation, a second motivation for avoiding model selection is that selection between alternative models can be expected to increase variance. This is because selection between models allows a learning system to more closely fit the training data. In consequence, changes in the training data will lead to greater changes in the model formed, which leads in turn to greater variance (see, for example, Hastie, Tibshirani, & Friedman, 2001). In contrast, under approaches such as naive Bayes where there is no choice in the form of the model, all that changes when the training data changes is the underlying conditional probability tables which tends to result in relatively gradual changes in the pattern of classification. Model selection avoidance may minimize the variance component of a classifier's error.

However, while avoiding model selection appears desirable, it appears to conflict with the desire to use 1-dependence classifiers. These require each attribute to depend on one other attribute and the precise such attribute must surely be selected. Our solution is to select a limited class of 1-dependence classifiers and to aggregate the predictions of all qualified classifiers within this class. The class we select is all 1-dependence classifiers where there is a single attribute that is the parent of all other attributes. However, we wish to avoid including models for which the base probability estimates are inaccurate. To this end, when classifying an object $\mathbf{x} = \langle x_1, \ldots, x_n \rangle$, we exclude models where the training data contain fewer than m examples of the value for $\mathbf{x}$ of the parent attribute x_i. In the current research we use $m = 30$, this being a widely utilized minimum on sample size for statistical inference purposes.

By application of the product rule it follows that for any attribute value x_i

$$P(y, \mathbf{x}) = P(y, x_i) P(\mathbf{x} \mid y, x_i). \tag{9}$$

As this equality holds for every x_i, it follows that it also holds for the mean over any group of attribute values. Hence,

$$P(y, \mathbf{x}) = \frac{\sum_{i:1 \leq i \leq n \wedge F(x_i) \geq m} P(y, x_i) P(\mathbf{x} \mid y, x_i)}{|\{i : 1 \leq i \leq n \wedge F(x_i) \geq m\}|} \tag{10}$$

where $F(x_i)$ is a count of the number of training examples having attribute-value x_i and is used to enforce the limit m that we place on the support needed in order to accept a conditional probability estimate. In the presence of estimation error, if the inaccuracies of the estimates are unbiased the mean can be expected to factor out that error.

Equation (10) provides a new strategy for estimating class probabilities. We call the resulting classifiers Averaged One-Dependence Estimators (AODE). As the denominator of Eq. (10) is invariant across classes it need not be calculated. In consequence, substituting probability estimates for the probabilities in Eq. (10) and seeking the class that maximizes the resulting term, these classifiers select the class

$$\underset{y}{\operatorname{argmax}} \left(\sum_{i:1\leq i\leq n\wedge F(x_i)\geq m} \hat{P}(y, x_i) \prod_{j=1}^{n} \hat{P}(x_j \mid y, x_i) \right). \tag{11}$$

If $\neg\exists i : 1 \leq i \leq n \wedge F(x_i) \geq m$, AODE defaults to NB.

AODE can be extended to provide direct class probability estimates by normalizing the numerator of Eq. (10) across all classes:

$$\hat{P}(y \mid X) = \frac{\sum_{i:1\leq i\leq n\wedge F(x_i)\geq m} \hat{P}(y, x_i) \prod_{j=1}^{n} \hat{P}(x_j \mid y, x_i)}{\sum_{y'\in Y} \sum_{i:1\leq i\leq n\wedge F(x_i)\geq m} \hat{P}(y', x_i) \prod_{j=1}^{n} \hat{P}(x_j \mid y', x_i)}. \tag{12}$$

At training time AODE need only form the tables of joint attribute-value, class frequencies from which the probability estimates $\hat{P}(y, x_i)$ and $\hat{P}(y, x_i, x_j)$ are derived that are required for estimating $\hat{P}(y, x_i)$ and $\hat{P}(x_j \mid y, x_i)$. The space complexity of these tables is $O(k(nv)^2)$. Derivation of the frequencies required to populate these tables is of time complexity $O(tn^2)$. There is no model selection. Classification requires the tables of probability estimates formed at training time of space complexity $O(k(nv)^2)$. Classification of a single example requires calculation of Eq. (11) and is of time complexity $O(kn^2)$. Table 1 displays the relative complexity of each of the algorithms discussed.

Table 1. Computational complexity.

	Training		Classification	
Algorithm	Time	Space	Time	Space
NB	$O(nt)$	$O(knv)$	$O(kn)$	$O(knv)$
TAN	$O(tn^2 + kn^2v^2 + n^2 \log n)$	$O(k(nv)^2)$	$O(kn)$	$O(knv^2)$
SP-TAN	$O(tkn^3)$	$O(tn + k(nv)^2)$	$O(kn)$	$O(knv^2)$
LBR	$O(tn)$	$O(tn)$	$O(tkn^3)$	$O(tn)$
AODE	$O(tn^2)$	$O(k(nv)^2)$	$O(kn^2)$	$O(k(nv)^2)$

k is the number of classes.
n is the number of attributes.
v is the average number of values for an attribute.
t is the number of training examples.

A further computational advantage of AODE compared to TAN or SP-TAN is that it lends itself directly to incremental learning. To update an AODE classifier with evidence from a new example requires only incrementing the relevant entries in the tables of joint attribute-value and class frequencies.

