# To Select or To Weigh: A Comparative Study of Model Selection and Model Weighing for SPODE Ensembles

Ying Yang †, Geoff Webb †, Jesús Cerquides ‡, Kevin Korb †, Janice Boughton †, and Kai Ming Ting †

† Clayton School 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.** An ensemble of Super-Parent-One-Dependence Estimators (SPODEs) offers a powerful yet simple alternative to naive Bayes classifiers, achieving significantly higher classification accuracy at a moderate cost in classification efficiency. Currently there exist two families of methodologies that ensemble candidate SPODEs for classification. One is to select only helpful SPODEs and uniformly average their probability estimates, a type of *model selection*. Another is to assign a weight to each SPODE and linearly combine their probability estimates, a methodology named *model weighing*. This paper presents a theoretical and empirical study comparing model selection and model weighing for ensembling SPODEs. The focus is on maximizing the ensemble's classification accuracy while minimizing its computational time. A number of representative selection and weighing schemes are studied, providing a comprehensive research on this topic and identifying effective schemes that provide alternative trades-off between speed and expected error.

## 1 Introduction

Semi-naive Bayesian classifiers reduce error by relaxing the attribute independence assumption of naive Bayes [1–17]. Among alternative semi-naive forms, Super-Parent-One-Dependence Estimators (SPODEs) [2, 3], and particularly ensembles thereof [13] have received a lot of attention [18–22] because they offer a combination of high training efficiency, high classification efficiency and high classification accuracy. 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. This paper identifies approaches that can maximize a SPODE ensemble's classification accuracy while minimizing its computational time. This leads to accurate and fast classification algorithms with immediate and significant impact on real-world applications.

### 1.1 Terminology and Notation

This paper addresses the problem of classification learning using an ensemble of Bayesian probabilistic classifiers. The following terminology and notation will be used throughout the paper. 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 $X_i$ ($i \in [1, m]$). It 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*. *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 $\phi_{ir}$. When applicable, $h$ indicates a SPODE in general and $h_i$ indicates a particular SPODE whose superparent is $X_i$.

### 1.2 SPODE and SPODE Ensemble

A SPODE [2, 3] relaxes the naive Bayes (NB) classifier's attribute independence assumption by allowing all attributes to depend on a common attribute, the *superparent*, in addition to the class, as depicted in Figure 1.

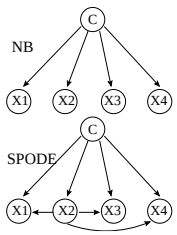



**Fig. 1.** Illustration of SPODE versus NB. An arc points from a parent to a child. A child only depends on its parents. NB assumes each attribute only depends on the class $Y$ and is independent of other attributes given the class. SPODE assumes that each attribute can depend on both the superparent $X_2$ and the class.

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

A SPODE with superparent $X_p$ finds $\text{argmax}_y\, \hat{P}(y, \mathbf{x})$ using $\hat{P}(y, \mathbf{x}) = \hat{P}(y, x_p)\hat{P}(\mathbf{x} \mid y, x_p) = \hat{P}(y, x_p)\prod_{i=1}^{m} \hat{P}(x_i \mid y, x_p)$. The final formula results from SPODEs' assumption that all attributes are independent of each other given $Y$ and $X_p$.

A SPODE ensemble is a linear combination of multiple SPODEs' probability estimates. It seeks $\text{argmax}_y\, \hat{P}(y, \mathbf{x})$ using: $\hat{P}(y, \mathbf{x}) \approx \sum_{i=1}^{m} w_i \hat{P}_i(y, \mathbf{x})$, where each $\hat{P}_i(y, \mathbf{x})$ is calculated by a SPODE whose superparent being $X_i$. For a training data set with $m$ attributes, there can be $m$ candidate SPODEs, each taking a different attribute as its superparent.

