Wrapper discretization by means of estimation of distribution algorithms

J.L. Flores, I. Inza and P. Larrañaga

We present a supervised wrapper approach to discretization. In contrast to many classical approaches, the discretiza-tion process is multivariate: all variables are discretized simultaneously, and the proposed discretization is evaluated with the Naive-Bayes classifier. The search for the optimal discretization is carried out as an optimization process with the learning model estimated accuracy guiding it. The global optimization algorithm is based on estimation of distribution algorithms, a set of novel algorithms which are special kinds of evolutionary algorithms. In order to evaluate the behaviour of the algorithm, an analysis of different parameters is performed by means of analysis of variance (ANOVA). The evaluation was carried out using artificial datasets, and with UCI datasets. The results suggest that the proposed method provides an effective and robust technique for discretizating variables.

K

1. Introduction

Machine learning tasks involve the handling of continuous and discrete information. Some limitations can exist with learning tasks with continuous variables. Some of these limitations are related to the nature of the data assumed the induction algorithms. For example, some learning algorithms make use of assumptions about normality that are frequently violated in real datasets. On the other hand, other limitations are related to the use of algorithms that solely require discrete information.

A common method of dealing with this problem is to discretize the continuous variables, i.e. by breaking variable values into ranges. This process has several advantages: a reduction in time to induce a classifier [11,17,21], a higher interpretability of the models [25] and an improvement in accuracy [30].

As a result, different discretization methods have been developed [21,22,35,56,69]. Some of them, known as supervised methods, make use of class information to perform the discretization process. The use of this information improves the search of intervals that discriminates class distribution better, i.e. intervals where the entropy is the lowest possible, as opposed to those in which the entropy is higher [21].

In addition, if it is possible to select intervals where the entropy is minimal, then ability of the classifier to discriminate instances is improved. Thus, different discretizations can increase or decrease classifier ability to discriminate instances, i.e. its accuracy. The increase or decrease in the accuracy of the classifiers will reflect the quality of the discretization.

The majority of the actual proposed methods do not take advantage of the classifier accuracy to carry out the discretization. Traditional approaches only take into account classifier accuracy to assess the quality of the algorithm. To the best of our knowledge, the methods that make use of this information are oriented to the induction of rules [1,26,39].

Our proposal makes use of classifier estimated accuracy to carry out a search of the best discretization. The classifier accuracy will be used as a guiding measure to search for the best discretization. The meta-heuristic process of search is carried out globally, i.e. it takes into account all variables simultaneously. This search can be supported by any optimization method due to the fact that the search for an optimal discretization is considered a NP-problem [13]. In our case estimation of distribution algorithms will be used as a heuristic to perform the search. The classification model will be a Naive-Bayes [18,40,49].

In this search there are many other factors that can influence the accuracy of the classifier: the number of intervals, the size of the database and, specifically in our algorithm, the population size. To analyze the influence of these parameters, a set of experiments with artificial data was carried out. All gathered results were analyzed by means of analysis of variance (ANOVA) [58]. This allows the testing of significant differences between means. Finally, a set of experiments with real data is performed to assess the effectiveness of the proposed approach.

The outline of the paper is as follows. In Section 2 the basis of the estimation of distribution algorithms is introduced. Section 3 presents the proposed wrapper discretization approach with detail. Section 4 describes the methodology of the experiments presented as well as the analysis performed with the results. Related works are presented in Section 5. Conclusions and future work are presented in Section 6.

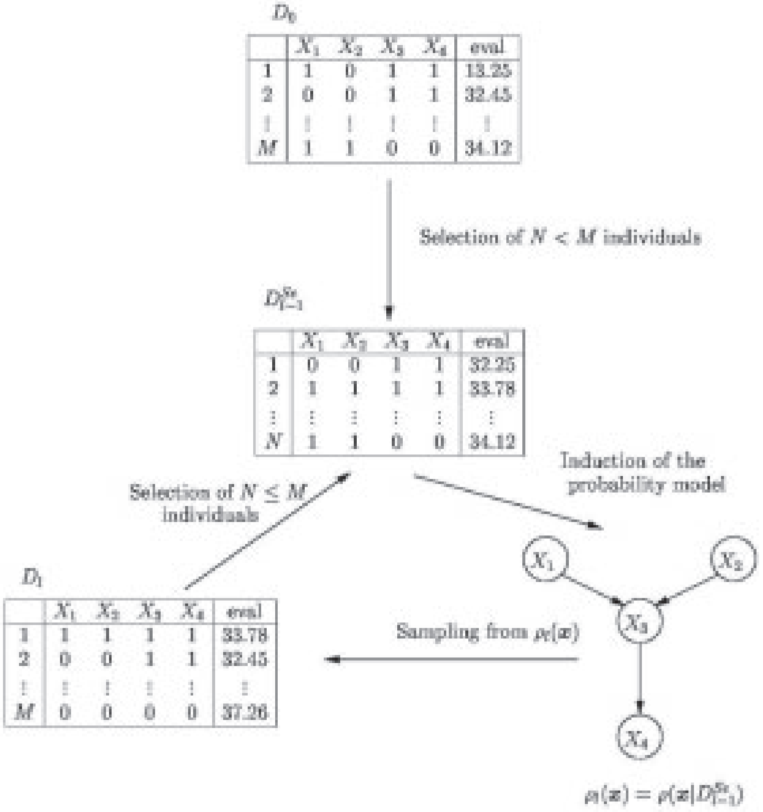
2. Estimation of Distribution Algorithms

Estimation of Distribution Algorithms (EDAs) [48] are a new evolutionary computation approach as an alternative to genetic algorithms. Our proposed algorithm will be supported by these kinds of evolutionary algorithms as an optimization tool in the search for the best discretization.

2.1. Overview

EDAs were introduced in the field of evolutionary computation in [48], although similar approaches can be found in [73]. In EDAs there are neither crossover nor mutation operators. Instead, the new population of individuals is sampled from a probability distribution, which is estimated from a database that contains the selected individuals from the previous generation. Thus, the interrelations between the different variables that represent the individuals are explicitly expressed through the joint probability distribution associated with the individuals selected at each generation. In order to understand the behavior of this heuristic better, a common outline for all EDAs follows:

1. Generate the first population of M individuals and evaluate each of them. Usually this generation is made assuming a uniform distribution on each variable.
2. N individuals are selected from the set of M, following a given selection method.
3. A n (size of the individual) dimensional probability model that shows the interdependencies among the variables is induced from the N selected individuals.



