GAODE and HAODE: Two Proposals based on AODE to Deal with Continuous Variables

M. Julia Flores JULIA@DSI.UCLM.ES
José A. Gámez JGAMEZ@DSI.UCLM.ES
Ana M. Martínez ANAMARTINEZ@DSI.UCLM.ES
José M. Puerta JPUERTA@DSI.UCLM.ES
Computing Systems Department, Intelligent Systems and Data Mining group, i3A
Campus Universitario s/n 02071, Albacete, SPAIN

Abstract

AODE (Aggregating One-Dependence Estimators) is considered one of the most interesting representatives of the Bayesian classifiers, taking into account not only the low error rate it provides but also its efficiency. Until now, all the attributes in a dataset have had to be nominal to build an AODE classifier or they have had to be previously discretized. In this paper, we propose two different approaches in order to deal directly with numeric attributes. One of them uses conditional Gaussian networks to model a dataset exclusively with numeric attributes; and the other one keeps the superparent on each model discrete and uses univariate Gaussians to estimate the probabilities for the numeric attributes and multinomial distributions for the categorical ones, it also being able to model hybrid datasets. Both of them obtain competitive results compared to AODE, the latter in particular being a very attractive alternative to AODE in numeric datasets.

1. Introduction

Supervised classification is a very common task, not only in machine learning applications but also in many aspects of daily life, such as spam detection in mail, recommendations for a specific product according to previous purchases, determination of the folder for incoming e-mail, etc. Bayesian network classifiers (Langley et al., 1992) offer significant advantages over other approaches for classification tasks. They are able to deal naturally with uncertainty, and estimate not only the label assigned to every object but also the probability distribution over the different labels of the class variable.

NB (Duda et al., 1973) is the simplest Bayesian classifier and one of the most efficient inductive algorithms for machine learning (Wu et al., 2007). Despite the strong independence assumption between predictive attributes given the class value, it provides a surprisingly high level of accuracy, even compared to other more sophisticated models (Domingos & Pazzani, 1996). However, in many real applications it is not only necessary to be accurate in the classification task, but also to produce a ranking as precise as possible with the probabilities of the different class values.

During the last few years, attention has focused on developing NB variants in order to alleviate the independence assumption between attributes, and so far AODE has come out as the most attractive option due to its capability for improving NB's accuracy with only a slight increase in time complexity (from $\mathcal{O}(n)$ to $\mathcal{O}(n^2)$), where n is the number of attributes. An extensive study comparing different semi-naive Bayes techniques (Zheng & Webb, 2005) proves that AODE is significantly better in terms of error reduction compared to the rest of semi-naive techniques, with the exception of Lazy Bayesian Rules (LBR) (Zheng & Webb, 2000) and Super-Parent TAN (SP-TAN)(Keogh & Pazzani, 1999), which obtain similar results but with a higher time complexity.

Bayesian networks (BNs) in general assume all random variables are multinomial. Most of the algorithms and procedures designed for Bayesian classifiers are only able to handle discrete variables, and when a numeric variable is present, it must be discretized. In (Pérez

Appearing in *Proceedings of the 26th International Conference on Machine Learning*, Montreal, Canada, 2009.

et al., 2006), wrapper and filter approaches are designed to adapt four well-known paradigms of discrete classifiers for handling continuous variables (namely NB, Tree-augmented NB, k-dependence Bayesian classifier and semi NB). So far, the only way of training an AODE classifier with a dataset containing numeric attributes has been to discretize this dataset before building the model, which can be a handicap in many situations as this process, by definition, entails an inherent loss of information.

Nevertheless, when numerical variables are considered, the problem arises of how to model the probability distribution for a variable conditioned, not only by the class (which is discrete), but also by another numeric attribute. Gaussian networks (Geiger & Heckerman, 1994) have been proposed as a good alternative to the direct discretization of continuous attributes. A Gaussian network is similar to a BN, but it assumes all attributes are sampled from a Gaussian density distribution, instead of a multinomial distribution. Despite this strong assumption, Gaussian distributions usually provide reasonable approximations to many real-world distributions.

In our study, two approaches have been developed to handle continuous variables in AODE: GAODE and HAODE, which both inherit the same structure as AODE. In the first one, we make use of CGNs to model the relationship between a numeric attribute conditioned to a discrete class and another numeric attribute; hence, it is restricted to numerical datasets. In the second one, a discrete version of the superparent attribute is considered in every model, so the previous relationship can be estimated by a univariate Gaussian distribution. The latter approach applies multinomials for nominal children, being able to deal directly with all kinds of datasets.

This paper is organized as follows: sections 2 and 3 present an overview of AODE and CGNs respectively, which are the main basis of the new classifiers developed. Section 4 provides a detailed explanation of the two algorithms designed. In section 5, we describe the experimental setup and results. And finally, section 6 summarizes the main conclusions of our paper and outlines the future work related with this study.