It has been shown that a SPODE, being a one-dependence estimator, can provide better probability estimates than NB because it involves a weaker attribute independence assumption [1–3, 10, 13]. It has also been shown that a SPODE ensemble can further improve upon the classification accuracy of a single SPODE by decreasing the classification variance [13, 18]. The first approach to ensembling SPODEs was AODE [13] which used equal weight combination of all SPODEs whose parent occurred with a user-specified minimum frequency in the training data. Subsequent research suggested that frequency is not a useful model selection criterion and that appropriate weighting can substantially improve upon equal weighting, proposing weighting schemes such as MAPLMG [18]. On the other hand, it has also been shown that model selection can be very effective when ensembling SPODEs [22]. This paper presents a comprehensive investigation into the relative merits of alternative approaches to weighting and selecting.

## 2 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. Previous research has suggested that cross validation, forward sequential addition and lazy elimination are more effective than alternative selection methods for SPODEs [22, 23]

**Cross Validation (CV)** [22] scores each individual SPODE by its cross validation error on the training data. 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 a SPODE should 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. Given a sequence of $m$ SPODEs ordered by their CV values, $m$ ensembles are candidates, from size 1 to size $m$. Starting with an empty ensemble, each ensemble in turn includes further one SPODE in the queue. Every ensemble's leave-one-out cross validation error is calculated. The ensemble with the lowest error is the one to be selected.[1]

**Forward Sequential Addition (FSA)** [22] begins with an empty ensemble. It then uses hill-climbing search to iteratively add SPODEs whose individual inclusion results in the lowest classification error. In each iteration, suppose the current ensemble is $E_{current}$ with $k$ SPODEs. FSA in turn adds each candidate

[1] If there are multiple ensembles that attain the lowest error, the one with the largest ensemble size is selected as a means to reduce classification variance caused by model selection. The same rule also applies to FSA.

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 $E_{test}$ with the lowest error is retained and the corresponding added SPODE is permanently deleted from the candidate list and included into the ensemble. 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 which achieves the lowest error in the adding process is the one to be selected.

**Lazy Elimination (LE)** CV and FSA 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 [23] 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 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 for every training instance containing $x_j$, $x_i$ is also present. 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.

## 3 Model Weighing Schemes

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

**Information-Theoretic Metrics** provide a combined score for a proposed explanatory model (a SPODE in our context) and for the data given the model. Since they rely upon Shannon information theory [24] for their motivation and interpretation, they should support the inversion of Shannon's law to derive the posterior probability of a model given the data as to be the model's probabilistic weight for purpose of prediction. In principle, the weight $w$ for a SPODE $h$ is[2]:

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

where $H(X_i, \Phi(i))$ is the mutual information between $X_i$ and its parents: $H(X_i, \Phi(i)) = \sum_{j=1}^{v_i}\sum_{r=1}^{|\phi_i|}\left(P(x_{ij}, \phi_{ir}) \log \frac{P(x_{ij},\phi_{ir})}{P(x_{ij})P(\phi_{ir})}\right)$; and $I(h)$ ***varies*** among different schemes, of which two representative ones are presented below.

[2] For simplicity, $X_i$ represents the class variable when $i = m + 1$. Generally the log base does not matter. A common practice is to use $e$ or 2.

***Bayesian Information Criterion* (BIC)** According to Schwarz [25]: $I_{BIC}(h) = (\log n)\left(\sum_{i=1}^{m+1}(v_i - 1)\prod_{j\in\Phi(i)} v_j\right)$. For any root node $X_i$ (where $\Phi(i) = \emptyset$), the product term on the right should be replaced by 1.

***Minimum Message Length* (MML)** According to Korb and Nicholson [26]: $I_{MML}(h) = \log(m+1)! + C_2^{m+1} - \log(m-1)! + \sum_{i=1}^{m+1}\frac{v_i-1}{2}(\log\frac{\pi}{6}+1) - \log\prod_{i=1}^{m+1}\prod_{j=1}^{|\phi_i|}\left(\frac{(v_i-1)!}{(S_{ij}+v_i-1)!}\prod_{l=1}^{v_i}\alpha_{ijl}!\right)$, where $S_{ij}$ is the number of training instances where the parents $\Phi(i)$ take their joint $j$-th value, and $\alpha_{ijl}$ 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 $\alpha_{ijl}$.