D_0

	X_1	X_2	X_3	X_4	eval
1	1	0	1	1	13.25
2	0	0	1	1	32.45
⋮	⋮	⋮	⋮	⋮	⋮
M	1	1	0	0	34.12

D_{l-1}^{Se}

	X_1	X_2	X_3	X_4	eval
1	0	0	1	1	32.25
2	1	1	1	1	33.78
⋮	⋮	⋮	⋮	⋮	⋮
N	1	1	0	0	34.12

D_l

	X_1	X_2	X_3	X_4	eval
1	1	1	1	1	33.78
2	0	0	1	1	32.45
⋮	⋮	⋮	⋮	⋮	⋮
M	0	0	0	0	37.26

Selection of $N < M$ individuals
Selection of $N \leq M$ individuals
Induction of the probability model
Sampling from $p_l(\mathbf{x})$
X_1
X_2
X_3
X_4
$p_l(\mathbf{x}) = p(\mathbf{x}|D_{l-1}^{Se})$

Fig. 1. EDA approach to optimization.

4. Finally, a new population of M individuals is generated based on the sampling of the probability distribution learnt in the previous step.

Steps 2, 3 and 4 are repeated until some stop criterion is met (e.g., a maximum number of generations, a homogeneous population or no improvement after a certain number of generations). A diagram of this process (for $n = 4$) can be seen in Fig. 1, and the pseudocode in Fig. 2.

The probabilistic graphical model learnt at each step has a significant influence on the behavior of the EDA (computing times and obtained results). Below we provide a classification of EDAs that uses as criterion the complexity of this probability model and the dependencies it considers:

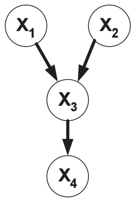
– Without dependencies: It is assumed that the n – dimensional joint probability distribution factorizes as a product of n univariate and independent probability distributions. Algorithms that use this model are, among others, *UMDA* [47].
– Bivariate dependencies: Only the dependencies between pairs of variables are taken into account. In this way, estimation of the joint probability can be done quickly. This group includes *MIMIC* [4].
– Multiple dependencies: All possible dependencies among the variables are considered without taking into account required complexity.
EBNA$_{\mathrm{BIC}}$ [19], *BOA* [51], Learning Factorized Distribution Algorithm (LFDA) [46] or *EGNA*$_{EE}$ [42] are some algorithms that belong to this group.

For detailed information about the characteristics and different algorithms that constitute the family of EDAs, see [43,54]. Theoretical results related to convergency and stability properties of EDAs can be

```
$D_0$ ← Generate $M$ individuals (the initial population) at random
**Repeat for** l=1,2,... until the stopping criterium is met
    $D_{l-1}^{Se}$ ← Select $N \leq M$ individuals from $D_{l-1}$ according to the
    selection method
    $p_l(\boldsymbol{x}) = p(\boldsymbol{x}|D_{l-1}^{Se})$ Estimate the joint probability distribution
    $D_l$ ← Sample M individuals (the new population) from $p_l(\boldsymbol{x})$
```

Fig. 2. EDA pseudocode.

Structure



Local probabilities

$$
\begin{aligned}
\boldsymbol{\theta_1} &= (\theta_{1-1}, \theta_{1-2}) & & p(x_1^1), p(x_1^2) \\
\boldsymbol{\theta_2} &= (\theta_{2-1}, \theta_{2-2}, \theta_{2-3}) & & p(x_2^1), p(x_2^2), p(x_2^3) \\
\boldsymbol{\theta_3} &= (\theta_{311}, \theta_{321}, \theta_{331}, & & p(x_3^1|x_1^1, x_2^1), p(x_3^1|x_1^1, x_2^2), p(x_3^1|x_1^1, x_2^3), \\
&\quad \theta_{341}, \theta_{351}, \theta_{361}, & & p(x_3^1|x_1^2, x_2^1), p(x_3^1|x_1^2, x_2^2), p(x_3^1|x_1^2, x_2^3), \\
&\quad \theta_{312}, \theta_{322}, \theta_{332}, & & p(x_3^2|x_1^1, x_2^1), p(x_3^2|x_1^1, x_2^2), p(x_3^2|x_1^1, x_2^3), \\
&\quad \theta_{342}, \theta_{352}, \theta_{362}, & & p(x_3^2|x_1^2, x_2^1), p(x_3^2|x_1^2, x_2^2), p(x_3^2|x_1^2, x_2^3), \\
\boldsymbol{\theta_4} &= (\theta_{411}, \theta_{421}, \theta_{412}, \theta_{422}) & & p(x_4^1|x_3^1), p(x_4^1|x_3^2), p(x_4^2|x_3^1), p(x_4^2|x_3^2)
\end{aligned}
$$

Factorization of the joint mass-probability

$$p(x_1, x_2, x_3, x_4) = p(x_1)p(x_2)p(x_3|x_1, x_2)p(x_4|x_3)$$

Fig. 3. Structure, local probabilities and resulting factorization for a Bayesian network with four variables (X_1, X_3, and X_4 with two possible values, and X_2 with three possible values).

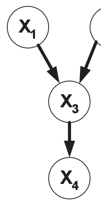
consulted in [27,71,72].

The algorithms of the last group (multiple dependencies) use different paradigms to codify the probabilistic model. The handling of dependencies leads to the estimation of a higher number of parameters and therefore a higher number of instances in order to correctly estimate all the parameters (see Figs 3 and 4). We will be interested in those probability models that do not consider dependencies between variables. The absence of dependencies enables considering simpler models, fewer parameters and even fewer instances to correctly estimate the parameters. Therefore, in some situations it will be not possible to build a model which is able to handle dependencies due. Because there are not enough instances to correctly estimate the parameters.

2.2. *The Univariate Marginal Distribution Algorithm* (*UMDA*)

Different approximations can be taken in order to perform the factorization of the model. We shall focus on those without dependencies due to the previously cited advantages, specifically on the Univariate

Structure



Local densities

$$\boldsymbol{\theta_1} = (m_1, -, v_1) \quad f_{X_1} \hookrightarrow \mathcal{N}(x_1; m_1, v_1)$$

$$\boldsymbol{\theta_2} = (m_2, -, v_2) \quad f_{X_2} \hookrightarrow \mathcal{N}(x_2; m_2, v_2)$$

$$\begin{array}{ll} \boldsymbol{\theta_3} = (m_3, \boldsymbol{b_3}, v_3) & f_{X_3|X_1=x_1,X_2=x_2} \hookrightarrow \\ \boldsymbol{b_3} = (b_{13}, b_{23})' & \quad \mathcal{N}(x_3; m_3 + b_{13}(x_1 - m_1) + b_{23}(x_2 - m_2), v_3) \end{array}$$

$$\begin{array}{ll} \boldsymbol{\theta_4} = (m_4, \boldsymbol{b_4}, v_4) & f_{X_4|X_3=x_3} \hookrightarrow \\ \boldsymbol{b_4} = (b_{34})' & \quad \mathcal{N}(x_4; m_4 + b_{34}(x_3 - m_3), v_4) \end{array}$$

