International Journal of Information Technology (IJIT)
Volume 1, Issue 2, Jan-Dec 2018, pp. 1-18, Article ID: IJIT_01_02_001
Available online at https://iaeme.com/Home/issue/IJIT?Volume=1&Issue=2
ISSN Online: 2251-2809, Journal ID: 4573-3410
Impact Factor (2018): 3.85 (Based on Google Scholar Citation)

AVERAGED ONE DEPENDENCE ESTIMATORS FOR NAIVE BAYESIAN CLASSIFICATION

ANNE VAMSYPRIYA
Department of Computer Science and Engineering,
R.V.R & J.C College of Engineering, Chowdavaram,
Andhra Pradesh, 522019, India

G. SWETHA
Department of Computer Science and Engineering,
R.V.R & J.C College of Engineering, Chowdavaram,
Andhra Pradesh, 522019, India

G. SAHITHI
Department of Computer Science and Engineering,
R.V.R & J.C College of Engineering, Chowdavaram,
Andhra Pradesh, 522019, India

ABSTRACT

Naive Bayes (NB) is a simple, computationally efficient probabilistic approach to classification learning. It assumes that all attributes are conditionally independent of each other given the class. But in real time applications complete attribute independence is not possible, so to address this limitation of NB we go for ODE and AODE.

One Dependence Estimators (ODE) is based on the concept of selecting one attribute as the dependent attribute and it acts as the predictor to identify the class label. To improve the efficiency of ODE further we have AODE.

Averaged One-Dependence Estimators (AODE) is a popular and effective approach to Bayesian learning. It relaxes the attribute independence assumption by averaging all models that assume all attributes are conditionally dependent on the class and one common attribute, known as the super-parent. This often improves the classification performance significantly.

In the work, a new attribute selection approach is proposed for AODE. It can search in a large model space, while it requires only a single extra pass through the training data, resulting in a computationally efficient two-pass learning algorithm. Its low bias and computational efficiency make it an attractive algorithm for learning from big data.

Keywords: Naive Bayes, One-Dependence Estimators (ODE), Averaged One-Dependence Estimators (AODE), Attribute Selection, Bayesian Classification

Cite this Article: Anne Vamsypriya, G. Swetha, G. Sahithi, Averaged one Dependence Estimators for Naive Bayesian Classification, *International Journal of Information Technology (IJIT)*, 1(2), 2018, pp. 1-18
https://iaeme.com/Home/issue/IJIT?Volume=1&Issue=2

1. INTRODUCTION

1.1. Introduction to Domain

Data mining is the computing process of discovering patterns in large data sets involving methods at the intersection of machine learning, statistics, and database systems. An essential process where intelligent methods are applied to extract data patterns.

It is an interdisciplinary subfield of computer science. The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use. Aside from the raw analysis step, it involves database and data management aspects, data pre-processing, model and inference considerations, interestingness online updating metrics, complexity considerations, post-processing of discovered structures, visualization. Data mining is the analysis step of the "knowledge discovery in databases" process.

The manual extraction of patterns from data has occurred for centuries. Early methods of identifying patterns in data include Bayes theorem and regression analysis. The proliferation, ubiquity and increasing power of computer technology has dramatically increased data collection, storage, and manipulation ability. As data sets have grown in size and complexity, direct "hands-on" data analysis has increasingly been augmented with indirect, automated data processing, aided by other discoveries in computer science, such as neural networks, cluster analysis, genetic algorithms, decision trees and decision rules, and support vector machines.

Data mining is the process of applying these methods with the intention of uncovering hidden patterns in large data sets. It bridges the gap from applied statistics and artificial intelligence (which usually provide the mathematical background) to database management by exploiting the way data is stored and indexed in databases to execute the actual learning and discovery algorithms more efficiently, allowing such methods to be applied to ever larger data sets.

The knowledge discovery in databases (KDD) process is commonly defined with the stages:

(1) Selection
(2) Pre-processing
(3) Transformation
(4) Data mining
(5) Interpretation/evaluation.

CRISP-DM methodology is the leading methodology used by data miners.

CRISP-DM is simplified as:

Pre-processing, Data Mining and Results Validation.

1.2. Keywords

Classification:

Classification is a general process related to categorization, the process in which ideas and objects are recognized, differentiated, and understood.

A classification system is an approach to accomplishing classification.

In machine learning and statistics, **classification** is the problem of identifying to which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known.

An example would be assigning a given email into "spam" or "non-spam" classes or assigning a diagnosis to a given patient as described by observed characteristics of the patient (gender, blood pressure, presence or absence of certain symptoms, etc.). Classification is an example of pattern recognition.

Examples of classification algorithms include:

- Linear classifiers
  - Fisher's linear discriminant
  - Logistic regression
  - Naive Bayes classifier
  - Perceptron
- Support vector machines
  - Least squares support vector machines
- Quadratic classifiers
- Kernel estimation
  - k-nearest neighbor
- Boosting (meta-algorithm)
- Decision trees
  - Random forests
- Neural networks
- FMM Neural Networks
- Learning vector quantization

Naive Bayes:

Naive Bayes (NB) is a simple, computationally efficient probabilistic approach to classification learning. It assumes that all attributes are conditionally independent of each other given the class.

We wish to predict from a training sample of classified objects the class of an example $x=(x_1,...,x_n)$, where x_i is the value of the ith attribute. We can minimize error by selecting $\text{argmax}_y P(y|x)$, where $y \in c1,...ck$ are the k classes. To this end we seek an estimate $\hat{} P(y|x)$ of $P(y|x)$ and perform classification by selecting $\text{argmax}_y \hat{} P(y|x)$. From the definition of conditional probability we have

$$P(y|x) = P(y,x)/P(x)$$
$$\propto P(y,x)$$

Hence, $\text{argmax}_y P(y|x) = \text{argmax}_y P(y,x)$, and the latter is often calculated in practice rather than the former.