**Bayesian Model Averaging (BMA)** provides a mechanism to ensemble classification models by accounting for single models' uncertainty of generating the data [27]. Given an instance $\mathbf{x}$ and a set of classifiers $h_i$, BMA estimates the probability of each class label given $\mathbf{x}$ using: $\hat{P}(y \mid \mathbf{x}) = \sum_{i=1}^{m}\hat{P}(y \mid h_i)\hat{P}(h_i \mid D)$, where $\hat{P}(y \mid h_i)$ is the class probability estimated by a SPODE. One representative approach to estimating the weight $\hat{P}(h_i \mid D)$, used in BMA, was proposed by Cooper and Herskovits [28]: $w_i = \hat{P}(h_i \mid D) = \frac{\hat{P}(h_i,D)}{\sum_{i=1}^{m}\hat{P}(h_i,D)}$, where $P(h_i, D) = \hat{P}(h_i)\prod_{k=1}^{m+1}\prod_{j=1}^{|\phi_i|}\left(\frac{(v_k-1)!}{(S_{kj}+v_k-1)!}\prod_{l=1}^{v_k}\alpha_{kjl}!\right)$, $\hat{P}(h_i) = \frac{1}{m}$ if there are $m$ candidate SPODEs, and $S_{kj}$ and $\alpha_{kjl}$ have the same meanings as for MML.

**Maximum a Posteriori Linear Mixture of Generative Distributions (MAPLMG)** [18] 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 $\mathbf{w} = \text{argmax}_{\mathbf{w}}\,\hat{P}_{LMG}(\mathbf{w}|D)$ 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 is a constrained nonlinear optimization problem that can be solved by means of a sequence of unconstrained maximizations [29], each of them solved by a Newton-like procedure such as BFGS [30].

## 4 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 $v$.

The ***training time complexity*** of each scheme is listed as follows:

| **CV** | **FSA** | **LE** | **BIC** | **MML or BMA** | **MAPLMG** |
|---|---|---|---|---|---|
| $O(m^2nc)$ | $O(m^3nc)$ | $O(0)$ | $O(m^2v^2c)$ | $O(m^2n(v+\frac{n}{vc}))$ | $O(m^2nc + Kmnc)$ |

Note that LE does not require any additional information to be gathered at training time and hence has no impact on training time. In practice, MML and BMA often lead to arithmetic overflow when calculating very large exponentials or factorials. One solution is to use the java class *BigDecimal* which unfortunately can be very slow. This is why MML and BMA require large amount of training time as later illustrated in Figure 3. The 'K' in MAPLMG's complexity is a large fixed number that bounds the number of iterations in the maximization step. Since K is fixed, it does not affect the theoretical complexity. However it can dominate the computing time when m,n and c are not large enough.

As for ***classification time complexity***, each scheme's dominating complexity is the linear combination of SPODEs: $O(m^2c)$ that results from the $O(mc)$ SPODE algorithm applied over an $O(m)$ sized ensemble.

## 5 Experiments

Empirical tests and observations of each selection or weighing scheme for ensembling SPODEs are presented here.

