Pre-publication draft of paper accepted for publication in the Proceedings of AI-2005 which will be published as Springer LNAI

Paper Title

Ensemble Selection for SuperParent-One-Dependence Estimators

Author names

Ying Yang, Kevin Korb, Kai Ming Ting, Geoffrey I. Webb

Author affiliations

School of Computer Science and Software Engineering
Faculty of Information Technology
Monash University, Australia

Addresses

Building 75, Monash University, Clayton Campus, VIC 3800, Australia

Phone numbers

+61 3 99053298 (Yang), 99055198 (Korb), 99026241 (Ting), 99053296 (Webb)

Email address of the contact author

Ying.Yang@Infotech.Monash.edu.au

Abstract

SuperParent-One-Dependence Estimators (SPODEs) loosen Naive-Bayes' attribute independence assumption by allowing each attribute to depend on a common single attribute (superparent) in addition to the class. An ensemble of SPODEs is able to achieve high classification accuracy with modest computational cost. This paper investigates how to select SPODEs for ensembling. Various popular model selection strategies are presented. Their learning efficacy and efficiency are theoretically analyzed and empirically verified. Accordingly, guidelines are investigated for choosing between selection criteria in differing contexts.

Content areas

Bayesian networks, machine learning

Ensemble Selection for SuperParent-One-Dependence Estimators

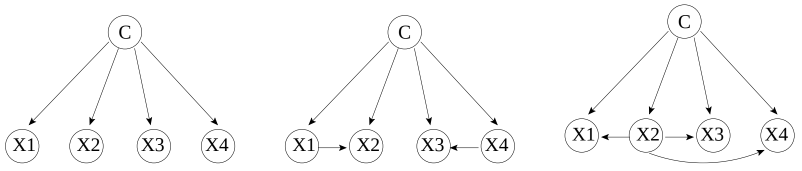
Ying Yang, Kevin Korb, Kai Ming Ting, and Geoffrey I. Webb

School of Computer Science and Software Engineering
Faculty of Information Technology
Monash University, VIC 3800, Australia
{Ying.Yang, Kevin.Korb, Kaiming.Ting, Geoff.Webb}@Infotech.Monash.edu.au

Abstract. SuperParent-One-Dependence Estimators (SPODEs) loosen Naive-Bayes' attribute independence assumption by allowing each attribute to depend on a common single attribute (superparent) in addition to the class. An ensemble of SPODEs is able to achieve high classification accuracy with modest computational cost. This paper investigates how to select SPODEs for ensembling. Various popular model selection strategies are presented. Their learning efficacy and efficiency are theoretically analyzed and empirically verified. Accordingly, guidelines are investigated for choosing between selection criteria in differing contexts.

1 Introduction

One-Dependence Estimators (ODEs) provides a simple, yet powerful, alternative to Naive-Bayes classifiers (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. Both theoretical analysis and empirical evidence have shown that ODEs can improve upon NB's accuracy when its attribute independence assumption is violated (Sahami, 1996; Friedman, Geiger, & Goldszmidt, 1997; Keogh & Pazzani, 1999). A SuperParent-One-Dependence Estimator (SPODE) is an ODE where all attributes depend on the *same* attribute (the *superparent*) in addition to the class (Keogh & Pazzani, 1999). Averaged One-Dependence Estimators (AODE) ensembles all SPODEs that satisfy a minimum support constraint (Webb, Boughton, & Wang, 2005) and estimate class conditional probabilities by averaging across them. This ensemble has demonstrated very high prediction accuracy with modest computational requirements.



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

Fig. 1. Examples of NB, ODE and SPODE. Assuming there are four attributes $X_1 \cdots X_4$ and one class C. An arc points from a parent to a child.

This paper addresses how to select an ensemble of SPODEs so as to minimize classification error. A data sample of m attributes can potentially have m SPODEs, each alternatively taking a different attribute as the superparent.

Finding answers to this SPODE selection problem is of great importance. Its solution will further improve classification accuracy while reducing classification time, albeit at a cost in additional training time.