Semi- naive Bayes:

A semi-naive Bayesian network is just a Bayesian network where the naive independence assumption is relaxed in some way. There are many ways of doing this, enabling one to represent some degree of dependence between the child nodes/features.

LBR & TAN:

Of the many approaches to obviating this problem cited in the introduction, two have demonstrated very low error: Lazy Bayesian Rules (LBR) and Super Parent Tree Augmented Naive Bayes (SP-TAN). Both rely on weaker attribute independence assumptions than NB.

LBR uses lazy learning. For each x = (xi,...,xn) to be classified, a set W of the attribute values is selected. Independence is assumed among the remaining attributes given W and y. Hence, x can be classified by selecting

$$\operatorname*{argmax}_y \left(\hat{P}(y \mid W) \prod_{i=1}^{n} \hat{P}(x_i \mid y, W) \right)$$

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

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

The parent function p(·) is developed at training time. TAN uses conditional mutual information to select the parent function. SP-TAN uses a simple heuristic wrapper approach that seeks to minimize error on the training sample. At training time both TAN and SP-TAN generate a three-dimensional table of probability estimates for each attribute-value.

J48:

J48 decision tree , by applying a decision tree like **J48** on that dataset would allow you to predict the target variable of a new dataset record. Decision tree **J48** is the implementation of algorithm ID3 (Iterative Dichotomiser 3) developed by the WEKA project team.

AODE:

As an improvement to NB, Averaged One-Dependence Estimators (AODE) relaxes the attribute independence assumption by averaging all models that assume all attributes are conditionally dependent on the class and one common attribute, known as the super-parent.

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

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

Attribute Selection:

In machine learning and statistics, attribute selection also known as variable subset selection, is the process of selecting a subset of relevant features(variables, predictors) for use in model construction.

Feature selection techniques are used for four reasons:

- simplification of models to make them easier to interpret by researchers/users
- shorter training times
- to avoid the curse of dimensionality

- enhanced generalization by reducing overfitting

2. NOT SO NAIVE BAYES: AGGREGATING ONE-DEPENDENCE ESTIMATORS

2.1. LBR:

Of the many approaches to obviating this problem cited in the introduction, two have demonstrated very low error: Lazy Bayesian Rules (LBR) and Super Parent Tree Augmented Naive Bayes (SP-TAN). Both rely on weaker attribute independence assumptions than NB.

LBR uses lazy learning. For each x = (xi,...,xn) to be classified, a set W of the attribute values is selected. Independence is assumed among the remaining attributes given W and y. Hence, x can be classified by selecting

$$\operatorname*{argmax}_y \left(\hat{P}(y \mid W) \prod_{i=1}^{n} \hat{P}(x_i \mid y, W) \right)$$

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

At training time, LBR simply stores the training data, an operation of time and space complexity O(tn). At classification time, however, LBR must select the attributes for inclusion in W, an operation of time complexity O(tkn2). In practice, the cumulative computation is reasonable when few examples are to be classified for each training set. When large numbers of examples are to be classified, the computational burden becomes prohibitive.

2.2. TAN & SP-TAN:

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

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

The parent function $p(\cdot)$ is developed at training time. TAN uses conditional mutual information to select the parent function. SP-TAN uses a simple heuristic wrapper approach that seeks to minimize error on the training sample. At training time both TAN and SP-TAN generate a three-dimensional table of probability estimates for each attribute-value conditioned by each other attribute-value and each class, space complexity O(k(nv)2).

SP-TAN must also store the training data, with additional space com- plexity O(tn). The time complexity of forming the three dimensional probability table required by both TAN and SP-TAN is O(tn2) as an entry must be updated for every training case and every combination of two attribute-values for that case.

To create the parent function TAN must first calculate the conditional mutual information, requiring consideration for each pair of attributes, every pairwise combination of their respective values in conjunction with each class value $O(kn^2v^2)$. A maximal spanning tree is then generated, time complexity $O(n^2 \log n)$. The time complexity of forming the parent function for SP-TAN is $O(tkn^3)$, as the selection of a single parent is order $O(tkn^2)$ and parent selection is performed repeatedly, potentially being repeated until every attribute has a parent.

At classification time both TAN and SP-TAN need only store the probability tables, space complexity $O(knv^2)$. This compression over the table required at training time is achieved by storing probability estimates for each attribute-value conditioned by the parent selected for that attribute, and the class. The time complexity of classifying a single example is O(kn).

3. AODE

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

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

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

Considering first the issue of probability estimation, it is clearly desirable to be able to pre-compute all required base probability estimates at training time, as does SP-TAN. Sahami introduces the notion of x-dependence estimators, whereby the probability of each attribute value is conditioned by the class and at most x other attributes.

In general, the probability estimates required for an x-dependence estimator can be stored in an (x + 2)-dimensional table, indexed by the target attribute-value, the class value, and the values of the x other attributes by which the target is conditioned. To maintain efficiency it appears desirable to restrict ourselves to 1-dependence classifiers.

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

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

However, while avoiding model selection appears desirable, it appears to conflict with the desire to use 1-dependence classifiers. These require each attribute to depend on one other attribute and the precise such attribute must surely be selected. Our solution is to select a limited class of 1-dependence classifiers and to aggregate the predictions of all qualified classifiers within this class. The class we select is all 1dependence classifiers where there is a single attribute that is the parent of all other attributes. However, we wish to avoid including models for which the base probability estimates are inaccurate. To this end, when classifying an object x = hx1,...,xni, we exclude models where the training data contain fewer than m examples of the value for x of the parent attribute xi. In the current research we use m = 30, this being a widely utilized minimum on sample size for statistical inference purposes.