### 5.1 Design and Results

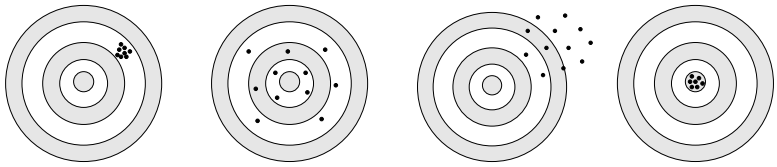
**Table 1.** Statistics of 57 experimental data sets

| Data | Ins. | Att. | Data | Ins. | Att. | Data | Ins. | Att. |
|---|---|---|---|---|---|---|---|---|
| Abalone | 4177 | 8 | Hypothyroid | 3772 | 29 | Postoperative | 90 | 8 |
| AE | 9961 | 12 | Ionosphere | 351 | 34 | PrimaryTumor | 339 | 17 |
| Annealing | 898 | 38 | IrisClassification | 150 | 4 | Promoter | 106 | 57 |
| Audiology | 226 | 69 | KRvsKP | 3196 | 36 | Satellite | 6435 | 36 |
| AutosImports85 | 205 | 25 | LaborNegotiations | 57 | 16 | Segment | 2310 | 19 |
| BalanceScale | 625 | 4 | LED | 1000 | 7 | SickEuthyroid | 3772 | 29 |
| Bands | 1078 | 36 | LetterRecognition | 20000 | 16 | Sign | 12546 | 8 |
| BreastCancer | 699 | 9 | LiverDisorders | 345 | 6 | Sonar | 208 | 60 |
| Chess | 551 | 39 | LungCancer | 32 | 56 | Soybean | 683 | 35 |
| CMC | 1473 | 9 | Lymphography | 296 | 18 | Splice | 3177 | 60 |
| CreditApproval | 690 | 15 | Mfeat-mor | 2000 | 6 | Syncon | 600 | 60 |
| Echocardiogram | 131 | 6 | Mushroom | 8124 | 22 | Thyroid | 9169 | 29 |
| German | 1000 | 20 | Musk | 476 | 166 | TicTacToe | 958 | 9 |
| GlassIdentification | 214 | 9 | NetTalkPhoneme | 5438 | 7 | Vehicle | 846 | 18 |
| HeartCleveland | 303 | 13 | NewThyroid | 215 | 5 | Vowel | 990 | 11 |
| Hepatitis | 155 | 19 | OpticalDigits | 5620 | 48 | Waveform | 5000 | 40 |
| HorseColic | 368 | 21 | PageBlocks | 10946 | 10 | Wine | 178 | 13 |
| HouseVotes84 | 435 | 16 | PenDigits | 10992 | 16 | Yeast | 1484 | 8 |
| Hungarian | 294 | 13 | PimaDiabetes | 768 | 8 | Zoo | 101 | 16 |

A large suite of 57 benchmark data sets from the UCI machine learning repository [31], as described in Table 1, are employed to test rival schemes. All missing

values for nominal and numeric attributes in a data set are replaced with the modes and means from the training data in order to facilitate calculating information metrics. Numeric attributes are discretized using entropy minimization discretization [32]. Each scheme is tested on each data set using a 10-trial 2-fold cross validation, where 5 performance measures are recorded: *training time*, *classification time* and *classification error* that can be decomposed into a *bias* term and a *variance* term [33–37]. We use Kohavi and Wolpert's [35] definitions of bias and variance, and estimate them using Webb's [37] cross-validation method.

It is useful to look into bias and variance of a classifier because they each offer a different perspective of view. 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. Moore and McCabe [38] illustrated bias and variance through shooting arrows at a target, as reproduced in Figure 2. We can think of the perfect classifier 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. High bias means that the arrows land consistently off the bull's-eye in the same direction. High variance means that repeated shots differ widely among themselves and are scattered on the target. A good learning scheme, like a good archer, should have both low bias and low variance.



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

**Fig. 2.** 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. (Moore and McCabe, 2002)

Statistically a win/lose/tie record (w/l/t) is calculated for each pair of competitors $A$ and $B$ with regard to a performance measure $M$. The record represents the number of data sets in which $A$ respectively beats, loses to or ties with $B$ on $M$. A one-tailed binomial sign test can be applied to wins versus losses. If its result is less than the critical level of 0.05, the wins against losses are statistically significant, supporting the claim that the winner has a systematic (instead of by chance) advantage over the loser.

