Non-Disjoint Discretization for Aggregating One-Dependence Estimator Classifiers

Ana M. Martínez[1], Geoffrey I. Webb[2], M. Julia Flores[1], and José A. Gámez[1]

[1] Computer Systems Department, Intelligent Systems & Data Mining,
University of Castilla-La Mancha, Albacete, Spain
{anamaria.martinez,julia.flores,jose.gamez}@uclm.es
[2] Faculty of Information Technology,
Monash University, Melbourne, Australia
geoff.webb@monash.edu

Abstract. There is still lack of clarity about the best manner in which to handle numeric attributes when applying Bayesian network classifiers. Discretization methods entail an unavoidable loss of information. Nonetheless, a number of studies have shown that appropriate discretization can outperform straightforward use of common, but often unrealistic parametric distribution (e.g. Gaussian). Previous studies have shown the Averaged One-Dependence Estimators (AODE) classifier and its variant Hybrid AODE (HAODE, which deals with numeric and discrete variables) to be robust towards the discretization method applied. However, all the discretization techniques taken into account so far formed non-overlapping intervals for a numeric attribute. We argue that the idea of non-disjoint discretization, already justified in Naive Bayes classifiers, can also be profitably extended to AODE and HAODE, albeit with some variations; and our experimental results seem to support this hypothesis, specially for the latter.

Keywords: AODE, HAODE, Non-Disjoint Discretization, Bayesian Classifiers.

1 Introduction

So far, the AODE classifier [1] has arisen as one of the most attractive alternative to naive Bayes (NB), as it has proved to be significantly better in terms of error reduction compared to many others semi-naive techniques, maintaining under control its time and space complexity in training and classification time [2]. Nevertheless, as most of the techniques based on Bayesian networks, a multinomial probability distribution is assumed. This gives rise to difficulties in the context of continuous variables, as there is a need to infer joint probability distributions, and this is difficult in the absence of very large quantities of data. Two different ways to tackle this issue for AODE were studied in [3], that led to the Gaussian AODE classifier, where Conditional Gaussian networks were used to deal with datasets containing exclusively numeric attributes; and

E. Corchado et al. (Eds.): HAIS 2012, Part II, LNCS 7209, pp. 151–162, 2012.

the Hybrid AODE classifier (HAODE), that resorted to the use of discretization only for the numeric values in the parents. However, these approaches also have their limitations, since the Gaussian assumption may be simply unrealistic. And discretization becomes a good alternative.

In this respect, we can find studies where the robustness of AODE and HAODE toward the discretization method is analyzed [4]. The conclusions in this work indicate that although the discretization method indeed matters when studying a particular dataset, it does not seem to be decisive when the aim is to compare a group of semi-naive Bayesian classifiers over a standard group of datasets. Nevertheless, only disjoint discretization techniques have been taken into account in that study. In [5], a novel non-disjoint discretization technique (NDD) is presented to cope with numeric attributes in NB by forming overlapping intervals. NDD forms overlapping intervals for a continuous attribute, always locating a value toward the middle of an interval to obtain more reliable probability estimations. Its use is based on the insight that while it is necessary to use a single discretization of each variable while classifying an instance, different discretizations can be applied when classifying different instances.

The results show a clear improvement in NB over other disjoint discretization methods, and we believe, that these results could also duplicate in AODE and HAODE, albeit with some modifications to the disjoint discretization method proposed. Compared to NB, AODE and HAODE could suffer more from creating a large number of intervals (from a variance increase), since their conditional probability tables (CPTs) are formed by the combination of a couple of attributes (the class and the parent). It is credible that NDD could help us to alleviate this problem by allowing larger intervals to be formed without greatly increasing the bias.

Hence, the main contributions of this paper are the following: to begin with, we redefine the original approach of NDD discretization for its use in AODE and HAODE, describing the corresponding modifications (Section 5). Furthermore, a new weighting system is included with the aim to decrease discretization bias. In Section 6, an experimental study compares the application of these *joint* discretization techniques in AODE and HAODE with the use of a traditional *disjoint* discretization method: equal frequency discretization (EFD)[1]. This study includes comparisons in terms of accuracy, but mainly focuses in results detailing bias and variance discretization records, as well as the combined error from both measures.

The rest of the paper is divided as follows: Section 2 and 3 introduces AODE and HAODE classifiers. Section 4 explains the main differences between disjoint and joint discretizations and finally, Section 7 provides our main conclusions from the study.