We expect AODE to achieve lower classification error than NB for the following reasons. First, as it involves a weaker attribute independence assumption, $P(y, x_i) \prod_{j=1}^{n} P(x_j \mid y, x_i)$ should provide a better estimate of $P(y, \mathbf{x})$ than $P(y) \prod_{i=1}^{n} P(x_i \mid y)$. Hence, the estimates from each of the one-dependence models over which AODE averages should be better than the estimate from NB, except insofar as the estimates of the base probabilities $P(y, x_i)$ and $P(x_j \mid y, x_i)$ in the one-dependence models are less accurate than the estimates of the base probabilities $P(y)$ and $P(x_i \mid y)$ used by NB. The only systematic cause for such a drop in accuracy might result from the smaller numbers of examples from which the AODE base probabilities are estimated. We seek to guard against negative impact from such a cause by restricting the base models to those for which the parent attribute-value occurs with sufficient frequency. Due to the extent to which AODE's estimates can be expected to grow in accuracy as the amount of data increases, we expect the magnitude of the advantage over NB to grow as the number of training examples grows. Second, there is a considerable evidence that aggregating multiple credible models leads to improved prediction accuracy (Ali, Brunk, & Pazzani, 1994; Breiman, 1996; Freund & Schapire, 1997; Nock & Gascuel, 1995; Oliver & Hand, 1995; Wolpert, 1992), and we expect to benefit from such an effect. Third, like NB, AODE avoids model selection and hence avoids the attendant variance.

5. Evaluation

We here evaluate our hypotheses that AODE will deliver efficient and accurate classification. We also evaluate our expectation that any one-dependence estimator should be more accurate than NB so long as there is sufficient data to accurately estimate the required probabilities. To this end we also consider ODE, a variant of AODE where, instead of averaging over all $x_i : F(x_i) \geq m$ as in Eq. (11), one $x_i : F(x_i) \geq m$ is randomly selected and we classify using

$$\underset{y}{\operatorname{argmax}} \left(\hat{P}(y, x_i) \prod_{j=1}^{n} \hat{P}(x_j \mid y, x_i) \right). \tag{13}$$

To assess the account of AODE as an approach to ensembling one-dependence estimators, we also consider bagged ODE.

We compare the prediction error, bias, variance, training time and classification time of AODE to those of NB, ODE, bagged ODE, LBR, TAN, and SP-TAN. In order to provide comparators with which many researchers will be familiar, we also provide results for a standard decision tree learner and a boosted decision tree learner. To this end we implemented AODE, ODE, TAN and SP-TAN in the Weka workbench version 3.35 (Witten & Frank, 2000). We used Weka's implementations of NB, LBR, the decision tree learner J48 (a reimplementation of C4.5), boosting and bagging. For all algorithms we employed Weka's default settings, in particular forming ensembles of ten base classifiers each for boosting and bagging. In keeping with Weka's NB and LBR, we estimated the base probabilities $P(y)$, $P(y, x_i)$ and $P(y, x_i, x_j)$ using the Laplace estimate as follows:

$$\hat{P}(y) = \frac{F(y) + 1}{K + k} \tag{14}$$

$$\hat{P}(y, x_i) = \frac{F(y, x_i) + 1}{K_i + kv_i} \tag{15}$$

$$\hat{P}(y, x_i, x_j) = \frac{F(y, x_i, x_j) + 1}{K_{ij} + kv_i v_j} \tag{16}$$

where $F(\cdot)$ is the frequency with which a combination of terms appears in the training data, K is the number of training examples for which the class value is known, K_i is the number of training examples for which both the class and attribute i are known, K_{ij} is the number of training examples for which all of the class, and attributes i and j are known, and v_a is the number of values for attribute a.

As LBR, TAN, SP-TAN and AODE require discrete valued data, all data were discretized using MDL discretization (Fayyad & Irani, 1993). MDL discretization was used in preference to techniques specifically optimized for naive Bayes (Yang & Webb, 2003) because the latter rely on the attribute independence assumption and hence are poorly adapted to the semi-naive approaches of LBR, TAN, SP-TAN and AODE that weaken that assumption. We evaluated J48 and boosted J48 with both discretized and the raw data. We report only results for the raw data as these are more favorable to those algorithms.

As we expect AODE to exhibit low variance, we compared the performance of the system using Weka's bias-variance decomposition utility which utilizes the experimental method proposed by Kohavi and Wolpert (1996). The training data are divided into training and test sets each containing half the data. 50 local training sets are sampled from the training set, each local set containing 50% of the training set, which is 25% of the full data set. A classifier is formed from each local training set and bias, variance, and error estimated from the performance of those classifiers on the test set.

Experiments were performed on a dual-processor 1.7 GHz Pentium 4 Linux computer with 2 Gb RAM. All algorithms were applied to the 37 data sets described in Table 2. These data sets are formed around a core of twenty-nine data sets used in previous related research (Domingos & Pazzani, 1996; Zheng & Webb, 2000) augmented by eight larger data sets added because the original data sets were all relatively small and AODE, LBR, TAN and SP-TAN have greatest scope to improve upon NB when more data is available. However, it should be noted that the bias-variance experimental method results in very small training sets, each only 25% of the size of the data set. Previous research suggests that NB enjoys particularly low relative error on small data sets (Zheng & Webb, 2000) and hence this experimental method can be expected to favor NB.

5.1. Error, bias and variance results

Table 3 presents for each data set the mean error for each algorithm. Tables 4 and 5 provide the mean bias and variance results respectively. For each algorithm the mean of each measure

Table 2. Data sets.

Name	Cases	Atts	Name	Cases	Atts
adult	48842	15	labor-neg	57	17
anneal	898	39	led	1000	8
balance-scale	625	5	letter-recognition	20000	17
bcw	699	10	lung-cancer	32	57
bupa	345	7	mfeat-mor	2000	7
chess	551	40	new-thyroid	215	6
cleveland	303	14	pendigits	10992	17
crx	690	16	post-operative	90	9
echocardiogram	131	7	promoters	106	58
german	1000	21	ptn	339	18
glass	214	10	satellite	6435	37
heart	270	14	segment	2310	20
hepatitis	155	20	sign	12546	9
horse-colic	368	22	sonar	208	61
house-votes-84	435	17	syncon	600	61
hungarian	294	14	ttt	958	10
hypothyroid	3163	26	vehicle	846	19
ionosphere	351	35	wine	178	14
iris	150	5			

across all data sets is also presented. The mean error, bias or variance across multiple data sets provides at best a very gross measure of relative performance as it is questionable whether error rates are commensurable across data sets. The geometric mean ratio is also presented. This is a standardized measure of relative performance. This is obtained by taking for each data set the ratio of the performance of the alternative algorithm divided by the performance of AODE. The geometric mean of these ratios is presented as this is the most appropriate average to apply to ratio data. A geometric mean ratio greater than 1.0 represents an advantage to AODE and a value lower than 1.0 represents an advantage to the alternative algorithm.

