Pre-publication version of a paper accepted in 2007 for publication in the proceedings of ECML'07

Finding the Right Family: Parent and Child Selection for Averaged One-Dependence Estimators

Fei Zheng and Geoffrey I. Webb

Faculty of Information Technology, Monash University, VIC 3800, Australia
{feizheng, Geoff.Webb}@infotech.monash.edu.au

Abstract. Averaged One-Dependence Estimators (AODE) classifies by uniformly aggregating all qualified one-dependence estimators (ODEs). Its capacity to significantly improve naive Bayes' accuracy without undue time complexity has attracted substantial interest. Forward Sequential Selection and Backwards Sequential Elimination are effective wrapper techniques to identify and repair harmful interdependencies which have been profitably applied to naive Bayes. However, their straightforward application to AODE has previously proved ineffective. We investigate novel variants of these strategies. Our extensive experiments show that elimination of child attributes from within the constituent ODEs results in a significant improvement in probability estimate and reductions in bias and error relative to unmodified AODE. In contrast, elimination of complete constituent ODEs and the four types of attribute addition are found to be less effective and do not demonstrate any strong advantage over AODE. These surprising results lead to effective techniques for improving AODE's prediction accuracy.

1 Introduction

Semi-naive Bayesian techniques further improve naive Bayes' accuracy by relaxing its assumption that the attributes are conditionally independent [1–16]. One approach to weakening this assumption is to use an *x-dependence classifier* [7], in which each attribute depends upon the class and at most x other attributes. Examples include Tree Augmented Naive Bayes (TAN) [9], Super Parent TAN (SP-TAN) [11], NBTree [5], Lazy Bayesian rules (LBR) [12] and Averaged One-Dependence Estimators (AODE) [13]. Among these techniques, TAN, SP-TAN and AODE restrict themselves to one-dependence classifiers, which readily admit to efficient computation. Another approach to remedying violations of the attribute independence assumption is to apply naive Bayes with a new attribute set by deleting or merging highly related attributes. Key such approaches include Backwards Sequential Elimination (BSE) [1], Forward Sequential Selection

(FSS) [4], Backward Sequential Elimination and Joining (BSEJ) [6] and Hierarchical Naive Bayes (HNB) [16]. An extensive comparative study of semi-naive Bayes techniques [17] shows that AODE has a significant advantage in error over many other semi-naive Bayesian algorithms, with the exceptions of LBR and SP-TAN. It shares similar levels of error with these two algorithms, while having considerably lower training time complexity relative to SP-TAN and test time complexity relative to LBR. AODE is a powerful alternative to naive Bayes, significantly reducing its error, while retaining much of its attractive simplicity and efficiency. Consequently it has received substantial attention. Indeed, at the time of writing, the paper introducing AODE [13] is the most cited paper from 2005 in the Machine Learning journal [18].

FSS and BSE use a simple heuristic wrapper approach that seeks to minimize error on the training set. Starting from the empty attribute set, FSS operates by iteratively adding attributes, each time adding the attribute whose addition best reduces training set error. BSE uses the opposite search direction and operates by iteratively removing attributes, each time removing the attribute whose elimination most improves training set accuracy. When applied to naive Bayes, FSS and BSE have proved to be beneficial in domains with highly correlated attributes. It is therefore surprising that two attempts to apply these approaches to AODE have proved ineffective [15, 19]. Where the training time overheads of attribute selection are not a major concern, attribute selection has the potential of two beneficial effects, both of improving accuracy and also of reducing test time due to the need to process fewer attributes. This paper investigates why previous approaches to attribute selection for AODE have proved ineffective, and develops novel attribute selection algorithms that do prove effective when applied to AODE and which have potential for wider application.

2 Averaged One-dependence Estimators (AODE)

The Bayesian classifier [20] predicts a class for an unseen example $\mathbf{x} = \langle x_1, \dots, x_n \rangle$ by selecting

$$\operatorname*{argmax}_{y} \left(\hat{P}(y \mid x_1, \dots, x_n) \right), \tag{1}$$

where $\hat{P}(\cdot)$ is an estimate of the probability $P(\cdot)$, x_i is a value of the ith attribute X_i, and $y \in \{c_1, \dots, c_k\}$ is a value of the class variable Y. Naive Bayes estimates $\hat{P}(y \mid x_1, \dots, x_n)$ by assuming that the attributes are independent given the class, and hence classifies $\mathbf{x}$ by selecting

$$\operatorname*{argmax}_{y} \left(\hat{P}(y) \prod_{i=1}^{n} \hat{P}(x_i \mid y) \right). \tag{2}$$

Domingos and Pazzani (1996) point out that interdependencies between attributes will not affect naive Bayes' accuracy, so long as it can generate the correct ranks of conditional probabilities for the classes. However, the success of semi-naive Bayesian methods show that appropriate relaxation of the conditional independence assumption is effective.