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

$$P(y,x) = P(y,x_i)P(x|y,x_i)$$

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

$$P(y,\mathbf{x}) = \frac{\sum_{i:1\le i\le n\wedge F(x_i)\ge m} P(y,x_i)P(\mathbf{x}\,|\,y,x_i)}{|\{i : 1 \le i \le n \wedge F(x_i) \ge m\}|}$$

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

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

$$\operatorname*{argmax}_{y} \left(\sum_{i:1 \le i \le n \wedge F(x_i) \ge m} \hat{P}(y, x_i) \prod_{j=1}^{n} \hat{P}(x_j \mid y, x_i) \right)$$

If $\neg\exists i : 1 \le i \le n \wedge F(xi) \ge m$, AODE defaults to NB.

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

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

At training time AODE need only form the tables of joint attribute value, class frequencies from which the probability estimates Pˆ(y, xi) and Pˆ(y, xi , xj) are derived that are required for estimating Pˆ(y, xi) and Pˆ(xj | y, xi). The space complexity of these tables is $O(k(nv)^2)$. Derivation of the frequencies required to populate these tables is of time complexity $O(tn^2)$. There is no model selection. Classification requires the tables of probability estimates formed at training time of space complexity $O(k(nv)^2)$. Classification of a single example requires calculation of Eq and is of time complexity $O(kn^2)$.

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

We expect AODE to achieve lower classification error than NB for the following reasons. First, as it involves a weaker attribute independence assumption, P(y,xi) ∏n j=1 P(xj |y,xi) should provide a better estimate of P(y,x) than P(y) ∏ n i=1 P(xi |y). Hence, the estimates from each of the one-dependence models over which AODE averages should be better than the estimate from NB, except insofar as the estimates of the base probabilities P(y,xi) and P(xj |y,xi) in the one-dependence models are less accurate than the estimates of the base probabilities P(y) and P(xi |y) used by NB.

The only systematic cause for such a drop in accuracy might result from the smaller numbers of examples from which the AODE base probabilities are estimated. We seek to guard against negative impact from such a cause by restricting the base models to those for which the parent attribute-value occurs with sufficient frequency.

Due to the extent to which AODE's estimates can be expected to grow in accuracy as the amount of data increases, we expect the magnitude of the advantage over NB to grow as the number of training examples grows. Second, there is a considerable evidence that aggregating multiple credible models leads to improved prediction accuracy and we expect to benefit from such an effect. Third, like NB, AODE avoids model selection and hence avoids the attendant variance.

4. LITERATURE REVIEW

4.1. AODE:

The classification task can be described as follows, given a training sample T of t classified objects, we are required to predict the probability P(y | x) that a new example x = ⟨x1,...,xa⟩ belongs to some class y, where xi is the value of the attribute xi and y ∈{c1,...,ck}.

In the following sections, we describe AODE for this classification task and a number of its key variants.

From the definition of conditional probability, we have P(y | x) = P(y,x)/P(x) . As $\mathrm{P}(\mathbf{x}) = \sum_{i=1}^{k} \mathrm{P}(c_i, \mathbf{x})$ and $y \in \{c_1, \ldots, c_k\}$, it is reasonable to consider P(x) as the normalizing constant and estimate only the joint probability P(y,x) in the remainder of this work.

Since the example x does not appear frequently enough in the training data, we cannot directly derive an accurate estimate of P(y,x) and must extrapolate this estimate from observations of lower-dimensional probabilities in the data.

Applying the definition of conditional probabilities again, we have P(y,x) = P(y)P(x | y) . The first term P(y) on the right side can be sufficiently accurately estimated from the sample frequencies, if the number of classes, k, is not too large. For the second term P(x | y), AODE assumes every attribute depends on the same parent attribute, the super-parent, thus obtains an one-dependence estimator (ODE), and then averages all eligible ODEs.

The joint probability P(y,x) is estimated as follows:

$$\hat{\mathrm{P}}(y, \mathbf{x}) = \frac{\sum_{i:1 \le i \le a \wedge \mathrm{F}(x_i) \ge m} \hat{\mathrm{P}}(y, x_i) \prod_{j=1}^{a} \hat{\mathrm{P}}(x_j \mid y, x_i)}{|\{i : 1 \le i \le a \wedge \mathrm{F}(x_i) \ge m\}|}$$

where |·| denotes the cardinality of a set, ˆ P(·) represents an estimate of P(·),F(xi) is the frequency of xi and m is the minimum frequency to accept xi as a super parent. The current research uses m = 1.

4.2. WAODE:

In the classification of AODE, each ODE is treated equally, that is, all eligible models are averaged and contribute uniformly to the classification rule. However, in many real world applications, attributes do not play the same role in classification. This observation inspires the weightily AODE, in which the joint probability is estimated as,

$$\hat{\mathrm{P}}(y, \mathbf{x}) = \frac{\sum_{i:1 \le i \le a \wedge \mathrm{F}(x_i) \ge m} W_i \hat{\mathrm{P}}(y, x_i) \prod_{j=1}^{a} \hat{\mathrm{P}}(x_j \mid y, x_i)}{\sum_{i:1 \le i \le a \wedge \mathrm{F}(x_i) \ge m} W_i}$$

In practice, mutual information between the super-parent and the class is often used as the weight Wi.