The following notations will be used throughout the paper. The training data D are composed of n instances. Each instance $\mathbf{X} < X_1, X_2, \cdots, X_m, C >$ is composed of m attributes and can have one class. The attributes are nominal rather than numeric. Numeric attributes will be discretized beforehand. Each attribute X_i $(i \in [1, m])$ takes v_i distinct values. The average number of values for an attribute is v. The class variable takes c values. The parents of X_i are referred to by $\Pi(i)$. [1] ϕ_{ir} is the r-th joint state (jointly instantiated values) of the parents of X_i. $|\phi_i|$ is the number of joint states of X_i's parents.

2 SuperParent-One-Dependance Estimators (SPODE)

The notion of k-dependence estimators was introduced by Sahami (1996). A k-dependence estimator is a Bayesian network in which each attribute has the class C and a maximum of k other attributes as parents. For example, naive-Bayes classifiers are 0-dependence estimators.

A One-Dependence Estimator (ODE) allows each attribute to depend on at most *one* other attribute in addition to the class, such as TAN (Friedman et al., 1997) and SP-TAN (Keogh & Pazzani, 1999). ODEs have attracted much attention for providing a good trade-off between classification efficiency and efficacy (Sahami, 1996; Friedman et al., 1997; Keogh & Pazzani, 1999).

A SuperParent-One-Dependence Estimator (SPODE) requires all attributes to depend on the *same* attribute, namely the *superparent*, in addition to the class (Keogh & Pazzani, 1999). A SPODE with superparent X_p will estimate the probability of each class label c given an instance $\mathbf{x}$ as follows. Denote the value of X_p in $\mathbf{x}$ by x_p.

$$\begin{aligned} P(c \,|\, \mathbf{x}) &= P(c, \mathbf{x}) / P(\mathbf{x}) \\ &\propto P(c, \mathbf{x}) \\ &= P(c, x_p, \mathbf{x}) \\ &= P(c, x_p) \times P(\mathbf{x} \,|\, c, x_p) \\ &= P(c, x_p) \times \prod_{i=1}^{m} P(x_i \,|\, c, x_p). \qquad (1) \end{aligned}$$

Averaged One-Dependence Estimators (AODE) uses an ensemble of all SPODEs. Since the equality (1) holds for every SPODE, it also holds for the

[1] For a SPODE, the class is the root and has no parents. The superparent has a single parent: the class. Other attributes have two parents: the class and the superparent.

mean over any subset. A group of k SPODEs corresponding to the superparents $X_{p_1}, \cdots, X_{p_k}$ estimate the class probability by averaging their results as follows.

$$P(c \,|\, \mathbf{x}) \propto \frac{\sum_{i=1}^{k} P(c, x_{p_i}) \times \prod_{j=1}^{m} P(x_j \,|\, c, x_{p_i})}{k}. \tag{2}$$

To classify an instance where $X_{p_i} = x_{p_i}$, AODE selects every SPODE i for which there are 30 or more training instances that satisfy $X_{p_i} = x_{p_i}, i \in [1, m]$. The reason behind is that 30 is a widely utilized minimum on sample size for statistical inference purposes. With fewer training data, a SPODE may incur unreliable probability estimation and hence sub-optimal classification accuracy.

This simple selection criterion of AODE has rendered surprisingly good performance. The resulting SPODE ensemble has been demonstrated to deliver competitive prediction accuracy together with improved efficiency in comparison to TAN (Friedman et al., 1997) and SP-TAN (Keogh & Pazzani, 1999). However, one should expect that selecting *out* poorly predictive SPODEs would both improve classification accuracy and classification speed. The goal here is to find out what selection criteria for SPODEs suit what scenarios and why.

3 Selecting SPODEs

The general problem for model selection is, given training instances, how to decide which the best explanatory model(s) is within some model space. Given m attributes, the model space here consists of m SPODEs, each appointing one superparent. To select SPODEs, two key factors are the ordering metric and the stopping criterion. The former orders each SPODE on merit. The latter decides when SPODEs of sufficient merit are no longer to be found for the ensemble.