We do not apply significance tests to pairwise comparisons of performance on a data set by data set basis, as the 888 (37 data sets × 3 metrics × 8 comparator algorithms) such comparisons would result in substantial risk of a large number of false positive outcomes. Nor do we present the standard deviations of the individual error outcomes as the number of outcomes makes interpretation of such information infeasible. Rather, we perform a win/draw/loss summary to compare overall performance of AODE against each other algorithm on each measure. The results are presented in Table 6. For each pairwise comparison we present first the number of data sets for which AODE obtained lower average error than the comparator algorithm, the number for which the algorithms obtained the same average error, and the number for which the alternative algorithm obtained lower average error. The

Table 3. Error.

				Bag			SP-		**Boost**
Data	AODE	NB	ODE	ODE	LBR	TAN	TAN	**J48**	**J48**
adult	0.152	0.168	0.172	0.170	0.140	0.147	0.147	**0.146**	**0.169**
anneal	0.065	0.082	0.064	0.059	0.064	0.067	0.067	**0.157**	**0.101**
balance-scale	0.302	0.303	0.310	0.267	0.302	0.303	0.300	**0.274**	**0.238**
bcw	0.027	0.030	0.038	0.036	0.030	0.050	0.030	**0.087**	**0.048**
bupa	0.424	0.424	0.424	0.426	0.424	0.424	0.424	**0.428**	**0.395**
chess	0.140	0.143	0.144	0.146	0.141	0.128	0.137	**0.144**	**0.124**
cleveland	0.176	0.174	0.175	0.170	0.174	0.176	0.178	**0.260**	**0.227**
crx	0.163	0.171	0.165	0.162	0.172	0.177	0.172	**0.172**	**0.179**
echocardiogram	0.382	0.389	0.382	0.364	0.392	0.388	0.388	**0.372**	**0.366**
german	0.262	0.268	0.268	0.262	0.269	0.277	0.268	**0.296**	**0.291**
glass	0.299	0.300	0.299	0.275	0.303	0.300	0.295	**0.288**	**0.257**
heart	0.216	0.215	0.217	0.201	0.215	0.236	0.218	**0.269**	**0.254**
hepatitis	0.140	0.139	0.137	0.129	0.140	0.143	0.138	**0.173**	**0.163**
horse-colic	0.219	0.221	0.227	0.217	0.210	0.213	0.219	**0.226**	**0.229**
house-votes-84	0.054	0.086	0.089	0.087	0.069	0.068	0.082	**0.040**	**0.044**
hungarian	0.173	0.169	0.173	0.177	0.173	0.179	0.172	**0.211**	**0.212**
hypothyroid	0.021	0.024	0.025	0.025	0.016	0.025	0.018	**0.013**	**0.015**
ionosphere	0.102	0.119	0.122	0.102	0.119	0.099	0.118	**0.166**	**0.143**
iris	0.058	0.058	0.058	0.054	0.058	0.056	0.058	**0.060**	**0.059**
labor-neg	0.150	0.150	0.150	0.135	0.196	0.168	0.154	**0.239**	**0.192**
led	0.258	0.255	0.268	0.270	0.257	0.271	0.259	**0.318**	**0.318**
letter-recognition	0.193	0.292	0.266	0.259	0.220	0.212	0.210	**0.208**	**0.103**
lung-cancer	0.556	0.556	0.556	0.540	0.557	0.562	0.555	**0.616**	**0.608**
mfeat-mor	0.311	0.317	0.321	0.312	0.313	0.312	0.314	**0.300**	**0.305**
new-thyroid	0.074	0.074	0.084	0.088	0.074	0.077	0.075	**0.119**	**0.093**
pendigits	0.037	0.132	0.067	0.059	0.065	0.066	0.055	**0.065**	**0.021**
post-operative	0.366	0.366	0.366	0.360	0.364	0.383	0.386	**0.317**	**0.416**
promoters	0.130	0.130	0.130	0.146	0.132	0.315	0.134	**0.247**	**0.208**
ptn	0.572	0.559	0.581	0.584	0.571	0.593	0.571	**0.635**	**0.635**
satellite	0.120	0.178	0.164	0.152	0.148	0.128	0.155	**0.164**	**0.119**
segment	0.071	0.112	0.116	0.094	0.092	0.082	0.090	**0.065**	**0.041**
sign	0.302	0.362	0.295	0.290	0.280	0.292	0.297	**0.206**	**0.175**
sonar	0.275	0.274	0.277	0.261	0.274	0.293	0.279	**0.316**	**0.269**
syncon	0.059	0.069	0.086	0.060	0.069	0.058	0.069	**0.191**	**0.106**
ttt	0.261	0.296	0.295	0.295	0.291	0.294	0.295	**0.240**	**0.147**
vehicle	0.383	0.444	0.438	0.406	0.385	0.382	0.428	**0.334**	**0.277**
wine	0.042	0.040	0.042	0.029	0.040	0.053	0.040	**0.143**	**0.094**
Mean	0.204	0.219	0.216	0.207	0.209	0.216	0.211	**0.230**	**0.206**
Geo mean ratio		1.124	1.115	1.048	1.049	1.102	1.056	**1.225**	**1.026**

Table 4. Bias.