One natural extension to naive Bayes is to relax the independence assumption by utilizing a one-dependence classifier (ODE) [7], such as TAN [9], in which each attribute depends upon the class and at most one other attribute. To avoid model selection, AODE [13] selects a limited class of ODEs and aggregates the probability estimates of all qualified classifiers within this class. A single attribute, called the *parent* attribute, is selected as the parent of all the other attributes in each ODE. In order to avoid unreliable base probability estimates, when classifying an object $\langle x_1, \ldots, x_n \rangle$ the original AODE excludes ODEs with parent x_i where the frequency of the value x_i is lower than limit m=30, a widely used minimum on sample size for statistical inference purposes. However, subsequent research [14] shows that this constraint actually increases error and hence the current research uses m=1.

From the definition of conditional probability we have

$$P(y \mid \mathbf{x}) = P(y, \mathbf{x})/P(\mathbf{x}) \propto P(y, \mathbf{x}), \tag{3}$$

and for any attribute value x_i,

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

This equality holds for every x_i. Therefore,

$$P(y, \mathbf{x}) = \frac{\sum_{i:1 \leq i \leq n \wedge F(x_i) \geq m} P(y, x_i)P(\mathbf{x} \mid y, x_i)}{|\{i : 1 \leq i \leq n \wedge F(x_i) \geq m\}|}, \tag{5}$$

where $F(x_i)$ is the frequency of attribute-value x_i in the training sample.

To this end, AODE classifies by selecting:

$$\underset{y}{\operatorname{argmax}} \left(\sum_{i:1 \leq i \leq n \wedge F(x_i) \geq m} \hat{P}(y, x_i) \prod_{j=1}^{n} \hat{P}(x_j \mid y, x_i) \right). \tag{6}$$

At training time AODE generates a three-dimensional table of probability estimates for each attribute-value, conditioned by each other attribute-value and each class. The resulting space complexity is $O(k(nv)^2)$, where v is the mean number of values per attribute. The time complexity of forming this table is $O(tn^2)$, where t is the number of training examples, as an entry must be updated for every training case and every combination of two attribute-values for that case. Classification requires the tables of probability estimates formed at training time of space complexity $O(k(nv)^2)$. The time complexity of classifying a single example is $O(kn^2)$ as we need to consider each pair of qualified parent and child attribute within each class.

AODE maintains the robustness and much of the efficiency of naive Bayes, and at the same time exhibits significantly higher classification accuracy for many data sets. Therefore, it has the potential to be a valuable substitute for naive Bayes over a considerable range of classification tasks.

3 Attribute Selection

In naive Bayes, all attributes are used during prediction, and hence all influence classification. When two attributes are strongly related, the influence from these two attributes may be given too much weight, and the influence of the other attributes may be reduced, which can result in prediction bias. Selecting an appropriate attribute subset, which excludes highly correlated attributes, might alleviate this problem.

Since there are 2^n candidate subsets of n attributes, an exhaustive search of the space is prohibitive. This necessitates the use of heuristic search. Greedy hill climbing is a simple and widely used technique, which adds or removes an attribute irrevocably at each step. That is, once an attribute is added or removed, it cannot be respectively removed from or added to the set. To measure the goodness of alternative attribute subsets, we need an evaluation function, which commonly measures the discriminating ability of an attribute or an attribute set among classes. The *Wrapper* [22] approach uses accuracy estimates on the target induction algorithm as the evaluation function. Leave-one-out cross validation is an attractive technique for estimating accuracy from the training set in Bayesian classifier, as it can be efficiently performed by simply modifying the frequency tables.

Another two issues in hill climbing search are the direction of search and stopping criteria. Forward Sequential Selection (FSS) [4] begins with the empty attribute set and successively adds attributes, while Backwards Sequential Elimination (BSE) [1] starts with the complete attribute set and successively removes attributes. There are three commonly used options for halting the search. We call the first strategy Stop on First Nonimprovement (SFN), as it terminates the search when there is no classification accuracy improvement [1, 6]. The second option, called Stop on First Reduction (SFR), considers performing selection continually so long as the accuracy is not reduced [4]. The third, called Continue Search and Select Best (CSSB), continues the search until all attributes have been added or removed and then selects the attribute subset with the highest accuracy evaluation [19].