2. Averaged One-Dependence Estimators

The AODE classifier (Webb et al., 2005) is considered an improvement on NB and a good alternative to other attempts such as LBR and SP-TAN, as they offer similar accuracy ratios, but AODE is significantly more efficient at classification time compared to the first one and at training time compared to the second.

Back in 1996, Sahami (Sahami, 1996) introduced the notion of k-dependence estimators, through which the probability of each attribute value is conditioned by the class and, at most, k other attributes. In order to maintain efficiency, AODE is restricted to exclusively use 1-dependence estimators (ODEs). Specifically, AODE makes use of SPODEs, as every attribute depends on the class and another shared attribute, designated as superparent.

Considering the MAP (maximum a posteriori) hypothesis, the classification of a single example in a SPODE classifier (and hence, a 1-dependence ODE) is defined in equation 1:

$$c_{MAP} = argmax_{c \in \Omega_C}\ p(c, a_j) \prod_{i=1, i \neq j}^{n} p(a_i | c, a_j) \quad (1)$$

where C represents the class variable, Ω_C the set of class labels, A_j the superparent attribute and $\{a_1, a_2, \cdots, a_n\}$ the instance to be classified.

The BN corresponding to an SPODE classifier is depicted in figure 1. In order to avoid selection between models or, in other words, trying to take advantage of all the created ones, AODE averages the n SPODE classifiers with every different attribute as superparent, as is shown in equation 2:

$$argmax_{c \in \Omega_C} \left(\sum_{j=1, N(a_j) > m}^{n} p(c, a_j) \prod_{i=1, i \neq j}^{n} p(a_i | c, a_j) \right) \quad (2)$$

The $N(a_j) > m$ condition is used as a threshold to avoid predictions from attributes with insufficient samples. If there is not any value which exceeds this threshold, the results are equivalent to NB.

At *training time*, AODE has a $\mathcal{O}(tn^2)$ time complexity, where t is the number of training examples; whereas

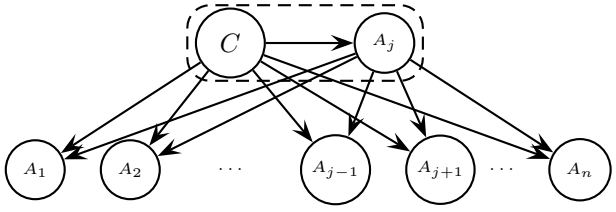

Figure 1. Generalized structure of SPODE classifiers.

the space complexity is $\mathcal{O}(k(nv)^2)$, where v is the average number of values per attribute and k the number of classes. The resulting time complexity at *classification time* is $\mathcal{O}(kn^2)$, while the space complexity is $\mathcal{O}(k(nv)^2)$.

3. Conditional Gaussian Networks

Continuous nodes in a BN can be modeled by a Gaussian distribution function, also called Normal distribution. Any Gaussian distribution may be defined by two parameters, location and scale: the mean ("average", μ) and variance (standard deviation squared, σ^2) respectively. Likewise, every continuous node can have a Gaussian distribution for every configuration of its discrete parents. If a continuous node has one or more continuous nodes as parents, the mean can be linearly dependent over the states of these continuous parents. This is the basic idea underlying CGNs (Lauritzen & Jensen, 2001). Note that discrete nodes are not allowed to have continuous parents though.

In this case, a parametrical learning process is carried out, where the estimation of the parameters is made from data. These parameters are modeled by the dependency relationships between variables, represented by the structure of the corresponding classifier or BN. A noteworthy property of CGNs is that they offer a frame where exactitude in inference is guaranteed. Another advantage of Gaussian networks is that they only need $\mathcal{O}(n^2)$ parameters to model a complete graph.

In general, every node stores a local density function (linear regression model) where the distribution for a continuous variable X with discrete parents $\boldsymbol{Y}$ and continuous parents $\mathbf{Z} = \{Z_1, \ldots, Z_s\}$ (with s the number of continuous parents) is a one-dimensional Gaussian distribution over the states of its parents (DeGroot, 1970):

$$f(X|\mathbf{Y} = y, \mathbf{Z} = z; \Theta) =$$

$$= \mathcal{N}(x : \mu_X(y) + \sum_{j=1}^{s} b_{XZ_j}(y)(z_j - \mu_{Z_j}(y)), \sigma^2_{X|\mathbf{Z}}(y)) \tag{3}$$

where:

- $\mu_X(y)$ is the mean of X with the configuration $Y = y$ of its discrete parents.
- $\mu_{Z_j}(y)$ is the mean of Z_j with the configuration $Y = y$ of its discrete parents.
- $\sigma^2_{X|\mathbf{Z}}(y)$ is the conditional variance of X over its continuous parents $\mathbf{Z}$ and also according to the configuration $Y = y$ of its discrete parents .
- $b_{XZ_j}(y)$ is a regression term that individually measures the strength of the connection between X and every continuous parent (it will be equal to 0 if there is not an edge between them).

