To Select or To Weigh: A Comparative Study of Linear Combination Schemes for SuperParent-One-Dependence Estimators

Ying Yang†, Geoffrey I. Webb†, Jesús Cerquides‡, Kevin Korb†, Janice Boughton† and Kai Ming Ting†
† Faculty of Information Technology, Monash University, Australia
{ying.yang,geoff.webb,kevin.korb,janice.boughton,kaiming.ting}@infotech.monash.edu.au
‡ Departament de Matemàtica Aplicada i Anàlisi, Universitat de Barcelona, Spain
cerquide@maia.ub.es

***Abstract*— We conduct a large-scale comparative study on linearly combining superparent-one-dependence estimators (SPODEs), a popular family of semi-naive Bayesian classifiers. Altogether 16 model selection and weighing schemes, 58 benchmark data sets, as well as various statistical tests are employed. This paper's main contributions are three-fold. First, it formally presents each scheme's definition, rationale and time complexity; and hence can serve as a comprehensive reference for researchers interested in ensemble learning. Second, it offers bias-variance analysis for each scheme's classification error performance. Third, it identifies effective schemes that meet various needs in practice. This leads to accurate and fast classification algorithms with immediate and significant impact on real-world applications. Another important feature of our study is using a variety of statistical tests to evaluate multiple learning methods across multiple data sets.**

***Index Terms*— Classification learning, Bayesian probabilistic learning, ensemble learning, model selection, model weighing, superparent-one-dependence estimator (SPODE).**

I. Introduction

Ensemble learning is a popular method in classification learning. It combines multiple learning models' decisions to produce more accurate results than single models [1]–[5]. This paper focuses on two particular aspects of ensemble learning, selection and weighing of models for linear model combination. The goal is to study formally alternative selection or linear weighing schemes in theory and to identify effective and efficient ones for practical use.

The general problem for model selection is, given some sample data, how to decide which are the most effective models within some model space. The general problem of linear model weighing focuses on calculating the weight associated with each model within some model space and accordingly weighing their decisions when ensembling.

This paper looks at the model space of Bayesian network classifiers. In particular, superparent-one-dependence estimators (SPODEs) [6], [7], a popular family of semi-naive Bayesian classifiers, are taken as a vehicle of illustration throughout the research.

This paper presents 16 alternative model selection or weighing schemes. Selection schemes include Akaike's information criterion (AIC), Bayesian information criterion (BIC), minimum description length (MDL), minimum message length (MML), random selection (RAN), cross validation (CV), forward sequential addition (FSA), backward sequential elimination (BSE), lazy elimination (LE). Weighing schemes include Akaike's information criterion (AIC), Bayesian information criterion (BIC), minimum description length (MDL), minimum message length (MML), Bayesian model averaging (BMA), maximum a posteriori linear mixture of discriminative distributions (MAPLMD), and maximum a posteriori linear mixture of generative distributions (MAPLMG). A large-scale empirical comparison using 58 benchmark data sets is conducted to test the classification accuracy and efficiency of ensembles that result from using alternative schemes. A variety of statistics are employed to thoroughly evaluate and rank their performances.

By doing this research, we seek answers to the following questions:

1) What are every scheme's strength and weakness for ensemble learning?
2) Which scheme is consistently among the best algorithms for our large suite of data sets?
3) In general, which is more effective and/or more efficient, model selection or model weighing?
4) How to choose which scheme to use in practice?

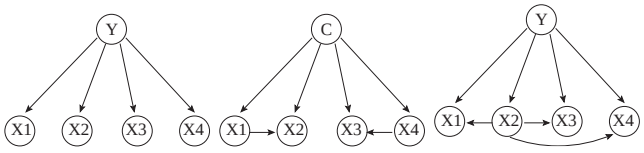
II. Background

This section defines terminology and notation that will be used throughout this paper. It also explains how a SPODE and an ensemble of SPODEs carry out classification.

A. Terminology and Notation

This paper addresses the problem of classification learning using Bayesian network classifiers. The following terminology and notation will be used.

An *instance* $\mathbf{x} \langle x_1, x_2, \cdots, x_m \rangle$ is a vector of m attribute values x_i, each observed for an attribute variable X_i ($i \in [1, m]$). As SPODEs currently require discrete-valued data, numeric attributes are discretized. An instance can also have a class label y corresponding to the class variable Y. If its class label is known, an instance is *labeled*. Otherwise, it is *unlabeled*. Whenever applicable, for the purpose of uniformity in formulae, X_i represents the class variable when $i = m+1$.



(a) NB (b) ODE (c) SPODE

Fig. 1. Illustration of NB, ODE and SPODE. An arc points from a parent to a child. A child only depends on its parents. NB assumes each attribute only depends on the class Y and is independent of other attributes given the class. ODE allows each attribute depends on at most one other attribute in addition to the class. SPODE assumes that each attribute can depend on a *common* attribute (the superparent X_2) in addition to the class.

Training data D is a set of labeled instances from which a *classifier* is learned to predict the class labels of unlabeled instances. The number of training instances is n. The number of values for X_i is v_i. X_i's parent variables are $\Phi(i)$. The number of joint states (joint instantiated values) of parents of X_i is $|\phi(i)|$. The r-th joint state of the parents is ϕ_{ir}. When applicable, h indicates a SPODE in general and h_i indicates a particular SPODE whose superparent is X_i. Generally the $\log$ base in information metrics does not matter. A common practice is to use e or 2.

B. SPODE

Bayesian network classifiers have long been a core technique in predictive learning. The naive-Bayesian (NB) classifier is among the first Bayesian networks introduced into machine learning. NB assumes attributes conditionally independent of each other given the class. It is very efficient with reasonable prediction accuracy [8]–[15]. In recent years, there has also been considerable interest in developing variants of NB that weaken the attribute independence assumption in order to further improve the prediction accuracy [6], [7], [16]–[30]. For instance, one-dependence estimators (ODEs) [23] such as tree-augmented naive Bayes (TAN) [16] provide a powerful alternative to NB. As depicted in Figure 1, an ODE is similar to an NB except that each attribute is allowed to depend on at most one other attribute in addition to the class. Among ODEs, SPODEs [6], [7] have received a lot of attention because they offer a combination of high training efficiency, high classification efficiency and high classification accuracy [30]–[47]. Those merits give SPODEs a great potential to substitute for naive Bayes classifiers in numerous real-world classification systems, including medical diagnosis, fraud detection, email filtering, document classification and webpage prefetching. As illustrated in Figure 1, a SPODE relaxes NB's attribute independence assumption by allowing all attributes to depend on a common attribute, the *superparent*, in addition to the class.

To classify an instance $\mathbf{x}$, a Bayesian network classifier calculates $\hat{P}(y \mid \mathbf{x})$ for each $y \in Y$, an estimate of the probability of the class label given this instance $P(y \mid \mathbf{x})$. The label attaining the highest probability will be assigned to $\mathbf{x}$. Since $P(y \mid \mathbf{x}) = \frac{P(y,\mathbf{x})}{P(\mathbf{x})}$ and $P(\mathbf{x})$ is invariant across different class labels, one only needs to estimate $P(y, \mathbf{x})$ as:

$$\text{argmax}_y \; P(y \mid \mathbf{x}) = \text{argmax}_y \; P(y, \mathbf{x}). \tag{1}$$

A SPODE with superparent X_p uses Formula (2) to calculate $\hat{P}(y, \mathbf{x})$. The second equation results from SPODEs' assumption that all attributes are independent of each other given the class Y and the superparent X_p.

$$\begin{aligned} \hat{P}(y, \mathbf{x}) &= \hat{P}(y, x_p)\hat{P}(\mathbf{x} \mid y, x_p) \\ &= \hat{P}(y, x_p) \prod_{i=1}^{m} \hat{P}(x_i \mid y, x_p) \end{aligned} \tag{2}$$

C. SPODE Ensemble

There has been a strong interest in ensembling SPODEs because it can decrease a single SPODE's classification variance, and attain high classification accuracy with moderate time requirement [30]–[47].

For a training data set with m attributes, there can be m candidate SPODEs, each taking a different attribute as its superparent. A SPODE ensemble is a linear combination of multiple SPODEs' probability estimates. It classifies $\mathbf{x}$ using Formula (3), where each $\hat{P}_j(y, \mathbf{x})$ is calculated by a SPODE using Formula (2) with $p = j$.

$$\hat{P}(y, \mathbf{x}) = \sum_{j=1}^{m} w_j \hat{P}_j(y, \mathbf{x}) \tag{3}$$

The first approach to ensembling SPODES used equal weight combination of all SPODEs whose parent value occurred above a user-specified minimum frequency in the training data [30]. Subsequent research suggested that frequency is not a useful model selection criterion and that appropriate weighing can substantially improve upon equal weighing, such as in MAPLMD and MAPLMG weighing schemes [31]. On the other hand, it has also been shown that model selection can be effective when ensembling SPODEs [35], [44]. This paper presents a comprehensive investigation into the relative merits of alternative approaches to weighing and selecting.

III. Model selection Schemes

The general problem for model selection is, given some sample data, how to decide which are the most effective models within some model space. This paper looks at the space of SPODE models. Only selected SPODEs will be included in the ensemble. Strictly speaking, model selection is an extreme form of model weighing where the weights are either 1 or 0. That is,

$$w_j = \begin{cases} 1 & \text{if SPODE}_j \text{ is selected} \\ 0 & \text{otherwise} \end{cases}$$

However, because information-theoretic schemes take different forms when used in model selection versus weighing, this study differentiates selection from weighing.