In the context of naive Bayes, FSS and BSE select a subset of attributes using leave-one-out cross validation error as a selection criterion and apply naive Bayes to the new attribute set. The subset of selected attributes is denoted as S. Independence is assumed among the resulting attributes given the class. Hence, FSS and BSE classify $\mathbf{x}$ by selecting

$$\underset{y}{\operatorname{argmax}} \left(\hat{P}(y) \prod_{x \in S} \hat{P}(x|y) \right). \tag{7}$$

4 Attribute Selection for AODE

In theory, AODE would appear to be a promising candidate for attribute selection. While an individual ODE can factor out harmful attribute inter-dependencies in which the parent is involved, it will not help when the parent is not. When there are many more attributes than those that participate in a particular inter-dependency, the majority of ODEs will not factor out the inter-dependency, and hence it is credible that deleting one of the attributes should be beneficial. Why then have previous attempts [15,19] to apply attribute-selection to AODE proved unfruitful?

One difference between applying attribute selection in NB compared to AODE may be the greater complexity of an AODE model, resulting in greater variance in estimates of performance as the model is manipulated through attribute elimination and hence reduced reliability in these estimates. Another difference may be that attributes play multiple roles in an AODE model (either a parent or a child) whereas they play only a single role of child in an NB model.

To explore the first issue, we evaluate the use of a statistical test to assess whether an observed difference in holdout evaluation scores should be accepted as meaningful during the attribute selection process.

To explore the second issue, we investigate the separate selection of attributes in each of the parent and child roles, as well as in both roles together.

In the context of AODE, FSS and BSE use leave-one-out cross validation error on AODE as a selection criterion. Each available selection is attempted and the one that results in the lowest error is implemented. The process is repeated for successive attributes until the decrease in error fails a one-tailed binomial sign test at a significance level of 0.05.

To formalize the various attribute selection strategies we introduce into AODE the use of a *parent* (p) and a *child* (c) set, each of which contains the set of indices of attributes that can be employed in respectively a parent or child role in the AODE. The number of indices in each set is denoted respectively as $\|p\|$ and $\|c\|$. We define AODE$_{p,c}$ as

$$\operatorname*{argmax}_{y}\left(\sum_{i\in p:F(x_i)\geq m}\hat{P}(y,x_i)\prod_{j\in c}\hat{P}(x_j\mid y,x_i)\right). \tag{8}$$

Assume that attribute x_i is related to other attributes, and that these harmful interdependencies can be detected and repaired by FSS or BSE. The exclusion of x_i from c may have influence on $\|p\|$ - 1 ODEs, while the exclusion of x_i from p may only factor out the effect of the single ODE in which x_i is the parent. In AODE$_{p,c}$, a linear function is used to combine constituent ODEs, and a multiplicative function is used to combine attributes within each ODE. Large improvements are possible because of the multiplicative influence, and hence exclusion of a child may have greater effect than exclusion of a parent.

4.1 FSS for AODE

There are four different types of attribute addition. The first type of attribute addition, called *parent addition* (PA), starts with p and c initialized to the empty and full sets of $\{1\dots n\}$ respectively. It adds attribute indexes to p, effectively

Table 1. Data sets

No.	Domain	Case	Att	Class	No.	Domain	Case	Att	Class
1	Abalone	4177	9	3	29	Liver Disorders (bupa)	345	7	2
2	Adult	48842	15	2	30	Lung Cancer	32	57	3
3	Annealing	898	39	6	31	Lymphography	148	19	4
4	Audiology	226	70	24	32	Mfeat-mor	2000	7	10
5	Autos Imports-85	205	26	7	33	Mushrooms	8124	23	2
6	Balance Scale	625	5	3	34	Nettalk(Phoneme)	5438	8	50
7	Breast Cancer (Wisconsin)	699	10	2	35	New-Thyroid	215	6	3
8	Car Evaluation	1728	7	4	36	Optical Digits	5620	49	10
9	Chess	551	40	2	37	Page Blocks	5473	11	5
10	Contact Lenses	24	5	3	38	Pen Digits	10992	17	10
11	Credit Approval	690	16	2	39	Pima Indians Diabetes	768	9	2
12	Dmplexer	1000	15	2	40	Postoperative Patient	90	9	3
13	Echocardiogram	131	7	2	41	Primary Tumor	339	18	22
14	German	1000	21	2	42	Promoter Gene Sequences	106	58	2
15	Glass Identification	214	10	3	43	Satellite	6435	37	6
16	Heart	270	14	2	44	Segment	2310	20	7
17	Heart Disease (cleveland)	303	14	2	45	Sign	12546	9	3
18	Hepatitis	155	20	2	46	Sonar Classification	208	61	2
19	Horse Colic	368	23	2	47	Splice-junction Gene Sequences	3190	62	3
20	House Votes 84	435	17	2	48	Syncon	600	61	6
21	Hungarian	294	14	2	49	Sick-euthyroid	3772	30	2
22	Hypothyroid(Garavan Institute)	3772	30	4	50	Tic-Tac-Toe Endgame	958	10	2
23	Ionosphere	351	35	2	51	Vehicle	846	19	4
24	Iris Classification	150	5	3	52	Volcanoes	1520	4	4
25	King-rook-vs-king-pawn	3196	37	2	53	Vowel	990	14	11
26	Labor negotiations	57	17	2	54	Waveform-5000	5000	41	3
27	LED	1000	8	10	55	Wine Recognition	178	14	3
28	Letter Recognition	20000	17	26	56	Zoo	101	18	7