Factorization of the joint density function

$$\begin{array}{l} f(x_1, x_2, x_3, x_4) = f(x_1)f(x_2)f(x_3|x_1, x_2)f(x_4|x_3) = \\ \frac{1}{\sqrt{2\pi v_1}} e^{-\frac{1}{2v_1}(x_1 - m_1)^2} \frac{1}{\sqrt{2\pi v_2}} e^{-\frac{1}{2v_2}(x_2 - m_2)^2} \\ \frac{1}{\sqrt{2\pi v_3}} e^{-\frac{1}{2v_3}(x_3 - (m_3 + b_{13}(x_1 - m_1) + b_{23}(x_2 - m_2)))^2} \\ \frac{1}{\sqrt{2\pi v_4}} e^{-\frac{1}{2v_4}(x_4 - (m_4 + b_{34}(x_3 - m_3)))^2} \end{array}$$

Fig. 4. Structure, local probabilities and resulting factorization for a Gaussian network with four variables.

Marginal Distribution Algorithm for continuous domains (*UMDA*$_c$). This algorithm was introduced by Larrañaga et al. [42,44] to learn the joint density function.

$$f_l(\boldsymbol{x}; \boldsymbol{\theta}^l) = \prod_{i=1}^{n} f_l(x_i; \theta_i^l) \tag{1}$$

It is usual to assume that the joint probability distribution follows a n-dimensional normal distribution, which can be factorized according to previous assumption by a product of n unidimensional and independent normal densities. The parameters of the joint probability distribution will be n means (μ_i) and n standard deviations (σ_i).

$$f(\boldsymbol{x}; \boldsymbol{\theta}) = \prod_{i=1}^{n} f_{\mathcal{N}}(x_i; \mu_i, \sigma_i^2) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{1}{2}(\frac{x_i - \mu_i}{\sigma_i})^2} \tag{2}$$

This particular case where all univariate distributions are normal will be denoted as *UMDA*$_c^G$ (Univariate Marginal Distribution Algorithm for Gaussian models). In *UMDA*$_c^G$ several steps are carried out. First a selection step, secondly an estimation of joint density function. The two parameters to be estimated at each generation and for each variable are the mean, μ_i^l, and the standard deviation, σ_i^l. It is well known that their respective maximum likelihood estimates are:

$$\widehat{\mu_i^l} = \overline{X}_i^l = \frac{1}{N} \sum_{r=1}^{N} x_{i,r}^l \, ; \, \widehat{\sigma_i^l} = \sqrt{\frac{1}{N} \sum_{r=1}^{N} (x_{i,r}^l - \overline{X_i^l})^2} \tag{3}$$

Finally, and in order to obtain a new population of individuals a sampling is performed based on the new joint density function.

```
Generate Initial Population
While not (Stopping Criterium is met) Do
        Foreach Individual Do
                Discretize Database
                Invididual(Score)=Classifier 10-CV accuracy
        End For
        Select the best 50% population
        Estimate $UMDA_c^G$ model
        Sample population from the $UMDA_c^G$ model
End While
```

Fig. 5. WEDA pseudocode.

3. Wrapper discretization by means of EDAs

3.1. Algorithm

Wrapper discretization by means of EDAs (WEDA) makes use of $UMDA_c^G$ search tool for discretization tasks in a wrapper way. The adaptation will consist of adding a new step in the $UMDA_c^G$ algorithm.

This new step will be previous to estimating the model. It will be comprised of two phases that will be applied to every individual of the population. The first phase will consist of discretizing the input database. In the second phase, previously discretized dataset plus class information associated to each instance is used to build a classifier. The classifier estimated accuracy will be the score of the applied discretization policy. We must point out that without class information it is not possible to build a classifier and, therefore, to score a discretization policy. Complete WEDA pseudocode is described in Fig. 5, and it will be described with detail hereinafter.

The main risk of the iterative process for WEDA is the possible overfitting that can occur leading to a low bias but a high variance [36]. So, to assess in a fair way the estimated predictive accuracy of WEDA, an external stratified 5-fold cross validation is performed (see Fig. 6). An external validation leads to a partitioning of the database. This division generates 5 parts of approximately the same size. Each part verifies that it has approximately the same proportion of class distribution of the original database class distribution. These 5 disjoint parts are combined to generate 5 new databases. Each new database is formed by two partial databases: one for learning a discretization policy including 80% of the original databas and the other 20% for testing purposes (see Fig. 6). The testing database contains unseen instances by WEDA.

Therefore, WEDA receives as input a partial training database. And the results are a discretized database and the policy used to discretize it. The discretization policy induced by WEDA is used to discretize the testing database. Finally, a Naive-Bayes is learnt with the partial training discretized database and tested with the testing discretized database.

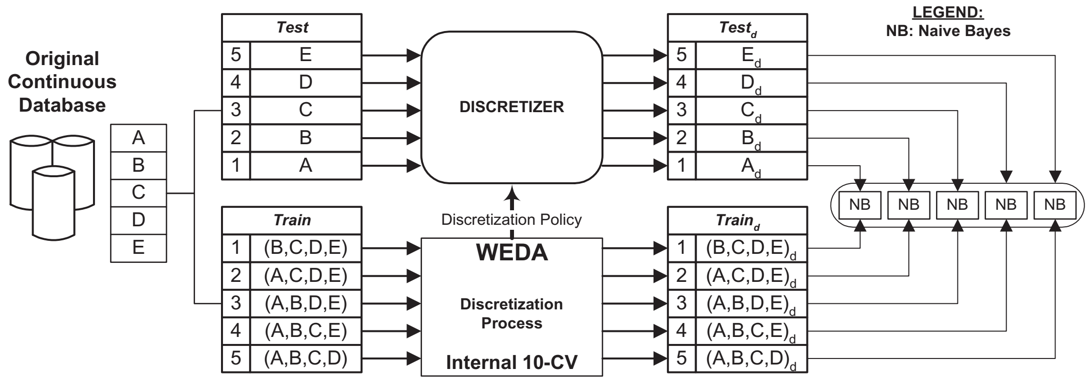


Fig. 6. WEDA external validation.

The whole process is repeated 5 times, once for each generated database. So, five Naive-Bayes are trained and tested, and the average accuracy is the individual accuracy.

3.2. Representation

In order to understand the way the discretization is represented, two key definitions will be provided. This representation will not be based on the database directly.

Let's suppose that we have a database with M examples and each of them is labeled with a class. This database can be denoted as $D = \{(\mathbf{x}^{(w)}, c^{(w)}), w = 1, ..., M\}$.