Five ordering metrics are studied here, including popular information-theoretic metrics and accuracy-based empirical metrics: Minimum Description Length (MDL), Minimum Message Length (MML), Leave One Out (LOO), Backward Sequential Elimination (BSE), and Forward Sequential Addition (FSA). The stopping criterion coupled with each metric may vary.

3.1 Information-theoretic metrics

Information-theoretic metrics have gained considerable popularity in the machine learning community. They provide a combined score for a proposed explanatory model and for the data given the model: $I(h) + I(D|h)$, where h is a SPODE and D are the training data. All such metrics aim to find a balance between goodness of fit ($I(D|h)$) and model simplicity ($I(h)$), and thereby achieve good modeling performance without overfitting the data.

The first term, $I(D|h)$, is shared by information-theoretic metrics and is: [2]

$$I(D|h) = n(\sum_{i=1}^{m+1} H(X_i) - \sum_{i=1}^{m+1} H(X_i, \Pi(i)))$$

[2] For uniformity, X_i represents the class variable when $i = m + 1$.

where $H(X_i)$ is the entropy of X_i, and $H(X_i, \Pi(i))$ is the mutual information between X_i and its parent variables: [3]

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

$$H(X_i, \Pi(i)) = H(X_i) - H(X_i|\Pi(i)) = \sum_{j \leq v_i} \sum_{m \leq |\phi_i|} P(x_{ij}, \phi_{ir}) \log \frac{P(x_{ij}, \phi_{ir})}{P(x_{ij})P(\phi_{ir})}.$$

Minimum description length (MDL) Suzuki (1996) developed MDL for learning Bayesian networks and calculates $I(h)$ as follows. For any root node X_i (where $\Pi(i) = \emptyset$), the product term on the right should be replaced by 1.

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

From the perspective of ordering models, MDL differs only insignificantly from Akaike's information criterion (AIC) (Akaike, 1974) and the Bayesian information criterion (BIC) (Schwarz, 1978), which respectively calculate $I(h)$ by $2(\sum_{i=1}^{m+1}(v_i - 1) \prod_{j \in \Pi(i)} v_j)$ and $\log n(\sum_{i=1}^{m+1}(v_i - 1) \prod_{j \in \Pi(i)} v_j)$. Hence, the analysis and evaluation of MDL here also represent those of AIC and BIC.

Minimum message length (MML) The MDL metric above is not strictly efficient for encoding Bayesian networks. The MML metric is. Also, it is a basic principle of MML that there is a relation between the precision of estimated parameters and the volume of data. So the MML score for the model $I(h)$ takes the data volume into account. The result is: [4]

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

where S_{ij} is the number of training instances where the parents $\Pi(i)$ take their joint j-th value, and α_{ijl} is the number of training instances where X_i takes its l-th value and $\Pi(i)$ 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}.

3.2 Accuracy-based empirical metrics

In contrast to theoretic metrics, empirical metrics select individual SPODEs, or their ensembles, strictly on how well they perform in predictive tests.

[3] Generally the log base does not matter and this paper assumes natural logs (base-e).

[4] C_2^{m+1} is combination of 2 out of $(m+1)$. See Korb and Nicholson (2004, Chapter 8) for details. Discrepancies in the first part of the formula are due to the restriction here to SPODEs, rather than the full range of Bayesian networks.

Leave one out (LOO) LOO scores each individual SPODE with superparent X_p by its accuracy on leave-one-out cross validation in the training data. Given this SPODE, LOO loops through the training data n times, each time training the SPODE from $(n-1)$ instances and using it 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.

Backward sequential elimination (BSE) Backward sequential elimination starts out with a full ensemble including every SPODE. It then uses hill-climbing search to iteratively eliminate SPODEs whose individual exclusion most lowers 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 error of E_{test}. [5] The E_{test} which yields the lowest error is retained and 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.

Forward sequential addition (FSA) Forward sequential addition begins with an empty ensemble of SPODEs. 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 accuracy of E_{test}. The E_{test} who obtains the lowest error is retained and the corresponding added SPODE is permanently 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.