4.3. AODE with subsumption resolution:

Semi-naive Bayesian techniques seek to improve the accuracy of naive Bayes (NB) by relaxing the attribute independence assumption. We present a new type of semi-naive Bayesian operation, Subsumption Resolution (SR), which efficiently identifies occurrences of the specialization-generalization relationship and eliminates generalizations at classification time.

We extend SR to Near-Subsumption Resolution (NSR) to delete near–generalizations in addition to generalizations. We develop two versions of SR: one that performs SR during training, called eager SR (ESR), and another that performs SR during testing, called lazy SR (LSR). We investigate the effect of ESR, LSR, NSR and conventional attribute elimination (BSE) on NB and Averaged One-Dependence Estimators (AODE), a powerful alternative to NB. BSE imposes very high training time overheads on NB and AODE accompanied by varying decreases in classification time overheads.

ESR, LSR and NSR impose high training time and test time overheads on NB. However, LSR imposes no extra training time overheads and only modest test time overheads on AODE, while ESR and NSR impose modest training and test time overheads on AODE. Our extensive experimental comparison on sixty UCI data sets shows that applying BSE, LSR or NSR to NB significantly improves both zero-one loss and RMSE, while applying BSE, ESR or NSR to AODE significantly improves zero-one loss and RMSE and applying LSR to AODE significantly improves zero-one loss.

The tests show that AODE with ESR or NSR have a significant zero-one loss and RMSE advantage over Logistic Regression and a zero-one loss advantage over Weka's LibSVM implementation with a grid parameter search on categorical data. AODE with LSR has a zero-one loss advantage over Logistic Regression and comparable zero-one loss with LibSVM. Finally, we examine the circumstances under which the elimination of near-generalizations proves beneficial.

One extreme type of inter-dependence between attributes results in a value of one being a generalization of a value of the other. For example, consider Gender and Pregnant as two attributes, then Pregnant = yes implies that Gender = female. Therefore, Gender = female is a generalization of Pregnant = yes. Likewise, Pregnant = no is a generalization of Gender = male. Where one value x_i is a generalization of another, x_j, $P(y|xi,xj) = P(y|xj)$. In consequence dropping the more general value from any calculations should not harm any posterior probability estimates, whereas assuming independence between them may.

Motivated by this observation, Subsumption Resolution (SR) identifies pairs of attribute values such that one appears to subsume the other and deletes the generalization. Suppose that the set of indices of the resulting attribute subset is denoted by R, the joint probability is estimated as,

$$\hat{\mathrm{P}}(y,\mathbf{x}) = \frac{\sum_{i:i\in R\wedge \mathrm{F}(x_i)\geq m}\hat{\mathrm{P}}(y,x_i)\prod_{j\in R}\hat{\mathrm{P}}(x_j\mid y,x_i)}{|\{i:i\in R\wedge \mathrm{F}(x_i)\geq m\}|}$$

4.4. Forward and Backward Attribute Selection in AODE:

In order to repair harmful inter-dependencies among highly correlated attributes, proposed to select an appropriate attribute subset by hill climbing search. Two different search strategies can be used: FSS begins with the empty attribute set and successively adds attributes, while BSE starts with the complete attribute set and successively removes attributes. Both strategies greedily select the attribute whose addition or elimination best reduces the leave-one-out cross validation error on the training set. The process is terminated if there is no error improvement.

To differentiate the selection of parent or child, they introduce the use of a parent (p) and a child (c) set, each of which contains the set of indices of attributes that can be employed in, respectively, a parent or a child role in AODE. The joint probability is estimated as,

$$\hat{\mathrm{P}}(y,\mathbf{x}) = \frac{\sum_{i:i\in p\wedge \mathrm{F}(x_i)\geq m} \hat{\mathrm{P}}(y,x_i)\prod_{j\in c}\hat{\mathrm{P}}(x_j \mid y,x_i)}{|\{i : i\in p\wedge \mathrm{F}(x_i)\geq m\}|}$$

The performance of BSE is better than FSS, so we focus on BSE in this work. Four types of attribute elimination are considered, parent elimination (PE), child elimination (CE), parent and child elimination (P∧CE), parent or child elimination (P∨CE) which performs the former three types of attribute eliminations in each iteration, selecting the option that best reduces the error.

The last strategy allows flexible selection of parents and children, but comes at a high cost, since it needs to scan the training data 2a times in the worst case.

4.5. AnDE:

The last extension to AODE we review here is AnDE, which allows children to depend on not just one super-parent, but a combination of n parents. The joint probability P(y,x) is estimated as follows,

$$\hat{\mathrm{P}}(y,\mathbf{x}) = \frac{\sum_{s:s\in\binom{\mathcal{A}}{n}\wedge \mathrm{F}(x_s)\geq m} \hat{\mathrm{P}}(y,x_s)\prod_{j=1}^{a}\hat{\mathrm{P}}(x_j \mid y,x_s)}{|\{s : s\in\binom{\mathcal{A}}{n}\wedge \mathrm{F}(x_s)\geq m\}|}$$

where $\binom{\mathcal{A}}{n}$ indicates the set of all size-n subsets of {1,...,a} and x_s means the set of attribute values indexed by the element in s.

Note that AnDE is in fact a superclass of AODE and NB. That is, AODE is AnDE with n=1 (A1DE) and NB is AnDE with n = 0 (A0DE).

5. ATTRIBUTE SELECTIVE AODE

Previous work on attribute selection for AODE through BSE and FSS has demonstrated attribute selection did succeed in reducing the harmful influence of inter-dependencies among attributes.