				Bag			SP-		**Boost**
Data	AODE	NB	ODE	ODE	LBR	TAN	TAN	**J48**	**J48**
adult	0.139	0.156	0.160	0.161	0.127	0.129	0.119	**0.113**	**0.101**
anneal	0.045	0.053	0.043	0.041	0.041	0.043	0.043	**0.094**	**0.050**
balance-scale	0.172	0.175	0.187	0.174	0.173	0.172	0.172	**0.140**	**0.117**
bcw	0.025	0.028	0.031	0.029	0.028	0.027	0.028	**0.040**	**0.027**
bupa	0.292	0.292	0.292	0.249	0.292	0.292	0.292	**0.232**	**0.213**
chess	0.101	0.104	0.102	0.103	0.097	0.062	0.090	**0.068**	**0.069**
cleveland	0.127	0.127	0.121	0.128	0.127	0.117	0.126	**0.129**	**0.122**
crx	0.138	0.147	0.140	0.141	0.147	0.130	0.143	**0.113**	**0.111**
echocardiogram	0.247	0.249	0.246	0.260	0.253	0.246	0.250	**0.232**	**0.227**
german	0.195	0.203	0.192	0.188	0.202	0.174	0.196	**0.191**	**0.157**
glass	0.168	0.169	0.168	0.165	0.167	0.164	0.166	**0.121**	**0.119**
heart	0.156	0.156	0.154	0.151	0.156	0.165	0.154	**0.134**	**0.139**
hepatitis	0.096	0.098	0.089	0.089	0.094	0.078	0.095	**0.085**	**0.073**
horse-colic	0.179	0.188	0.175	0.177	0.177	0.170	0.183	**0.194**	**0.154**
house-votes-84	0.043	0.077	0.075	0.072	0.046	0.044	0.071	**0.024**	**0.017**
hungarian	0.156	0.156	0.155	0.159	0.157	0.134	0.155	**0.163**	**0.138**
hypothyroid	0.018	0.021	0.022	0.021	0.013	0.022	0.014	**0.013**	**0.012**
ionosphere	0.068	0.077	0.078	0.071	0.077	0.063	0.076	**0.096**	**0.088**
iris	0.037	0.037	0.037	0.035	0.037	0.034	0.038	**0.047**	**0.043**
labor-neg	0.046	0.046	0.046	0.039	0.068	0.057	0.048	**0.084**	**0.067**
led	0.211	0.209	0.224	0.225	0.208	0.221	0.208	**0.229**	**0.229**
letter-recognition	0.133	0.230	0.182	0.182	0.103	0.124	0.110	**0.080**	**0.039**
lung-cancer	0.311	0.311	0.311	0.299	0.312	0.375	0.309	**0.319**	**0.325**
mfeat-mor	0.240	0.246	0.248	0.247	0.231	0.235	0.234	**0.181**	**0.183**
new-thyroid	0.040	0.039	0.043	0.055	0.039	0.028	0.039	**0.059**	**0.041**
pendigits	0.023	0.111	0.040	0.041	0.025	0.035	0.025	**0.021**	**0.008**
post-operative	0.299	0.299	0.299	0.299	0.300	0.315	0.306	**0.309**	**0.291**
promoters	0.043	0.043	0.043	0.052	0.044	0.134	0.044	**0.077**	**0.060**
ptn	0.348	0.346	0.353	0.347	0.352	0.370	0.342	**0.345**	**0.343**
satellite	0.095	0.162	0.119	0.121	0.085	0.094	0.131	**0.075**	**0.065**
segment	0.044	0.075	0.059	0.058	0.047	0.039	0.056	**0.031**	**0.019**
sign	0.260	0.324	0.257	0.262	0.218	0.245	0.235	**0.108**	**0.096**
sonar	0.180	0.181	0.178	0.178	0.181	0.169	0.182	**0.141**	**0.121**
syncon	0.037	0.046	0.038	0.036	0.046	0.027	0.046	**0.065**	**0.036**
ttt	0.191	0.234	0.215	0.212	0.207	0.195	0.199	**0.110**	**0.051**
vehicle	0.255	0.315	0.304	0.297	0.248	0.231	0.300	**0.155**	**0.147**
wine	0.016	0.015	0.016	0.014	0.015	0.017	0.016	**0.040**	**0.024**
Mean	0.140	0.155	0.147	0.145	0.139	0.140	0.142	**0.126**	**0.111**
Geo mean ratio		1.160	1.084	1.074	1.003	0.998	1.026	**0.963**	**0.773**

Table 5. Variance.