3.3 Stopping criterion

For MDL, MML and LOO, the lower its metric value, the higher priority is given to using a SPODE. The stopping criterion used here is the mean value of a metric over all candidate SPODEs. SPODEs whose metric values are lower than the mean will be included in the ensemble, while those of higher values will not.

For BSE, the process produces m SPODE ensembles, from size m to 1. Each ensemble is the one that achieves the lowest classification error among all ensembles of its size. Across these m ensembles, the one with the lowest classification error, E_{min}, gives the stopping point. Following the reverse order of the elimination order, one should first include the last SPODE to be eliminated and so on until the ensemble reaches the set of SPODEs that delivered E_{min}.

For FSA, the stopping criterion is similar to BSE's, being the ensemble that achieves the lowest classification error during the addition process. The difference

[5] A SPODE ensemble does classification by Formula 2.

is that one should follow the same order of the addition order to include SPODEs until the ensemble reaches the set of SPODEs that delivered E_{min}.

4 Time complexity analysis

Recall that the number of training instances and attributes are n and m. The average number of values for an attribute is v. The number of classes is c.

4.1 Training overhead

MDL The complexity of calculating $I(D|h)$ is $O(mv^2c)$. The dominating part is from $H(X_i, \Pi(i))$ which iterates through each value ($O(v)$), and then each joint value of the superparent and the class ($O(vc)$). The complexity of calculating $I(h)$ is $O(m)$. [6] Since the selection repeats for each attribute ($O(m)$), the overall complexity is $O(m * (mv^2c + m)) = O(m^2v^2c)$.

MML Although it looks complex, MML for SPODEs can be computed in polynomial time despite that it is exponential for Bayesian networks (Cooper & Herskovits, 1992; Korb & Nicholson, 2004). The dominating complexity of MML 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 ($O(v) + O(n)$). On top of that it loops through each attribute value ($O(v)$) for which a third factorial is calculated ($O(n)$). Hence the complexity is $O(m*vc*(v+n)*v*n) = O(mv^3n^2c)$. This repeats for each attribute ($O(m)$) and the overall complexity is hence $O(m^2v^3n^2c)$.

LOO 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 $O(m^2cn)$.

BSE The hill climbing procedure of reducing a SPODE ensemble of size m to 0 will render a complexity of $O(m^3)$. In the first round, it alternatively eliminates each of m SPODEs, each time testing a SPODE set of size $(m-1)$. In the second round, it alternatively eliminates each of $(m-1)$ SPODEs, each time testing a SPODE set of size $(m-2)$. Following this line of reasoning, the total number of probing a SPODE is $m(m-1) + (m-1)(m-2) + \cdots + 2*1 + 1*0 = O(m^3)$. As explained for LOO, to test each SPODE by leave-one-out cross validation will incur complexity of $O(mcn)$. As a result, the overall complexity is $O(m^4cn)$.

[6] Although MDL has an extra loop $\prod_{j \in \Pi(i)} v_j$, in case of SPODE, $|\Pi(i)|$ 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.

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, each time testing a SPODE set of size 1. In the second round, it alternatively adds each of $(m-1)$ SPODEs, each time testing a SPODE set of size 2. Following this line of reasoning, the total number of probing a SPODE is $m*1+(m-1)*2+\cdots+2*(m-1)+1*m = O(m^2)$. As explained for LOO, to test each SPODE by leave-one-out cross validation will incur complexity of $O(mcn)$. As a result, the overall complexity is $O(m^3cn)$.

4.2 Classification overhead

No matter what selection metric is applied, the result is a linear combination of SPODEs. Hence, each metric's complexity is of the same order $O(m^2c)$, resulting from an $O(mc)$ classifying algorithm applied over an $O(m)$ sized ensemble.

5 Experiments

Experiments are conducted to find out the classification efficacy and efficiency for each selection metric.