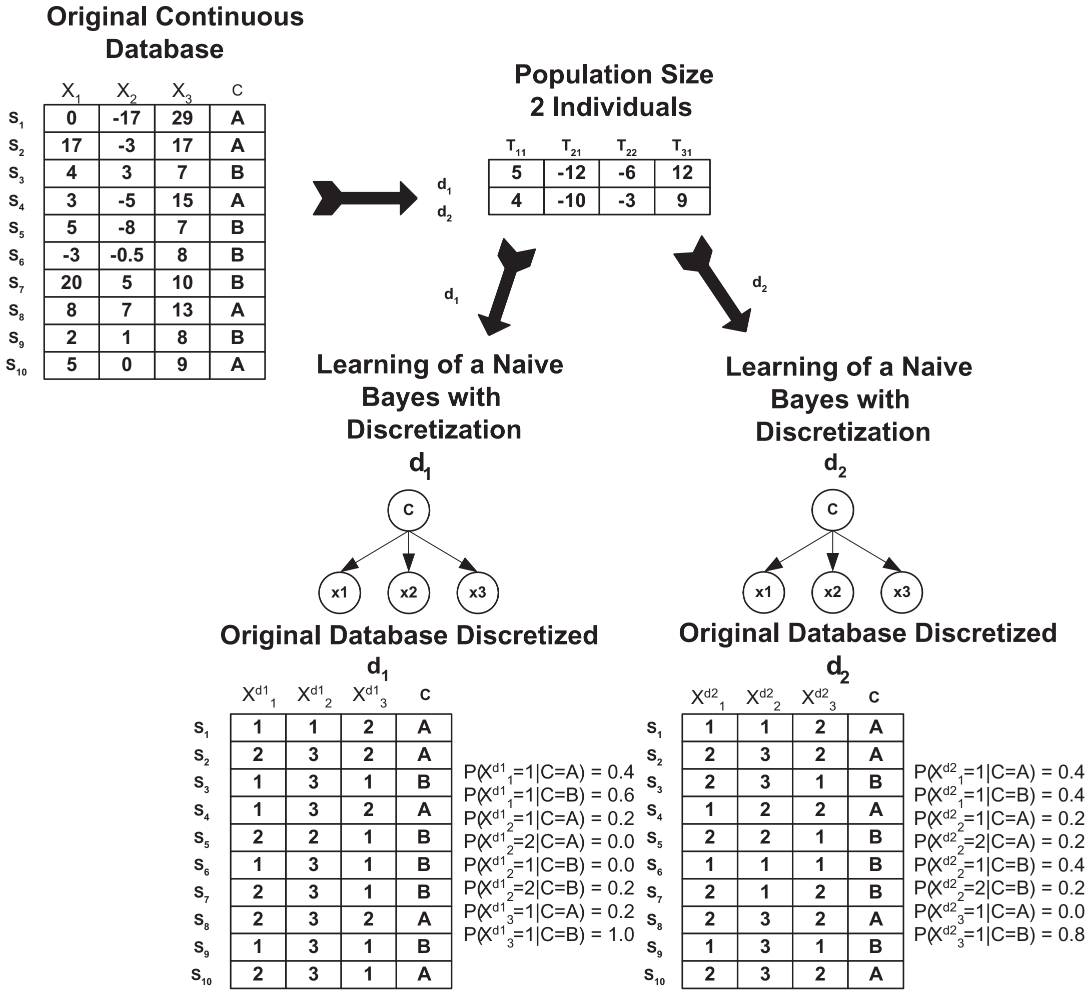
Definition 1. A **discretization sequence** for variable X_i is denoted as $\lambda^i =< t_1^i, \ldots, t_{k_i}^i >$ and it is an increasing sequence of real numbers (or cut-points). The sequence verifies $t_1^i < t_2^i < \ldots < t_{k_i}^i$. Based on this definition a function for each variable is defined $f_{\lambda^i} : \Omega_{X_i} \rightarrow \{1, \ldots, k_i\}$ as:

$$f_{\lambda^i}(x_i) = \begin{cases} 1 & \text{if } x < t_1^i \\ j & \text{if } t_j^i \leqslant x < t_{j+1}^i \quad j = 1, \ldots, k_i - 1 \\ k_i & \text{if } x \geqslant t_{k_i}^i \end{cases} \tag{4}$$

Definition 2. A **discretization policy** Λ is defined as a set of n discretization sequences, each of them related to each original continuous variable. When a policy $\Lambda =< \lambda^1, \ldots, \lambda^n >$ is applied to a database D, a new discretized database, $D^d = \Lambda(D)$, is obtained with $D^d = \{(\mathbf{x}^{(w)d}, c^{(w)}); w = 1, ..., M\}$. Note that class label remains invariable.

An individual of WEDA is defined as a discretization policy. Assuming that the original database has n continuous variables, we state the following:

- An individual represents n different discretization sequences, each of them corresponding to a univariate variable component. Each element of the discretization sequence t_j^i is considered as the value of a variable represented by WEDA as T_{ij}. The first subindex i in T_{ij} corresponds to superindex i in t_j^i, and the second subindex j corresponds to subindex j. It is important to note that T_{ij} variables are independent of each other given our factorization model.
- The number of variables used to represent an individual is $\sum_{i=1}^{n} k_i$.



Original Continuous Database

	X_1	X_2	X_3	C
s_1	0	-17	29	A
s_2	17	-3	17	A
s_3	4	3	7	B
s_4	3	-5	15	A
s_5	5	-8	7	B
s_6	-3	-0.5	8	B
s_7	20	5	10	B
s_8	8	7	13	A
s_9	2	1	8	B
s_{10}	5	0	9	A

Population Size 2 Individuals

	T_{11}	T_{21}	T_{22}	T_{31}
d_1	5	-12	-6	12
d_2	4	-10	-3	9

d_1
d_2
Learning of a Naive Bayes with Discretization d_1
Learning of a Naive Bayes with Discretization d_2
C
x1
x2
x3

Original Database Discretized d_1

	X^{d1}_1	X^{d1}_2	X^{d1}_3	C
s_1	1	1	2	A
s_2	2	3	2	A
s_3	1	3	1	B
s_4	1	3	2	A
s_5	2	2	1	B
s_6	1	3	1	B
s_7	2	3	1	B
s_8	2	3	2	A
s_9	1	3	1	B
s_{10}	2	3	1	A

$P(X^{d1}_1=1|C=A) = 0.4$
$P(X^{d1}_1=1|C=B) = 0.6$
$P(X^{d1}_2=1|C=A) = 0.2$
$P(X^{d1}_2=2|C=A) = 0.0$
$P(X^{d1}_2=1|C=B) = 0.0$
$P(X^{d1}_2=2|C=B) = 0.2$
$P(X^{d1}_3=1|C=A) = 0.2$
$P(X^{d1}_3=1|C=B) = 1.0$

Original Database Discretized d_2

	X^{d2}_1	X^{d2}_2	X^{d2}_3	C
s_1	1	1	2	A
s_2	2	3	2	A
s_3	2	3	1	B
s_4	1	2	2	A
s_5	2	2	1	B
s_6	1	1	1	B
s_7	2	1	2	B
s_8	2	3	2	A
s_9	1	3	1	B
s_{10}	2	3	2	A

$P(X^{d2}_1=1|C=A) = 0.4$
$P(X^{d2}_1=1|C=B) = 0.4$
$P(X^{d2}_2=1|C=A) = 0.2$
$P(X^{d2}_2=2|C=A) = 0.2$
$P(X^{d2}_2=1|C=B) = 0.4$
$P(X^{d2}_2=2|C=B) = 0.2$
$P(X^{d2}_3=1|C=A) = 0.0$
$P(X^{d2}_3=1|C=B) = 0.8$

Fig. 7. Discretization policy.