Data	AODE	NB	ODE	Bag ODE	LBR	TAN	SP- TAN	**J48**	**Boost J48**
adult	0.012	0.011	0.011	0.009	0.013	0.018	0.027	**0.032**	**0.067**
anneal	0.020	0.028	0.021	0.018	0.023	0.023	0.024	**0.062**	**0.050**
balance-scale	0.128	0.125	0.121	0.090	0.126	0.128	0.126	**0.131**	**0.118**
bcw	0.002	0.002	0.008	0.007	0.002	0.023	0.002	**0.046**	**0.021**
bupa	0.130	0.130	0.130	0.173	0.130	0.130	0.130	**0.192**	**0.178**
chess	0.038	0.038	0.041	0.042	0.043	0.065	0.046	**0.074**	**0.054**
cleveland	0.048	0.046	0.053	0.041	0.046	0.059	0.051	**0.129**	**0.103**
crx	0.025	0.024	0.025	0.020	0.024	0.047	0.028	**0.058**	**0.067**
echocardiogram	0.133	0.137	0.133	0.102	0.137	0.139	0.135	**0.137**	**0.136**
german	0.066	0.063	0.075	0.072	0.066	0.101	0.071	**0.103**	**0.132**
glass	0.129	0.129	0.129	0.108	0.134	0.134	0.127	**0.163**	**0.135**
heart	0.058	0.058	0.062	0.049	0.058	0.070	0.063	**0.132**	**0.113**
hepatitis	0.043	0.040	0.047	0.039	0.044	0.064	0.043	**0.086**	**0.088**
horse-colic	0.040	0.032	0.051	0.039	0.033	0.042	0.035	**0.031**	**0.074**
house-votes-84	0.010	0.009	0.014	0.014	0.022	0.024	0.011	**0.016**	**0.026**
hungarian	0.017	0.013	0.017	0.018	0.016	0.044	0.016	**0.047**	**0.073**
hypothyroid	0.003	0.003	0.004	0.004	0.002	0.003	0.004	**0.001**	**0.003**
ionosphere	0.033	0.041	0.043	0.030	0.041	0.036	0.041	**0.068**	**0.053**
iris	0.021	0.021	0.021	0.018	0.021	0.021	0.019	**0.013**	**0.015**
labor-neg	0.102	0.102	0.102	0.094	0.126	0.109	0.104	**0.152**	**0.123**
led	0.046	0.045	0.043	0.044	0.048	0.049	0.050	**0.087**	**0.087**
letter-recognition	0.058	0.061	0.083	0.075	0.114	0.086	0.098	**0.126**	**0.063**
lung-cancer	0.240	0.240	0.240	0.236	0.240	0.184	0.241	**0.291**	**0.277**
mfeat-mor	0.070	0.070	0.072	0.064	0.081	0.075	0.079	**0.116**	**0.119**
new-thyroid	0.034	0.034	0.040	0.032	0.034	0.049	0.035	**0.059**	**0.051**
pendigits	0.014	0.020	0.026	0.018	0.039	0.030	0.029	**0.043**	**0.012**
post-operative	0.065	0.065	0.065	0.060	0.064	0.067	0.078	**0.009**	**0.122**
promoters	0.085	0.085	0.085	0.093	0.086	0.177	0.088	**0.166**	**0.145**
ptn	0.219	0.210	0.224	0.232	0.215	0.218	0.225	**0.284**	**0.287**
satellite	0.025	0.016	0.044	0.030	0.062	0.034	0.025	**0.087**	**0.053**
segment	0.026	0.036	0.056	0.036	0.044	0.043	0.034	**0.034**	**0.021**
sign	0.041	0.037	0.037	0.028	0.060	0.045	0.061	**0.096**	**0.078**
sonar	0.093	0.092	0.096	0.081	0.092	0.122	0.095	**0.172**	**0.145**
syncon	0.022	0.022	0.046	0.024	0.022	0.030	0.022	**0.123**	**0.069**
ttt	0.068	0.061	0.079	0.081	0.083	0.097	0.094	**0.128**	**0.094**
vehicle	0.126	0.126	0.132	0.107	0.134	0.148	0.125	**0.176**	**0.127**
wine	0.025	0.024	0.026	0.014	0.024	0.036	0.024	**0.101**	**0.068**
Mean	0.063	0.062	0.068	0.061	0.069	0.075	0.068	**0.102**	**0.093**
Geo mean ratio		0.980	1.178	0.995	1.131	1.375	1.119	**1.735**	**1.717**

Table 6. Win/draw/loss records, AODE vs. alternatives.

	NB		ODE		Bagged ODE	
	W/D/L	p	W/D/L	p	W/D/L	p
Error	22/7/8	0.008	23/10/4	<0.001	19/2/16	0.736
Bias	24/9/4	<0.001	19/8/10	0.136	22/1/14	0.243
Variance	6/15/16	0.026	23/10/4	<0.001	15/0/22	0.324
	LBR		TAN		SP-TAN	
	W/D/L	p	W/D/L	p	W/D/L	p
Error	19/6/12	0.281	27/2/8	0.002	23/3/11	0.058
Bias	18/4/15	0.728	13/2/22	0.175	18/3/16	0.864
Variance	19/8/10	0.136	31/4/2	<0.001	25/5/7	0.002
	J48				Boosted J48	
	W/D/L	p			W/D/L	p
Error	25/0/12	0.047			21/0/16	0.511
Bias	15/0/22	0.324			10/0/27	0.008
Variance	33/0/4	<0.001			32/1/4	<0.001

p value is the outcome of a binomial sign test and represents the probability that AODE would obtain the observed or more extreme ratio of wins to losses. The p value for NB is one-tailed because a specific prediction is made about the direction of the result. For all other algorithms the reported p value is the result of a two-tailed test because no specific prediction about relative performance has been made. We assess a difference as significant if $p \leq 0.05$. Using only one-tailed or only two-tailed tests would in each case only change one assessment of significance, each of which is noted below.

Considering first the error outcomes, AODE achieves the lowest mean error, its mean error being substantially (0.010 or more) lower than that of NB, ODE, TAN and J48 and the geometric mean error ratio showing a substantially (1.10 or greater) advantage with respect to NB, ODE and J48. The win/draw/loss record indicates a significant advantage over NB, ODE, TAN and J48. While the mean and geometric mean ratios might suggest marginal advantage over the remaining algorithms, the win/draw/loss tables do not reveal any of these to be statistically significant. Note, however, that a one-tailed p for the win/draw/loss record with respect to SP-TAN is 0.029, which would be accepted as significant.

With respect to bias and variance, all measures indicate that AODE obtains lower bias and higher variance than NB, the win/draw/loss records being significant in each case. Note, however, that the two-tailed p for the win/draw/loss record with respect to variance is 0.052, which is only marginally significant. Compared to LBR, TAN and SP-TAN, AODE obtains lower mean and geometric mean ratio outcomes for variance and similar outcomes for bias. Turning to the win/draw/loss records, the advantage in variance is significant with respect to TAN and SP-TAN, but not LBR. The win/draw/loss records for bias do not indicate a significant difference with respect to any of these algorithms.

All measures suggest that AODE has higher bias but lower variance than J48 and Boosted J48, the win/draw/loss outcomes being significant in all cases except for bias with respect to J48.