The local parameters are given by $\Theta = (\mu_X(y), b_X(y), \sigma^2_{X|\mathbf{Z}}(y))$, where $b_X(y) = (b_{XZ_1}(y), \ldots, b_{XZ_{s(y)}})^t$ is a column vector.

Then, if we focus on the bivariate case, where the X variable is only conditioned by one continuous variable Z and the discrete variables mentioned, the conditional variance and the regression term would be easily obtained as shown in equations 4 and 5 [1]:

$$\sigma^2_{X|Z}(y) = \sigma^2_X(y) - b^2_{XZ}(y)\sigma^2_Z(y) \tag{4}$$

$$b_{XZ}(y) = \frac{\sigma_{XZ}(y)}{\sigma^2_Z(y)} \tag{5}$$

Figure 2 shows an example of factorization of the density function in a SPODE structure, as in the model depicted on the left. Following the former notation, in this case $b_X(y) = b_{XZ}(y)$ as there is just one continuous variable.

Equation 3 has been obtained following the guidelines in (Larrañaga et al., 1999) and (Neapolitan, 2003). However, in the Hugin tool (Andersen et al., 1989), the estimation of the Gaussian distribution of interest, is carried out in a slightly different way. In the estimation of the final mean for the CGN, Hugin does not take into account the means of the continuous parents, and the variance is constant for every state's configuration of the discrete parents. Hence, the corresponding equation according to Hugin principles would be as follows:

$$f(X|\mathbf{Y} = y, \mathbf{Z} = z; \Theta) =$$

$$= \mathcal{N}(x : \mu_X(y) + \sum_{j=1}^{s} b_{XZ_j}(y)z_j, \sigma^2_X(y)) \tag{6}$$

On the other hand, in (Pérez et al., 2006) the authors consider equation 3 but the variance for the CGN is constant for every state's configuration of the discrete parents.

[1] The estimate in equations 4 and 5 has been obtained by working out the value of $\sigma^2_{X|Z}(y)$ and $b_{XZ}(y)$ when the inverse of the precision matrix (W^{-1}) and the covariance matrix (Σ) from the Gaussian network are matched:

$$W^{-1} = \begin{pmatrix} \sigma^2_Z(y) & b_{XZ}(y)\sigma^2_Z(y) \\ b_{XZ}(y)\sigma^2_Z(y) & \sigma^2_{X|Z}(y) + b^2_{XZ}(y)\sigma^2_Z(y) \end{pmatrix} =$$

$$= \begin{pmatrix} \sigma^2_Z(y) & \sigma_{XZ}(y) \\ \sigma_{XZ}(y) & \sigma^2_X(y) \end{pmatrix} = \Sigma$$