So, an individual will be a set of different variables representing different discretization sequences that form a discretization policy. Each variable will be sampled in order to provide a cut-point. This sampling is controlled by the population sampling step.

Let's use an example to clarify previous ideas. In Fig. 7 there is an example where we have 3 variables, X_1, X_2, X_3, in the original continuous database. Let's suppose that the first variable and the third variable are discretized in two intervals and the second in three intervals. This means that an individual will be composed of 3 different discretization sequences. The first discretization sequence will be composed of one variable T_{11}. This variable will be associated with variable X_1. The second discretization sequence will be composed of 2 variables T_{21}, T_{22}. These variables are associated with variable X_2. Finally the last discretization sequence will be composed of one variable T_{31}, whose value is associated with variable X_3.

Whole discretization policy is composed of the variables: $T_{11}, T_{21}, T_{22}, T_{31}$. Now, these variables will be sampled twice because in our example there are only two individuals that make up the population. Each individual representing a different discretization policy will discretize the original database, and subsequently a Naive-Bayes is learnt by means of 10-fold cross validation (cross validation process has

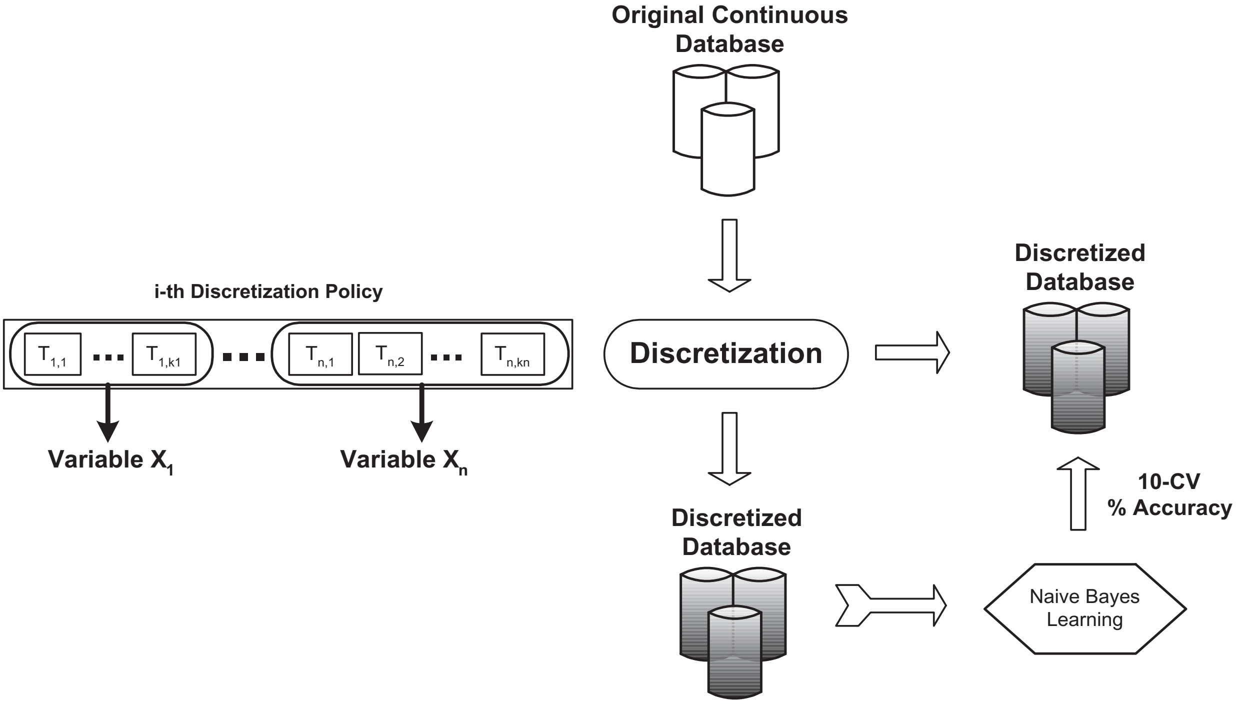


Fig. 8. Individual evaluation process.

been omitted due to space reasons).

3.3. Generating initial population

The mechanism to perform the generation of the initial population is carried out in several steps. The first step is to determinethe range of values for each X_i, obtaining its minimum and maximum values. The next step consists of building a uniform model for each T_{ij} associated with X_i, where the parameters are determined by the corresponding minimum and maximum values. In the last step each T_{ij} is sampled producing the initial population. This process is repeated for each X_i variable of the database.

3.4. Scoring a discretization policy

In order to evaluate an individual, two steps are followed (see Fig. 8). First, the original database is discretized by the proposed discretization policy. Secondly, the previously discretized database is used to perform a Naive-Bayes 10-fold cross validation with the selected classifier. The result of the previous 10-fold cross validation will be the score of the discretization policy and ,therefore, of the individual.

3.5. Selecting population and estimating model

After scoring the whole population, the best individuals are selected to build the model. Typically, 50% of the population is considered to be selected. The information taken into account to build the model is based on the different cut-points that integrate the different discretization sequences.

As every cut-point is a variable in WEDA, our model is represented by:

$$f_l(\mathtt{t}, \boldsymbol{\theta}^l) = \prod_{i=1}^{n}\prod_{j=1}^{k_i} f_l(t_{ij}, \boldsymbol{\theta}^l_{ij}) = \prod_{i=1}^{n}\prod_{j=1}^{k_i} \frac{1}{\sqrt{2\pi}\sigma^l_{ij}} e^{-\frac{1}{2}\left(\frac{t_{ij}-\mu^l_{ij}}{\sigma^l_{ij}}\right)^2} \tag{5}$$

where $\boldsymbol{\theta}^l$ is given by:

$$\boldsymbol{\theta}^l = (\mu^l_{11}, \ldots, \mu^l_{nk_n}, \sigma^l_{11}, \ldots, \sigma^l_{nk_n}) \tag{6}$$

3.6. Population sampling

The model previously estimated is used for generating the new population. To acomplish this process for each original variable X_i, it is necessary to simulate WEDA variables $(T_{i1}, \ldots, T_{ik_i})$ associated with it. The simulation will start with T_{i1} and will finish with the variable T_{ik_i}. However, a special situation can occur in this simulation: a variable T_{ij+1} can produce a cut-point with a value lower than the value produced by the variable T_{ij}, generating an unsorted and therefore invalid sequence.