[1] As we will see below, this selection has not been made at random, but equal frequency division with 5 bins has shown to be the most beneficial for AODE [4].

2 AODE

AODE [1] is considered an improvement over NB and an interesting alternative to other attempts such as Lazy Bayesian Rules (LBR) [6] and Super-Parent TAN (SP-TAN) [7], since they offer similar error values, but AODE is significantly more efficient at classification time compared with the first one and at training time compared with the second one. In order to maintain efficiency, AODE is restricted to exclusively use 1-dependence estimators. Specifically, AODE can be considered as an ensemble of SPODEs (Superparent One-Dependence Estimators), because every attribute depends on the class and another shared attribute, designated as super-parent.

AODE computes the average of the n possible SPODE classifiers (one for each attribute in the database) and hence, the MAP (maximum a posteriori) hypothesis is as follows:

$$c_{MAP} = argmax_{c \in \Omega_C} \left(\sum_{j=1, N(x_j)>q}^{n} p(c, x_j) \prod_{i=1, i \neq j}^{n} p(x_i | c, x_j) \right), \qquad (1)$$

where x_i, x_j are the label of the predictive attributes and c the class label. The condition $N(x_j) > q$ is used as a threshold to avoid making predictions from attributes with few observations. In our experiments this q value has been set to 1, which is the default value in the data mining tool WEKA [8].

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

NB can deal with hybrid (discrete and numeric variables) datasets by means of Gaussian and multinomial distributions. On the contrary, this is not possible for AODE, as a numeric variable (super-parent) cannot be the parent of a discrete variable. This is the reason why AODE can only be applied after discretizing numeric variables. HAODE [3] restricts the use of discretization to only the variable which acts as super-parent in every model, keeping it numeric when it is playing the role of *child*. Thus, multinomial distributions are estimated for discrete variables and the super-parent, together with one univariate Gaussian distribution (one for each configuration in the Cartesian product between the class and the super-parent) for each numeric variable which acts as child.

Classification is performed according to the following equation then:

$$c_{MAP} = argmax_{c \in \Omega_C} \left(\sum_{j=1, N(x_j)>m}^{n} p(x_j, c) \prod_{i=1 \wedge i \neq j}^{n} \mathcal{N}(x_i : \mu_i(c, x_j), \sigma_i^2(c, x_j)) \right), \qquad (2)$$

where $\mu_i(c, x_j)$ and $\sigma_i^2(c, x_j)$ are the mean and variance of X_i conditioned to the values c for the class and x_j for X_j. $\mathcal{N}(x_i : ., .)$ is the resulting value of the normal density function of x_i with the corresponding mean and variance.

HAODE presents the same time complexity as AODE, but achieves a slight reduction in spatial complexity because HAODE requires only two parameters (mean and variance) for Gaussian distributions, independently of the number of states in which this variable has been discretized when it acts as super-parent.

4 Disjoint vs. Non-Disjoint Discretization

Formally, given the numeric attribute values $x_i, x_j \in \mathbb{R}$, any disjoint discretization method would create a unique interval $(a, b] \ni x_i$ and $(d, e] \ni x_j$ for every value so that AODE's statistics, $p(X_j = x_j, C = c)$ and $p(X_i = x_i | C = c, X_j = x_j)$ would be estimated by

$$p(X_j = x_j, C = c) \approx p(d < X_j \leq e, C = c) \tag{3}$$

$$p(X_i = x_i | C = c, X_j = x_j) \approx p(a < X_i \leq b | C = c, d < X_j \leq e) \tag{4}$$

In disjoint discretization techniques (EFD, equal width division, MDL, etc) every numeric sample belongs to a single interval. I.e., considering $x_i < x_j$, if $a \neq d$ (they do not fall in the same interval) then $d \geq b$. This implies that for those cases where the original numeric value falls around the center of the interval assigned, we could expect more distinguishing information than when it falls near one of the boundaries of the interval.

In contrast, NDD creates bins that overlap. So long as a single bin is used consistently when classifying a single object, it does not matter whether inconsistent bins are used when classifying different objects.

4.1 Equal Frequency Discretization

EFD is an unsupervised technique where the values are ordered and divided into b disjoint bins so that each one contains approximately the same number of training instances.

Therefore, every bin contains m/b instances with adjacent values, where m is the total number of samples. This type of discretization method provides bins containing equal numbers of examples and hence the variance of the estimates formed from the bins should be more stable than alternatives. As a group of values with identical values must be placed in the same bin, it is not always possible to generate b intervals with exactly the same number of values.