5.1 Design and results

Experimental data involve a comprehensive suite of 41 often-used data sets from the UCI machine learning repository (Blake & Merz, 2004). The statistics of each data set are presented in Table 3 in Appendix. To test a selection strategy, a 3-fold cross validation is conducted. Each candidate selection metric selects SPODEs according to evidence offered by the training data, and uses the resulting SPODE ensemble to classify the test data.

The classification error rate on each data set produced by each selection metric is presented in Table 3 in Appendix. The resulting win/lose/draw record of each metric compared against each other metric is presented in Table 1. A binomial sign test can be applied to each record to suggest whether the wins are by chance or systematic. The training time and classification time of each metric on each data set are presented in Table 4 in Appendix.

	NB	AODE	MDL	MML	LOO	BSE	FSA
NB	0/0/41	10/27/4	13/25/3	11/27/3	9/26/6	8/28/5	8/29/4
AODE	**27/10/4**	0/0/41	17/14/10	11/19/11	10/22/9	10/24/7	11/25/5
MDL	**25/13/3**	14/17/10	0/0/41	11/16/14	13/20/8	8/25/8	10/25/6
MML	**27/11/3**	19/11/11	16/11/14	0/0/41	17/19/5	12/24/5	13/21/7
LOO	**26/9/6**	**22/10/9**	20/13/8	19/17/5	0/0/41	9/23/9	10/24/7
BSE	**28/8/5**	**24/10/7**	**25/8/8**	**24/12/5**	**23/9/9**	0/0/41	13/11/17
FSA	**29/8/4**	**25/11/5**	**25/10/6**	21/13/7	**24/10/7**	11/13/17	0/0/41

Table 1. Win-lose-draw record of each method in column compared against each in row. A **bold face** indicates that the wins against losses are statistically significant using a two-tailed binomial sign test at the critical level 0.05, and hence the corresponding method has a systematic (instead of by chance) advantage over its counterpart.

5.2 Observations and analysis

Selection makes a difference Compared with AODE, all selection metrics except MDL win more often than not across the 41 data sets. In particular, LOO, BSE and FSA achieve win/lose/tie records of 22/10/9, 24/10/7 and 25/11/5 respectively, all of which are statistically significant at the 0.05 critical level according to the binomial sign test. This suggests that their advantages over AODE is systematic rather than due to chance. Hence model selection for SPODEs is advisable.

MML is more effective than MDL Among the information-theoretic metrics, MML wins against MDL more often than not. It also achieves lower arithmetic and geometric mean error than MDL. Compared with AODE, MML outperforms AODE with a win/lose/tie record of 19/11/11. By contrast, MDL loses to AODE (and so also AIC and BIC). The plausible explanation lies in MML providing a more efficient encoding of network structure, as well as taking the precision of its parameter estimates more seriously.

Empirical metrics are more effective than theoretic metrics All three empirical metrics outperform their theoretic counterparts. Compared with the most effective theoretic metric MML, BSE achieves a win/lose/draw record of 24/12/5, FSA of 21/13/7 and LOO of 19/17/5. A possible reason is that not only do empirical metrics consider interactions among the class, the superparent and other attributes within the model, they also consider the interaction's impact on the classification accuracy. When selecting a classifier it is always desirable to optimize the thing that one wants to optimize, that is, the accuracy.

Measuring ensemble outperforms measuring single SPODEs MDL, MML and LOO measure each individual SPODE in isolation. [7] BSE and FSA measures a SPODE ensemble as a whole. BSE and FSA outperform the best theoretic metric MML with their win/lose/tie records being 24/12/5 and 21/13/7 respectively. They also outperform their empirical peer LOO with the win/lose/tie records being 23/9/9 and 24/10/7. Most of these wins are statistically significant at the 0.05 critical level. The reason is that the eventual classification task is carried out by a team of SPODEs. A SPODE that achieves high accuracy in isolation does not necessarily mean it is the most valuable one to include in an ensemble. Measuring the collective merit of a SPODE ensemble directly assesses what one is trying to learn, giving better results. For example, consider five SPODEs A, B, C, D, E and their classification on each training instances as illustrated in Table 2, where X indicates a misclassification and $\surd$ indicates a correct classification. Using a majority vote, this ensemble will misclassify instance 1. However, A, B and C are redundant, and replicate each other's errors. Hence it is better to eliminate two out of the three A, B or C instead of either D or E, despite their lower individual errors.