This success may be attributed to their ability to search in a large model space. For P∨CE, the search space is of size 2a+1, as it includes all subsets of attributes in parent role coupled with all subsets of attributes in child role.

Nevertheless, this is achieved at a high computational overhead. The strategy of P∨CE needs to scan the training data 2a times, as each time either one child or one parent can be deleted. This is impractical for data sets with a large number of attributes.

In order to explore a large space of models in a single additional pass through the data, we propose a new attribute selection approach for AODE. Our proposal is based on the observation that it is possible to nest a large space of alternative models such that each is a trivial extension to another.

Let p and c be the set of indices of parent and child attributes, respectively. For every attribute xi, the AODE models that use attributes in p as parents and attributes in c∪{i} as children are minor extensions of a model that uses attributes in p as parents and attributes in c as children. The same is true of models that use attributes in p∪{i} as parents and attributes in c as children.

Importantly, multiple models that build upon one another in this way can be efficiently evaluated in a single set of computations. Using this observation, we create a space of models that are nested together, and then select the best model using leave-one-out cross validation in single extra pass through the training data.

Step by step information of the algorithm is provided in the following sections:

5.1. Ranking the Attributes:

Our method for nesting models depends on a ranking of the attributes. Models containing lower ranked attributes will be built upon models containing higher ranked attributes.

The mutual information between an attribute and the class measures how informative this attribute is about the class, and thus it is a suitable metric to rank the attributes.

The advantage of using mutual information is that it can be computed very efficiently after one pass through the training data.

Although the mutual information between an attribute and the class can help to identify the attributes that are individually most discriminative, it is important to note that it does not directly assess the discriminative power of an attribute in combination with other attributes.

Nevertheless, the ranking of attributes based on mutual information with the class will permit the search over a large space of possible models and the deficiencies of this discriminative approach will be mitigated by the richness of the search space that is evaluated in a discriminative fashion.

5.2. Building the Model Space:

Without loss of generality, in the following we assume that the attributes are ordered by mutual information. That is, xi represents the attribute with the ith greatest mutual information with the class.

As the attributes have been ranked, we can create, in total, a^2 nested submodels of attribute subsets. To be more specific, suppose we select top r attributes as parents and top s attributes as children, where $1 \leq r,s \leq a$, the candidate AODE model would be,

$$\hat{\mathrm{P}}(y, \mathbf{x})_{r,s} = \frac{\sum_{i:1 \leq i \leq r \wedge \mathrm{F}(x_i) \geq m} \hat{\mathrm{P}}(y, x_i) \prod_{j=1}^{s} \hat{\mathrm{P}}(x_j \mid y, x_i)}{|\{i : 1 \leq i \leq r \wedge \mathrm{F}(x_i) \geq m\}|}$$

Figure 1 gives an example of the model space with 3 attributes. For instance, model m21 considers the two attributes{x1,x2}as parents and a single attribute {x1}as a child. Then, when the attribute x2 is considered to be added as a child, we obtain a new model m22.

When instead the attribute x3 is considered to be added as a parent, we obtain a new model m31. Both of these models are minor extensions to the existing model m21 and all three (and all their extensions) can be applied to a test instance in a single nested computation. Consequently all models can be efficiently evaluated in a single set of nested computations.

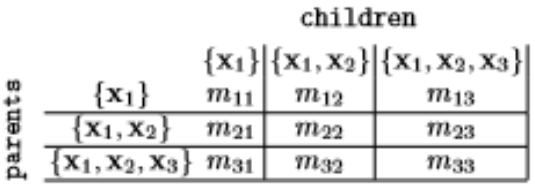

Fig. 1. An example of the model space with 3 attributes.

5.3. Selecting the Best Model:

Once we have built the model space, we can perform model selection within this space. To evaluate the goodness of an alternative model, an evaluation function is required, which commonly measures the discriminative ability of the model among classes.

We use leave-one-out cross validation error to measure the performance of each model. Rather than building a new model for every fold, we use incremental cross validation, in which the contribution of the training example being left out in each fold is simply subtracted from the count table, thus producing a model without that training example. This method allows the model to be evaluated quickly, whilst obtaining a good estimate of the generalization error.

There are several loss functions to measure model performance for leaveone-out cross validation, zero-one loss and root mean squared error (RMSE) are among the most common and effective. Zero-one loss simply assigns a loss of '0' to correct classification, and '1' to incorrect classification, treating all misclassifications as equally undesirable. RMSE, however, accumulates for each example the squared error, which is the probability of incorrectly classifying the example, and then computes the root mean of the sum. As RMSE gives a finer grained measure of the calibration of the probability estimates compared to zero-one loss, with the error depending not just on which class is predicted, but also on the probabilities estimated for each class, we use RMSE to evaluate the candidate models in this research.

5.4. Algorithm:

Based on the methodology presented above, we develop the training algorithm for attribute selective AODE shown in Algorithm 1.

Algorithm 1: Training algorithm for attribute selective AODE.
1: Form the table of joint frequencies of pairwise attribute-values and class
2: Compute the mutual information
3: Rank the attributes
4: **for all** example in T **do**
5: Build all a^2 models while leaving the current example out
6: Predict the current example using a^2 models
7: Accumulate the squared error for each model
8: **end for**
9: Compute the root mean squared error for each model
10: Select the model with the lowest RMSE

As in AODE, we need to form the table of joint frequencies of pairs of attribute-values and class from which the probability estimates $P^{\wedge}(y,xi)$, $P^{\wedge}(xj\ |y,x\ i)$ and the mutual information between the attributes and class are derived.

This is done in one pass through the training data (line 1). Note that this provides all of the information needed to create any selective AODE model with any sets of parent and child attributes.