Time complexity for this technique is $\mathcal{O}(m \log m)$ as it is necessary to perform an ordering of the data.

4.2 Non-Disjoint Discretization

NDD is also an unsupervised technique that forms t atomic intervals $B_0 = [a'_1, b'_1]$, $B_1 = (a'_2, b'_2], \ldots, B_t = (a'_t, b'_t]$ (where $b'_i = a'_{i+1}, \forall i$), with equal frequency. In its definition for NB [5], one operational interval or label is formed

then for each set of three consecutive atomic intervals, such that the rth ($1 \leq r \leq t-2$) interval $(a_r, b_r]$ satisfies $a_r = a'_r$ and $b_r = b'_{r+2}$. Each numeric value x is assigned to interval $(a'_{i-1}, b'_{i+1}]$ where i is the index of the atomic interval $(a'_i, b'_i]$ such that $a'_i < x \leq b'_i$, except when $i = 1$ in which it is assigned to interval $(a'_1, b'_3]$ and when $i = t$ that it is assigned to interval $(a'_{t-2}, b'_t]$. Here t and the number of instances per atomic interval are selected proportionally to the number of training instances, following the idea of Proportional k-Interval Discretization [9].

NDD is dominated by sorting as well, and hence, its complexity is also $\mathcal{O}(m \log m)$.

5 NDD Adapted to AODE and HAODE

By dividing the ranges of numeric attributes into overlapping intervals in AODE and HAODE, we not only intend to reduce discretization bias [10] by always locating a value toward the middle of an interval and, in general, creating a larger number of intervals; but also maintaining discretization variance, since the number of samples from which the CPTs will be estimated should be similar.

Intuitively, discretization resulting in large interval numbers tends to have low bias (any given interval is less likely to include a decision boundary of the original numeric attribute). Discretization resulting in intervals with a large number of instances tends to have low variance (as the probability estimations are more stable and reliable). The problem is that supposing there is a fixed dataset size, the larger the number of intervals, the smaller the number of instances per interval is.

The application of NDD to AODE involves discretizing the whole dataset into non-disjoint intervals before training the classifier, whereas in the case of HAODE, just the cases where a numeric attribute plays the role of super-parent will be discretized.

However, and for the reasons that we detail next, some changes are introduced to the original definition of NDD as specified in Yang and Webb's paper [5]:

1. A threshold is considered to mark the minimum frequency from which an atomic interval will not be merged with its neighbours. This should prevent us from increasing bias when sufficient samples are already provided. See Figure 1 for an example on interval formation having each atomic interval frequency into account. Since it is possible the presence of multiple instances with the same value, the number of final samples per atomic attribute may vary, and it usually does[2].
2. In the original definition of NDD, the interval size is equal to the interval number ($\approx \sqrt{m}$) with the aim to give equal importance to discretization bias and discretization variance reduction. Even though it provides very good results for NB, it is not the case for AODE or HAODE, where in general, a

[2] The way in which this is handled is the same for NDD and EF5, check WEKA's equal frequency discretization method for more details.

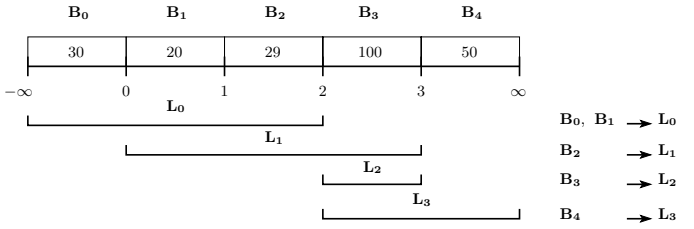

Fig. 1. Example of NDD division, the minimum frequency to merge atomic intervals into a single label (L) is equal to 100. Labels selected when classifying samples belonging to atomic bins B_0, B_1, B_2, B_3 and B_4 are indicated at the bottom right corner.

smaller number of intervals is desired because it is necessary to estimate the probability of an interval on one attribute conditioned by both an interval on another attribute and the class, whereas in NB it is necessary only to estimate the probability of an interval given the class. Previous experiments have shown that Proportional Discretization (PD), tailored to NB, where a number of $\sqrt{m}$ instances is selected, is not generally beneficial for AODE [4].