When this occurs, the cut-point that invalidates the sequence is discarded. Then T_{ij+1} is resampled in order to obtain a cut-point that preserves the sorted and, therefore valid sequence. However, if succesive resamplings are not able to preserve the sequence valid, a different strategy is carried out. This strategy consists of discarding the whole discretization sequence, i.e. all cut-points produced by $T_{i1}, \ldots, T_{ij+1}$ are discarded. Then, the process restarts again simulating variables T_{i1},. . . , T_{ik_i}. This strategy is carried out only when the number of intervals is very high at the first iterations of the algorithm and the unsorted sequences tend to disspear quickly.

Finally, when all discretization sequences have been generated, a global discretization policy is obtained. This discretization policy is associated with an individual. Consequently, this process is repeated for all the individuals of the population.

4. Experiments

In this section, we present an analysis of WEDA under different parameters, and subsequently a set of results of applying WEDA in real datasets from UCI with the aim to assess its effectiveness.

4.1. Analysis

The analysis of WEDA behaviour was accomplished with a classical method for testing significant differences between means: ANalysis Of VAriance. ANOVA is one of the most suitable tools to carry out these kinds of tasks.

A 3-way ANOVA model was used to reflect our analysis. In our model we have three factors in ANOVA terminology:

- α – Number of intervals: 2, 4, 8.
- β – Database size: 50, 100, 200.
- γ – Population size: 10, 25, 75.

Table 1
ANOVA Results

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	17009,148[a]	26	654,198	1652,463	,000
Intercept	1925586,675	1	1925586,7	4863912	,000
α	8954,943	2	4477,471	11309,813	,000
β	4814,908	2	2407,454	6081,079	,000
γ	1824,063	2	912,032	2303,735	,000
$\alpha\beta$	872,673	4	218,168	551,079	,000
$\alpha\gamma$	137,746	4	34,437	86,985	,000
$\beta\gamma$	196,554	4	49,138	124,121	,000
$\alpha\beta\gamma$	208,261	8	26,033	65,757	,000
Error	96,202	243	0,396		
Total	1942692,025	270			
Corrected Total	17105,350	269			

[a] R Squared = 0.994 (Adjusted R Squared = 0.994).

The model with all parameters is:

$$Y_{ijkl} = \mu + \alpha_i + \beta_j + \gamma_k + (\alpha\beta)_{ij} + (\alpha\gamma)_{jk} + (\beta\gamma)_{jk} + (\alpha\beta\gamma)_{ijk} + E_{ijkl} \tag{7}$$

To acomplish this analysis it was necessary to generate databases with different sizes. Each generated database was based on an artificial model: a Naive-Bayes with a fixed structure, and a fixed set of parameters.

Once all databases were generated with suitable parameters, the experiments consisted of executing WEDA for each parameter combination using an external 5-fold cross validation for each parameter combination. Each combination was repeated 10 times due to the stochastic nature of WEDA. All gathered results have been grouped by the population size (see Fig. 9).

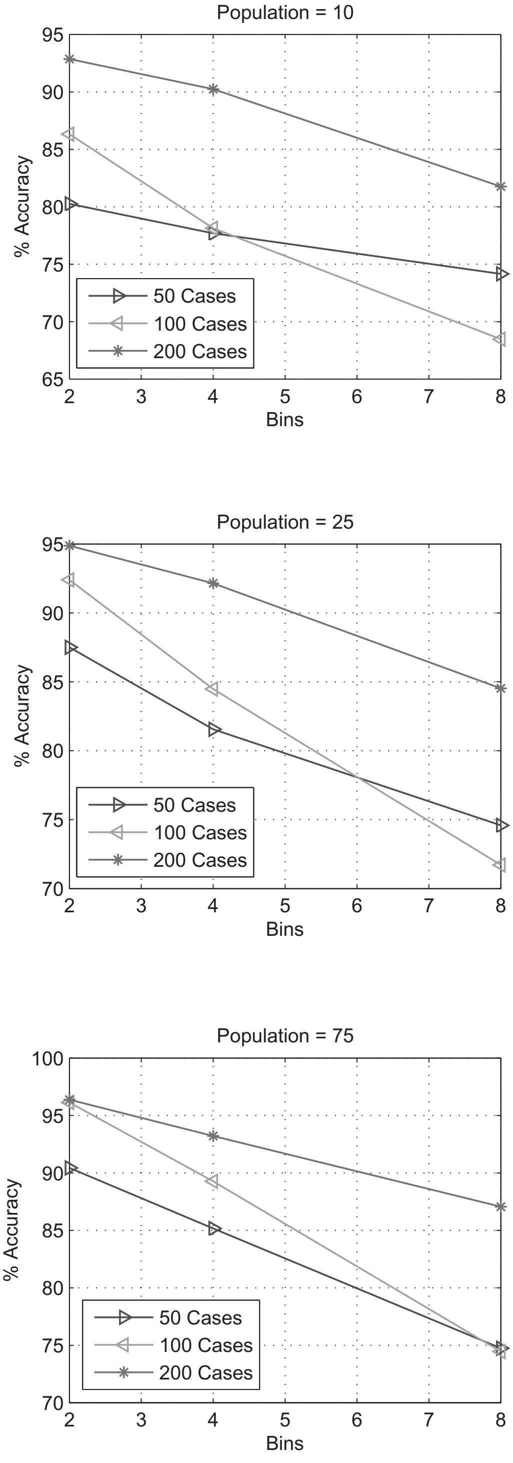
After gathering all the results, we proceed with the formal analysis of relationships between the different parameters. This task is acomplished performing all tests associated to our ANOVA model. The result was that all null hypothesis were rejected (see Table 1). In order to proceed with the statistical contrast, we should perform several tests, each of them related with the combination of the parameters. But, if we take into account the lowest value (the worst case) with only one test with a significance level of 5% at we can conclude if there are relationships between the parameters. Because observed F values are much bigger than F(2,243) (the worst and lowest value), it can be concluded that there are relationships between the different parameters and the different treatments of the parameters. As a result a change of the population size, a change of the number of intervals, and finally a change of the size of the dataset will have consequences on the classifier accuracy. In general, a change of whatever combination of the parameters will have consequences on the classifier accuracy.