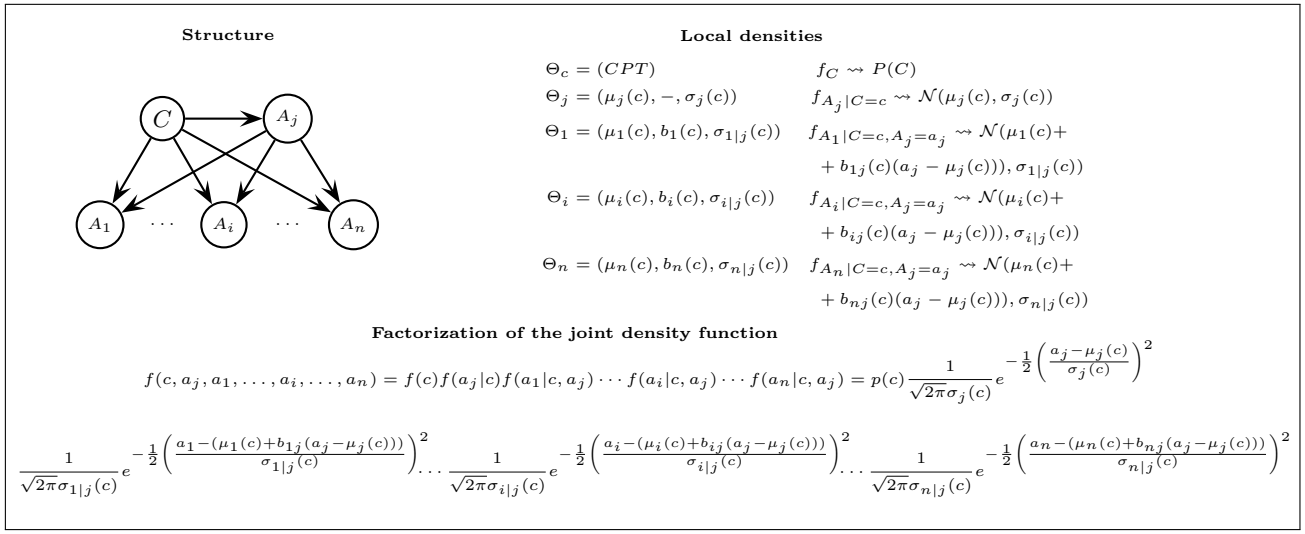

Figure 2. Structure, local densities and result from the factorization of the joint density function in a network with the SPODE structure where all the predictive attributes are continuous.

4. New proposals for the treatment of continuous attributes with AODE

4.1. Gaussian AODE (GAODE) classifier

The underlying idea of this classifier consists in using CGNs to deal with continuous attributes in AODE. In fact, as the class variable is discrete, if we restrict all the predictive attributes to be continuous, we can make use of the Bayes rule to combine Bayesian and Gaussian networks to encode the joint probability distributions among the domain variables, based on the conditional independencies defined by AODE.

In this particular case of AODE, the density function for every predictive attribute has to be estimated over a node with a single discrete parent, that is, the class C and another continuous parent, which is the superparent attribute in every model, A_j. The adaptation of the conditional Gaussian (CG) density function in equation 3 to this case is:

$$f(A_i = a_i|C = c, A_j = a_j) = \\ = \mathcal{N}(a_i : \mu_i(c) + b_{ij}(c)(a_j - \mu_j(c)), \sigma^2_{i|j}(c)) \quad (7)$$

The Bayesian structure for GAODE would remain the same as AODE, and its MAP hypothesis is obtained when we replace the multinomial probability distributions in equation 2 on page 2 with the corresponding CG distribution function defined in equation 7. Whereas the relationship between every predictive attribute conditioned to the class and the corresponding superparent is modeled by a CG distribution, the relationship between every superparent and the class is modeled by a univariate Gaussian distribution. Hence, assuming all the predictive variables are continuous, GAODE selects the class label which maximizes the following summation:

$$argmax_c\left(\sum_{j=1}^{n}\mathcal{N}(a_j : \mu_j(c), \sigma^2_j(c))p(c)\cdot \right. \\ \left. \prod_{i=1\wedge i\neq j}^{n}\mathcal{N}(a_i : \mu_i(c) + b_{ij}(c)(a_j - \mu_j(c)), \sigma^2_{i|j}(c))\right) \quad (8)$$

As we can deduce from our definition of this classifier with the application of CGNs, it is not possible to define the corresponding probability function for a discrete variable conditioned to a numeric attribute. As in AODE, all the attributes play the superparent role in one model, none of the children attributes are allowed to be discrete and therefore, GAODE is only defined to deal with datasets exclusively formed by numeric attributes (plus, of course, the discrete class).

In this case, the space complexity at *training* and *classification time* becomes independent of the number of values per attribute v, and equals $\mathcal{O}(kn^2)$. Furthermore, as the number of necessary parameters is independent of v, the probabilities estimated can be more reliable compared to the multinomial version as they are modeled from more samples, specially when the size of the Conditional Probability Tables (CPTs) is very large.

The time complexity undergo no variation as the parameters of the different Gaussian and CG distributions can be computed incrementally.

4.2. Hybrid AODE (HAODE) classifier

As we have seen, the GAODE classifier defined is only able to deal with datasets which exclusively contain continuous attributes. In order to include the possibility of handling all kind of datasets, we decided to consider every superparent as discrete in its corresponding model, in principle, by means of any discretization method. However, only the superparent will be discretized, for the rest of attributes their numeric value will be considered. In this way, there is no need to resort to CGNs, as all the parents in the network will be discrete, but at the same time, we keep most of the original precision from the numeric data.

This can also be seen as an even more simple way of solving the problem of dealing with the continuous superparents on each model in AODE, as there is no need to use CGNs, but only univariate Gaussian distributions.

Hence, equation 2 is developed in the following way:

$$argmax_c\left(\sum_{j=1,N(a_j)>m}^{n} p(a_j,c)\prod_{i=1\wedge i\neq j}^{n}\mathcal{N}(a_i:\mu_i(c,a_j),\sigma_i^2(c,a_j))\right) \tag{9}$$

This means the relationship between the superparent and the class is modeled with a multinomial probability distribution, whereas the rest of relationships, where every other attribute is conditioned to the class and the superparent, are modeled by normal Gaussian distributions, as long as they are continuous.