A. Information-Theoretic Metrics

Information-theoretic metrics including AIC, BIC, MDL and MML [48]–[51], provide a combined score, as in Formula (4), for a proposed explanatory model (a SPODE in our context) and for the data given the model. They aim to find a balance between goodness of fit (minimizing $I(D|h)$) and model simplicity (minimizing $I(h)$), and thereby achieve good modeling performance without overfitting the data. The best score is the smallest. Hence the lower the score a SPODE gets, the higher its priority to appear in the ensemble.

$$score = I(D|h) + I(h). \tag{4}$$

The term $I(D|h)$ is ***shared*** by information-theoretic metrics and is:

$$I(D|h) = n\left(\sum_{i=1}^{m+1} H(X_i) - \sum_{i=1}^{m+1} H(X_i, \Phi(i))\right) \tag{5}$$

where $H(X_i)$ is the entropy of X_i, and $H(X_i, \Phi(i))$ is the mutual information between X_i and its parents:

$$H(X_i) = -\sum_{j=1}^{v_i} (P(X_i = x_{ij}) \log P(X_i = x_{ij})) \tag{6}$$

$$H(X_i, \Phi(i)) = \sum_{j=1}^{v_i}\sum_{r=1}^{|\phi_i|}\left(P(x_{ij}, \phi_{ir}) \log \frac{P(x_{ij}, \phi_{ir})}{P(x_{ij})P(\phi_{ir})}\right).$$

How to compute $I(h)$ ***varies*** among different schemes and is presented below.

a) Akaike's Information Criterion (AIC): According to Akaike [48],

$$I_{AIC}(h) = 2\left(\sum_{i=1}^{m+1}(v_i - 1)\prod_{j\in\Phi(i)} v_j\right). \tag{7}$$

For any root node X_i (where $\Phi(i) = \emptyset$), the product term on the right should be replaced by 1. The same principle also applies to BIC and MDL, below.

b) Bayesian Information Criterion (BIC): According to Schwarz [49],

$$I_{BIC}(h) = (\log n)\left(\sum_{i=1}^{m+1}(v_i - 1)\prod_{j\in\Phi(i)} v_j\right). \tag{8}$$

c) Minimum Description Length (MDL): According to Suzuki [50],

$$I_{MDL}(h) = (\frac{1}{2}\log n)\left(\sum_{i=1}^{m+1}(v_i - 1)\prod_{j\in\Phi(i)} v_j\right). \tag{9}$$

d) Minimum Message Length (MML): According to Korb and Nicholson [51],

$$\begin{aligned} I_{MML}(h) &= \log(m+1)! + C_2^{m+1} - \log(m-1)! \\ &+ \sum_{i=1}^{m+1}\frac{v_i - 1}{2}(\log\frac{\pi}{6} + 1) \\ &- \log\prod_{i=1}^{m+1}\prod_{j=1}^{|\phi_i|}\left(\frac{(v_i-1)!}{(S_{ij}+v_i-1)!}\prod_{l=1}^{v_i}\alpha_{ijl}!\right) \end{aligned} \tag{10}$$

where S_{ij} is the number of training instances where the parents $\Phi(i)$ take their joint j-th value, and α_{ijl} is the number of training instances where X_i takes its l-th value and $\Phi(i)$ take their j-th joint value. For any root X_i, $|\phi_i|$ should be treated as 1 and every instance should be treated as matching the parents for the purposes of computing S_{ij} and α_{ijl}. Formula (10) looks complicated, but it can be computed in polynomial time [52].

Each information-theoretic metric can order a sequence of SPODEs by their supposed merits. One should then expect that excluding poorly predictive SPODEs could improve the classification accuracy. For instance, after it has reached the optimal classification accuracy, an ensemble should not proceed to include additional SPODEs that are counterproductive, even when there are some left. To decide when SPODEs of sufficient merit are no longer to be found for the ensemble given an ordered sequence of m SPODEs, m ensembles are tested. Starting with an empty ensemble, each ensemble in turn includes further one SPODE in the queue. Every ensemble's leave-one-out cross validation accuracy is calculated. The ensemble with the lowest error is the one to be selected.

B. Random Selection (RAN)

RAN randomly orders SPODEs. Following the practice with information-theoretic metrics, it then tests m ensembles from size 1 to size m; and the one with the lowest leave-one-out cross validation error is selected. RAN has low computational overhead and offers a useful comparator against which to judge the impact on classification error of other selection schemes.

C. Cross Validation (CV)

CV [35] scores each individual SPODE by its cross validation error in the training data. Particularly in this study, leave-one-out cross validation is employed. Given a SPODE, CV loops through the training data n times, each time training the SPODE from $(n-1)$ instances to classify the remaining 1 instance. The misclassifications are summed and averaged over n iterations. The resulting classification error rate is taken as the metric value of the SPODE. The lower the metric, the higher priority for the SPODE to be used. This process is very efficient as the model need only be updated for each instance that is left out, rather than recalculated from scratch.

Following the practice with information-theoretic metrics, after CV orders SPODEs according to their merits, it tests m ensembles from size 1 to size m; and the one with the lowest leave-one-out cross validation error is selected.

D. Forward Sequential Addition (FSA)

Inspired by the forward sequential selection strategy for attribute selection in NB [21], FSA [35] begins with an empty ensemble. It then uses hill-climbing search to iteratively add SPODEs most helpful for lowering the ensemble's classification error. In each iteration, suppose the current ensemble is $E_{current}$ with k SPODEs. FSA in turn adds each candidate SPODE, one that has not been included into $E_{current}$, and obtains an ensemble E_{test} of size $(k+1)$. It then calculates the leave-one-out cross validation error of E_{test}. The E_{test}

who obtains the lowest error is retained. The corresponding added SPODE is permanently included into the ensemble and deleted from the candidate list. The same process is applied to the new SPODE ensemble of size $(k+1)$ and so on, until every SPODE has been included. The order of addition produces a ranking order for SPODEs. The earlier a SPODE is added, the more merit it possesses and the higher its priority to be used.

The ensemble that achieves the lowest leave-one-out cross validation error in training during the addition process is selected. If multiple ensembles attain the lowest error, the one that includes most SPODEs is chosen, as a means to reduce variance caused by model selection [30].

E. Backward Sequential Elimination (BSE)

Inspired by the backward sequential elimination strategy for attribute selection in NB [21], BSE [35] starts out with a full ensemble including every SPODE. It then uses hill-climbing search to iteratively eliminate SPODEs whose individual exclusion most helpful for lowering the classification error. In each iteration, suppose the current ensemble is $E_{current}$ involving k SPODEs. BSE eliminates each member SPODE in turn from $E_{current}$ and obtains an ensemble E_{test} of size $(k-1)$. It then calculates the leave-one-out cross validation error of E_{test}. The E_{test} which yields the lowest error is retained. The corresponding eliminated SPODE is permanently deleted from the ensemble. The same process is applied to the new SPODE ensemble of size $(k-1)$ and so on, until the ensemble is empty. The order of the elimination produces a ranking order for SPODEs. The earlier a SPODE is eliminated, the less merit it possesses and the lower its priority to be used.

The ensemble that achieves the lowest leave-one-out cross validation error in training during the elimination process is selected. If multiple ensembles attain the lowest error, the one that includes most SPODEs is chosen, as a means to reduce variance caused by model selection [30].

F. Lazy Elimination (LE)

The above schemes studied sofar select at training time a subset of SPODEs that are used to classify all test instances. An alternative approach delays selection until classification time. LE [44] is based on the observation that $\forall a, b, c : P(a \mid b) = 1.0$ entails $P(c \mid a, b) = P(c \mid b)$. Hence, if it can be inferred that one attribute value entails another, assuming conditional independence between the values is likely to be harmful and the more general value a may safely be deleted. To this end, before a test instance is classified LE deletes any attribute value x_i of the instance that occurs in the training data more than a user-defined minimum number of times (in this research, 30) and for which there is another value $x_j (j \neq i)$ such that x_i is present in every training instance containing x_j. If x_i and x_j are identical, only one is deleted. Effectively, LE performs lazy selection, by not using SPODEs whose superparents are generalizations of other values of the instance to be classified. Note however that it also deletes children from within SPODEs and hence is not solely a SPODE selection algorithm.

IV. Linear Model Weighing Schemes

Linear model weighing focuses on calculating the weight associated with each SPODE to linearly combine their probability estimates of $P(y, \mathbf{x})$ as in Formula (3).

A. Information Theoretic Metrics

Since the information-theoretic metrics AIC, BIC, MDL and MML as defined in Section III-A rely upon Shannon information theory [53] for their motivation and interpretation, it is appropriate to ask what kind of probabilistic weight they imply for purpose of prediction. In principle, they should support the inversion of Shannon's law to derive the posterior probability of a model given the data for such purposes. Hence, the weight w for a SPODE h is:

$$\begin{aligned} w &= \hat{P}(h|D) \\ &= e^{-I(h|D)} \\ &= e^{-(I(D|h)+I(h)-I(D))} \end{aligned} \tag{11}$$

where $I(D) = n \sum_{i=1}^{m+1} H(X_i)$ is the entropy of data whose $H(X_i)$ calculated by Equation (6); $I(D|h)$ is calculated by Equation (5); and $I(h)$ is calculated by Equations (7, 8, 9, 10) respectively for AIC, BIC, MDL and MML to be weights.

B. Bayesian Model Averaging (BMA)

BMA [54], [55] is theoretically the optimal method for combining learned models. It provides a coherent mechanism to ensemble classification models by accounting for single models' uncertainty of generating the data. In the Bayesian view, using a single model to make predictions ignores the uncertainty caused by training data as to which is the correct model; thus all possible models in the model space under consideration should be used when making predictions, with each model weighted by its probability of being the correct model $P(h_i \mid D)$.