ODE, and bagged ODE were included in the experiments in order to evaluate the interpretation of the power of AODE in terms of ensembling one-dependence classifiers. Comparing ODE first to NB, ODE has lower mean error and bias but higher mean variance. The win/draw/loss records of NB compared to ODE show that the advantage is not significant (11/7/19, one-tailed $p = 0.100$) for error but is significant for bias (21/6/10, one-tailed $p = 0.035$) and variance (4/9/24, one-tailed $p < 0.001$). Bagging ODE can be seen to bring the error, bias and variance toward that of AODE, lending credibility to an explanation of the effectiveness of AODE in terms of ensembling one-dependence estimators.

5.2. Learning curves

Cross data set experimental studies of the traditional form presented above are of only limited value for gaining deep understanding of the relative prediction characteristics of alternative algorithms. Demonstrating a significant benefit for one algorithm across a group of data sets provides evidence only that the algorithm is likely to perform better with respect to subsequent data sets with similar characteristics. Unfortunately, however, the science of machine learning has made little progress in identifying characteristics that are likely to affect relative classification performance, and hence we have limited ability to generalize from results on one group of data sets to expected results on further data. One proposal that has been made is that data set size interacts with the bias-variance characteristics of an algorithm to affect prediction performance (Brain & Webb, 2002). In particular, it is hypothesized that low variance algorithms tend to enjoy an advantage with small data sets while low bias algorithms tend to enjoy an advantage with larger data sets. The descriptors 'small' and 'large' here are clearly imprecise and impossible to exactly quantify, as the rate at which bias comes to dominate error will depend upon the complexity of each classification task. Nonetheless, this framework does provide us with a precise expectation, that for two algorithms one with lower bias and the other with lower variance, the lower variance algorithm will exhibit lower error at very small data set sizes and that learning curves for the algorithms will eventually cross so that at some larger data set size the low bias algorithm will achieve lower error.

The experiments reported above suggest that AODE, LBR, TAN and SP-TAN all share similar levels of bias. However, as already noted, the data sets are primarily small and the bias-variance evaluation procedure utilizes training sets containing only 25% of each data set. Hence many of the training sets are quite small. We expect the bias of LBR, TAN and SP-TAN to decrease as training set sizes increase, as more data will lead to more accurate probability estimates and hence to more appropriate model selection. Of these three algorithms LBR has the greatest potential to benefit from increases in the quantity of training data as it is able to utilize higher order conditional probabilities where there is sufficient data to obtain accurate estimates thereof. As NB and AODE do not perform model

Table 7. Comparative error, bias and variance for ten largest datasets.

Data	AODE	NB	LBR	TAN	SP-TAN
Mean error	0.173	0.211	0.180	0.181	0.181
Geo mean error ratio		1.335	1.076	1.110	1.081
Mean bias	0.136	0.174	0.126	0.132	0.133
Geo mean bias ratio		1.440	0.920	1.020	0.990
Mean variance	0.036	0.036	0.053	0.048	0.048
Geo mean variance ratio		1.003	1.408	1.352	1.384

Table 8. Win/draw/loss records for ten largest data sets.

	NB		LBR		TAN		SP-TAN	
	W/D/L	*p*	W/D/L	*p*	W/D/L	*p*	W/D/L	*p*
Error	9/0/1	0.011	6/0/4	0.377	8/0/2	0.055	7/0/3	0.172
Bias	9/0/1	0.011	3/0/7	0.172	3/0/7	0.172	4/0/6	0.377
Variance	3/2/5	0.363	8/1/1	0.020	9/1/0	0.002	9/1/0	0.002

selection we do not expect their bias to decrease with increased data in the same manner. In an attempt to assess these predictions we recalculated the mean, geometric mean ratio and win/draw/loss records of NB, LBR, TAN and SP-TAN relative to AODE over the ten largest data sets (those with 1000 or more cases and hence for which the training sets contained 250 or more cases). The mean and geometric mean ratios are presented in Table 7 and the win/draw/loss records are presented in Table 8. All p values are one-tailed as specific predictions are made. However, in no case would the use of two-tailed in place of one-tailed tests affect significance at the 0.05 level. As can be seen, the mean bias over these larger data sets does favor LBR, TAN and SP-TAN, although with the small number of data sets the win/draw/loss records are not significant.

If our reasoning about the expected bias profiles of these algorithms is accepted, it leads to the expectation that naive Bayes should excel compared to AODE, LBR, TAN and SP-TAN at very small data set sizes and then as the quantity of data increases AODE should then come to the fore (with intermediate bias and variance) and then at even larger data set sizes LBR, TAN and SP-TAN should achieve the lowest error, with LBR enjoying an advantage for very large data sets.

To assess this expectation we formed learning curves for the largest data set in our collection, adult, starting with training sets of size 23 and then doubling up to 47104, this particular sequence being contrived to maximize the final term within the constraint that the final term must be less than the total data set size. We repeated 50 experiments. For each experiment 1000 objects were selected at random as a test set and then successive training sets were sampled from the remaining objects and each algorithm was evaluated on the resulting training-test set pairs. The learning curves thereby generated are presented in figure 1, with each point representing the mean error over all 50 experiments and bounded