As we have pointed out above, this new classifier offers the additional advantage of dealing with any kind of dataset, including the ones that contain a mixture of discrete and continuous variables. In those cases where the child attribute is discrete, a multinomial distribution will be used, as in AODE. This feature represents a significant advantage with respect to the use of CGNs proposed in the previous section, as well as an evident simplification in the calculation of parameters.

The models constructed are still 1-dependent, which is why the CPTs required to store the different probability distributions, when necessary for HAODE, are still three-dimensional, as in AODE. In this case, space complexity will increase with the number of discrete variables in the dataset, the top level being the same as for AODE ($\mathcal{O}(k(nv)^2)$).

As in AODE, in both classifiers the model selection between the n SPODE models is unnecessary, thus avoiding the computational cost required by this task and hence maintaining AODE's efficiency and minimizing the variability in the error obtained.

5. Experimental Methodology and Results

5.1. Datasets with only continuous attributes

In order to evaluate the performance of the two classifiers developed, we have carried out experiments over a total of 26 numeric datasets, downloaded from the University of Waikato homepage (Wek08, 2008). We gathered together all the datasets on this web page, originally from the UCI repository (Asuncion & Newman, 2007), which are aimed at classification problems and exclusively contain numeric attributes according to Weka (Witten & Frank, 2005). Table 1 displays these datasets and their main characteristics.

Table 1. Main characteristics of the datasets: number of predictive variables (n), number of classes (k) and number of instances (I).

Id	Datasets	n	k	I	Id	Datasets	n	k	I
1	balance-scale	4	3	625	14	mfeat-fourier	76	10	2000
2	breast-w	9	2	699	15	mfeat-karh	64	10	2000
3	diabetes	8	2	768	16	mfeat-morph	6	10	2000
4	ecoli	7	8	336	17	mfeat-zernike	47	10	2000
5	glass	9	7	214	18	optdigits	64	9	5620
6	hayes-roth	4	4	160	19	page-blocks	10	5	5473
7	heart-statlog	13	2	270	20	pendigits	16	9	10992
8	ionosphere	34	2	351	21	segment	19	7	2310
9	iris	4	3	150	22	sonar	60	2	208
10	kdd-JapanV	14	9	9961	23	spambase	57	2	4601
11	letter	16	26	20000	24	vehicle	18	4	946
12	liver-disorders	6	2	345	25	waveform-5000	40	3	5000
13	mfeat-factors	216	10	2000	26	wine	13	3	178

Table 2 shows the accuracy results obtained when using a 5x2 cross validation (cv) to evaluate the different classifiers. Each value represents the arithmetical mean from the 10 executions. The bullet next to certain outputs means the corresponding classifier on this particular dataset either obtains the highest accuracy or is not significantly worse than the classifier which does. The results were compared using the 5x2 cv F Test defined by (Alpaydin, 1999), which has lower type I error and higher power than the 5x2 cv t-test. The level of significance was fixed at 95% ($\alpha = 0.05$). The 5x2 cv F Test is more conservative than the 5x2 cv t-test, so a higher number of ties will be obtained with the same level of significance.

Besides GAODE and HAODE, 3 other classifiers were included in the comparison. From left to right: NB classifier with Gaussian distributions to deal with continuous attributes (NB-G); and NB and AODE classifiers with the datasets previously discretized (NB and AODE) using Fayyad and Irani's MDL method (Fayyad & Irani, 1993). The corresponding discretization of the superparent attributes in HAODE was also

Table 2. Accuracy results obtained for NB with Gaussians (NB-G), NB, AODE ($m = 1$), GAODE and HAODE ($m = 1$) in continuous datasets.