3. When the number of cut-points is lower than 3, then equal frequency discretization will be kept.
4. Weighting importance: note that by using NDD as defined above, there are some numeric samples that fall within two or three labels. Given a numeric sample x_i discretized by NDD into the labels $L_1 = (a'_1, b'_1]$, $L_2 = (a'_2, b'_2]$ and $L_3 = (a'_3, b'_3]$ in training time; L_2 would be the final label assigned to another sample $x_j \in \mathbb{R}$, $x_j = x_i$ in classification time. The contribution of L_2 to the CPT will be greater (it is given more importance when training) than the contribution provided by the other two bins. This is carried out by the use of weights. There exist several forms in which these weights could be distributed, in this first approach we have adopted the simplest one (apart from uniform distribution, being equivalent to non-weighting). Since a single sample can be allocated at most in three atomic bins, the weight distribution could be set as 0.75 for the centred label and the rest equally divided into the other labels (if there is more than one)[3]. In AODE, the combination of weights when both the parent and the child involved in a CPT come from a joint discretization is carried out by multiplying its corresponding weights (so that the sum remains equal to one).

   Figure 2 shows an example for a training instance I with two numeric attributes: X_0 and X_1. This instance is discretized using the NDD procedure indicated in Section 5, obtaining I_{NDD}. Hence, the value 3.5 for X_0 falls within three labels: L_0, L_1 and L_2 (specifically centred in L_1, that is why it is given the highest weight), whereas the value 2 for X_1 falls

[3] Further experiments have been carried out by slightly altering the weight assignment obtaining very similar results. This study has been performed using 3 atomic bins per interval, and we believe that this result may not be extrapolated to any higher odd number.

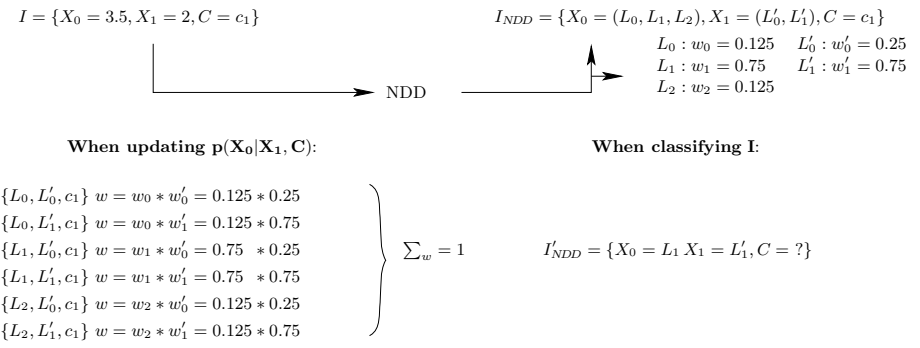

Fig. 2. Example on how weighted NDD works in AODE: first of all, the instance is discretized using NDD and weights are assigned to every label. When training, instance I would contribute to the CPT for X_0 given X_1 and C as shown in the left hand side. If classifying I, only the main labels (so that the sample is in the center) are considered.

within labels L'_0 and L'_1 (centred in L'_1 in this case). These weights are then used to indicate the contribution of each pair of values when updating the CPTs[4]. When the same instance I were to be classified (the class is missing), then I'_{NDD} would be used, where the centred labels for both attributes are considered. Then the MAP equation would be the following: $argmax_{c \in \Omega_C} \left(\sum_{j=1, N(x_j)>q}^{n} p(c, L^*_j) \prod_{i=1, i \neq j}^{n} p(L^*_i | c, L^*_j) \right)$, where L^*_j, is the centred label for X_j; and L^*_i, the centred label for X_i.

As NDD is dominated by sorting, no increase in the complexity is induced.

6 Experiments

We run our experiments on 28 datasets from the UCI machine learning repository [11] and KDD archive [12], listed in Table 1. As in [5], this experimental suite comprises 3 parts. The first part is composed of all the UCI datasets used by [13] when publishing the entropy minimization heuristic discretization. The second part is composed of all the datasets with numeric attributes used by [14] for studying NB classification. The third part is composed of larger datasets employed in [9].

To begin with, we have pre-processed the datasets using an unsupervised filter to replace all the missing values with the modes and means from the existing data in the corresponding column, and another one to remove useless attributes that do not vary at all or whose variation percentage is lower than 99%[5]. This is

[4] Note that in a multinomial distribution, the combination of values from an instance to be incorporated in a CPT contribute with a `unit`, whereas here we consider the contribution of the `weight` for each label (that always sums to one for each instance).