[7] Note that the information-theoretic metrics can be extended to apply to ensembles rather than individual models. It is a future work.

Instance ID.	A	B	C	D	E
1	X	X	X	√	√
2	√	√	√	X	√
3	√	√	√	X	√
4	√	√	√	√	X
5	√	√	√	√	X

Table 2. Measuring a SPODE in isolation is less effective.

Backward elimination and forward addition Although previous work suggested that backward elimination tends to be more effective than forward addition for feature selection(Koller & Sahami, 1996; Kohavi & John, 1996; Wu & Urpani, 1999), there is no significant difference of performance between BSE and FSA. In terms of classification accuracy, BSE wins slightly more than not compared with FSA (win 13 loss 11). It also achieves the lowest arithmetic and geometric mean error among alternative metrics. In terms of training efficiency, FSA is faster than BSE by an order of m.

Theoretic metrics are more efficient for training Consistent with the analysis of training time complexity in Section 4, for training efficiency, from the fastest to the slowest are MDL, MML, LOO, FSA and BSE. In general, empirical metrics are slower than theoretic ones because they need to loop through training data for leave-one-out cross validation. Metrics that measure SPODE ensembles are slower than those that measure individual SPODEs because they need to probe different aggregations of individual SPODEs.

All metrics are fast for classification Consistent with the analysis of classification time complexity in Section 4, for classification efficiency, every selection metric is equally fast. In many real-world scenarios, classification efficiency is more important than training efficiency. The experimental results suggest that when training time is not taken into consideration, high classification accuracy and high classification efficiency are not necessarily exclusive. This observation suggests that metrics like BSE hold considerable promise as practical, accurate and feasible selection strategies.

6 Conclusion

An ensemble of SuperParent-One-Dependence-Estimators (SPODEs) retains the simplicity and direct theoretical foundation of naive Bayes while alleviating the limitations of its attribute independence assumption. In consequence it delivers effective classification with modest computational overhead (Webb et al., 2005). This paper focuses on how to select SPODEs for ensembling so as to further improve the classification accuracy. Popular information-theoretic metrics like MDL and MML, and accuracy-based empirical metrics like LOO, BSE and FSA have been applied for model selection.

Evidence obtained from theoretical analysis and empirical trials suggests that appropriate selection of the SPODEs to be included in an ensemble can further improve the classification accuracy. Empirical metrics that involve testing

SPODEs' classification performance on training data can outperform theoretic metrics at a cost of higher training time overhead. Metrics that measure the ensemble as a whole can outperform metrics that measure SPODEs in isolation, also at a cost of higher training time overhead. For classification time, various selection criteria all produce a linear combination of SPODEs. Hence, they have the same order of time complexity for classification.

As a result, if the training time is limited, it is suggested to employ MML, LOO, FSA and BSE in that order, corresponding to the increasing order of time allowance. If the training time is not a concern, given a choice amongst the metrics studied here, BSE is the best.

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *AC-19*, 716–23.

Blake, C., & Merz, C. J. (2004). UCI repository of machine learning databases. [Machine-readable data repository]. University of California, Department of Information and Computer Science, Irvine, CA.

Cooper, G. F., & Herskovits, E. (1992). A Bayesian method for the induction of probabilistic network from data. *Machine Learning*, *9*, 309–347.

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning*, *29*(2-3), 131–163.

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

Kohavi, R., & John, G. H. (1996). Wrappers for feature subset selection. *Artificial Intelligence, Special Issue on Relevance*, *97*(1-2), 273–324.

Koller, D., & Sahami, M. (1996). Toward optimal feature selection. In *Proceedings of the 13th International Conference on Machine Learning*, pp. 284–292. Morgan Kaufmann Publishers.