adding a single ODE at each step. The second type of attribute addition, called *child addition* (CA), begins with p and c initialized to the full and empty sets respectively. It adds attribute indexes to c, effectively adding an attribute to within every ODE at each step. Starting with the empty set for both p and c, *Parent and child addition* (P∧CA) at each step adds the same value to both p and c , hence selecting it for use in any role in the classifier. *Parent or child addition* (P∨CA) performs any one of the other types of attribute additions in each iteration, selecting the option that most improves the accuracy.

4.2 BSE for AODE

All four types of attribute elimination start with p and c initialized to the full set. The first approach, called *parent elimination* (PE), deletes attribute indexes from p, effectively deleting a single ODE at each step. The second approach, called *child elimination* (CE), deletes attribute indexes from c, effectively deleting an attribute from within every ODE at each step. *Parent and child elimination* (P∧CE) [15] at each step deletes the same value from both p and c, thus eliminating it from use in any role in the classifier. *Parent or child elimination* (P∨CE) performs any one of the other types of attribute eliminations in each iteration, selecting the option that best reduces error.

4.3 Complexity

As child selection requires modifying the probability estimates for $\|p\|$ ODEs at each step, it has higher training time complexity than that of parent selection, which only considers one ODE at each step. At training time PA and PE generate a three-dimensional table of probability estimates, as AODE does. They must also store the training data, with additional space complexity $O\big(tn\big)$, to perform leave-one-out cross validation on AODE. A three-dimensional table, indexed by instance, class and attribute, is introduced to speed up the process of evaluating the classifiers, with space complexity $O\big(tkn\big)$. Therefore, the resulting space complexity is $O\big(tkn + k(nv)^2\big)$. Deleting attributes has time complexity of $O\big(tkn^2\big)$, as a single leave-one-out cross validation is order $O(tk)$ and it is performed at most $O\big(n^2\big)$ times. They have identical time and space complexity with AODE at classification time. For the strategies involving child selection, they have identical space complexity and classification time complexity with PA and PE, but higher training time complexity of $O\big(tkn^3\big)$, as a single leave-one-out cross validation is order $O(tkn)$.

4.4 Statistical test

It is quite likely that small improvements in leave-one-out error may be attributable to chance. In consequence it may be beneficial to use a statistical test to assess whether an improvement is significant. We employ a standard binomial sign test. Treating the examples for which an attribute addition or deletion corrects a misclassification as a *win* and one for which it misclassifies a previously correct example as a *loss*, a change is accepted if the number of wins exceeds the number of losses and the probability of obtaining the observed number of wins and losses if they were equiprobable was no more than 0.05.

5 Empirical comparison

The main goal in this comparison is to assess the efficacy of the statistical test and study the influence of the use of different types of attribute selection in AODE. The fifty-six natural domains from the UCI Repository of machine learning [23] used in our experiments are shown in Table 1. Continuous attributes were discretized using MDL discretization [24] and missing values were replaced with the modes and means from the training data. The base probabilities were estimated using Laplace estimation [25]. Algorithms are implemented in the Weka workbench [26], and the experiments were performed on a dual-processor 1.7 GHz Pentium 4 Linux computer with 2 Gb RAM.

We compare the classification error of AODE with different attribute selection techniques on AODE using the repeated cross-validation bias-variance estimation method proposed by Webb (2000). This is preferred to the default method in Weka, which uses 25% of the full data set as training sets, because it results in the use of substantially larger training sets. In order to maximize the variation in the training data from trial to trial we use two-fold cross validation. The training data are randomly divided into two folds. Each fold is used as a test set for a classifier generated from the other fold. Hence, each available example is classified once for each two-fold cross-validation. Bias and variance are estimated by fifty runs of two-fold cross-validation in order to give a more accurate estimation of the average performance of an algorithm. The advantage of this technique is that it uses the full training data as the training set and test set, and every case in the training data is used the same number of times in each of the roles of training and test data. In addition to the classification error, we use the information loss function to evaluate the probabilistic prediction of each technique.

Two variants of attribute selection were evaluated, one employing a binomial sign test and the other not. Algorithms using a binomial sign test are superscripted by ^{S} and those without by ^{NS}. We use Stop on First Reduction with attribution addition algorithms and Stop on First Nonimprovement with attribute elimination algorithms as these produce the best performance (results not presented due to lack of space). The number of times that an algorithm performs better, worse or equally to the others is summarized into pairwise win/loss/draw records which are presented in Table 2. Algorithms are sorted in descending order on the value of wins minus losses against AODE on each metric. Each entry compares the algorithm with which the row is labelled (L) against the algorithm with which the column is labelled (C). We assess a difference as significant if the outcome of a one-tailed binomial sign test is less than 0.05. For space reason, we only present bias, variance and information loss results for the attribute elimination algorithms.