Id	NB-G	NB	AODE	**GAODE**	**HAODE**
1	•88, 864	77, 632	76, 992	•89, 088	•87, 68
2	•96, 0801	•97, 1102	•96, 6237	•95, 9662	•95, 0787
3	•74, 974	•74, 6875	•74, 5573	•74, 7917	•75, 9115
4	•83, 9881	•80, 7738	•81, 0119	•84, 5238	•84, 3452
5	49, 7196	•60	•60, 7477	•52, 8037	•60, 6542
6	•65, 375	•57, 5	•57, 5	•65, 625	•68, 5
7	•83, 4815	•81, 2593	•80, 8148	•83, 7778	•83, 037
8	•82, 963	•88, 8889	•90, 7123	•92, 0228	•91, 7379
9	•95, 0667	•93, 4667	•93, 3333	•97, 4667	•95, 6
10	85, 7444	84, 5758	90, 3885	91, 8442	•93, 9966
11	64, 06	73, 296	•86, 292	71, 235	•86, 138
12	•54, 2609	•58, 6087	•58, 6087	•57, 3333	•54, 2029
13	92, 29	92, 36	•96, 08	•95, 94	•96, 31
14	75, 7	75, 87	79, 25	•79, 39	•80, 69
15	93, 16	90, 48	•93, 83	•96, 15	•95, 92
16	•69, 32	68, 03	68, 9	•70, 79	•69, 95
17	72, 99	70, 21	74, 63	•77, 42	•78, 1
18	91, 1317	91, 7544	•96, 3167	93, 637	•96, 9181
19	•87, 7142	93, 1336	•96, 6307	•90, 9446	•91, 8144
20	85, 7041	87, 3362	•97, 1161	94, 2085	•97, 5182
21	80, 6753	90, 4416	•94, 1732	86, 6667	•95, 1602
22	67, 5	•75, 6731	•75, 5769	•71, 4423	•75, 9615
23	79, 5131	89, 8544	•92, 7277	79, 8566	77, 3658
24	43, 1678	58, 6052	67, 4704	•68, 5106	•72, 9787
25	80	79, 968	•84, 508	•4, 46	•84, 22
26	97, 4157	96, 9663	•96, 9663	•98, 427	97, 4157
Av	78, 4842	80, 3262	83, 1445	82, 4739	84, 1233

carried out using this method [2].

Table 3 shows, in the upper half of each cell, the comparison between every pair of algorithms, where each entry *w-t-l* in row i and column j means that the algorithm in row i wins in w datasets, ties in t (ties means no statistical difference according to the 5x2 cv F Test) and loses in l datasets, compared to the algorithm in column j. The lower half of each cell contains the results from the Wilcoxon tests (Demšar, 2006), with $\alpha = 0.05$, which compare every pair of algorithms with the 26 datasets: whenever the test result represented a significant improvement in favor of one of the tests over the other, the name of the winner is shown, otherwise `NO` is shown.

Table 3. Accuracy comparison between pairs of algorithms.

Ftest / **Wilcoxon**	NB-G	NB	AODE	**GAODE**
NB	7-16-3			
	NO			
AODE	11-14-1	14-12-0		
	AODE	AODE		
GAODE	12-14-0	12-12-2	5-16-5	
	GAODE	GAODE	NO	
HAODE	13-13-0	13-12-1	6-19-1	6-18-2
	HAODE	HAODE	HAODE	HAODE

In terms of the arithmetical mean obtained, NB with discretization might be thought to work better than NB-G, but the number of datasets where NB-G is not significantly worse than the best method, or actually is the best method, is 11, versus the 10 for NB. In fact, the Wilcoxon test returned no significant difference between these two methods for these datasets. We might expect the same reasoning to be extendable to the comparison between AODE and GAODE. However, this is not entirely true as the difference between means from the two algorithms is lower and the number of datasets where they are not significantly worse than the other classifiers is exactly the same. In this case, the Wilcoxon test also failed to show a significant difference.

Analyzing these scores, we can confirm that both the GAODE and HAODE classifiers are significantly better than NB in any of its versions. As far as HAODE is concerned, not only does it obtain the highest accuracy mean, but also the highest number of datasets whose accuracies are not significantly different from the best provided by any of the other classifiers. Likewise, according to the Wilcoxon test, it is significantly better than AODE and GAODE for this group of numerical datasets despite the considerable number of ties.

Furthermore, a Friedman test was performed for the 5 classifiers, yielding statistical difference. The posterior Nemenyi tests (Demšar, 2006; García & Herrera, 2009) only rejected the hypothesis that two algorithms are not significantly different in favor of GAODE and HAODE over NB-G and NB, whereas AODE could not be proved to be significantly better than any of them.

5.2. Hybrid Datasets

So far, we have seen the great capacity of HAODE as an alternative to AODE for numeric datasets. As opposed to GAODE, HAODE is able to deal with all kinds of datasets, hence we have also carried out experiments with 16 hybrid datasets included in a standard group of 36 UCI repository datasets, whose main characteristics are summarized in table 4. All the numeric datasets in these group were included in the previous set of experiments and for the discrete datasets both classifiers are equal. That is the reason why in this block we just focus on hybrid ones.

Table 5 shows the accuracy results with NB (estimating Gaussians or multinomials according to the attribute type), AODE and HAODE using a 5x2cv on the evaluation and applying the discretization method previously mentioned in all the cases. The reason why the order of the datasets was altered will be given below.

Taking each *w-t-l* notation to mean that HAODE wins in w datasets, ties in t and loses in l datasets compared

[2]Further experiments have been performed with different discretization methods, and the results obtained follow the same tendency.

Table 4. Characteristics of the hybrid datasets: number of attributes (n), number of classes (k), number of instances (I), number of discrete and continuous attributes ($\#D$ and $\#C$) and % of missing values ($\%M$).