In the second pass through the training data (line 4-8), the squared error is accumulated for each model. After this pass, the RMSE will be computed and used to select the best model.

At training time, the space complexity of the table of joint frequencies of attribute-values and class is $O(k(av)^2)$ as in AODE, where v is the average number of values per attribute. Attribute selection will not require more memory.

Derivation of the frequencies required to populate this table is of time complexity $O(ta^2)$. Attribute selection needs one more pass through the training data, the time complexity of which is $O(tka^2)$, since for each example we need to compute the joint probability in (1) for each class. So the overall time complexity is $O(t(k + 1)a^2)$.

Classification requires the table of probability estimates formed at training time of space complexity $O(k(av)^2)$. The time complexity of classifying a single example is $O(ka^2)$ in the worst-case scenario, because some attributes may be omitted after attribute selection.

6. EXPERIMENTAL ANALYSIS

6.1. Metrics to be Calculated:

We use leave-one-out cross validation error to measure the performance of each model. Rather than building a new model for every fold, we use incremental cross validation, in which the contribution of the training example being left out in each fold is simply subtracted from the count table, thus producing a model without that training example. This method allows the model to be evaluated quickly, whilst obtaining a good estimate of the generalization error.

There are several loss functions to measure model performance for leave one-out cross validation, zero-one loss and root mean squared error (RMSE) are among the most common and effective.

Zero-one loss simply assigns a loss of ‘0’ to correct classification, and ‘1’ to incorrect classification, treating all misclassifications as equally undesirable.

RMSE, however, accumulates for each example the squared error, which is the probability of incorrectly classifying the example, and then computes the root mean of the sum.

As RMSE gives a fine grained measure of the calibration of the probability estimates compared to zero-one loss, with the error depending not just on which class is predicted, but also on the probabilities estimated for each class, we use RMSE to evaluate the candidate models in this research.

6.2. Data sets Used:

We run the above algorithms on 71 data sets from the UCI repository. Table 1 presents the detailed characteristics of data sets in ascending order on the number of instances. We run the experiments on a single CPU single core virtual Linux machine running on a Sun grid node with dual 6 core Intel Xeon L5640 processors running at 2.27 GHz with 96 GB RAM.

Table 1. Data sets.

No.	Name	Inst	Att	Class
1	contact-lenses	24	4	3
2	lung-cancer	32	56	3
3	labor-negotiations	57	16	2
4	post-operative	90	8	3
5	zoo	101	16	7
6	promoters	106	57	2
7	echocardiogram	131	6	2
8	lymphography	148	18	4
9	iris	150	4	3
10	teaching-ae	151	5	3
11	hepatitis	155	19	2
12	wine	178	13	3
13	autos	205	25	7
14	sonar	208	60	2
15	glass-id	214	9	3
16	new-thyroid	215	5	3
17	audio	226	69	24
18	hungarian	294	13	2
19	heart-disease-c	303	13	2
20	haberman	306	3	2
21	primary-tumor	339	17	22
22	ionosphere	351	34	2
23	dermatology	366	34	6
24	horse-colic	368	21	2
25	house-votes-84	435	16	2
26	cylinder-bands	540	39	2
27	chess	551	39	2
28	syncon	600	60	6
29	balance-scale	625	4	3
30	soybean	683	35	19
31	credit-a	690	15	2
32	breast-cancer-w	699	9	2
33	pima-ind-diabetes	768	8	2
34	vehicle	846	18	4
35	anneal	898	38	6
36	tic-tac-toe	958	9	2
37	vowel	990	13	11
38	german	1000	20	2
39	led	1000	7	10
40	contraceptive-mc	1473	9	3
41	yeast	1484	8	10
42	volcanoes	1520	3	4
43	car	1728	6	4
44	segment	2310	19	7
45	hypothyroid	3163	25	2
46	splice-c4.5	3177	60	3
47	kr-vs-kp	3196	36	2
48	abalone	4177	8	3
49	spambase	4601	57	2
50	phoneme	5438	7	50
51	wall-following	5456	24	4
52	page-blocks	5473	10	5
53	optdigits	5620	64	10
54	satellite	6435	36	6
55	musk2	6598	166	2
56	mushrooms	8124	22	2
57	thyroid	9169	29	20
58	pendigits	10992	16	10
59	sign	12546	8	3
60	nursery	12960	8	5
61	magic	19020	10	2
62	letter-recog	20000	16	26
63	adult	48842	14	2
64	shuttle	58000	9	7
65	connect-4	67557	42	3
66	ipums.la.99	88443	60	19
67	waveform	100000	21	3
68	localization	164860	5	11
69	census-income	299285	41	2
70	poker-hand	1025010	10	10
71	record-linkage	5749132	11	2

6.3. Methods to be Compared:

Because ASAODE explores a larger space of models than AODE and BSEAODE explores a larger space of models than ASAODE, we expect BSEAODE to have the lowest bias, followed by ASAODE then AODE and this order to be reversed for their relative variance. Hence we expect AODE to deliver the lowest error on smaller datasets, ASAODE to dominate at some intermediate data size, and for BSEAODE to deliver the lowest error on very large data. The bias and variance of ASAODE relative to WAODE, AODESR and A2DE can be expected to vary from dataset to dataset as these all embody different learning biases and none of their spaces of models subsumes the other.

In order to assess these expectations, we first perform bias variance decomposition using the experimental method proposed. As this study is more meaningful with more data, we run these experiments only on the largest 28 data sets which have at least 2000 examples. For each data set, 1000 training examples and 1000 test examples are randomly selected. The bias variance decomposition is calculated from the error on the test examples. This process is repeated 10 times to obtain the mean bias and variance.