[5] These two filters have been applied with the default settings provided by WEKA.

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

Id	Datasets	n	d	k	m	Id	Datasets	n	d	k	m
1	labor-negotiations	8	8	2	57	15	annealing	6	32	6	898
2	echocardiogram	5	1	2	74	16	german	7	13	2	1000
3	iris	4	0	3	150	17	multiple-features	3	3	10	2000
4	hepatitis	6	13	2	155	18	hypothyroid	7	18	2	2163
5	wine-recognition	13	0	3	178	19	satimage	36	0	6	6435
6	sonar	60	0	2	208	20	musk	166	0	2	6598
7	glass-identification	9	0	3	214	21	pioneer-mobile-robot	29	7	57	9150
8	heart-disease	7	6	2	270	22	handwritten-digits	16	0	10	10992
9	liver-disorders	6	0	2	345	23	sign-language	8	0	3	12546
10	ionosphere	34	0	2	351	24	letter-recognition	16	0	26	20000
11	horse-colic	7	14	2	368	25	adult	6	8	2	48842
12	credit-screening	6	9	2	690	26	impums.la.99	20	40	13	88443
13	prima-indians-diabetes	8	0	2	768	27	census-income	8	33	2	299285
14	vehicle	18	0	4	846	28	forest-covertype	10	44	7	581012

in order to make the group of datasets uniform and suitable for all the classifiers considered in the comparison.

In order to evaluate the experimental results we use two methods: accuracy (for the sake of comparison with previous works and to facilitate the frame to reproduce experiments) and error in terms of bias and variance according to [15], using five times 2-fold cross validation (5x2cv). 5x2cv entails a reasonable trade-off between precision and execution time of the experiments, providing a better partition for the posterior statistical analysis, as in addition, the degree of overlapping between the different folds is lower [16]. The bias-variance decomposition has been performed using the sub-sampled cross-validation procedure as specified by [17]. The global error obtained by this procedure is the sum of the bias and variance results.

The discretization technique selected as the basis for comparison is EFD with 5 bins (EF5), as it has shown to provide slightly better results compared to other methods [4] such as Minimum Description Length [13], equal width discretization or equal frequency discretization with a different number of bins.

As advanced in section 4, the labels formed in NDD will comprise at most 3 atomic bins[6]. To provide a fair comparison with EF5, the initial number of atomic bins considered is 15. This means that the labels created (groups of three atomic bins) will be of approximately the same average size as the bins for EF5. The minimum frequency from which an atomic interval will not be merged with its neighbours will be 100 (approximately 30 per atomic bin[7]).

[6] In theory any odd number would be acceptable (the larger the better to allocate a sample in the middle of an interval), but for simplicity we take 3 as in [5].

[7] The figure 30 has been selected motivated by the 30-sample rule-of-thumb very recurrent in statistics. Still, further experiments were carried out with different values; although the results were not significantly different, the best values were obtained with 30 and $33.\overline{3}$.

Table 2. Results in terms of `accuracy±sample standard deviation` obtained for AODE and HAODE using EF5, NDD and NDDw