5.1 Error

CE^{S}, P$\vee$CE^{S} and P$\wedge$CE^{S}, enjoy a significant advantage in error over AODE ($p = 0.011$, $p = 0.011$ and $p = 0.048$ respectively), while attribute addition (both with and without statistical test) always has a significant disadvantage to AODE. The rest of the algorithms share a similar level of error with AODE.

The algorithms using attribute elimination share a similar level of error with the exception that CE^{S} and P$\vee$CE^{S} outperform PE^{S}, PE^{NS} outperforms CS^{NS} and P$\wedge$CE^{NS} outperforms CE^{NS}. The advantage of all the attribute elimination algorithms is significant compared with all the attribute addition algorithms but PA^{NS}. PA^{NS} has a significant advantage over CA, P$\wedge$CA and P$\vee$CA (with and without statistical test). The reason the performances of CA, P$\wedge$CA and P$\vee$CA are disappointing might be that they are susceptible to getting trapped into poor selections by local minima during the first several child additions.

5.2 Bias and Variance

All the attribute elimination algorithms, except PE^{S}, have a significant advantage in bias over AODE and PE^{S}. P$\wedge$CE^{NS}, CE^{NS} and P$\vee$CE^{NS} outperform PE^{NS} and the remaining four algorithms with statistical tests. The advantage of P$\wedge$CE^{NS} is significant compared with CE^{NS}. AODE enjoys a significant advantage over all the algorithms with respect to variance. The algorithms with statistical tests have a significant advantage over the algorithms without a statistical test. PE^{S} has a significant advantage over P$\wedge$CE^{S}.

5.3 Information Loss

Two algorithms, P∧CE^S and CE^S, significantly improve AODE's probability estimate. PE^{NS} is the only algorithm that has a significant disadvantage over AODE. It also has a significant disadvantage over P∧CE^S, CE^S, P∨CE^S and PE^S. The advantage of P∨CE^S is marginal compared with AODE and is significant compared with CE^S and PE^{NS}.

5.4 Continue Search and Select Best (CSSB)

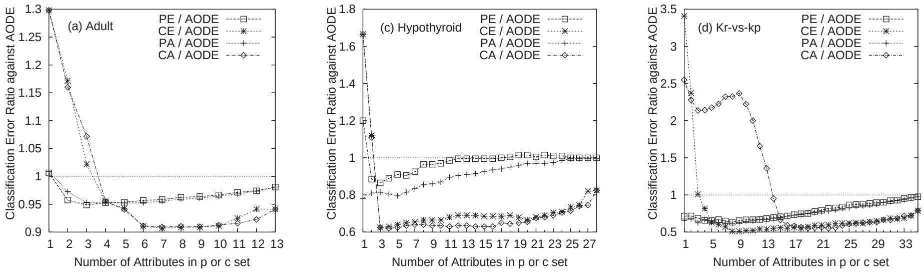

Fig. 1. Error ratio of parent and child selection using CSSB against AODE, as function of the number of attributes

To observe the behaviors of parent and child selection, we also examine the attribute selection techniques with CSSB. Due to the significantly increasing variance, all of these selection approaches have proved ineffective. Figure 1 shows the error ratio of PA [19], PE, CA and CE against AODE as a function of the number of attributes on 3 data sets with more than 3000 instances, in which both selection of parent and child have lower error compared with AODE (for the space reason, the other 6 data sets are not presented). The values on the x-axis are the number of attributes in the p set for PA and PE, and the number of attributes in the c set for CA and CE. The values on the y-axis are the classification error of each selection algorithm divided by that for AODE. The smaller the ratio, the more accuracy improvement will be.

Slight error differences between PA and PE are observed as shown in the graph (win/draw/loss being 25/8/23). Notice that PA tends to achieve the minima at an early stage, while PE appears to reach it at a late stage. CE has greater error reduction compared with PE until there are a small number of children left, after which it increases error sharply. The error ratios for PE and CE for the first attribute elimination are 0.98 and 0.94, 0.99 and 0.96, 1 and 0.83, and 0.98 and 0.79 for Adult, Nettalk, Hypothyroid and King-rook-vs-king-pawn respectively. The performance of CA fluctuates over the first several attribute additions for King-rook-vs-king-pawn. Similar behavior is observed for many other data sets in our collection.

Table 2. Win/Loss/Draw records on 56 data sets with binomial sign test.