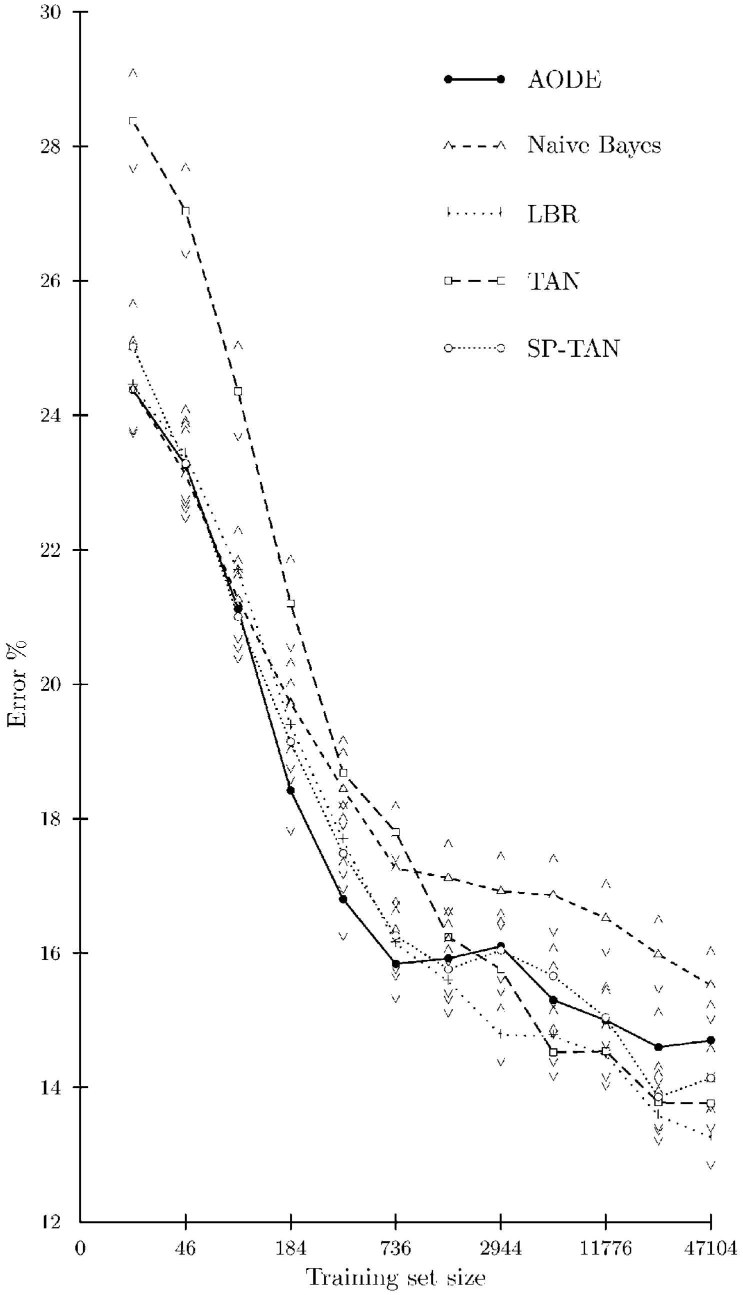

Figure 1. Learning curves on adult dataset.

by error bars that delimit one standard deviation from the mean. Note, due to the large number of comparators, the upper bound of each error bar is indicated by $\wedge$ and the lower bound by $\vee$, with the point of the arrow in each case resting on the bound.

At the smallest training set size AODE defaults to NB and shares equal error lower than TAN and SP-TAN and marginally lower than LBR. At the next two larger sizes there is little separation between the error of NB, AODE, LBR and SP-TAN, all of which have substantially lower error than TAN. At the fourth and fifth training set sizes AODE comes to the fore with error more than one standard error below the next best algorithm. LBR, TAN and SP-TAN then come to overtake AODE, with LBR attaining error that is one standard error below the next closest algorithm at data set sizes 2944 and 47104 and LBR and TAN both exhibiting similar low error rates at the data set sizes in between.

These learning curves correspond well to our predictions, with AODE achieving lower error than NB, LBR, TAN and SP-TAN at intermediate data set sizes but being overtaken by LBR and TAN at larger data set sizes.

5.3. Compute time results

Tables 9 and 10 present summaries of the average CPU time in seconds for each of AODE, NB, LBR, TAN and SP-TAN on each task, broken down into training time and test time. Note that the Weka bias-variance evaluation method results in the use of test sets that are twice the size of the training sets, and hence that the test time is greatly amplified compared with most alternative evaluation methods. Note further that both training and test times include a substantial overhead for discretization. It should also be emphasized that there may be differences in the efficiency of the various implementations, and hence that specific timing results should be regarded at best as broadly indicative. Importantly, the implementation of

Table 9. Training time.

	AODE	NB	LBR	TAN	SP-TAN
Mean	4.42	3.41	4.72	8.60	557.45
Geo mean		0.79	0.97	1.96	17.73
Wins		11	19	30	35
Losses		25	17	3	2
p		0.029	0.868	<0.001	<0.001

Table 10. Testing time.

	AODE	NB	LBR	TAN	SP-TAN
Mean	22.80	2.92	85648.39	3.17	2.04
Geo mean		0.41	156.37	0.39	0.32
Wins		6	30	4	2
Losses		30	7	32	34
p		<0.001	<0.001	<0.001	<0.001

LBR does not include caching that can very substantially reduce classification time (Zheng & Webb, 2000).

The first row of these tables presents the mean time across all data sets. The next row presents the geometric mean across all data sets of the ratio obtained by dividing the training or test time on a data set for the alternative algorithm by that of AODE. A value less than 1.0 indicates that AODE tends to be slower than the alternative while a value greater than 1.0 indicates that AODE tends to be faster. The next row presents the number of data sets for which AODE obtained lower compute time than the alternative algorithm and the final row the number of data sets for which the time for AODE was higher. The final row presents the outcome of a two-tailed binomial sign test presenting the probability that the observed or more extreme record of wins and losses would be obtained if wins and losses were equiprobable.

Comparing AODE to NB, all measures indicate that AODE is slower than NB, being slightly slower at training time and substantially slower at test time. Note that while AODE scores a number of wins over NB, particularly with respect to training time, these are very marginal and represent data sets with small numbers of attributes for which the difference in compute time between the two algorithms is so small that random variations in the compute time dominate the result.

Comparing AODE to LBR, there is little difference between the compute times of the algorithms at training time. In contrast there is a very clear and substantial advantage to AODE at test time.

This profile is reversed when AODE is compared to TAN. AODE has a consistent advantage over TAN at training time and a consistent and substantial disadvantage at test time.

AODE enjoys an even greater advantage at training time compared to SP-TAN, while the suffering the same test-time disadvantage as for TAN.