Each entry in cell [i,j] compares the algorithm in row i against the algorithm in column j. The p value following each win/draw/loss record is the outcome of a binomial sign test and represents the probability of observing the given number of wins and losses if each were equally likely. The reported p value is the result of a two-tailed test. We consider a difference to be significant if $p \leq 0.05$. All such p values have been changed to boldface in the table.

6.4. Analysis:

The study shows that all five variants to AODE achieve significant reductions in bias relative to AODE. While ASAODE achieves lower bias than WAODE and AODESR more often than not, the reverse is true for BSEAODE and A2DE; although these differences are not significant.

Next, we conduct 10-fold cross validation experiments to obtain the error of the alternative algorithms. As attribute selection is based on the RMSE metric, we are inclined to evaluate the error by RMSE. The win/draw/loss records of alternative algorithms for RMSE on 71 data sets are to be tabulated.

From the study all five improvements to AODE have achieved significant reductions in RMSE relative to AODE. ASAODE has also achieved significant reductions in RMSE relative to WAODE and AODESR.

But the advantages of BSEAODE over WAODE and AODESR are not as significant as those of ASAODE over WAODE and AODESR. While A2DE achieves significant reductions in RMSE relative to AODE, WAODE, AODESR and BSEAODE, its advantage over ASAODE is not significant.

The fact that ASAODE obtains, in general, lower bias and higher variance compared with WAODE and AODESR, indicates that it will perform better on larger datasets, since it will be able to capture more complex relationships from large amount of data. In order to demonstrate this hypothesis, we also compile the win/draw/loss results in terms of RMSE on the 43 smallest data sets and the 28 largest data sets.

We can see that the performance of ASAODE is better on large data sets than on small data sets. While for even larger data sets BSEAODE and A2DE might outperform ASAODE for the same reason, both have high computational complexity that can be prohibitive for large data, since BSEAODE requires 2a passes on the whole training set and A2DE's memory requirements and classification time are very high.

6.5.1. Error, Bias and Variance:

For each algorithm the mean of each measure across all data sets is also presented. The mean error, bias or variance across multiple data sets provides at best a very gross measure of relative performance as it is questionable whether error rates are commensurable across data sets. The geometric mean ratio is also presented.

This is a standardized measure of relative performance. This is obtained by taking for each data set the ratio of the performance of the alternative algorithm divided by the performance of AODE. The geometric mean of these ratios is presented as this is the most appropriate average to apply to ratio data. A geometric mean ratio greater than 1.0 represents an advantage to AODE and a value lower than 1.0 represents an advantage to the alternative algorithm.

We do not apply significance tests to pairwise comparisons of performance on a data set by data set basis, as the 888 (37 data sets × 3 metrics × 8 comparator algorithms) such comparisons would result in substantial risk of a large number of false positive outcomes. Nor do we present the standard deviations of the individual error outcomes as the number of outcomes makes interpretation of such information infeasible.

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

ODE, and bagged ODE were included in the experiments in order to evaluate the interpretation of the power of AODE in terms of ensembling one-dependence classifiers. Comparing ODE first to NB, ODE has lower mean error and bias but higher mean variance. The win/draw/loss records of NB compared to ODE show that the advantage is not significant for error but is significant for bias and variance. Bagging ODE can be seen to bring the error, bias and variance toward that of AODE, lending credibility to an explanation of the effectiveness of AODE in terms of ensembling one-dependence estimators.

6.5.2. Learning Curves:

Cross data set experimental studies of the traditional form presented above are of only limited value for gaining deep understanding of the relative prediction characteristics of alternative algorithms. Demonstrating a significant benefit for one algorithm across a group of data sets provides evidence only that the algorithm is likely to perform better with respect to subsequent data sets with similar characteristics.

Unfortunately, however, the science of machine learning has made little progress in identifying characteristics that are likely to affect relative classification performance, and hence we have limited ability to generalize from results on one group of data sets to expected results on further data. One proposal that has been made is that data set size interacts with the bias-variance characteristics of an algorithm to affect prediction performance.

The descriptors ‘small’ and ‘large’ here are clearly imprecise and impossible to exactly quantify as the rate at which bias comes to dominate error will depend upon the complexity of each classification task. Nonetheless, this framework does provide us with a precise expectation, that for two algorithms one with lower bias and the other with lower variance, the lower variance algorithm will exhibit lower error at very small data set sizes and that learning curves for the algorithms will eventually cross so that at some larger data set size the low bias algorithm will achieve lower error.

The experiments reported above suggest that AODE, LBR, TAN and SP-TAN all share similar levels of bias. However, as already noted, the data sets are primarily small and the bias-variance evaluation procedure utilizes training sets containing only 25% of each data set. Hence many of the training sets are quite small. We expect the bias of LBR, TAN and SP-TAN to decrease as training set sizes increase as more data will lead to more accurate probability estimates and hence to more appropriate model selection.

As NB and AODE do not perform model selection we do not expect their bias to decrease with increased data in the same manner. In an attempt to assess these predictions we recalculated the mean, geometric mean ratio and win/draw/loss records of NB, LBR, TAN and SP-TAN relative to AODE over the ten largest data sets (those with 1000 or more cases and hence for which the training sets contained 250 or more cases). The mean and geometric mean ratios are presented in Table VII and the win/draw/loss records are presented in Table VIII.

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

Note that the Weka bias-variance evaluation method results in the use of test sets that are twice the size of the training sets, and hence that the test time is greatly amplified compared with most alternative evaluation methods.