Error

W/L/D	CE^{S}	$\mathrm{P}\vee\mathrm{CE}^{S}$	$\mathrm{P}\wedge\mathrm{CE}^{S}$	PE^{NS}	PE^{S}	CE^{NS}	$\mathrm{P}\vee\mathrm{CE}^{NS}$	$\mathrm{P}\wedge\mathrm{CE}^{NS}$	PA^{NS}	$\mathrm{P}\wedge\mathrm{CA}^{NS}$	PA^{S}	CA^{NS}	$\mathrm{P}\vee\mathrm{CA}^{NS}$	$\mathrm{P}\vee\mathrm{CA}^{S}$	$\mathrm{P}\wedge\mathrm{CA}^{S}$	CA^{S}
CE^{S}																
$\mathrm{P}\vee\mathrm{CE}^{S}$	6/5/45															
$\mathrm{P}\wedge\mathrm{CE}^{S}$	8/6/42	3/8/45														
PE^{NS}	24/23/9	24/23/9	24/23/9													
PE^{S}	**4/13/39**	**4/14/38**	6/12/38	20/25/11												
CE^{NS}	24/27/5	24/27/5	24/27/5	**19/34/3**	25/26/5											
$\mathrm{P}\vee\mathrm{CE}^{NS}$	24/29/3	24/29/3	24/29/3	23/28/5	24/28/4	32/18/6										
$\mathrm{P}\wedge\mathrm{CE}^{NS}$	25/28/3	24/29/3	24/29/3	21/31/4	23/29/4	**33/17/6**	19/31/6									
PA^{NS}	22/32/2	22/31/3	22/31/3	21/33/2	23/31/2	29/23/4	21/32/3	26/26/4								
$\mathrm{P}\wedge\mathrm{CA}^{NS}$	**17/37/2**	**17/37/2**	**17/37/2**	**13/41/2**	**17/37/2**	**15/38/3**	**10/44/2**	**12/40/4**	**13/41/2**							
PA^{S}	**13/41/2**	**12/42/2**	**12/42/2**	**10/43/3**	**14/40/2**	**13/41/2**	**10/44/2**	**11/43/2**	**4/49/3**	24/29/3						
CA^{NS}	**14/40/2**	**14/40/2**	**14/40/2**	**10/44/2**	**13/41/2**	**8/44/4**	**7/47/2**	**7/46/3**	**10/44/2**	**15/35/6**	**18/36/2**					
$\mathrm{P}\vee\mathrm{CA}^{NS}$	**12/42/2**	**12/42/2**	**12/42/2**	**12/42/2**	**12/42/2**	**17/37/2**	**16/38/2**	**16/38/2**	**15/39/2**	21/32/3	23/31/2	29/25/2				
$\mathrm{P}\vee\mathrm{CA}^{S}$	**11/43/2**	**11/43/2**	**11/43/2**	**10/44/2**	**10/44/2**	**15/39/2**	**14/40/2**	**14/40/2**	**13/41/2**	**19/35/2**	**19/35/2**	28/26/2	**4/20/32**			
$\mathrm{P}\wedge\mathrm{CA}^{S}$	**9/46/1**	**9/46/1**	**9/46/1**	**9/46/1**	**10/45/1**	**7/48/1**	**6/49/1**	**6/49/1**	**7/48/1**	**4/50/2**	**10/44/2**	**14/41/1**	**18/37/1**	**20/34/2**		
CA^{S}	**6/49/1**	**6/49/1**	**6/49/1**	**6/49/1**	**7/48/1**	**5/50/1**	**6/49/1**	**6/49/1**	**6/49/1**	**3/51/2**	**7/47/2**	**6/47/3**	**13/42/1**	**16/39/1**	**19/34/3**	
AODE	**3/13/40**	**3/13/40**	**5/13/38**	20/25/11	4/3/49	26/24/6	28/24/4	29/23/4	31/23/2	**37/17/2**	**40/14/2**	**41/13/2**	**42/12/2**	**44/10/2**	**45/10/1**	**48/7/1**

Bias

W/L/D	$\mathrm{P}\wedge\mathrm{CE}^{NS}$	CE^{NS}	$\mathrm{P}\vee\mathrm{CE}^{NS}$	PE^{NS}	$\mathrm{P}\vee\mathrm{CE}^{S}$	CE^{S}	$\mathrm{P}\wedge\mathrm{CE}^{S}$	PE^{S}
$\mathrm{P}\wedge\mathrm{CE}^{NS}$								
CE^{NS}	**14/36/6**							
$\mathrm{P}\vee\mathrm{CE}^{NS}$	28/26/2	32/23/1						
PE^{NS}	**12/40/4**	**12/40/4**	**12/42/2**					
$\mathrm{P}\vee\mathrm{CE}^{S}$	**6/47/3**	**7/45/4**	**8/47/1**	19/28/9				
CE^{S}	**6/47/3**	**7/45/4**	**9/46/1**	19/28/9	3/8/45			
$\mathrm{P}\wedge\mathrm{CE}^{S}$	**6/47/3**	**7/45/4**	**8/47/1**	19/28/9	2/7/47	7/5/44		
PE^{S}	**6/47/3**	**7/45/4**	**9/46/1**	**14/30/12**	**1/16/39**	**2/16/38**	**2/15/39**	
AODE	**6/47/3**	**7/45/4**	**9/46/1**	**14/30/12**	**1/16/39**	**2/16/38**	**2/15/39**	2/5/49