Korb, K., & Nicholson, A. (2004). *Bayesian Artificial Intelligence*. Chapman & Hall/CRC, Boca Raton, FL.

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

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, *6*, 461–5.

Suzuki, J. (1996). 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*, pp. 463–470.

Webb, G. I., Boughton, J., & Wang, Z. (2005). Not so naive Bayes: Averaged one-dependence estimators. *Machine Learning*, *58*(1), 5–24.

Wu, X., & Urpani, D. (1999). Induction by attribute elimination. *IEEE Transactions on Knowledge and Data Engineering*, *11*(5), 805–812.

Appendix

Data Set	Size	Att	NB	AODE	MDL	MML	LOO	BSE	FSA
adult	48842	14	16.2	14.8	14.7	14.8	14.7	14.0	14.0
anneal	898	38	4.1	3.5	3.5	3.5	3.1	3.0	2.4
balance-scale	625	4	24.3	27.5	25.8	24.3	27.2	24.0	24.0
bands	1078	36	28.4	27.8	26.5	26.2	27.5	26.7	27.5
bcw	699	9	2.7	3.4	3.0	3.6	3.3	3.0	3.0
bupa	345	6	42.9	42.9	42.9	42.9	42.9	42.9	42.9
chess	551	39	13.1	12.2	12.7	13.2	12.0	11.1	11.8
cleveland	303	13	16.5	17.2	17.2	17.5	16.8	17.2	16.8
crx	690	15	13.9	12.8	12.9	13.2	13.0	13.9	12.9
echo74	74	6	24.3	25.7	23.0	23.0	24.3	23.0	23.0
german	1000	20	26.4	26.0	26.9	27.1	26.2	25.3	25.5
glass	214	9	23.4	22.9	22.9	22.4	22.9	22.9	22.4
heart	270	13	17.0	16.7	15.9	16.3	16.7	17.4	17.4
hepatitis	155	19	14.8	14.2	13.5	13.5	15.5	15.5	14.8
horse-colic	368	21	23.9	21.5	22.0	20.9	21.5	20.7	20.9
house-votes-84	435	16	9.9	6.2	5.7	5.7	5.5	4.8	6.0
hungarian	294	13	16.3	16.3	16.3	16.3	16.3	16.0	16.0
hypo	3772	29	1.9	2.0	2.2	2.2	1.9	1.2	1.2
ionosphere	351	34	9.7	9.7	10.3	8.8	9.4	9.7	9.1
iris	150	4	5.3	6.0	6.0	6.0	6.0	6.0	6.0
kr-vs-kp	6393	36	12.3	8.8	8.8	8.6	7.0	5.4	5.3
labor-neg	57	16	14.0	10.5	10.5	10.5	12.3	12.3	14.0
led	1000	7	26.6	26.7	26.3	26.2	26.8	26.8	26.8
letter-recognition	20000	16	26.1	12.1	13.9	12.3	11.7	11.4	11.3
lyn	296	18	15.5	13.5	12.2	12.8	12.2	12.2	12.2
mfeat-mor	2000	6	32.0	31.5	31.5	31.4	30.5	30.7	30.8
musk1	476	166	18.9	18.3	17.2	17.4	18.3	16.6	16.0
new-thyroid	215	5	7.0	7.0	8.4	8.4	8.4	6.5	6.5
pendigits	10992	16	12.6	2.6	3.0	2.6	2.7	2.6	2.6
pid	768	8	25.0	24.7	25.0	24.5	24.6	24.9	24.7
post-operative	90	8	30.0	27.8	32.2	26.7	28.9	28.9	28.9
promoters	106	57	9.4	17.9	15.1	16.0	15.1	9.4	9.4
ptn	339	17	52.2	53.4	52.8	52.8	53.4	52.2	54.3
sign	12546	8	36.3	28.9	28.5	28.5	28.6	28.0	28.1
sonar	208	60	26.9	26.0	27.4	28.8	26.9	27.4	28.8
soybean	683	35	11.3	7.6	7.8	7.6	7.5	7.5	8.2
thyroid	9169	29	11.7	8.3	8.7	8.3	8.3	7.9	8.0
ttt	958	9	29.1	25.3	25.5	25.5	26.6	25.6	25.3
vehicle	846	18	39.7	31.3	31.7	31.3	30.9	31.3	31.6
vowel-context	990	11	42.3	28.4	32.8	30.7	26.4	24.1	24.1
wine	178	13	2.2	2.8	2.8	2.8	2.2	3.9	3.9
Arithmetic Mean	-	-	19.9	18.1	18.2	17.9	18.0	17.4	17.5
Geometric Mean	-	-	20.0	18.2	18.3	18.0	18.0	17.5	17.6