The first row of these tables presents the mean time across all data sets. The next row presents the geometric mean across all data sets of the ratio obtained by dividing the training or test time on a data set for the alternative algorithm by that of AODE. A value less than 1.0 indicates that AODE tends to be slower than the alternative while a value greater than 1.0 indicates that AODE tends to be faster.

The next row presents the number of data sets for which AODE obtained lower compute time than the alternative algorithm and the final row the number of data sets for which the time for AODE was higher.

The final row presents the outcome of a two-tailed binomial sign test presenting the probability that the observed or more extreme record of wins and losses would be obtained if wins and losses were equiprobable.

6.5.3. Computation Time:

The logarithmic means of training and classification time on the 71 data sets for all algorithms are to be plotted We have added 1 to each mean before computing the logarithm to avoid negative bars. ASAODE requires more training time than such one pass algorithms as AODE, WAODE and AODESR. This is because ASAODE involves two passes through the training data. As BSEAODE needs at most 2a passes, it requires significantly more training time than ASAODE.

As for the classification time, ASAODE, AODESR and BSEAODE require, in general, less time than AODE and WAODE because they might eliminate some attributes. Fig. 3 also shows that ASAODE requires even less classification time than AODESR and BSEAODE.

A2DE requires more training and classification time than AODE, as it needs to compile a more complicated table at training time and requires more computation at classification time.

7. CONCLUSION AND FUTURE WORK

In this work, a new attribute selection algorithm is proposed for AODE. It is a two-pass algorithm, so compared to AODE, it just requires one more pass through the training data. The alternative attribute selection methods, such as FSA and BSE, need a number of passes that is linear to the number of attributes to obtain similar results.

The empirical results show that the new algorithm is significantly more accurate than AODE, WAODE and AODESR, has comparable error to BSEAODE, and as we expected, worse than A2DE. It requires significantly less training time than BSEAODE, and less classification time than AODE and all other variants, especially than A2DE.

It is worthwhile to note that the technique proposed in this work is of squared complexity in the number of attributes, so it is not scalable to high dimensional data. On the other hand, it is compatible with weighting, subsumption resolution and higher orders of AnDE.

Consequently, it might be possible to further improve the accuracy by combining it with weighting, subsumption resolution and A2DE. This is a promising direction for future research.

REFERENCES

[1] Duda, R.O., Hart, P.E.: Pattern Classification and Scene Analysis. 1 edn. John Wiley & Sons Inc (1973)

[2] Webb, G.I., Boughton, J.R., Wang, Z.: Not so naive Bayes: Aggregating onedependence estimators. Machine Learning 58(1) (2005) 5–24

[3] Zheng, F., Webb, G.I.: A comparative study of semi-naive Bayes methods in classification learning. In: AusDM. (2005) 141–156

[4] Yang, Y., Webb, G.I., Cerquides, J., Korb, K.B., Boughton, J., Ting, K.M.: To select or to weigh: a comparative study of linear combination schemes for superparent-one-dependence estimators. IEEE Transactions on Knowledge and Data Engineering 19(12) (2007) 1652–1665

[5] Zheng, F., Webb, G.I.: Finding the right family: parent and child selection for averaged one-dependence estimators. In Kok, J.N., Koronacki, J., de Mantaras, R.L., Matwin, S., Mladeni, D., Skowron, A., eds.: ECML. Springer (2007) 490–501

[6] Webb, G.I., Boughton, J.R., Zheng, F., Ting, K.M., Salem, H.: Learning by extrapolation from marginal to full-multivariate probability distributions: decreasingly naive Bayesian classification. Machine Learning 86(2) (2012) 233–272

[7] Cerquides, J., de M´antaras, R.L.: Robust Bayesian linear classifier ensembles. In Gama, J., Camacho, R., Brazdil, P.B., Jorge, A.M., Torgo, L., eds.: ECML, Springer (2005) 72–83

[8] Jiang, L., Zhang, H.: Weightily averaged one-dependence estimators. In Yang, Q., Webb, G.I., eds.: PRICAI: trends in artificial intelligence. Springer (2006) 970–974

[9] Zheng, F., Webb, G.I., Suraweera, P., Zhu, L.: Subsumption resolution: an efficient and effective technique for semi-naive Bayesian learning. Machine Learning 87(1) (2012) 93–125

[10] Langley, P., Sage, S.: Induction of selective Bayesian classifiers. In: Proceedings of the tenth international conference on uncertainty in artificial intelligence, Morgan Kaufmann Publishers Inc. (1994) 399–406

[11] Kittler, J.: Feature selection and extraction. Handbook of pattern recognition and image processing (1986) 59–83

[12] MacKay, D.J.: Information theory, inference and learning algorithms. Cambridge university press (2003)

[13] Kohavi, R.: The power of decision tables. In Lavrac, N., Wrobel, S., eds.: ECML, Springer (1995) 174–189

[14] Fayyad, U.M., Irani, K.B.: Multi-interval discretization of continuous-valued attributes for classification learning. In: IJCAI. (1993) 1022–1027

[15] Cestnik, B.: Estimating probabilities: a crucial task in machine learning. In: ECAI. Volume 90. (1990) 147–149

[16] Bache, K., Lichman, M.: UCI machine learning repository (2013)

[17] Kohavi, R., Wolpert, D.H.: Bias plus variance decomposition for zero-one loss functions. In: ICML. (1996) 275–283

[18] Brain, D., Webb, G.I.: The need for low bias algorithms in classification learning from large data sets. In Elomaa, T., Mannila, H., Toivonen, H., eds.: Principles of Data Mining and Knowledge Discovery. Springer (2002) 62–73