	AODE			HAODE		
Id	EF5	NDD	NDDw	EF5	NDD	NDDw
1	93.3333±3.88	•94.3860±4.49	•94.0351±5.35	90.8772±8.59	•91.2281±9.76	•91.2281±9.48
2	68.9189±4.81	•72.9730±4.77	•72.1622±7.10	74.3243±4.64	72.9730±5.41	•76.7568±5.44
3	92.9333±2.74	•93.8667±2.20	•93.0667±1.97	95.8667±1.83	•96.0000±2.08	95.4667±1.80
4	82.1935±2.81	•82.9677±3.86	•82.4516±3.54	83.2258±2.23	82.0645±3.23	82.7097±3.09
5	96.4045±1.28	•96.8539±1.66	○96.4045±1.48	98.0899±0.93	○98.0899±0.76	97.8652±0.98
6	81.4423±3.63	•81.6346±4.19	80.7692±4.03	82.7885±3.91	82.5000±3.76	•84.6154±3.74
7	68.1308±5.07	•68.5047±4.06	•70.2804±3.76	69.1589±4.13	•69.5327±4.83	•70.0000±4.77
8	81.4815±2.42	•83.4815±2.59	•81.7037±1.60	81.0370±1.95	•81.5556±1.96	○81.0370±2.84
9	60.3478±3.47	•65.1014±3.46	•63.5942±2.76	62.0290±3.56	59.5942±3.81	61.1014±3.40
10	91.3390±2.19	89.4017±2.55	90.3134±2.42	92.2507±2.33	•92.7066±1.68	•92.4217±1.37
11	79.5652±1.23	•80.1087±1.94	•80.7609±1.18	65.6522±4.65	•66.3043±4.42	•66.4674±3.95
12	86.4638±1.23	•86.5797±0.95	•86.5507±1.18	80.7826±1.16	•80.0870±0.88	80.0870±1.14
13	75.2083±1.78	•75.5208±1.33	74.1927±1.65	75.6250±0.90	75.2344±0.94	75.0260±1.08
14	69.2199±1.14	68.3215±1.39	67.9433±1.58	73.3806±2.05	•73.5225±2.13	72.2695±2.25
15	87.9955±1.77	86.3474±1.65	•90.0668±1.07	82.9176±1.61	81.9822±2.81	82.5167±2.64
16	74.1600±1.08	•74.3800±0.93	•74.3400±1.19	73.7400±1.27	•74.7800±1.16	•74.1800±1.10
17	66.2600±1.22	•68.1700±1.26	•68.3600±1.31	69.1800±1.37	•69.9400±1.68	•70.6800±1.60
18	97.3000±0.21	•98.1979±0.27	•98.2548±0.22	98.1284±0.23	•98.3181±0.26	•98.3307±0.33
19	87.4219±0.57	•88.4444±0.40	•88.4444±0.40	83.9254±0.98	•85.9176±0.79	•85.9176±0.78
20	85.2743±0.85	•93.2404±0.32	•93.2555±0.30	83.5920±1.09	•87.5750±0.71	•87.5720±0.71
21	90.5268±0.47	•93.5432±0.86	•93.5016±0.87	89.1607±0.86	•94.3607±0.67	•94.3497±0.67
22	97.0287±0.17	96.8013±0.32	96.8013±0.32	97.1634±0.33	•97.6638±0.21	•97.6638±0.21
23	71.3678±0.70	•73.2680±0.51	•73.2855±0.51	66.3399±1.01	•67.1433±0.86	•67.1242±0.88
24	83.4580±0.21	•85.4120±0.37	•85.4720±0.37	84.5250±0.21	•88.1870±0.32	•88.2030±0.32
25	83.9347±0.25	•84.1677±0.29	•84.2771±0.29	84.0830±0.31	•83.9237±0.37	83.9892±0.35
26	92.3890±0.08	92.3854±0.08	•92.3928±0.08	87.0904±0.44	•87.7243±0.58	•87.7017±0.57
27	92.1766±0.09	•92.4165±0.07	•92.4171±0.07	93.4646±0.11	•93.6628±0.09	•93.6666±0.09
28	71.3988±0.11	•73.9682±0.09	•73.9682±0.09	69.9027±0.13	•70.8710±0.09	•70.8710±0.09
Av.	82.4169±1.62	•85.5873±1.67	•83.5381±1.67	81.7251±1.89	•82.2658±2.01	•82.4935±1.99

Table 2 shows the accuracy results obtained for AODE and HAODE using EF5, NDD and NDDw; along with the sample standard deviation for each dataset. The bullet next to certain outputs (in NDD and NDDw) indicates that the corresponding result improves the output provided when EF5 is used. The circle, in turn, indicates a draw. These results lead us to think that the use of NDD or NDDw is competitive over EF5 (and by extension, other traditional disjoint discretization techniques), especially for the former. Nevertheless, standard deviation is, on average, higher for NDD and NDDw compared to EF5, this indicates that the latter is more robust with respect to the income data, although the values provided in terms of accuracy are lower, in spite of that.