Table 3. Data sets and classification error (%)

	Training Time					Classification Time				
Data Set	MDL	MML	LOO	BSE	FSA	MDL	MML	LOO	BSE	FSA
adult	3,320	11,630	12,640	133,530	114,240	580	570	670	450	460
anneal	240	360	1,650	78,220	53,290	40	20	40	40	40
balance-scale	20	30	20	60	70	30	0	30	20	10
bands	90	580	1,150	37,380	30,110	20	10	40	40	10
bcw	50	50	70	440	500	30	10	20	20	30
bupa	10	40	20	50	70	10	0	0	0	10
chess	40	670	1,120	50,380	40,780	30	40	40	30	20
cleveland	10	50	140	590	1,070	20	30	40	20	10
crx	60	130	400	3,060	1,770	30	10	40	20	10
echo74	10	10	20	20	30	0	0	0	0	10
german	10	430	450	8,340	6,770	20	60	20	40	20
glass	20	80	50	190	150	10	0	0	10	10
heart	20	90	70	600	470	20	10	10	10	10
hepatitis	20	60	80	810	1,350	0	0	0	10	20
horse-colic	30	110	220	3,390	2,920	0	20	10	40	10
house-votes-84	20	90	140	1,730	1,410	30	20	10	20	20
hungarian	20	50	80	870	560	10	20	10	20	20
hypo	260	5,080	4,280	181,750	126,870	110	250	180	90	80
ionosphere	220	470	670	18,340	16,090	10	10	20	10	10
iris	20	0	10	20	0	10	10	20	0	0
kr-vs-kp	350	3,510	5,590	224,960	169,020	310	120	90	80	80
labor-neg	10	20	30	150	270	0	10	0	10	0
led	20	80	80	760	610	30	40	10	30	30
letter	4,290	11,930	21,790	1,282,030	760,180	3,440	5,650	3,830	4,300	4,180
lyn	60	40	80	1,630	1,420	10	10	10	10	0
mfeat-mor	230	270	270	1,130	950	60	60	50	60	40
musk1	1,220	11,040	138,540	18,842,398	15,084,140	380	170	340	60	50
new-thyroid	0	30	40	20	50	10	10	10	0	20
pendigits	2,050	3,820	8,250	252,520	165,620	610	780	780	840	830
pid	40	70	120	360	340	10	30	20	10	20
post-operative	10	0	30	40	30	10	0	0	0	0
promoters	290	370	1,190	34,050	37,250	10	10	0	0	0
ptn	90	80	200	10,230	6,470	50	30	40	20	20
sign	770	3,260	1,520	6,750	6,010	140	280	120	140	160
sonar	160	600	1,390	106,300	72,010	40	20	30	10	20
soybean	540	660	2,260	594,910	252,010	230	260	360	130	100
thyroid	1,090	6,540	15,470	2,591,640	1,302,000	3,920	750	1,340	680	910
ttt	10	100	60	640	480	20	10	30	20	30
vehicle	110	300	390	8,550	5,790	40	40	50	10	30
vowel-context	300	360	400	3,960	2,940	50	70	40	40	20
wine	30	60	60	440	370	0	10	0	10	0
Arithmetic Mean	394	1,540	5,391	597,152	445,524	253	230	204	179	179

Table 4. Training and classification time (milliseconds)