Please be noted that different from our previous research, we no longer impose a frequency threshold on SPODEs. Previously as a means to reduce classifica-

tion variance, a SPODE was considered a candidate for ensembling only if its frequency was above 30 [22]. However, subsequent research demonstrated better results when the minimum frequency was reduced to 1 [18]. Accordingly, the experimental results of CV and FSA can be different from the previous report [22]. AODE, a complete SPODE ensemble without any selection or weighing applied, is also included to offer a baseline in comparing alternative schemes.

### 5.2 Observations and Analysis

Experimental results are summarized in Figures 3, 4 and Table 2. Empirical observations reveal the following knowledge.

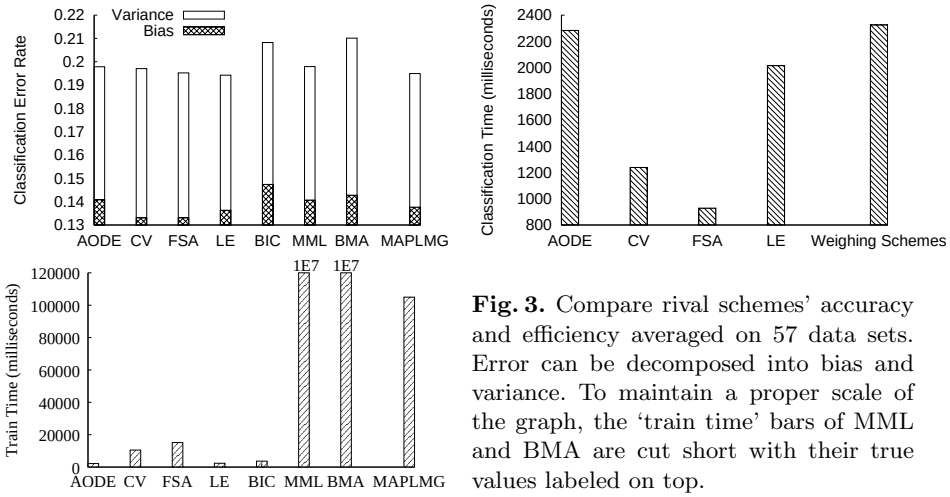



**Fig. 3.** Compare rival schemes' accuracy and efficiency averaged on 57 data sets. Error can be decomposed into bias and variance. To maintain a proper scale of the graph, the 'train time' bars of MML and BMA are cut short with their true values labeled on top.

**LE, best model selection** According to Table 2(a), LE significantly wins against AODE and CV at the 0.05 critical level (w/l/t being 28/6/23 and 34/15/8 respectively). It also wins more often than not when compared with FSA (w/l/t being 31/20/6). As shown in Figure 3, LE achieves the lowest mean error among alternative selection methods. It is also the most efficient method in terms of training time.

**MAPLMG, best model weighing** Among model weighing schemes, the best one is MAPLMG. According to Table 2(a), it significantly wins against AODE and every other single weighing scheme. As shown in Figure 3, MAPLMG achieves the lowest mean error among weighing schemes. One factor that may contribute to its low error is that MAPLMG optimizes multiple weights simultaneously, while others calculate the weights for individual SPODEs in isolation.

**Table 2.** Compare rival schemes' win/lose/tie records with regard to classification error, bias and variance respectively. Each entry indicates that the scheme of the row compares against the scheme of the column. A statistically significant record (at the 0.05 critical level) is indicated in a bold face.