Variance

W/L/D	PE^{S}	CE^{S}	$\mathrm{P}\wedge\mathrm{CE}^{S}$	$\mathrm{P}\vee\mathrm{CE}^{S}$	PE^{NS}	CE^{NS}	$\mathbf{P}\wedge\mathbf{CE}^{NS}$	$\mathrm{P}\vee\mathrm{CE}^{NS}$
PE^{S}								
CE^{S}	6/13/37							
$\mathrm{P}\wedge\mathrm{CE}^{S}$	**5/14/37**	4/8/44						
$\mathrm{P}\vee\mathrm{CE}^{S}$	6/11/39	4/9/43	6/4/46					
PE^{NS}	**14/32/10**	**16/32/8**	**16/31/9**	**17/31/8**				
CE^{NS}	**10/42/4**	**11/42/3**	**11/42/3**	**10/43/3**	**14/39/3**			
$\mathrm{P}\wedge\mathrm{CE}^{NS}$	**11/42/3**	**11/42/3**	**11/41/4**	**11/41/4**	**12/41/3**	20/27/9		
$\mathrm{P}\vee\mathrm{CE}^{NS}$	**11/40/5**	**11/40/5**	**11/40/5**	**11/40/5**	**16/37/3**	**31/16/9**	**37/14/5**	
AODE	**6/0/50**	**14/5/37**	**15/5/36**	**13/5/38**	**33/14/9**	**43/10/3**	**42/11/3**	**40/11/5**

Info Loss

W/L/D	CE^{NS}	$\mathrm{P}\wedge\mathrm{CE}^{NS}$	$\mathrm{P}\wedge\mathrm{CE}^{S}$	CE^{S}	$\mathrm{P}\vee\mathrm{CE}^{NS}$	$\mathrm{P}\vee\mathrm{CE}^{S}$	PE^{S}	PE^{NS}
CE^{NS}								
$\mathrm{P}\wedge\mathrm{CE}^{NS}$	26/27/3							
$\mathrm{P}\wedge\mathrm{CE}^{S}$	21/33/2	22/32/2						
CE^{S}	21/33/2	21/33/2	7/8/41					
$\mathrm{P}\vee\mathrm{CE}^{NS}$	28/27/1	27/28/1	33/23/0	32/24/0				
$\mathrm{P}\vee\mathrm{CE}^{S}$	21/33/2	21/32/3	9/5/42	**11/2/43**	23/33/0			
PE^{S}	22/32/2	22/32/2	7/14/35	6/13/37	24/32/0	7/13/36		
PE^{NS}	26/28/2	25/29/2	**16/36/4**	**16/36/4**	25/31/0	**16/36/4**	**16/35/5**	
AODE	22/32/2	22/32/2	**6/15/35**	**5/13/38**	24/32/0	7/14/35	2/6/48	**35/16/5**

6 Conclusion

AODE efficiently induces classifiers that have competitive classification performance with other state-of-the-art semi-naive Bayes algorithms. Its accuracy and classification time complexity might be further improved if harmful ODEs are excluded. In view of their effectiveness with naive Bayes, it is surprising that previous applications of FSS and BSE to AODE have proved ineffective. In this paper we explore two explanations of this phenomenon. One is that AODE has higher variance compared with naive Bayes, and hence appropriate variance management is required. Another is that child selection appears to have greater effect than parent selection, as ODEs are combined using a linear function but attributes within an ODE are combined using a multiplicative function.

Our extensive experiments suggest that the types of attribute elimination that remove child attributes from within the constituent ODEs can significantly reduce bias and error, but only if a statistical test is employed to provide variance management. In contrast, elimination of complete constituent ODEs does not consistently provide error reduction. CE and P$\wedge$CE also significantly improve probability estimates when used with a statistical test. The types of attribute addition that add child attributes to within the constituent ODEs do not provide any positive benefits, possibly due to being mislead early in the search by local minima. These results suggest that the elimination of a child is more effective than the elimination of a parent, leading to effective approaches to further enhance AODE's accuracy.