Id.	Dataset	n	k	I	$\#D$	$\#C$	$\%M$
1	anneal.ORIG	38	6	898	29	9	63, 32
2	anneal	38	6	898	29	9	0
3	autos	25	7	205	10	15	11, 06
4	colic.ORIG	27	2	368	20	7	18, 7
5	colic	27	2	368	15	7	22, 77
6	credit-a	15	2	690	9	6	5
7	credit-g	20	2	1000	13	7	0
8	heart-c	13	2	303	7	6	0, 17
9	heart-h	13	2	294	7	6	19
10	hepatitis	19	2	155	13	6	5, 39
11	hypothyroid	29	4	3772	22	7	5, 4
12	labor	16	2	57	8	8	0
13	lymph	18	4	148	15	3	0
14	sick	29	2	3772	22	7	5, 4
15	vowel	13	11	990	10	3	0
16	zoo	17	7	101	16	1	0

Table 5. Accuracy results obtained with NB, AODE and HAODE classifiers in hybrid datasets.

Id	NB	AODE	HAODE	$\%M$
16	•90, 495	•91, 6832	•94, 2574	0
13	•81, 0811	•80, 8108	•82, 5676	0
15	50, 6667	61, 0505	•78, 4444	0
7	•74, 16	•74, 44	•75, 32	0
12	•88, 4211	•87, 7193	•88, 0702	0
2	•95, 1448	•96, 7483	•92, 784	0
8	•83, 3003	•83, 3003	•83, 7624	0, 17
6	•86, 029	•86, 2609	78, 8696	5
10	•82, 3226	•83, 0968	•84, 3871	5, 39
11	•97, 7253	•98, 0011	95, 6416	5, 4
14	97, 0891	•97, 2057	94, 5652	5, 4
3	•58, 7317	•64, 1951	•57, 561	11, 06
4	•69, 6196	•69, 7826	60, 8696	18, 7
9	•83, 8776	•83, 9456	•83, 4014	19
5	•79, 3478	•81, 087	•78, 8043	22, 77
1	•93, 1403	•93, 9866	88, 7751	63, 32
Av	•81, 947	•83, 3321	•82, 3801	

to AODE at a 95% confidence level, the hybrid classifier significantly improves on AODE in 1 of them, loses in 5 others and draws in 10 of them (1-10-5). Even though these are not the results we expected, considering only these hybrid datasets, it cannot be proved that there exists a significant advantage of AODE over HAODE, as Wilcoxon does not guarantee statistical difference.

Looking for a plausible explanation of this fact, specially taking into account the good results obtained by HAODE vs AODE in numerical datasets (table 2), we analyzed the percentage of numerical variables with respect to discrete ones in hybrid datasets, but no significant pattern was found. Then, we turned to study the impact of missing values and, in this case, a relevant pattern can be obtained: *the presence of missing values punishes HAODE vs AODE*. Thus, in table 5, hybrid datasets have been ordered according to their percentage of missing values. Above the line of the 2nd column in table 5 we have the ones with almost no missing values. In fact, the Wilcoxon test shows statistical difference when only the datasets with missing values are considered. Based on the apparent tendency of HAODE to punish datasets with missing values, we then preprocessed all the datasets with an unsupervised filter to replace missing values with the modes and means from the existing data in the corresponding column. The same experiments were executed obtaining a result of 2-12-2. For the first group of numeric datasets the results are the same, as only breast-w has a $0, 23\%$ of missing values. These results lead us to the conclusion that HAODE can be more sensitive to missing values than the other classifiers included in the comparison. It seems that the repeated use of different estimators (average of n models) made from few data when using Gaussian networks is more damaging than when they are made from multinomials.

6. Conclusions and Future Work

In this paper, we have proposed two alternatives to the AODE classifier in order to deal with continuous attributes without performing a direct discretization process over the whole data. The first classifier, GAODE, applies CGNs to model the relationships between each predictive attribute and its parents, obtaining competitive results compared to AODE. GAODE implies a reduction in the space complexity and the parameters can be computed *a priori* in a single pass over the data, maintaining AODE's time complexity as well. This approach can also provide a more reliable and robust computation of the necessary statistics as the parameters are exclusively class-conditioned.

Furthermore, we have also presented a "hybrid" classifier, HAODE, which keeps the superparent attribute discrete in every model. This approach offers the clear advantage of dealing with any kind of dataset. Nonetheless, even though it is in general competitive when compared with AODE, it has shown a clear preference for datasets with continuous attributes and the absence of missing data, where it is significantly better than AODE. The proper treatment of these missing values in HAODE is beyond the scope of this paper, so we shall tackle it in future work.