**(a) ERROR**

| w/l/t | AODE | CV | FSA | LE | BIC | MML | BMA |
|---|---|---|---|---|---|---|---|
| CV | 22/26/9 | | | | | | |
| FSA | 27/24/6 | 26/15/16 | | | | | |
| LE | **28/6/23** | **34/15/8** | 31/20/6 | | | | |
| BIC | **10/40/7** | **8/41/8** | **7/41/9** | **6/47/4** | | | |
| MML | 10/10/37 | 21/22/14 | 21/27/9 | **7/31/19** | **39/11/7** | | |
| BMA | **7/46/4** | **6/45/6** | **5/47/5** | **4/50/3** | **16/29/12** | **8/46/3** | |
| MAPLMG | **34/7/16** | **37/12/8** | **33/17/7** | 26/19/12 | **44/9/4** | **35/9/13** | **49/6/2** |

**(b) BIAS**

| w/l/t | AODE | CV | FSA | LE | BIC | MML | BMA |
|---|---|---|---|---|---|---|---|
| CV | **47/4/6** | | | | | | |
| FSA | **48/2/7** | 22/16/19 | | | | | |
| LE | **35/1/21** | **13/35/9** | **11/35/11** | | | | |
| BIC | **13/34/10** | **4/45/8** | **6/45/6** | **10/41/6** | | | |
| MML | 15/11/31 | **4/46/7** | **4/48/5** | **5/35/17** | **35/14/8** | | |
| BMA | 25/25/7 | **6/43/8** | **8/44/5** | **12/38/7** | 28/20/9 | 22/28/7 | |
| MAPLMG | **37/2/18** | **11/38/8** | **5/37/15** | 19/26/12 | **43/11/3** | **38/6/13** | **32/18/7** |

**(c) VARIANCE**

| w/l/t | AODE | CV | FSA | LE | BIC | MML | BMA |
|---|---|---|---|---|---|---|---|
| CV | **3/49/5** | | | | | | |
| FSA | **7/44/6** | **30/12/15** | | | | | |
| LE | **7/21/29** | **44/5/8** | **42/8/7** | | | | |
| BIC | **18/33/6** | 27/22/8 | 25/21/11 | 19/31/7 | | | |
| MML | 8/17/32 | **47/3/7** | **43/7/7** | 21/14/22 | **32/17/8** | | |
| BMA | **6/46/5** | **15/34/8** | **10/39/8** | **7/46/4** | **10/36/11** | **7/45/5** | |
| MAPLMG | **11/22/24** | **48/2/7** | **47/4/6** | 23/16/18 | **32/19/6** | 14/23/20 | **45/7/5** |

On the other hand, this optimization demands time and hence MAPLMG is slower than BIC on training (but still faster than MML and BMA as has been reasoned in Section 4).

**AODE, best variance reduction** AODE does not incur sophisticated selection or weighing. It instead simply uniformly averages every SPODE's probability estimate. This simplicity turns out to be the best scheme in terms of variance reduction, as shown in Figure 3. Also according to Table 2(c), AODE always achieves lower variance than every other single scheme, most of which are statistically significant at the critical level of 0.05. In contrast, schemes like CV and FSA are more capable of reducing bias. However, their bias reduction is overshadowed by AODE's variance reduction. As a result, they cannot significantly outperform AODE on error reduction.

**To select or to weigh** The best of model selection, LE, and the best of model weighing, MAPLMG both beat AODE at the statistical significance level of 0.05 in terms of classification error. Figure 4 graphs the relative bias, variance and error of the three classifiers. The values on the y-axis are the outcome for LE divided by that for AODE. The values of the x-axis are the outcome for MAPLMG divided by that for AODE. Each point on the graph represents one of the 57 data sets. Points on the left of the vertical line at MAPLMG/AODE=1 in each subgraph are those of which MAPLMG outperforms AODE. Points below the horizontal line at LE/AODE=1 indicate that LE outperforms AODE. Points below the diagonal line X=Y represent that MAPLMG outperforms LE. It is observed that both LE and MAPLMG frequently reduce bias compared with AODE as the majority of points fall within the boundaries X=1 and Y=1 in Figure 4(a). Although AODE is better at reducing variance as shown in Figure 4(b), LE and MAPLMG's bias reduction dominates AODE's variance reduction. Hence both can significantly beat AODE on error reduction as in Figure 4(c).