Given an instance $\mathbf{x}$ and a set of classifiers h_i, BMA estimates the probability of each class label given $\mathbf{x}$ using:

$$\hat{P}(y \mid \mathbf{x}) = \sum_{i=1}^{m} \hat{P}(y \mid h_i) \hat{P}(h_i \mid D) \tag{12}$$

where $\hat{P}(y \mid h_i)$ is the class probability estimated by a SPODE as in Formula (2). One common approach to estimating the weight was proposed by Cooper and Herskovits [52]:

$$w_i = \hat{P}(h_i \mid D) = \frac{\hat{P}(h_i, D)}{\sum_{i=1}^{m} \hat{P}(h_i, D)} \tag{13}$$

where

$$\hat{P}(h_i, D) = \hat{P}(h_i) \prod_{k=1}^{m+1} \prod_{j=1}^{|\phi_i|} \left(\frac{(v_k - 1)!}{(S_{kj} + v_k - 1)!} \prod_{l=1}^{v_k} \alpha_{kjl}! \right),$$

$$\hat{P}(h_i) = \frac{1}{m} \text{ if there are } m \text{ candidate SPODEs,}$$

and S_{kj} and α_{kjl} have the same meanings as in Equation (10).

C. Maximum a Posteriori Linear Mixture of Generative Distributions (MAPLMG)

The method of maximum a posteriori (MAP, or posterior mode) estimation can be used to obtain a point estimate of an unobserved quantity on the basis of empirical data. It is closely related to Fisher's method of maximum likelihood (ML), but employs an augmented optimization objective which incorporates a prior distribution over the quantity one wants to estimate. MAPLMG and MAPLMD both assume as prior distribution a Dirichlet over the SPODE ensemble weights. Once this is done, they use MAP estimation to find the most probable set of weights for a SPODE ensemble given a concrete dataset. The difference between MAPLMG and MAPLMD is that the former finds the MAP weights for an ensemble of generative probabilistic models whilst the latter finds the MAP weights for an ensemble of discriminative probabilistic models.

MAPLMG [31] constructs a SPODE ensemble that maximizes the supervised posterior probability of the weights given the training data. It determines the weighing vector $\mathbf{w}\,\langle w_1, \ldots, w_m\rangle$ as

$$\mathbf{w} = \text{argmax}_{\mathbf{w}}\ \hat{P}_{LMG}(\mathbf{w}|D) \tag{14}$$

where

$$\hat{P}_{LMG}(\mathbf{w}|D) = \prod_{\langle \mathbf{x},y\rangle \in D} \left(\frac{\sum_{i=1}^{m} w_i \hat{P}_i^{LOO}(y,\mathbf{x})}{\sum_{y\in Y}\sum_{i=1}^{m} w_i \hat{P}_i^{LOO}(y,\mathbf{x})} \prod_{i=1}^{m} w_i \right)$$

and $\hat{P}_i^{LOO}(y,\mathbf{x}) = \hat{P}(x_i,y)\prod_{j=1}^{m}\hat{P}(x_j \mid x_i, y)$ whose right hand side is estimated from $(D - \{\langle \mathbf{x}, y\rangle\})$ for h_i. The maximization appearing in (14) is a constrained nonlinear optimization problem that can be solved by means of a sequence of unconstrained maximizations [56], each of them solved by a Newton-like optimization procedure such as BFGS [57].

D. Maximum a Posteriori Linear Mixture of Discriminative Distributions (MAPLMD)

A scheme closely related to MAPLMG is MAPLMD. It also constructs a SPODE ensemble that maximizes the supervised posterior probability of the weights. It differs from MAPLMG in that the ensemble constructed linearly combines $\hat{P}_i(y \mid \mathbf{x})$ instead of $\hat{P}_i(y,\mathbf{x})$ in Formula (3):

$$\hat{P}(y \mid \mathbf{x}) \approx \sum_{i=1}^{m} w_i \hat{P}_i(y \mid \mathbf{x}).$$

It determines weights as

$$\mathbf{w} = \text{argmax}_{\mathbf{w}}\ \hat{P}_{LMD}(\mathbf{w}|D) \tag{15}$$

where

$$\hat{P}_{LMD}(\mathbf{w}|D) \propto \prod_{\langle \mathbf{x},y\rangle \in D} \left(\sum_{i=1}^{m} w_i \hat{P}_i^{LOO}(y|\mathbf{x}) \prod_{i=1}^{m} w_i \right)$$

and $\hat{P}_i^{LOO}(y|\mathbf{x})$ is h_i's probability estimate for $\mathbf{x}$'s true class given $(D - \{\langle \mathbf{x}, y\rangle\})$. The maximization appearing in (15) can be computed by means of the Expectation-Maximization algorithm [58].

V. Time Complexity Analysis

Assume that the number of training instances and attributes are n and m, and number of classes is c. Let the average number of values for an attribute be v.

A. Training Overhead

The time complexity of each scheme to order SPODEs by their merits or to calculate their weights is as follows.

1) AIC, BIC and MDL: The complexity of calculating $I(D|h)$ is $O(mv^2c)$. The dominating part is from $H(X_i, \Phi(i))$ which iterates through every attribute ($O(m)$), and then every value ($O(v)$), and then every joint value of the superparent and the class ($O(vc)$). The complexity of calculating $I(h)$ is $O(m)$.[1] Since the selection repeats for each attribute ($O(m)$), the overall complexity is $O(m \times (mv^2c + m)) = O(m^2v^2c)$.

2) MML and BMA: The dominating complexity of MML as well as BMA for SPODEs is from $\prod_{i=1}^{m+1}\prod_{j=1}^{|\phi_i|}\frac{(v_i-1)!}{(S_{ij}+v_i-1)!}\prod_{l=1}^{v_i}\alpha_{ijl}!$. MML iterates through each attribute ($O(m)$); and then each joint value of the superparent and the class ($O(vc)$) for which two factorials are calculated ($O(v) + O(\frac{n}{vc})$). On top of that it loops through each attribute value ($O(v)$) for which a third factorial is calculated ($O(\frac{n}{v^2c})$). Hence the complexity is $O(m * vc * (v + \frac{n}{vc}) * v * \frac{n}{v^2c}) = O(mn(v + \frac{n}{vc}))$. This repeats for each attribute ($O(m)$) and the overall complexity is hence $O(m^2n(v + \frac{n}{vc}))$.

3) CV: To classify an instance, a SPODE will multiply the conditional probability of each attribute value given each class label and one (constant) superparent value. This results in $O(mc)$. To do leave-one-out cross validation, the classification will repeat n times. Hence the complexity is $O(mcn)$. This repeats for each attribute ($O(m)$) and the overall complexity is hence $O(m^2cn)$.

4) FSA: The hill climbing procedure of increasing a SPODE ensemble from empty to size m will render a complexity of $O(m^2)$. In the first round, it alternatively adds each of m SPODEs. In the second round, it alternatively adds each of $(m-1)$ SPODEs. Following this line of reasoning, the total number of probing a SPODE is $m + (m-1) + \cdots + 2 + 1 = O(m^2)$. As explained for CV, to test each SPODE by leave-one-out cross validation will incur complexity of $O(mcn)$. As a result, the overall complexity is $O(m^3cn)$.

5) BSE: The hill climbing procedure of reducing a SPODE ensemble of size m to 0 will render a complexity of $O(m^2)$. In the first round, it alternatively eliminates each of m SPODEs. In the second round, it alternatively eliminates each of $(m-1)$ SPODEs. Following this line of reasoning, the total number of probing a SPODE is $m + (m-1) + \cdots + 2 + 1 = O(m^2)$. As explained for CV, to test each SPODE by leave-one-out cross validation will incur complexity of $O(mcn)$. As a result, the overall complexity is $O(m^3cn)$.

6) LE: LE does not require any additional information to be gathered at training time and hence has no impact on training time.

[1] Although MDL has an extra loop $\prod_{j\in\Phi(i)} v_j$, in case of a SPODE, $|\Phi(i)|$ is of maximum value 2 (the superparent and the class). Hence it can be treated as a constant and does not increase the order of the complexity.

Data set	Ins.	Att.	Data	Ins.	Att.	Data	Ins.	Att.
Abalone	4177	8	Ionosphere	351	34	PrimaryTumor	339	17
JapaneseVowels	9961	12	IrisPlant	150	4	Promoter	106	57
Annealing	898	38	KRvsKP	3196	36	Satellite	6435	36
Audiology	226	69	LaborRelations	57	16	ImageSegmentation	2310	19
Automobile	205	25	LEDDisplay	1000	7	SickEuthyroid	3772	29
BalanceScale	625	4	LetterRecognition	20000	16	AustralianSignLanguage	12546	8
Bands	539	36	LiverDisorders	345	6	Sonar	208	60
BreastCancer	699	9	LungCancer	32	56	Soybean	683	35
Chess	551	39	Lymphography	148	18	Spambase	4601	57
ContraceptiveMethodChoice	1473	9	MultipleFeaturesMorphological	2000	6	SpliceJunction	3177	60
CreditScreening	690	15	Mushrooms	8124	22	SyntheticControl	600	60
Echocardiogram	131	6	Musk	476	166	Thyroid	9169	29
German	1000	20	NettalkPhoneme	5438	7	TicTacToe	958	9
GlassIdentification	214	9	NewThyroid	215	5	Vehicle	846	18
HeartDiseaseCleveland	303	13	OpticalDigits	5620	48	Vowel	990	11
Hepatitis	155	19	PageBlocks	5473	10	Waveform	5000	40
HorseColic	368	21	PenBasedRecognition	10992	16	Wine	178	13
CongressionalVoting	435	16	PimaIndiansDiabetes	768	8	Yeast	1484	8
HeartDiseaseHungarian	294	13	Postoperative	90	8	Zoo	101	16
Hypothyroid	3772	29						

TABLE I

STATISTICS OF EXPERIMENTAL DATA SETS. 'INS.' AND 'ATT.' ARE THE NUMBER OF INSTANCES AND ATTRIBUTES RESPECTIVELY.

7) MAPLMD: The computation of the optimal weights can be implemented in two steps. In the first step, $P_i^{LOO}(y|\mathbf{x})$ of each h_i is computed for each training instance. This takes $O(m^2cn)$, as reasoned in Section V-A.3. After that, the EM algorithm iterates until convergence or until a maximum number of 10000 iterations is reached. Each EM iteration takes $O(nmc)$. The complete computational complexity is therefore $O(m^2cn + Kmnc)$ where K is the bound of the number of iterations in the maximization algorithm. Since K is fixed, it does not affect the theoretical computational complexity, but influences the computing time when m, n and c are not relatively large enough. Hence we keep the large constant K in the complexity expression.