Even though Gaussian networks often provide a reasonable approximation to many real-world distributions, they assume variables are sampled from Gaussian distributions. In the future, we are planning to explore more general distribution probabilities, specifically the application of Mixtures of Truncated Exponentials (MTEs) (Moral et al., 2001)) to AODE, which entail a more precise estimation, being also able to model Bayesian, Gaussian and hybrid networks.

Acknowledgments

We are grateful to the reviewers for their numerous suggestions and J. L. Mateo for his comments. This work has been partially supported by the Consejería de Educación y Ciencia (JCCM) under Project PCI08-0048-8577574, the Spanish Ministerio de Educación y Tecnología under Project TIN2007-67418-C03-01 and the FPU grant with reference number AP2007-02736.

References

Alpaydin, E. (1999). Combined 5 x 2 cv f test for comparing supervised classification learning algorithms. *Neural Comput.*, *11*, 1885–1892.

Andersen, S. K., Olesen, K. G., Jensen, F. V., & Jensen, F. (1989). HUGIN–A shell for building Bayesian belief universes for expert systems. *Proc. of the 11th Int. Joint Conf. on AI* (pp. 1080–1085).

Asuncion, A., & Newman, D. (2007). UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences. `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

DeGroot, M. H. (1970). *Optimal statistical decisions*. New York: McGraw-Hill.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, *7*, 1–30.

Domingos, P., & Pazzani, M. (1996). Beyond independence: Conditions for the optimality of the simple Bayesian classifier. *Proc. of the 13th Int. Conf. on Machine Learning* (pp. 105–112).

Duda, R. O., Hart, P. E., & Stork, D. G. (1973). *Pattern classification and scene analysis*. Wiley NY.

Fayyad, U. M., & Irani, K. B. (1993). Multi-interval discretization of continuous-valued attributes for classification learning. *Proc. of the 13th Int. Joint Conf. on Articial Intelligence* (pp. 1022–1027).

García, S., & Herrera, F. (2009). An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. *J. Mach. Learn. Res.*, *9*, 2677–2694.

Geiger, D., & Heckerman, D. (1994). Learning Gaussian networks. *Proc. of the 10th Annual Conf. on Uncertainty in AI* (pp. 235–243).

Keogh, E., & Pazzani, M. (1999). Learning augmented Bayesian classifiers: A comparison of distribution-based and classification-based approaches. *Proc. of the 7th Int. Workshop on AI and Statistics* (pp. 225–230).

Langley, P., Iba, W., & Thompson, K. (1992). An analysis of Bayesian classifiers. *Proc. of the 10th National Conf. on AI* (pp. 223–228).

Larrañaga, P., Etxeberria, R., Lozano, J., & Peña, J. M. (1999). *Optimization by learning and simulation of Bayesian and Gaussian networks* (Technical Report). University of the Basque Country.

Lauritzen, S. L., & Jensen, F. (2001). Stable local computation with conditional Gaussian distributions. *Statistics and Computing*, *11*, 191–203.

Moral, S., Rumí, R., & Salmerón, A. (2001). Mixtures of Truncated Exponentials in hybrid Bayesian networks. *Proc. of the 6th European Conf. on Symbolic and Quantitative Approaches to Reasoning with Uncertainty* (pp. 156–167).

Neapolitan, R. E. (2003). *Learning Bayesian networks*. Prentice Hall.

Pérez, A., Larrañaga, P., & Inza, I. (2006). Supervised classification with conditional gaussian networks: Increasing the structure complexity from naive Bayes. *Int. J. Approx. Reasoning*, *43*, 1–25.

Sahami, M. (1996). Learning limited dependence Bayesian classifiers. *Proc. of the 2nd Int. Conf. on Knowledge Discovery in Databases* (pp. 335–338).

Webb, G. I., Boughton, J. R., & Wang, Z. (2005). Not So Naive Bayes: Aggregating One-Dependence Estimators. *Mach. Learn.*, *58*, 5–24.

Wek08 (2008). Collection of datasets avalaibles from the weka official homepage. `http://www.cs.waikato.ac.nz/ml/weka/`.

Witten, I. H., & Frank, E. (2005). *Data mining: Practical machine learning tools and techniques*. Morgan Kaufmann. 2 edition.

Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Yu, P. S., Zhou, Z.-H., Steinbach, M., Hand, D. J., & Steinberg, D. (2007). Top 10 algorithms in data mining. *Knowl. Inf. Syst.*, *14*, 1–37.

Zheng, F., & Webb, G. (2005). A Comparative Study of Semi-naive Bayes Methods in Classification Learning. *Proc. of the 4th Australian Data Mining Conf.* (pp. 141–156).

Zheng, Z., & Webb, G. I. (2000). Lazy Learning of Bayesian Rules. *Mach. Learn.*, *41*, 53–84.