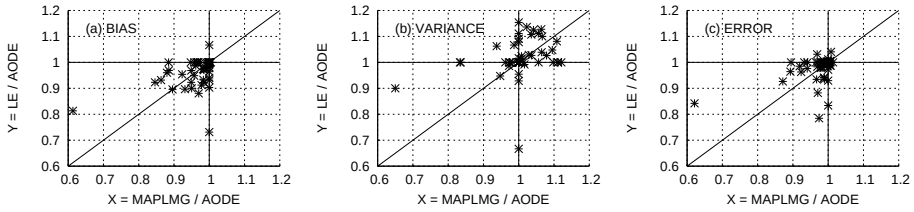



**Fig. 4.** LE and MAPLMG's performance relative to AODE

Between themselves, as shown in Table 2, LE wins against MAPLMG (although not significantly) on reducing bias (w/l/t being 26/19/12). MAPLMG wins against LE (although not significantly) on reducing variance (w/l/t being 23/16/18). The end effect is that MAPLMG beats LE (although not significantly) on reducing error (w/l/t being 26/19/12). Meanwhile, as shown in Figure 3, LE is more efficient than MAPLMG on both training and classification.

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. If one seeks both high learning accuracy and efficiency, we recommend LE. If one needs to minimize variance while obtaining a reasonable accuracy, we recommend AODE.

## 6 Conclusion

We have studied a number of representative model selection and model weighing schemes for SPODE ensemble learning. We have presented the definition, rationale and time complexity of each scheme. We have conducted comprehensive experiments across 57 UCI benchmark data sets to test each scheme's effect on

SPODE ensembles' learning accuracy and efficiency. LE delivers efficient learning and significantly higher accuracy than AODE and thus is identified as the method of choice for model selection. MAPLMG delivers significantly higher accuracy than AODE and thus is identified as the method of choice for model weighing.

**Acknowledgment**

This research was supported by Australian Research Council grant DP0556279.

## References

1. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Machine Learning **29**(2) (1997) 131–163
2. Keogh, E.J., Pazzani, M.J.: 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) 225–230
3. Keogh, E.J., Pazzani, M.J.: Learning the structure of augmented Bayesian classifiers. International Journal on Artificial Intelligence Tools **11**(40) (2002) 587–601
4. Kittler, J.: Feature selection and extraction. In Young, T.Y., Fu, K.S., eds.: Handbook of Pattern Recognition and Image Processing, New York (1986)
5. Kohavi, R.: Scaling up the accuracy of naive-Bayes classifiers: a decision-tree hybrid. In: Proceedings of the 2nd SIGKDD. (1996) 202–207
6. Kononenko, I.: Semi-naive Bayesian classifier. In: Proceedings of the 6th European Working Session on Machine learning. (1991) 206–219
7. Langley, P.: Induction of recursive Bayesian classifiers. In: Proceedings of the 4th ECML. (1993) 153–164
8. Langley, P., Sage, S.: Induction of selective Bayesian classifiers. In: Proceedings of the 10th UAI. (1994) 399–406
9. Pazzani, M.J.: Constructive induction of Cartesian product attributes. ISIS: Information, Statistics and Induction in Science (1996) 66–77
10. Sahami, M.: Learning limited dependence Bayesian classifiers. In: Proceedings of the 2nd SIGKDD. (1996) 334–338
11. Singh, M., Provan, G.M.: Efficient learning of selective Bayesian network classifiers. In: Proceedings of the 13th ICML. (1996) 453–461
12. Webb, G.I.: Candidate elimination criteria for lazy Bayesian rules. In: Proceedings of the 14th Australian AI. (2001) 545–556
13. Webb, G.I., Boughton, J., Wang, Z.: Not so naive Bayes: Aggregating one-dependence estimators. Machine Learning **58**(1) (2005) 5–24
14. Webb, G.I., Pazzani, M.J.: Adjusted probability naive Bayesian induction. In: Proceedings of the 11th Australian AI. (1998) 285–295
15. Xie, Z., Hsu, W., Liu, Z., Lee, M.L.: Snnb: A selective neighborhood based naive Bayes for lazy learning. In: Proceedings of the 6th PAKDD. (2002) 104–114
16. Zheng, Z., Webb, G.I.: Lazy learning of Bayesian rules. Machine Learning **41**(1) (2000) 53–84
17. Zheng, Z., Webb, G.I., Ting, K.M.: Lazy Bayesian rules: A lazy semi-naive Bayesian learning technique competitive to boosting decision trees. In: Proceedings of the 16th ICML. (1999) 493–502