8) MAPLMG: The computation of the optimal weights can be implemented in two steps. In the first step, $P_i^{LOO}(y, \mathbf{x})$ of each h_i is computed for each training instance. This takes $O(m^2cn)$, as reasoned in Section V-A.3. After that, the maximum is found by a sequence of applications of the BFGS minimization algorithm until convergence or a maximum number of 1000 iterations is reached. Each BFGS iteration computes both the value of the function it tries to maximize and the value of its derivative. In this case this can be done in $O(nmc)$. Following the same reasoning as for the above MAPLMD, The complete computational complexity is therefore $O(m^2cn + Kmnc)$.

B. Classification overhead

For selection schemes, the result is a linear combination of SPODEs. Hence, each scheme's complexity is of the same order $O(m^2c)$, resulting from the $O(mc)$ SPODE algorithm applied over an $O(m)$ sized ensemble. Please note that lazy elimination requires a test each time a pair of attribute values is considered to determine whether one is a generalization of the other, incurring an additional complexity $O(m^2)$.

For weighing schemes, following the above lines of reasoning, its classification complexity is $O(m^2c)$. More precisely, weighing's complexity is higher than selection's by O(1), resulting from multiplying each SPODE's probability estimate by its weight.

VI. EXPERIMENTS

Empirical tests, observations, analyses and evaluations are presented here for each selection or weighing scheme. The objective function is to maximize the learning accuracy and efficiency of resulting ensemble classifiers. AODE [30], a complete SPODE ensemble without any selection or weighing applied, is also included to offer a baseline in comparing alternative schemes.

A. Data

Rival schemes are implemented in the WEKA machine learning environment [59], and are validated using a large suite of 58 benchmark data sets from the UCI machine learning repository and KDD archive [60], as described in Table I. Because SPODEs currently require discrete-valued data, numeric attributes are discretized using the WEKA MDL discretizer [59]. Since part of the software (information metrics) does not handle missing values, following WEKA's practice, missing values for nominal and numeric attributes in a data set are replaced with the modes and means respectively.

B. Design

Each scheme is tested on each data set using a 30-trial 2-fold cross validation. An s-fold cross validation divides a data set into s equal-size subsets. Each subset is used in turn as a test set with the remaining $(s-1)$ data sets used for training. One may conduct s-fold cross validation for t trials, each trial shuffling the instances and forming s different

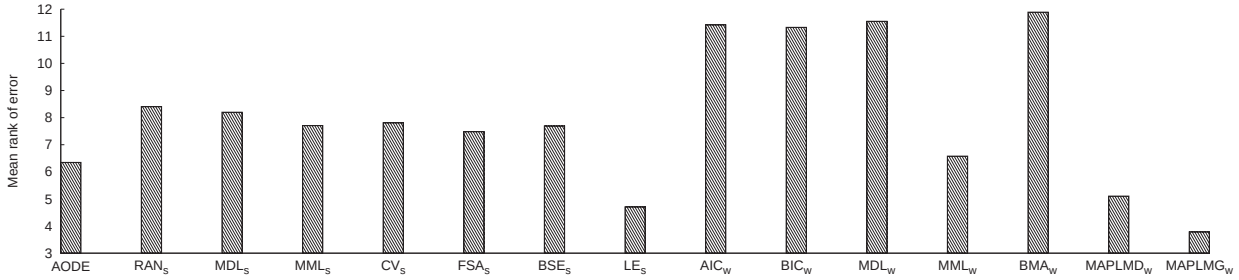


Fig. 2. Compare alternative methods' mean ranks of reducing error.

subsets. The reason that we use a substantial number (30) of trials is because we perform bias-variance decomposition analysis, which is more accurate when sufficient trials are conducted [61]. The reason that we use 2-fold cross validation is to maximize the variation in the training data from trial to trial.

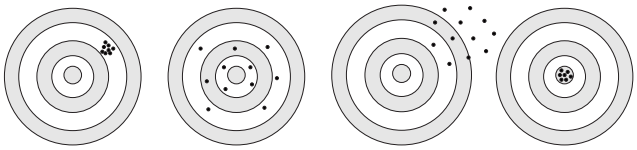
Five performance measures are recorded on each data set: *training time*, *classification time*, *classification error* which can be decomposed into a *bias* term and a *variance* term [61]–[65]. A third *irreducible* term is the error of an optimal algorithm (the level of noise in the data). In our study, following Kohavi and Wolpert's method, it is merged into bias [64].

Please note that varying from our previous research, we no longer impose a frequency threshold on SPODEs. Previously as a means to reduce classification variance, a SPODE was considered a candidate for ensembling only if the parent value's frequency was above 30 [35]. However, subsequent research demonstrated better results when the minimum frequency was reduced to 1 [31]. Accordingly, some experimental results differ from those obtained in previous otherwise equivalent experiments [35].

C. Bias-variance decomposition of error

It is useful to look into bias and variance of a classifier because they each offer a different perspective on classification error. Bias describes the component of error that results from systematic error of the learning algorithm. Variance describes the component of error that results from random variation in the training data and from random behavior in the learning algorithm, and thus measures how sensitive an algorithm is to changes in the training data. As the algorithm becomes more sensitive, the variance increases.

Moore and McCabe [66] illustrated bias and variance through shooting arrows at a target, as reproduced in Figure 3. We can think of the perfect model as the bull's-eye on a target, and the learned classifier as an arrow fired at the bull's-eye. Bias and variance describe what happens when an archer fires many arrows at the target. Bias means that the aim is off and the arrows land consistently off the bull's-eye in the same direction. Large variance means that repeated shots are widely scattered on the target. They do not give similar results but differ widely among themselves. A good learning scheme, like a good archer, should have both low bias and low variance. We use Kohavi and Wolpert's definitions of bias and variance [64].



(a) High bias, low variance (b) Low bias, high variance (c) High bias, high variance (d) Low bias, low variance

Fig. 3. Bias and variance in shooting arrows at a target. Bias means that the archer systematically misses the bull's eye in the same direction. Variance means that the arrows are scattered. [66]

Each instance is classified once in each trial and hence 30 times in all.

D. Statistics

A variety of statistics are employed to evaluate measured performance of each competing scheme.

- **Mean of ranks** Following the practice of Friedman [67], [68], for each data set, we rank competing algorithms. The one that attains the best performance is ranked 1, the second best ranked 2, so on and so forth. A method's mean rank is obtained by averaging its ranks across all data sets. Compared with mean value (the arithmetic mean of measured performance, such as error, across all data sets), mean rank can reduce the susceptibility to outliers that, for instance, allows a classifier's excellent performance on one data set to compensate for its overall bad performance [69].
- **Friedman test** As recommended by Demsar [69], the Friedman test is effective for comparing multiple algorithms across multiple data sets. It compares mean ranks of schemes to decide whether to reject the null-hypothesis, which states that all the schemes are equivalent and so their ranks should be equal.
- **Nemenyi test** If the Friedman test rejects its null-hypothesis, we can proceed with a post-hoc test, the Nemenyi test. It can be applied to mean ranks of competing schemes and indicate whose performances have statistically significant differences (here we use the 0.05 critical level).
- **Win/lose/tie record** (w/l/t) can be calculated for each pair of competitors A and B with regard to a performance

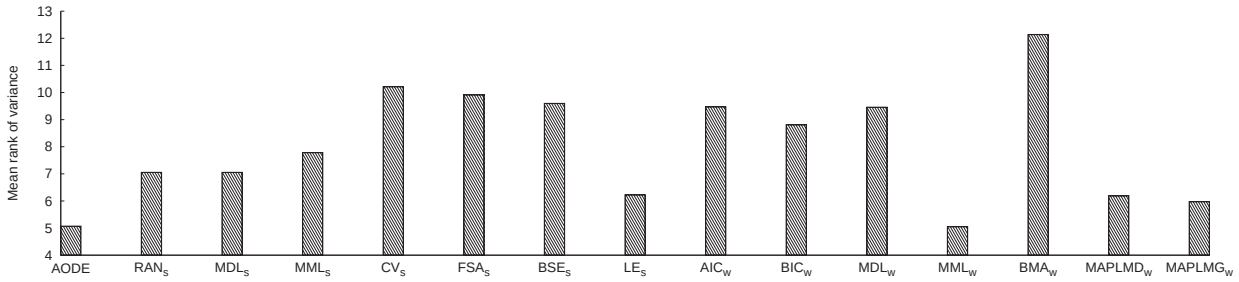


Fig. 4. Compare alternative methods' mean ranks of reducing variance.

measure M. The record represents the number of data sets in which A respectively beats, loses to or ties with B on M. To avoid breaking the flow of the main text, the w/l/t records on error, bias and variance are respectively listed in Tables II, III and IV in the Appendix.

E. Observations and analyses

Because information metrics can act as both selection and weighing schemes, we add a suffix '$_s$' to each selection scheme while a '$_w$' to each weighing scheme. When acting as selection schemes, AIC, BIC and MDL produce the same order of SPODEs and select the same ones. Hence MDL$_s$ represents the results for AIC$_s$ and BIC$_s$ as well.

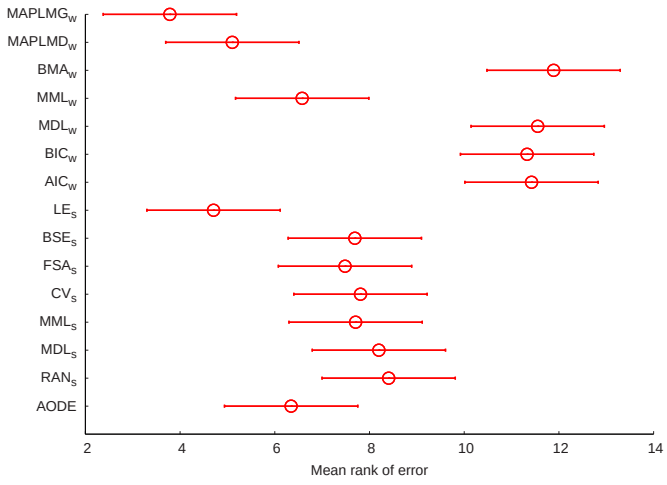


Fig. 6. Apply the Nemenyi test to alternative schemes' mean ranks of reducing error.