6. Conclusion

Naive Bayes delivers fast and effective classification with a clear theoretical foundation. It is hampered, however, by the limitations of the attribute independence assumption. The current work is motivated by the desire to obtain the accuracy improvements derived by LBR and SP-TAN from weakening the attribute independence assumption without those techniques' high computational overheads. Our new classification technique averages all models from a restricted class of one-dependence classifiers, the class of all such classifiers that have all other attributes depend on a common attribute and the class. Our experiments suggest that the resulting classifiers have substantially lower bias than naive Bayes at the cost of a very small increase in variance. AODE appears to deliver lower variance but higher bias than LBR, TAN, SP-TAN, a decision tree learner and a boosted decision tree learner. This error profile is achieved without the prohibitive training time of SP-TAN or test time of LBR. In all, we believe that we have been successful in our goal of developing a classification learning technique that retains the simplicity and direct theoretical foundation of naive Bayes while alleviating the limitations of its attribute independence assumption without incurring the same order of computational overhead as LBR and SP-TAN. AODE is particularly suited to incremental learning. Its low variance leads to an expectation of relatively low error for small data sets. Its low training time complexity may be computationally desirable when learning from large data sets.

The success of this approach suggests that it might be profitable to explore approaches to aggregating all of other restricted classes of models, as this strategy avoids model selection and hence minimizes variance, allowing the favorable bias of a low bias class of models to be exploited while reducing the high variance with which low bias is often accompanied.

Acknowledgments

We are very grateful to Fei Zheng and Shane Butler for valuable comments on drafts of this paper. We are indebted to our action editor Tom Fawcett and the anonymous reviewers for valuable feedback and suggestions that greatly improved the paper.

References

Ali, K., Brunk, C., & Pazzani, M. (1994). On learning multiple descriptions of a concept. In *Proceedings of Tools with Artificial Intelligence* (pp. 476–483). New Orleans, LA.

Brain, D., & Webb, G. I. (2002). The need for low bias algorithms in classification learning from large data sets. In *Proceedings of the Sixth European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 2002)* (pp. 62–73). Berlin: Springer-Verlag.

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*, 123–140.

Domingos, P., & Pazzani, M. (1996). Beyond independence: Conditions for the optimality of the simple Bayesian classifier. In *Proceedings of the Thirteenth International Conference on Machine Learning* (pp. 105–112). Morgan Kaufmann.

Fayyad, U. M., & Irani, K. B. (1993). Multi-interval discretization of continuous-valued attributes for classification learning In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-93)* (pp. 1022–1027). Morgan Kaufmann.

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting *Journal of Computer and System Sciences, 55:1*, 119–139.

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning, 29:2*, 131–163.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer.

Keogh, E., & Pazzani, M. (1999). Learning augmented Bayesian classifiers: A comparison of distribution-based and classification-based approaches. In *Proceedings of the International Workshop on Artificial Intelligence and Statistics* (pp. 225–230).

Kohavi, R. (1996). Scaling up the accuracy of naive-Bayes classifiers: A decision-tree hybrid. In *Proceedings of the Second ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-96)* (pp. 202–207). Portland, OR.

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

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

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

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

Nock, R., & Gascuel, O. (1995). On learning decision committees. In *Proceedings of the Twelfth International Conference on Machine Learning* (pp. 413–420). San Francisco: Morgan Kaufmann.

Oliver, J. J., & Hand, D. J. (1995). On pruning and averaging decision trees. In *Proceedings of the Twelfth International Conference on Machine Learning* (pp. 430–437). San Francisco: Morgan Kaufmann.

Pazzani, M. J. (1996). Constructive induction of Cartesian product attributes. In *ISIS: Information, Statistics and Induction in Science* (pp. 66–77). Singapore: World Scientific.

Sahami, M. (1996). Learning limited dependence Bayesian classifiers. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (pp. 334–338). Menlo Park, CA: AAAI Press.

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

Wang, Z., & Webb, G. I. (2002). Comparison of lazy Bayesian rule and tree-augmented Bayesian learning. In *Proceedings of the IEEE International Conference on Data Mining, ICDM-2002* (pp. 775–778). Maebashi, Japan.

Webb, G. I. (2001). Candidate elimination criteria for Lazy Bayesian Rules. In *Proceedings of the Fourteenth Australian Joint Conference on Artificial Intelligence* (pp. 545–556). Berlin: Springer.

Webb, G. I., & Pazzani, M. J. (1998). Adjusted probability naive Bayesian induction. In *Proceedings of the Eleventh Australian Joint Conference on Artificial Intelligence* (pp. 285–295). Berlin: Springer.

Witten, I. H., & Frank, E. (2000). *Data mining: Practical machine learning tools and techniques with Java implementations*. San Francisco, CA: Morgan Kaufmann.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks, 5*, 241–259.

Xie, Z., Hsu, W., Liu, Z., & Lee, M. L. (2002). SNNB: A selective neighborhood based naive Bayes for lazy learning In M.-S. Chen, P. S. Yu, & B. Liu (Eds.), *Advances in knowledge discovery and data mining, proceedings PAKDD 2002* (pp. 104–114). Berlin: Springer.

Yang, Y., & Webb, G. I. (2003). Discretization for naive-Bayes learning: Managing discretization bias and variance Tech. Rep. 2003/131, School of Computer Science and Software Engineering, Monash University.

Zheng, Z., & Webb, G. I. (2000). Lazy learning of Bayesian Rules. *Machine Learning, 41:1*, 53–84.

Zheng, Z., Webb, G. I., & Ting, K. M. (1999). Lazy Bayesian Rules: A lazy semi-naive Bayesian learning technique competitive to boosting decision trees. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML-99)* (pp. 493–502). Morgan Kaufmann.

Received October 30, 2003
Revised April 29, 2004
Accepted April 29, 2004
Final manuscript April 29, 2004