18. Cerquides, J., de Mántaras, R.L.: Robust bayesian linear classifier ensembles. In: Proceedings of the 16th ECML. (2005) 72–83
19. De Ferrari, L.: Mining housekeeping genes with a naive Bayes classifier. MSc Thesis, University of Edinburgh, School of Informatics (2005)
20. Flikka, K., Martens, L., Vandekerckhove, J., Gevaert, K., Eidhammeri, I.: 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)
21. Nikora, A.P.: Classifying requirements: Towards a more rigorous analysis of natural-language specifications. In: Proceedings of the 16th IEEE International Symposium on Software Reliability Engineering. (2005) 291–300
22. Yang, Y., Korb, K., Ting, K.M., Webb, G.I.: Ensemble selection for superparent-one-dependence estimators. In: Proceedings of the 18th Australian AI. (2005) 102–112
23. Zheng, F., Webb, G.I.: Efficient lazy elimination for averaged one-dependence estimators. In: Proceedings of the 23rd ICML. (2006)
24. Shannon, C.E.: A mathematical theory of communication. Bell System Technical Journal **27**(3) (1948) 379–423
25. Schwarz, G.: Estimating the dimension of a model. Annals of Statistics **6** (1978) 461–5
26. Korb, K., Nicholson, A.: Bayesian Artificial Intelligence. Chapman & Hall/CRC, Boca Raton, FL (2004)
27. Hoeting, J.A., Madigan, D., Raftery, A.E., Volinsky, C.T.: Bayesian model averaging: A tutorial. Statistical Science **14**(4) (1999) 382417
28. Cooper, G. F., Herskovits, E.: A Bayesian method for constructing Bayesian belief networks from databases. In: Proceedings of the 7th UAI. (1991) 86–94
29. Pedregal, P.: Introduction to Optimization. Number 46 in Texts in Applied Mathematics. Springer (2004)
30. Heath, M.T.: Scientific Computing: An Introductory Survey, 2nd Edition. McGraw-Hill, New York (2002)
31. Blake, C.L., Merz, C.J.: 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.
32. Fayyad, U.M., Irani, K.B.: Multi-interval discretization of continuous-valued attributes for classification learning. In: Proceedings of the 13th IJCAI. (1993) 1022–1027
33. Breiman, L.: Bias, variance and arcing classifiers, technical report 460, Statistics Department, University of California, Berkeley (1996)
34. Friedman, J.H.: On bias, variance, 0/1-loss, and the curse-of-dimensionality. Data Mining and Knowledge Discovery **1**(1) (1997) 55–77
35. Kohavi, R., Wolpert, D.: Bias plus variance decomposition for zero-one loss functions. In: Proceedings of the 13th ICML. (1996) 275–283
36. Kong, E.B., Dietterich, T.G.: Error-correcting output coding corrects bias and variance. In: Proceedings of the 12th ICML. (1995) 313–321
37. Webb, G.I.: Multiboosting: A technique for combining boosting and wagging. Machine Learning **40**(2) (2000) 159–196
38. Moore, D.S., McCabe, G.P.: Introduction to the Practice of Statistics, Fourth Edition. Michelle Julet (2002)