1) Reducing classification error: To compare each scheme's influence on the SPODE ensemble's classification error, their mean ranks of reducing error are illustrated in Figure 2. It indicates that among selection schemes, LE$_s$ is the most effective on reducing classification error; while among weighing schemes, MAPLMG$_w$ is the most effective. It also reveals an interesting point that AODE, which simply linearly combines every SPODE without any selection or weighing, is actually more effective than the majority of rival schemes. We partially attribute this to AODE's outstanding performance on reducing variance, which will be discussed in Section VI-E.2.

When we apply the Friedman test, with 15 algorithms[2] and 58 data sets, F_F is distributed according to the F distribution with $(15-1)=14$ and $(15-1)\times(58-1)=798$ degrees of freedom. The critical value of $F(14,798)$ at the 0.05 critical level is 1.7. F_F calculated from the mean ranks is 28.3. Since $28.3 > 1.7$, we can reject the null hypothesis and infer that there exists significant difference among rival schemes.

To find out exactly which schemes are significantly different, we proceed to the Nemenyi test, whose results are illustrated in Figure 6. In the graph, the mean rank of each scheme is pointed by a circle. The horizontal bar across each circle indicates the 'critical difference'. The performance of two methods is significantly different if their corresponding mean ranks differ by at least the critical difference. That is, two methods are significantly different if their horizontal bars are not overlapping. For instance, Figure 6 reveals that MAPLMG$_w$ is ranked best and is significantly better than RAN$_s$, MDL$_s$, MML$_s$, CV$_s$, FSA$_s$, BSE$_s$, AIC$_w$, BIC$_w$, MDL$_w$ and BMA$_w$.

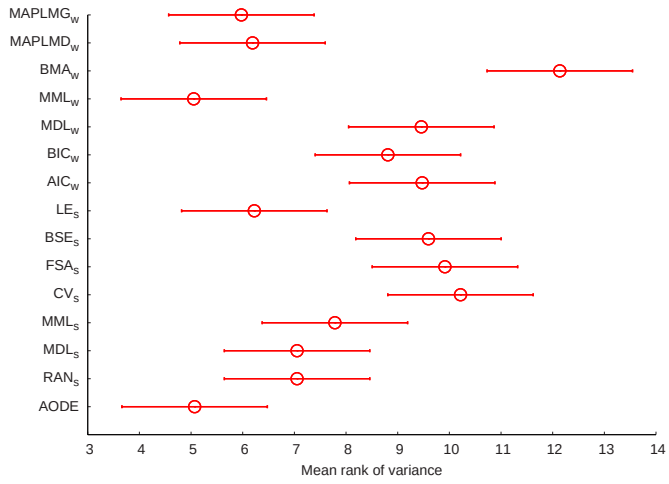


Fig. 7. Apply the Nemenyi test to alternative schemes' mean ranks of reducing variance.

2) Reducing classification variance: Figure 4 illustrates each scheme's mean rank of reducing variance. The Friedman test indicates that there exist significant differences among

[2]We have studied 16 schemes. For experimental purpose, AIC$_s$ and BIC$_s$ are presented by MDL$_s$ because they produce the same results. AODE is added as a benchmark algorithm. As a result, there are 15 algorithms tested.

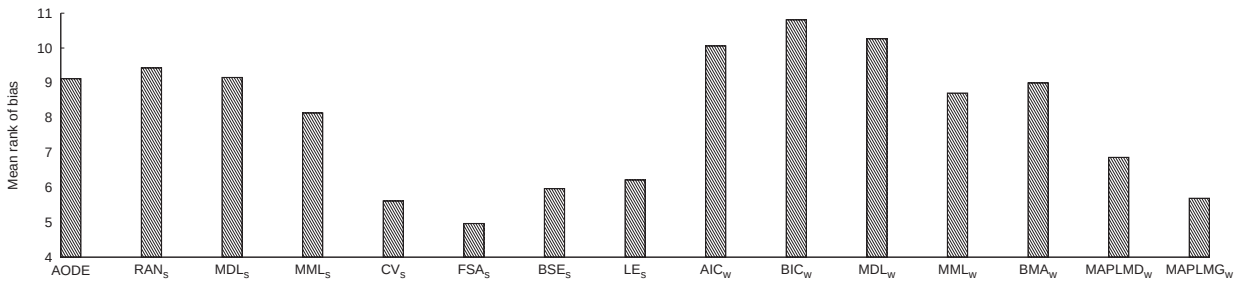


Fig. 5. Compare alternative methods' mean ranks of reducing bias.

schemes on reducing variance and Figure 7 depicts the results of the Nemenyi test to reveal what those differences are.

It is observed that AODE and MML$_w$ are the best at reducing classification variance among alternative methods. Between themselves, AODE beats MML$_w$ more often than not (w/l/t being 11/8/39) according to Table IV. We suggest the reason for AODE's outstanding performance on variance reduction is that selection and weighing will increase the classifier's sensitivity to training data because weights as well as selection metrics are calculated therefrom. In contrast, AODE minimizes dependence on training data and hence can minimize classification variance.

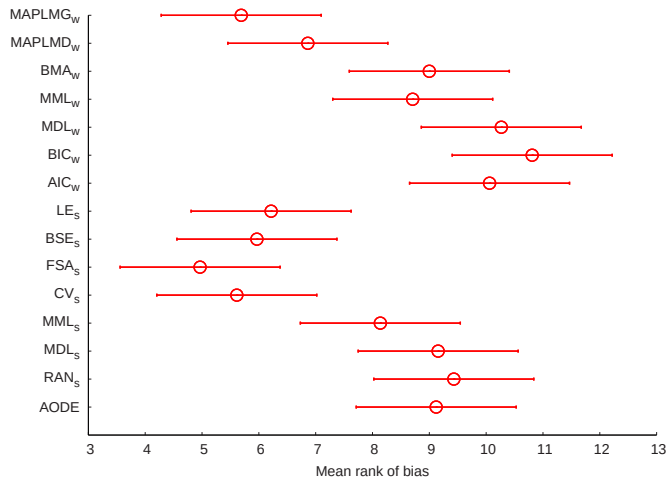


Fig. 8. Apply the Nemenyi test to alternative schemes' mean ranks of reducing bias.

3) Reducing classification bias: Figure 5 illustrates each scheme's mean rank of reducing bias. The Friedman test indicates that there exist significant differences among schemes on reducing bias and Figure 8 depicts the results of the Nemenyi test to reveal what those differences are.

It is observed that on reducing bias, model selection schemes like FSA$_s$, CV$_s$ and BSE$_s$ are the most effective. However, their outstanding capability for bias reduction is overshadowed by their inferior performance on variance reduction (refer to Figure 7). The net effect is that they are worse at reducing error for SPODE ensembles. In contrast, schemes like LE$_s$ and MAPLMG$_w$ reduce bias as well as control variance and turn out to be more effective at error reduction for SPODE ensembles.

4) Capability for fast training: Figure 9 illustrates the mean ranks of alternative schemes' training time. Consistent with our time complexity analyses in Section V-A, AODE and LE$_s$ that do not conduct any selection or weighing work in training are the most efficient. MAPLMD$_w$ and MAPLMG$_w$ optimize multiple weights simultaneously, which very likely contributes to their effectiveness since others calculate the weights for individual SPODEs in isolation. On the other hand, this optimization demands time and hence MAPLMD$_w$ and MAPLMG$_w$ are slower than every other scheme except MML$_w$ and BMA$_w$.

MML and BMA can return large values. In that case, when serving as weighing schemes, MML$_w$ and BMA$_w$ involve calculating large exponentials in Formula (11). This often leads to arithmetic overflow when using 32-bit computing machines. Our solution to this problem is to use the java class *BigDecimal* that implements arbitrary-precision signed decimal numbers. A BigDecimal consists of an arbitrary precision integer unscaled value and a non-negative 32-bit integer scale, which represents the number of digits to the right of the decimal point. Although BigDecimal solves the problem of overflowing, its calculation can be very slow when the numbers are large. This is why MML$_w$ and BMA$_w$ are ranked the worst in Figure 9 and require large amount of training time as in Figure 10.

Hence, although MML$_w$ is as effective as AODE at reducing classification variance and is ranked fifth at reducing classification error, it can be infeasible for modern real-world applications where large data sets are commonly involved.

5) Capability for fast classification: Figure 11 and Figure 12 illustrate alternative schemes' training time. Consistent with our time complexity analyses in Section V-B, model selection schemes identify a subset of SPODEs to carry out classification, and hence are faster than AODE that uses all SPODEs. Model weighing schemes uses all SPODEs and multiply each with calculated weights, and hence are slower than AODE. Among all schemes, FSA$_s$ delivers the fastest classification and CV$_s$ the second.

One very interesting issue to spot is that LE$_s$, a lazy method that conducts calculation in classification time, turns out to be faster than AODE. We suggest the reason is that on one hand, LE$_s$ requires a simple test each time when a pair of attribute values is considered to determine whether one is a generalization of the other, which causes a small increase in the compute time. On the other hand, once a generalization

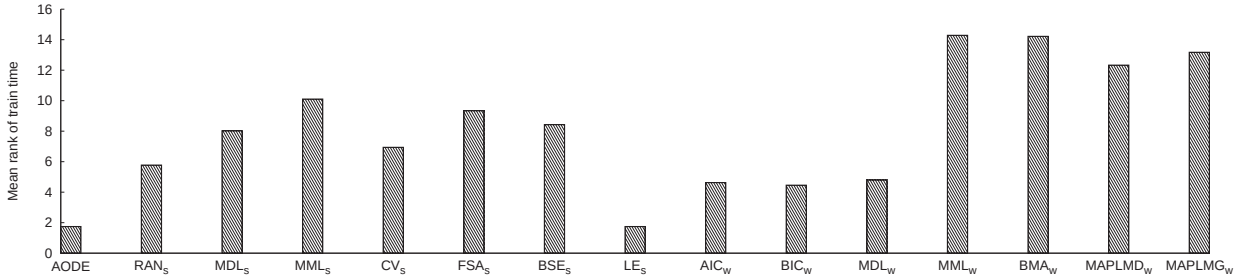


Fig. 9. Compare alternative schemes' mean ranks of training time.