Table 3 shows the number of datasets for which discretizing with NDD obtained better, equal or worse performance compared to using equal frequency with 5 bins. These records are complemented by the results from the Wilcoxon signed-rank tests [18], which compare every pair of algorithms considering the whole group of datasets. The first two columns depict the records when the samples are not weighted (i.e. weighted uniformly) according to the atomic bin to which they belong. In this case, NDD in AODE and HAODE is better at improving accuracy and global error. The improvement is clear also as far as bias is concerned for HAODE and variance for AODE. However, this advantage is not as clear in terms of bias in AODE and variance in HAODE, although they still

Table 3. Comparisons in terms of win-draw-lose records and Wilcoxon tests

	non-weighted		weighted	
w-t-l / Wilcoxon	AODE NDD vs EF5	HAODE NDD vs EF5	AODE NDDw vs EF5	HAODE NDDw vs EF5
Accuracy	23-0-5 / **< 0.05**	21-1-6 / **< 0.05**	22-1-5 / **< 0.05**	18-2-8 / **< 0.05**
Bias	14-3-11 / 0.2395	21-1-6 / **< 0.05**	15-3-10 / **< 0.1**(0.06)	22-0-6 / **< 0.05**
Variance	18-2-8 / **< 0.05**	14-4-10 / 0.3621	13-2-13 / 0.6	10-0-14 / 0.863
Error	21-1-6 / **< 0.05**	19-3-6 / **< 0.05**	16-3-9 / **< 0.05**	18-1-9 / **< 0.05**

Table 4. Average results in terms of accuracy/bias/variance/error (best value in bold)

	AODE		HAODE	
	EF5	NDD	EF5	NDD
Accuracy	82.4169	**83.5873**	81.7251	**82.2658**
Bias	0.1298	**0.1250**	0.1348	**0.1275**
Variance	0.0395	**0.0355**	0.0440	**0.0435**
Error	0.1737	**0.1643**	0.1836	**0.1758**

provide better records compared to EF5, no statistical difference is found. If we consider the weighted version of NDD, the results are slightly better in terms of bias (specially for AODE), at the expense of variance and overall worsening. Hence, from now on in the paper, we will just consider non-weighted NDD, although it is important to observe that the increase in variance may have less effect on error when larger data are provided.

Table 4 displays the average results in terms of accuracy, bias, variance and error obtained for the different classifiers, where NDD outperforms in every pair-to-pair comparison.

Note that execution time comparisons would show no interest information, since differences are minimum (same complexity order).

Hence, in the light of these results one question arises: why does NDD seems to improve more pronouncedly AODE's variance and HAODE's bias compared to applying equal frequency? The difference between the two classifiers lies in the "double use" (in parents and children nodes) of NDD in AODE, which seems to help in reducing variance at the expense of a bias sacrifice.

In this study, even though there is a slight improvement of HAODE over AODE (16-0-12 in terms of accuracy, see Table 2), this is not as striking as in the original study in 2009 [3], and this difference even shifts to 13-1-14 when NDD is applied. We believe this fact might be motivated by two reasons:

- HAODE aims to avoid information loss by resorting to the use of discretization only when necessary for the super-parents. However, that implies that Gaussian distributions are assumed in some cases, which can be a handicap if the real distributions in data are not Gaussians.
- In general, we should prefer high-bias, low-variance classifiers when the data are sparse; and low-bias, high-variance classifiers when data are numerous. Since we are now dealing with larger datasets, we could also deduce that

HAODE is more robust in small ones and AODE in larger ones, unless the normality condition is satisfied.

7 Conclusions

In this paper, we have studied the impact of applying NDD to AODE and HAODE compared to traditional disjoint discretization techniques. In this study we have chosen equal frequency division to represent the latter, as it was shown previously to provide better results among the most common disjoint discretization methods (EF, equal width division, MDL, etc).

We have introduced some modifications to the original definition of NDD [5] in order to fit into AODE and HAODE's context, as a smaller number of bins is usually desired compared to NB to avoid increasing variance. Furthermore, a new weighting system has been introduced at the counting process in order to increase the importance given to the bins created by NDD where samples are placed in the middle; which provided better results in terms of bias but worse overall records.

The results have been analyzed in terms of accuracy, bias, variance and global error obtaining the following conclusions:

- In general terms, an overall improvement is found for the two classifiers (AODE and HAODE) when NDD is used. Statistical differences according to the Wilcoxon test are found for both classifiers as far as accuracy and global error (sum of bias and variance) is concerned.
- The analysis on error descomposition in terms of bias and variance displays better results at all times when using NDD, being this improvement more marked for HAODE in terms of bias, and AODE in terms of variance.

The most important conclusion though, is the fact that whereas some of the most common disjoint discretization techniques have failed to demonstrate consistent improvement relative to alternatives, non-disjoint discretization demonstrates better win/draw/loss records and significant overall improvement. Still, we plan to extend the experimental part to a test bed of high dimensional datasets in order to corroborate these conclusions.