Now, we proceed to extract several conclusions of this analysis as well as making some comments about the results.

Firstly, we can conclude that accuracy of WEDA results will depend on the database size, on the number of intervals and finally on the population size. Therefore, taking advantage of these conclusions an increase in the population size will have relevant importance. The performance improves as can be seen in Fig. 9.

Secondly, a relationship exists between the dataset size and the number of intervals as a second relevant conclusion.

And finally, we must point out that there is an increase of classifier accuracy when the dataset size increases (see Fig. 9). This can be an expected result because when the number of instances increases



Population Size	10	Number of intervals		
		2	4	8
Data Size	50	80.25– 1.17	77.69– 0.72	74.15– 0.31
	100	86.32– 1.28	78.12– 0.89	68.47– 0.58
	200	92.86– 0.23	90.23– 0.40	81.76– 0.50

Population Size	25	Number of intervals		
		2	4	8
Data Size	50	87.15– 0.79	81.56– 0.85	74.59– 0.52
	100	92.39– 0.53	84.48– 0.78	71.71– 0.34
	200	94.87– 0.36	92.15– 0.33	84.52– 0.43

Population Size	75	Number of intervals		
		2	4	8
Data Size	50	90.44– 0.72	85.17– 0.84	74.74– 0.38
	100	96.13– 0.64	89.27– 0.55	74.49– 0.41
	200	96.36– 0.23	93.21– 0.20	87.06– 0.32

Fig. 9. Summary of results of ANOVA. Graphical results for Population sizes: 10, 25 and 75. Numeric results summarized in tables for population sizes: 10, 25 and 75. Each cell shows the average and the deviation of the estimated predictive accuracy of 10 executions in a 5-fold cross validation.

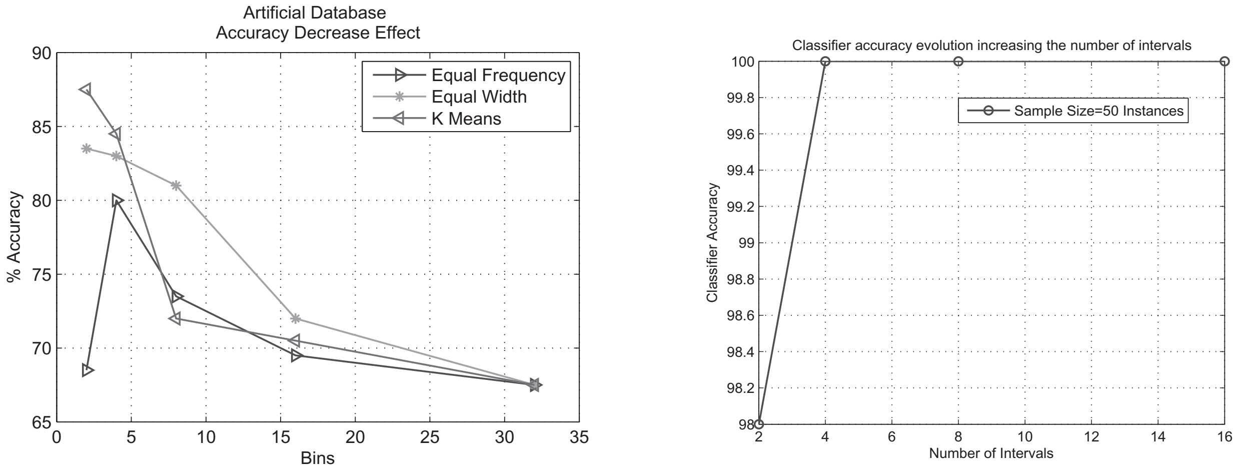


Fig. 10. Increase and decrease effect.

there is more information and the estimations can be closer to real data nature. There is also an increase of classifier accuracy when the population size increases; this is perhaps the most valuable result when WEDA is applied to real datasets.

On the other hand, we should note an unexpected effect detected. This effect is the decrease of accuracy when the number of intervals increases, contrary to theorical expected results [30]. This effect was also addressed by traditional algorithms such as Equal Frequency, Equal Width, and K-Means (see Fig. 10). This effect is due to the election of the parameters of the artificial Naive-Bayes model. The original parameters were modified in order to proceed with our study. The reason was that preliminary experiments (with 50 and 1600 instances) produced very high accuracies with classical algorithms with only two intervals. Furthermore, this accuracy reached 100% with more than two intervals with our proposal (see Fig. 10).

4.2. Real datasets

At this point, we present estimated accuracies obtained with WEDA proposed algorithm. We compare accuracies obtained with WEDA and the accuracies with respect to unsupervised approaches and supervised approaches. We have considered the following unsupervised approaches: Equal Frequency, Equal Width, K-Means and Unsupervised Monothetic Contrast. Supervised approaches were: Fayyad and Irani Entropy (available in WEKA [66]) and Class-Attribute Interdependence Maximization (CAIM) [38], a more modern supervised approximation to discretization.

The results were obtained in 10 UCI datasets, which only contain continuous predictor variables. The main characteristics of the datasets included are summarized in Table 2: dataset name, number of classes, number of variables and the number of instances.

Each dataset was tested with each algorithm by an external 5-fold stratified cross validation process using a classification paradigm in order to estimate the predictive accuracies. In the case of WEDA, experiments were carried out with a population of 75 individuals and repeated 10 times due to its stochastic nature. The whole process was repeated with three different classification paradigms: a Naive-Bayes, a C4.5, and a k-NN.

The results are summarized in two tables: Tables 3 and 4, where each column represents each discretization algorithm and each row represents each dataset. Therefore, each cell contains the average

Table 2
UCI Datasets

#	Data Set	Num. Class Values	Num. Variables	Num. Instances
1	Balance	3	4	625
2	Bupa	2	6	246
3	Hayes	3	4	160
4	Image	7	20	2310
5	Ionosphere	2	35	351
6	Iris	3	4	150
7	Liver	2	6	345
8	Pima	2	8	768
9	Vehicle	4	19	846
10	Wine	3	13	179

Table 3
Summary of the estimated predictive accuracy with 2 bins. Classical unsupervised approaches compared with WEDA