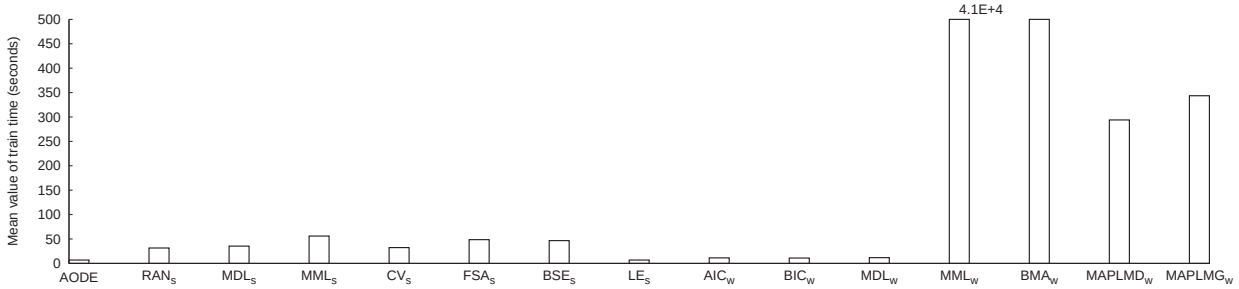


Fig. 10. Compare alternative schemes' mean values of training time. MML$_w$ and BMA$_w$ suffer from the overflowing problem in practice. To keep a readable scale, the bars of MML$_w$ and BMA$_w$ are cut short with their true values labeled on top.

relationship is detected, LE$_s$ need not calculate or multiply related conditional probabilities, which decreases the compute time. The time saved in the latter often exceeds the time cost in the former. Hence, LE$_s$, although 'lazy', can still deliver faster classification than AODE.

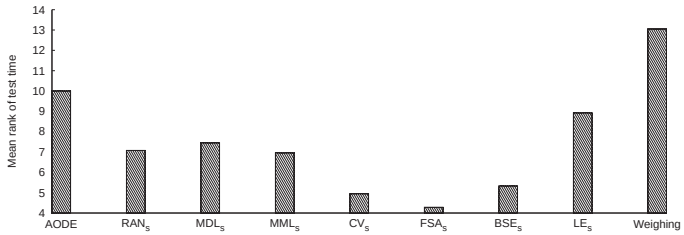


Fig. 11. Compare alternative schemes' mean ranks of test time.

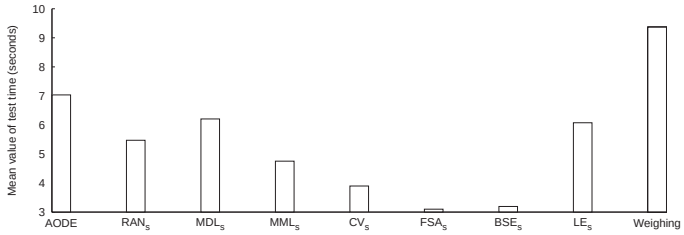


Fig. 12. Compare alternative schemes' mean values of test time.

6) Best schemes' relative performance: LE$_s$ and MAPLMG$_w$ are respectively the best model selection and model weighing schemes for reducing SPODE ensembles' classification error. Figure 13 graphs the relative bias, variance and error between LE$_s$, MAPLMG$_w$ and AODE. The values on the y-axis are the outcome for LE$_s$ divided by that for AODE. The values of the x-axis are the outcome for MAPLMG$_w$ divided by that for AODE. Each point on the graph represents one of the 58 data sets. Points on the left of the vertical line at MAPLMG$_w$/AODE=1 in each subgraph are those of which MAPLMG$_w$ outperforms AODE. Points below the horizontal line at LE$_s$/AODE=1 indicate that LE$_s$ outperforms AODE. Points below the diagonal line Y=X represent that MAPLMG$_w$ outperforms LE$_s$. It is observed that on one hand, both LE$_s$ and MAPLMG$_w$ frequently reduce bias compared with AODE as the majority of points fall within the boundaries X=1 and Y=1 in Figure 13 (a). On the other hand, AODE is better at reducing variance as the majority of points fall beyond the boundaries X=1 and Y=1 in Figure 13 (b). The end effect is that both LE$_s$ and MAPLMG$_w$ outperform AODE on reducing error (w/l/t being 27/8/23 and 33/6/19 respectively as in Table II).

Between LE$_s$ and MAPLMG$_w$ themselves, it is observed that MAPLMG$_w$ slightly outperforms LE$_s$ on both bias and variance reduction (w/l/t being 25/23/10 and 22/20/16 respectively as in Tables III and IV). The end effect is that MAPLMG$_w$ more frequently attains lower error than LE$_s$ (w/l/t being 28/19/11 as in Table II). On the other hand, in terms of training efficiency and classification efficiency, LE$_s$ is much faster than MAPLMG$_w$ as detailed in Sections VI-E.4 and VI-E.5.

VII. Conclusion and Future Work

This paper presents a comprehensive study, both theoretically and empirically, of 16 representative model selection and model weighing schemes for linearly ensembling superparent-one-dependence estimators, a popular family of semi-naive Bayesian classifiers.

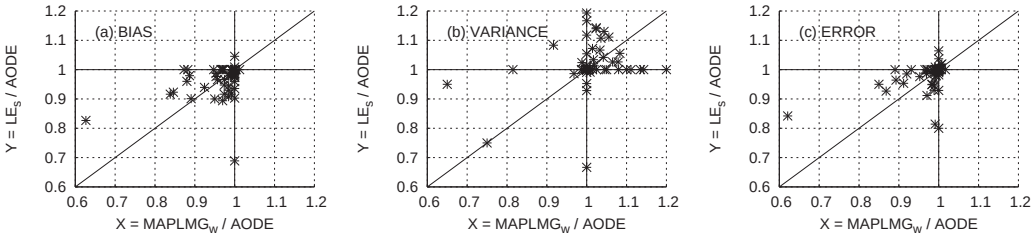


Fig. 13. Illustrate LE_s and MAPLMG_w's performance relative to AODE.

For each scheme, this paper provides its definition, rationale and time complexity. Comprehensive experiments across 58 UCI benchmark data sets are conducted to test each scheme's effect on ensemble learning's accuracy and efficiency.

The study results suggest the following answers to the questions we have asked at the beginning of the paper:

1) MAPLMG_w is ranked the best among all rival schemes on classification accuracy. It wins more often than not when compared with every other single scheme across the suite of 58 data sets. It is significantly better than RAN_s, MDL_s, MML_s, CV_s, FSA_s, BSE_s, AIC_w, BIC_w, MDL_w and BMA_w. However, its training takes longer than the majority of the schemes.
2) LE_s is ranked the best among model selection schemes and the second best among all rival schemes on classification accuracy. It wins more often than not when compared with every other single scheme except MAPLMG_w across the suite of 58 data sets. It is significantly better than RAN_s, MDL_s, MML_s, CV_s, BSE_s, AIC_w, BIC_w, MDL_w and BMA_w. Besides, it is the most efficient at training as well as very efficient at classification.
3) AODE is ranked the best among all rival methods on reducing classification variance. It wins more often than not compared with every other single scheme across the suite of 58 data sets. It is significantly better than CV_s, FSA_s, BSE_s, AIC_w, BIC_w, MDL_w and BMA_w. It is the most efficient at training. It is faster than weighing schemes and slower than selection schemes at classification.
4) Commonly used selection schemes such as CV_s, FSA_s and BSE_s turn out to be less effective than simply including every candidate classifier (AODE). The reason is that they incur high classification variance. Although they are ranked among the best on reducing classification bias, their wins in bias reduction are overshadowed by their losses in variance reduction. The end effect is that they are less effective at reducing error on the learning tasks investigated.
5) The observation that MAPLMD_w is less effective than MAPLMG_w suggests that combining joint (generic) probabilities $P_i(y, \mathbf{x})$ leads to more accurate classification than combining conditional (discriminative) probabilities $P_i(y \mid \mathbf{x})$. In practice, $P_i(y, \mathbf{x})$ is estimated from $count(y, \mathbf{x})$, the count of training instances $< \mathbf{x}, y >$; whereas $P_i(y \mid \mathbf{x})$ is estimated from the count of training instances $\mathbf{x}$ in addition to $count(y, \mathbf{x})$. Hence it is suggested that estimating $P_i(y \mid \mathbf{x})$ is less reliable than estimating $P_i(y, \mathbf{x})$ and is not preferred.
6) In general, information-theoretic metrics (either as selection or as weighing schemes) are not effective at reducing an ensemble's classification error. Although MML_w as a weighing scheme is ranked fairly well (5th), its high time requirement for calculating weights hinders its deployment in practice. A further thought is that currently information-theoretic metrics measure the merit of individual classifiers. It might help to generalize them so as to measure the collective merit of an ensemble. This can be an interesting research issue to further explore.
7) Hence, whether to use model selection or model weighing depends on the specific requirements of a particular classification task. If one needs to maximize accuracy, we recommend MAPLMG_w. If one seeks both high learning accuracy and efficiency, we recommend LE_s. If one needs to minimize variance while obtaining a reasonable accuracy, we recommend AODE.