Moreover, we believe that the positive results observed in AODE are a good motivation to think that the beneficial properties of NDD will be strengthen when applied to Aggregating n-dependence estimators (AnDE) [19], for values of n greater or equal to 2 (since when $n = 1$ it is equivalent to AODE).

One drawback of NDD is that it requires the user to select additional parameters apart from the number of bins to form (such as in equal frequency division), also the number of atomic bins per operational interval and the minimum frequency per interval must be chosen.

Acknowledgements. This work has been partially funded by FEDER funds, the Spanish Government (MICINN) and the Castilla-La Mancha regional Government (JCCM) through projects TIN2010-20900-C04-03 and PEII11-0100-7773,

the FPU grant with reference number AP2007-02736 and the Australian Research Council under grant DP110101427.

References

1. Webb, G.I., Boughton, J.R., Wang, Z.: Not So Naive Bayes: Aggregating One-Dependence Estimators. Mach. Learn. 58(1), 5–24 (2005)
2. Zheng, F., Webb, G.I.: A Comparative Study of Semi-naive Bayes Methods in Classification Learning. In: Simoff, S.J., Williams, G.J., Galloway, J., Kolyshkina, I. (eds.) Proc. of the 4th AusDM Conf., pp. 141–156 (2005)
3. Flores, M.J., Gámez, J.A., Martínez, A.M., Puerta, J.M.: GAODE and HAODE: two proposals based on AODE to deal with continuous variables. In: Danyluk, A.P., Bottou, L., Littman, M.L. (eds.) ICML. ACM Int. Conf. Proc. Series, vol. 382, p. 40. ACM (2009)
4. Flores, M.J., Gámez, J.A., Martínez, A.M., Puerta, J.M.: Handling numeric attributes when comparing bayesian network classifiers: does the discretization method matter? Appl. Intell. 34(3), 372–385 (2011)
5. Yang, Y., Webb, G.I.: Non-disjoint discretization for naive-bayes classifiers. In: Sammut, C., Hoffmann, A. (eds.) Proc. of the 9th Int. Conf. on Mach. Learn (ICML 2002), pp. 666–673. Morgan Kaufmann, San Francisco (2002)
6. Zheng, Z., Webb, G.I.: Lazy Learning of Bayesian Rules. Mach. Learn. 41(1), 53–84 (2000)
7. Keogh, E.J., Pazzani, M.J.: Learning augmented Bayesian classifiers: A comparison of distribution-based and classification-based approaches. In: Proc. of the 7th Int. Workshop on AI and Statistics, pp. 225–230 (1999)
8. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann (2005)
9. Yang, Y., Webb, G.I.: Proportional k-Interval Discretization for Naive-Bayes Classifiers. In: Flach, P.A., De Raedt, L. (eds.) ECML 2001. LNCS (LNAI), vol. 2167, pp. 564–575. Springer, Heidelberg (2001)
10. Yang, Y., Webb, G.I.: Discretization for Naive-Bayes Learning: Managing Discretization Bias and Variance. Mach. Learn. 74(1), 39–74 (2009)
11. Frank, A., Asuncion, A.: UCI machine learning repository (2010), `http://archive.ics.uci.edu/ml`
12. Hettich, S., Bay, S.D.: The UCI KDD Archive (1999), `http://kdd.ics.uci.edu`
13. Fayyad, U.M., Irani, K.B.: Multi-interval discretization of continuousvalued attributes for classification learning. In: 13th Int. Joint Conf. on AI, vol. 2, pp. 1022–1027. Morgan Kaufmann (1993)
14. Domingos, P., Pazzani, M.J.: On the optimality of the simple bayesian classifier under zero-one loss. Mach. Learn. 29(2-3), 103–130 (1997)
15. Kohavi, R., Wolpert, D.H.: Bias plus variance decomposition for zero-one loss functions. In: Proc. of the 13th Int. Mach. Learn., pp. 275–283 (1996)
16. Dietterich, T.G.: Approximate statistical tests for comparing supervised classification learning algorithms. Neural Comput. 10, 1895–1923 (1998)
17. Webb, G.I., Conilione, P.: Estimating bias and variance from data (2002)
18. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. 7, 1–30 (2006)
19. Webb, G.I., Boughton, J., Zheng, F., Ting, K.M., Salem, H.: Learning by extrapolation from marginal to full-multivariate probability distributions: decreasingly naive Bayesian classification. Machine Learning (in-press)