References

1. Kittler, J.: Feature selection and extraction. In Young, T.Y., Fu, K.S., eds.: Handbook of Pattern Recognition and Image Processing. Academic Press, New York (1986) 60–81
2. Kononenko, I.: Semi-naive Bayesian classifier. In: Proc. 6th European Working Session on Machine learning, Berlin: Springer-Verlag (1991) 206–219
3. Langley, P.: Induction of recursive Bayesian classifiers. In: Proc. 1993 European Conf. Machine Learning, Berlin: Springer-Verlag (1993) 153–164
4. Langley, P., Sage, S.: Induction of selective Bayesian classifiers. In: Proc. 10th Conf. Uncertainty in Artificial Intelligence, Morgan Kaufmann (1994) 399–406
5. Kohavi, R.: Scaling up the accuracy of naive-Bayes classifiers: a decision-tree hybrid. In: Proc. 2nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining. (1996) 202–207
6. Pazzani, M.J.: Constructive induction of Cartesian product attributes. ISIS: Information, Statistics and Induction in Science (1996) 66–77
7. Sahami, M.: Learning limited dependence Bayesian classifiers. In: Proc. 2nd Int. Conf. Knowledge Discovery in Databases, Menlo Park, CA: AAAI Press (1996) 334–338
8. Singh, M., Provan, G.M.: Efficient learning of selective Bayesian network classifiers. In: Proc. 13th Int. Conf. Machine Learning, Morgan Kaufmann (1996) 453–461
9. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Machine Learning **29**(2) (1997) 131–163

10. Webb, G.I., Pazzani, M.J.: Adjusted probability naive Bayesian induction. In: Proc. 11th Australian Joint Conf. Artificial Intelligence, Berlin:Springer (1998) 285–295
11. Keogh, E.J., Pazzani, M.J.: Learning augmented Bayesian classifers: A comparison of distribution-based and classification-based approaches. In: Proc. Int. Workshop on Artificial Intelligence and Statistics. (1999) 225–230
12. Zheng, Z., Webb, G.I.: Lazy learning of Bayesian rules. Machine Learning **41**(1) (2000) 53–84
13. Webb, G.I., Boughton, J., Wang, Z.: Not so naive Bayes: Aggregating one-dependence estimators. Machine Learning **58**(1) (2005) 5–24
14. Cerquides, J., Mántaras, R.L.D.: Robust Bayesian linear classifier ensembles. In: Proc. 16th European Conf. Machine Learning, Lecture Notes in Computer Science. (2005) 70–81
15. Zheng, F., Webb, G.I.: Efficient lazy elimination for averaged-one dependence estimators. In: Proc. 23th Int. Conf. Machine Learning (ICML 2006), ACM Press (2006) 1113–1120
16. Langseth, H., Nielsen, T.D.: Classification using hierarchical naive Bayes models. Machine Learning **63**(2) (2006) 135 – 159
17. Zheng, F., Webb, G.I.: A comparative study of semi-naive Bayes methods in classification learning. In: Proc. 4th Australasian Data Mining Conference (AusDM05). (2005) 141–156
18. Thomson ISI: Web of science. http://scientific.thomson.com/products/wos/ (2007)
19. Yang, Y., Webb, G., Cerquides, J., Korb, K., Boughton, J., Ting, K.M.: To select or to weigh: A comparative study of model selection and model weighing for SPODE ensembles. In: Proc. 18th European Conf. Machine Learning (ECML), 2006, Springer (2006) 533–544
20. Duda, R.O., Hart, P.E.: Pattern Classification and Scene Analysis. John Wiley and Sons, New York (1973)
21. Domingos, P., Pazzani, M.J.: Beyond independence: Conditions for the optimality of the simple Bayesian classifier. In: Proc. 13th Int. Conf. Machine Learning, Morgan Kaufmann (1996) 105–112
22. John, G.H., Kohavi, R., Pfleger, K.: Irrelevant features and the subset selection problem. In: Proc. 11th Int. Conf. Machine Learning, San Francisco, CA: Morgan Kaufmann (1994) 121–129
23. Newman, D., Hettich, S., Blake, C., Merz, C.: UCI repository of machine learning databases. [http://www.ics.uci.edu/ mlearn/mlrepository.html]. Irvine, CA: University of California, Department of Information and Computer Science. (1998)
24. Fayyad, U.M., Irani, K.B.: Multi-interval discretization of continuous-valued attributes for classification learning. In: Proc. 13th Int. Joint Conf. Artificial Intelligence (IJCAI-93), Morgan Kaufmann (1993) 1022–1029
25. Cestnik, B.: Estimating probabilities: A crucial task in machine learning. In: Proc. 9th European Conf. Artificial Intelligence, London: Pitman (1990) 147–149
26. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann (2005)
27. Webb, G.I.: Multiboosting: A technique for combining boosting and wagging. Machine Learning **40**(2) (2000) 159–196