The model selection and weighing schemes studied here can be generalized to other Bayesian network classifiers. RAN, CV, FSA and BSE only utilize a classifier's classifications. Hence they can be applied to any classifier. LD seeks generalization relationships among attribute values. If such a relationship exists, it deletes the more general value from the network structure. The calculations for AIC and BIC, MDL, MML, and BMA with MML have already been extended to arbitrary Bayesian network structures respectively [70] [71] [51] [72]. MAPLMG and MAPLMD were introduced as linear mixture inducers for either generative or discriminative probabilistic classifiers [31] and are then particularized to SPODE in this study. Hence, they can be directly applied to any Bayesian network classifier. An interesting direction for future research is to examine the extent to which our results generalize to other Bayesian network classifiers.

ACKNOWLEDGMENTS

We gratefully acknowledge Dr. Janez Demsar for his kind help on our statistical tests; and Dr. Gavin Brown for his kind help on the collection of ensemble learning publications. This research was supported by Australian Research Council grant DP0556279.

REFERENCES

[1] G. Brown, "The ensemble learning bibliography," 2006, http://www.cs.man.ac.uk/ gbrown/ensemblebib/.

[2] A. C. Achilles, "The collection of computer science bibliographies," 2006, http://liinwww.ira.uka.de/bibliography/Neural/EnsembleLearning.html.

[3] R. Caruana, A. Niculescu, G. Crew, and A. Ksikes, "Ensemble selection from libraries of models," in *Proceedings of the 21st International Conference on Machine Learning*, 2004.

[4] L. Kuncheva, "Switching between selection and fusion in combining classifiers: An experiment," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 32, no. 2, pp. 146–156, 2002.

[5] R. E. Banfield, L. O. Hall, K. W. Bowyer, and W. P. Kegelmeyer, "A comparison of decision tree ensemble creation techniques," *IEEE Transactions on Pattern Analysis and Machine Learning*, vol. 29, no. 1, pp. 173–180, 2007.

[6] E. J. Keogh and M. J. Pazzani, "Learning augmented Bayesian classifers: A comparison of distribution-based and classification-based approaches," in *Proceedings of the International Workshop on Artificial Intelligence and Statistics*, 1999, pp. 225–230.

[7] ——, "Learning the structure of augmented Bayesian classifiers," *International Journal on Artificial Intelligence Tools*, vol. 11, no. 40, pp. 587–601, 2002.

[8] B. Cestnik, I. Kononenko, and I. Bratko, "Assistant 86: A knowledge-elicitation tool for sophisticated users," in *Proceedings of the 2nd European Working Session on Learning*, 1987, pp. 31–45.

[9] P. Clark and T. Niblett, "The CN2 induction algorithm," *Machine Learning*, vol. 3, pp. 261–283, 1989.

[10] B. Cestnik, "Estimating probabilities: A crucial task in machine learning," in *Proceedings of the 9th European Conference on Artificial Intelligence*, 1990, pp. 147–149.

[11] I. Kononenko, "Comparison of inductive and naive Bayesian learning approaches to automatic knowledge acquisition," 1990.

[12] P. Langley, W. Iba, and K. Thompson, "An analysis of Bayesian classifiers," in *Proceedings of the 10th National Conference on Artificial Intelligence*, 1992, pp. 223–228.

[13] P. Domingos and M. J. Pazzani, "Beyond independence: Conditions for the optimality of the simple Bayesian classifier," in *Proceedings of the 13th International Conference on Machine Learning*. Morgan Kaufmann Publishers, 1996, pp. 105–112.

[14] ——, "On the optimality of the simple Bayesian classifier under zero-one loss," *Machine Learning*, vol. 29, pp. 103–130, 1997.

[15] H. Zhang, C. X. Ling, and Z. Zhao, "The learnability of naive Bayes," in *Proceedings of Canadian Artificial Intelligence Conference*, 2000, pp. 432–441.

[16] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Machine Learning*, vol. 29, no. 2, pp. 131–163, 1997.

[17] J. Kittler, "Feature selection and extraction," in *Handbook of Pattern Recognition and Image Processing*, T. Y. Young and K. S. Fu, Eds., New York, 1986.

[18] R. Kohavi, "Scaling up the accuracy of naive-Bayes classifiers: a decision-tree hybrid," in *Proceedings of the 2nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 202–207.

[19] I. Kononenko, "Semi-naive Bayesian classifier," in *Proceedings of the 6th European Working Session on Machine learning*, 1991, pp. 206–219.

[20] P. Langley, "Induction of recursive Bayesian classifiers," in *Proceedings of the 4th European Conference on Machine Learning*, 1993, pp. 153–164.

[21] P. Langley and S. Sage, "Induction of selective Bayesian classifiers," in *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence*, 1994, pp. 399–406.

[22] M. J. Pazzani, "Constructive induction of Cartesian product attributes," *ISIS: Information, Statistics and Induction in Science*, pp. 66–77, 1996.

[23] M. Sahami, "Learning limited dependence Bayesian classifiers," in *Proceedings of the 2nd International Conference on Knowledge Discovery in Databases*, 1996, pp. 334–338.

[24] M. Singh and G. M. Provan, "Efficient learning of selective Bayesian network classifiers," in *Proceedings of the 13th International Conference on Machine Learning*, 1996, pp. 453–461.

[25] Z. Xie, W. Hsu, Z. Liu, and M. L. Lee, "Snnb: A selective neighborhood based naive Bayes for lazy learning," in *Advances in Knowledge Discovery and Data Mining, Proceedings of the Pacific-Asia Conference (PAKDD 2002)*, 2002, pp. 104–114.

[26] Z. Zheng and G. I. Webb, "Lazy learning of Bayesian rules," *Machine Learning*, vol. 41, no. 1, pp. 53–84, 2000.

[27] Z. Zheng, G. I. Webb, and K. M. Ting, "Lazy Bayesian rules: A lazy semi-naive Bayesian learning technique competitive to boosting decision trees," in *Proceedings of the 16th International Conference on Machine Learning (ICML 1999)*, 1999, pp. 493–502.

[28] G. I. Webb, "Candidate elimination criteria for lazy Bayesian rules," in *Proceedings of the 14th Australian Joint Conference on Artificial Intelligence*, 2001, pp. 545–556.

[29] G. I. Webb and M. J. Pazzani, "Adjusted probability naive Bayesian induction," in *Proceedings of the 11th Australian Joint Conference on Artificial Intelligence*, 1998, pp. 285–295.

[30] G. I. Webb, J. Boughton, and Z. Wang, "Not so naive Bayes: Aggregating one-dependence estimators," *Machine Learning*, vol. 58, no. 1, pp. 5–24, January 2005.

[31] J. Cerquides and R. L. de Mántaras, "Robust bayesian linear classifier ensembles." in *Proceedings of the 16th European Conference on Machine Learning (ECML)*, 2005, pp. 72–83.

[32] L. De Ferrari, "Mining housekeeping genes with a naive Bayes classifier," MSc Thesis, University of Edinburgh, School of Informatics, 2005.

[33] K. Flikka, L. Martens, J. Vandekerckhove, K. Gevaert, and I. Eidhammeri, "Improving throughput and reliability of peptide identifications through spectrum quality evaluation," in *Proceedings of the 9th Annual International Conference on Research in Computational Molecular Biology*, 2005.

[34] A. P. Nikora, "Classifying requirements: Towards a more rigorous analysis of natural-language specifications," in *Proceedings of the 16th IEEE International Symposium on Software Reliability Engineering*, 2005, pp. 291–300.

[35] Y. Yang, K. Korb, K. M. Ting, and G. I. Webb, "Ensemble selection for superparent-one-dependence estimators," in *Proceedings of the 18th Australian Joint Conference on Artificial Intelligence*, 2005.

[36] F. Zheng and G. I. Webb, "A comparative study of semi-naive bayes methods in classification learning," in *Proceedings of the 4th Australasian Data Mining conference*, 2005, pp. 141–156.

[37] S. B. Kotsiantis and P. E. Pintelas, "Logitboost of simple bayesian classifier," *Informatica*, vol. 29, pp. 53–59, 2005.

[38] H. Zhang, L. Jiang, and J. Su, "Hidden naive bayes," in *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI)*, 2005, pp. 919–924.

[39] F. Birzele and S. Kramer, "A new representation for protein secondary structure prediction based on frequent patterns," *Bioinformatics*, vol. 22, no. 21, pp. 2628–2634, 2006.

[40] J. Su and H. Zhang, "Representing conditional independence using decision trees," in *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI)*, 2005, pp. 874–879.

[41] H. Zhang, L. Jiang, and J. Su, "Augmenting naive bayes for ranking," in *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, 2005, pp. 1020–1027.

[42] J. Abellan, "Application of uncertainty measures on credal sets on the naive bayesian classifier," *International Journal of General Systems*, vol. 35, no. 6, pp. 675–686, 2006.

[43] H. YIN, G. LI, T. Y. LEONG, V. KURALMANI, B. PANG, B. T. ANG, K. K. LEE, and I. NG, "Experimental analysis on severe head injury outcome prediction - a preliminary study," technical Report TRD9/06, School of Computing, The National University of Singapore.

[44] F. Zheng and G. I. Webb, "Efficient lazy elimination for averaged one-dependence estimators," in *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, 2006, pp. 1113–1120.

[45] D. Zeng, S. Zhang, Z. Cai, S. Jiang, and L. Jiang, "A novel one-dependence estimator based on multi-parents," in *Proceedings of the 6th International Conference on Intelligent Systems Design and Applications (ISDA)*, 2006, pp. 639–643.

[46] M. Ceci, "Naive bayesian learning from structural data," Ph.D. dissertation, Dipartimento di Informatica, University of Bari, Italy, 2005.

[47] Y. Yang, G. Webb, J. Cerquides, K. Korb, J. Boughton, and K. M. Ting, "To select or to weigh: A comparative study of model selection and model weighing for spode ensembles." in *Proceedings of the 17th European Conference on Machine Learning*, 2006, pp. 533–544.

[48] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. AC-19, pp. 716–23, 1974.

[49] G. Schwarz, "Estimating the dimension of a model," *Annals of Statistics*, vol. 6, pp. 461–5, 1978.

[50] J. Suzuki, "Learning Bayesian belief networks based on the MDL principle: an efficient algorithm using the branch and bound technique," in *Proceedings of the International Conference on Machine Learning*, 1996, pp. 463–470.

[51] K. Korb and A. Nicholson, *Bayesian Artificial Intelligence*. Boca Raton, FL: Chapman & Hall/CRC, 2004.

[52] G. F. Cooper and E. Herskovits, "A bayesian method for the induction of probabilistic networks from data," *Machine Learning*, vol. 9, pp. 309–347, 1992.

[53] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[54] J. A. Hoeting, D. Madigan, A. E. Raftery, and C. T. Volinsky, "Bayesian model averaging: A tutorial," *Statistical Science*, vol. 14, no. 4, p. 382417, 1999.

[55] P. Domingos, "Bayesian averaging of classifiers and the overfitting problem," in *Proceedings of the 17th International Conference on Machine Learning*, 2000, pp. 223–230.

[56] P. Pedregal, *Introduction to Optimization*, ser. Texts in Applied Mathematics. Springer, 2004, no. 46.

[57] M. T. Heath, *Scientific Computing: An Introductory Survey, 2nd Edition*. New York: McGraw-Hill, 2002.

[58] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. Wiley, 1997.

[59] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Second Edition*. San Francisco, CA: Morgan Kaufmann, 2005.

[60] C. L. Blake and C. J. Merz, "UCI repository of machine learning databases [http://www.ics.uci.edu/∼mlearn/mlrepository.html]," 1998, department of Information and Computer Science, University of California, Irvine.

[61] G. I. Webb, "Multiboosting: A technique for combining boosting and wagging," *Machine Learning*, vol. 40, no. 2, pp. 159–196, 2000.

[62] L. Breiman, "Bias, variance and arcing classifiers, technical report 460, Statistics Department, University of California, Berkeley," 1996.

[63] J. H. Friedman, "On bias, variance, 0/1-loss, and the curse-of-dimensionality," *Data Mining and Knowledge Discovery*, vol. 1, no. 1, pp. 55–77, 1997.

[64] R. Kohavi and D. Wolpert, "Bias plus variance decomposition for zero-one loss functions," in *Proceedings of the 13th International Conference on Machine Learning*, 1996, pp. 275–283.

[65] E. B. Kong and T. G. Dietterich, "Error-correcting output coding corrects bias and variance," in *Proceedings of the 12th International Conference on Machine Learning*, 1995, pp. 313–321.

[66] D. S. Moore and G. P. McCabe, *Introduction to the Practice of Statistics, Fourth Edition*. Michelle Julet, 2002.

[67] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the American Statistical Association*, vol. 32, pp. 675–701, 1937.

[68] ——, "A comparison of alternative tests of significance for the problem of m rankings," *Annals of Mathematical Statistics*, vol. 11, pp. 86–92, 1940.

[69] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[70] T. Silander and P. Myllymaki, "A simple approach for finding the globally optimal bayesian network structure," in *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence*.

[71] N. Friedman and M. Goldszmidt.

[72] R. O'Donnell, "Learning flexible causal models with MML," Dissertation, Monash University, 2007.

APPENDIX

Error w/l/t	AODE	RAN_s	MDL_s	MML_s	CV_s	FSA_s	BSE_s	LE_s	AIC_w	BIC_w	MDL_w	MML_w	BMA_w	$MAPLMD_w$
RAN_s	11/29/18													
MDL_s	10/28/20	16/16/26												
MML_s	14/25/19	23/16/19	16/8/34											
CV_s	18/27/13	18/19/21	16/14/28	13/17/28										
FSA_s	20/29/9	24/15/19	25/17/16	21/22/15	23/17/18									
BSE_s	17/30/11	24/17/17	22/16/20	19/20/19	21/20/17	17/16/25								
LE_s	27/8/23	40/5/13	37/6/15	36/6/16	36/11/11	40/13/5	38/13/7							
AIC_w	6/48/4	14/43/1	11/45/2	12/44/2	12/43/3	12/43/3	12/43/3	6/49/3						
BIC_w	6/47/5	13/43/2	10/43/5	12/43/3	12/43/3	13/42/3	12/42/4	6/48/4	14/12/32					
MDL_w	6/48/4	13/43/2	8/45/5	11/44/3	11/44/3	12/44/2	11/43/4	6/49/3	9/10/39	10/12/36				
MML_w	8/11/39	30/10/18	27/10/21	23/13/22	25/18/15	28/21/9	29/20/9	7/29/22	49/7/2	46/7/5	48/7/3			
BMA_w	7/49/2	9/46/3	10/46/2	9/47/2	10/45/3	9/47/2	11/45/2	4/52/2	21/26/11	24/25/9	22/25/11	7/49/2		
$MAPLMD_w$	25/15/18	39/6/13	39/6/13	38/8/12	34/13/11	31/15/12	35/14/9	21/22/15	48/7/3	48/6/4	50/5/3	27/13/18	52/5/1	
$MAPLMG_w$	33/6/19	47/2/9	45/5/8	44/4/10	43/7/8	39/7/12	44/7/7	28/19/11	50/5/3	49/6/3	49/5/4	36/6/16	50/5/3	26/10/22

TABLE II
WIN/LOSE/TIE RECORDS OF EACH PAIR OF COMPETING SCHEMES ON REDUCING CLASSIFICATION ERROR.

Bias w/l/t	AODE	RAN_s	MDL_s	MML_s	CV_s	FSA_s	BSE_s	LE_s	AIC_w	BIC_w	MDL_w	MML_w	BMA_w	$MAPLMD_w$
RAN_s	23/18/17													
MDL_s	20/20/18	20/18/20												
MML_s	28/17/13	27/13/18	17/6/35											
CV_s	40/12/6	40/6/12	38/2/18	36/8/14										
FSA_s	43/11/4	47/2/9	41/3/14	40/5/13	20/14/24									
BSE_s	43/13/2	45/5/8	38/4/16	32/9/17	14/19/25	5/23/30								
LE_s	33/2/23	35/12/11	34/12/12	30/15/13	24/29/5	22/31/5	27/29/2							
AIC_w	16/33/9	19/33/6	17/35/6	17/36/5	14/40/4	14/42/2	16/40/2	14/39/5						
BIC_w	11/37/10	16/36/6	14/36/8	15/37/6	12/43/3	13/43/2	15/40/3	11/40/7	7/18/33					
MDL_w	15/33/10	19/33/6	16/35/7	15/37/6	13/41/4	13/43/2	14/41/3	15/38/5	7/10/41	13/7/38				
MML_w	14/5/39	20/19/19	23/20/15	18/28/12	12/39/7	11/43/4	12/39/7	4/33/21	35/16/7	38/11/9	34/15/9			
BMA_w	19/31/8	25/27/6	25/27/6	25/30/3	18/36/4	20/34/4	21/33/4	14/39/5	27/21/10	31/18/9	27/20/11	17/32/9		
$MAPLMD_w$	30/7/21	37/13/8	34/12/12	29/18/11	20/33/5	15/36/7	19/32/7	21/24/13	38/12/8	41/11/6	38/12/8	31/11/16	33/18/7	
$MAPLMG_w$	36/2/20	44/6/8	41/10/7	34/16/8	22/26/10	17/31/10	22/24/12	25/23/10	41/10/7	44/10/4	43/10/5	37/7/14	37/15/6	22/7/29

TABLE III

WIN/LOSE/TIE RECORDS OF EACH PAIR OF COMPETING SCHEMES ON REDUCING CLASSIFICATION BIAS.

Variance w/l/t	AODE	RAN_s	MDL_s	MML_s	CV_s	FSA_s	BSE_s	LE_s	AIC_w	BIC_w	MDL_w	MML_w	BMA_w	$MAPLMD_w$
RAN_s	12/35/11													
MDL_s	10/28/20	16/19/23												
MML_s	9/32/17	11/24/23	7/18/33											
CV_s	9/44/5	5/37/16	2/33/23	3/31/24										
FSA_s	9/43/6	3/38/17	6/35/17	9/29/20	21/11/26									
BSE_s	8/44/6	3/35/20	8/32/18	13/28/17	27/16/15	21/14/23								
LE_s	8/21/29	31/17/10	26/16/16	31/16/11	43/9/6	42/11/5	39/14/5							
AIC_w	10/40/8	17/34/7	16/33/9	18/32/8	24/27/7	24/27/7	24/28/6	16/35/7						
BIC_w	14/34/10	18/32/8	16/29/13	18/30/10	24/25/9	26/24/8	24/24/10	18/33/7	14/9/35					
MDL_w	11/39/8	17/33/8	16/32/10	17/33/8	26/26/6	25/26/7	23/27/8	14/37/7	8/9/41	6/13/39				
MML_w	8/11/39	36/11/11	29/10/19	34/9/15	44/8/6	42/9/7	45/8/5	21/12/25	43/10/5	37/14/7	41/11/6			
BMA_w	5/48/5	8/45/5	7/47/4	8/47/3	15/40/3	13/41/4	12/40/6	6/47/5	9/35/14	11/37/10	13/36/9	5/47/6		
$MAPLMD_w$	9/27/22	27/15/16	28/17/13	29/15/14	41/8/9	44/9/5	42/8/8	20/23/15	38/11/9	33/16/9	37/13/8	10/28/20	47/6/5	
$MAPLMG_w$	9/25/24	27/12/19	25/15/18	31/13/14	43/7/8	45/7/6	43/7/8	22/20/16	38/12/8	34/15/9	38/12/8	10/25/23	47/6/5	14/15/29

TABLE IV

WIN/LOSE/TIE RECORDS OF EACH PAIR OF COMPETING SCHEMES ON REDUCING CLASSIFICATION VARIANCE.