	Equal Frequency	Equal Width	K Means	Unsupervised Monothetic Contrast	WEDA +C4.5	WEDA + k-NN	WEDA + Naive Bayes
# 1	73.02 ± 2.99	73.02 ± 2.99	77.50 ± 3.37	**77.82 ± 4.07**	77.64 ± 2.58	77.75 ± 0.41	77.40 ± 1.07
# 2	**66.38 ± 3.14**	56.23 ± 3.75	57.68 ± 1.21	58.84 ± 2.63	57.10 ± 0.70	58.14 ± 1.33	58.44 ± 0.59
# 3	56.45 ± 8.06	37.73 ± 5.79	51.85 ± 6.05	31.73 ± 5.80	**60.24 ± 2.39**	46.94 ± 4.11	57.60 ± 1.24
# 4	67.75 ± 0.83	71.17 ± 2.29	72.77 ± 1.26	72.42 ± 1.66	**92.15 ± 0.41**	91.38 ± 0.57	88.08 ± 0.27
# 5	66.49 ± 22.81	68.23 ± 26.54	63.08 ± 22.83	65.64 ± 22.93	90.52 ± 0.85	87.35 ± 0.52	**96.02 ± 0.61**
# 6	76.00 ± 2.81	73.99 ± 5.96	78.67 ± 3.80	78.67 ± 5.05	**95.47 ± 0.27**	95.20 ± 0.42	94.80 ± 0.00
# 7	**68.12 ± 8.39**	55.36 ± 4.15	56.23 ± 6.59	57.10 ± 4.53	57.71 ± 1.13	58.32 ± 1.64	60.33 ± 4.13
# 8	72.26 ± 3.39	68.49 ± 4.74	72.53 ± 1.83	74.15 ± 1.57	**74.53 ± 0.40**	73.62 ± 0.41	73.22 ± 0.53
# 9	45.68 ± 3.94	39.78 ± 2.87	43.28 ± 3.87	42.51 ± 2.66	66.36 ± 1.37	**66.38 ± 1.00**	61.34 ± 1.02
# 10	96.11 ± 3.85	88.55 ± 4.43	93.75 ± 6.75	93.17 ± 6.84	94.05 ± 1.03	94.87 ± 0.98	**96.38 ± 1.45**

and the deviation of the estimated predictive accuracy of the corresponding discretization algorithm with the dataset. The best values in the table are marked in bold.

As suggested by Demsar [15], a Wilcoxon paired signed-rank test has been used in order to compare our proposed algorithm with unsupervised classical approaches on the group of experimented datasets. To accomplish the comparison four tests have been carried out, all of them with a significance level of 5%. The first test compares Equal Frequency with WEDA using the results obtained by Equal Frequency in the 10 UCI datasets and the results obtained by WEDA in the same datasets, i.e. the results contained in the columns corresponding to Equal Frequency and to WEDA in Table 3. The second test compares Equal Width with WEDA in the same way, i.e. using the results obtained by Equal Width in the 10 UCI datasets and the results obtained by WEDA in the same datasets. The third test compares K Means and WEDA in the same way, and the fourth test compares Unsupervised Monothetic Contrast and WEDA also in the same way.

So, according to the Wilcoxon's test, for a confidence of $\alpha = 5\%$. We therefore reject all null hypothesis with the exception of Equal Frequency algorithm that in only two datasets (Bupa and Liver) that is ahead of the rest of the algorithms. WEDA seems to induce better discretization policies than classical approaches.

On the other hand, when we observe the results with supervised approximations (see Table 4), these show that supervised approximations seem to be more competitive than classical unsupervised approximations. But we have to point out that in many cases supervised approximations generate a very high number of intervals. And in a few cases, some datasets were discretized in only two intervals. An example is the image dataset where a variable is discretized in 14 intervals.

Table 4
Summary of the estimated predictive accuracy. Supervised approaches compared with WEDA

	Fayyad Irani Entropy	CAIM	WEDA + C4.5	WEDA + k-NN	WEDA + Naive Bayes
# 1	73.02 ± 2.99	73.02 ± 2.99	77.64 ± 2.58	**77.75 ± 0.41**	77.40 ± 1.07
# 2	52.17 ± 8.20	**62.90 ± 6.84**	57.10 ± 0.70	58.14 ± 1.33	58.44 ± 0.59
# 3	60.14 ± 2.56	53.87 ± 4.50	**60.24 ± 2.39**	46.94 ± 4.11	57.60 ± 1.24
# 4	88.48 ± 0.18	80.35 ± 3.31	**92.15 ± 0.41**	91.38 ± 0.57	88.08 ± 0.27
# 5	88.91 ± 3.87	89.75 ± 4.74	90.52 ± 0.85	87.35 ± 0.52	**96.02 ± 0.61**
# 6	93.33 ± 2.36	69.53 ± 2.58	**95.47 ± 0.27**	95.20 ± 0.42	94.80 ± 0.00
# 7	58.28 ± 0.68	**61.19 ± 4.30**	57.71 ± 1.13	58.32 ± 1.64	60.33 ± 4.13
# 8	**75.76 ± 3.58**	72.91 ± 5.02	74.53 ± 0.40	73.62 ± 0.41	73.22 ± 0.53
# 9	61.26 ± 4.02	46.55 ± 1.61	66.36 ± 1.37	**66.38 ± 1.00**	61.34 ± 1.02
# 10	98.24 ± 2.63	**98.24 ± 2.63**	94.05 ± 1.03	94.87 ± 0.98	96.38 ± 1.45

We can conclude that WEDA seems to perform better than classical unsupervised algorithms but can be less competitive than supervised approaches when the number of intervals is very high. This performance improvement is related to the use of classifier accuracy information and the global way of accomplishing discretization. But this global way of performing discretization leads to a cost in CPU that will be detailed next.

The cost in CPU is the most critical aspect in our approach. In order to assess this, Figs 11 and 12 are presented. In these figures we can see the accuracy evolution over time in all datasets. The accuracy evolution showed is the average accuracy obtained in all executions performed by WEDA.

Several conclusions can be extracted from these figures. The first one is that the accuracy evolution shows that in most cases WEDA requires only one internal iteration to reach the same accuracy as some classical approaches. And with 15 iterations WEDA reaches the same accuracy as all classical approaches in all datasets, excluding those cases where WEDA is inferior. General behaviour of WEDA indicates that the cost in CPU does not need to be necessarily high to outperform classical approaches as few iterations are required to reach the same accuracies.

5. Related work

Since 1986, when the first algorithms were published, the discretization methods have proved to be very useful in machine learning algorithms. Nowadays, many different kinds of approaches have been applied to discretization. Dougherty et al. [17] shows a summary of these methods but Yang [68] has perhaps the best taxonomy of the different discretization methods. We will describe them, and a new category will be added:

- *Supervised vs. unsupervised* [17]. Methods that use class information to improve the quality of their discretization or procedures which ignore this information.
- *Univariate vs. multivariate* [2]. Multivariate methods take the relationships between variables into account and univariate methods discretizes each variable in isolation.
- *Parametric vs. non-parametric*. Parametric methods require some input from the user such as the number of intervals, while non-parametric methods do not require any input, they only use information from data.
- *Hierarchical vs. non-hierarchical*. Hierarchical methods begin with a pre-defined discretization that use *split* or *merge* [33] operations. If methods that use "splitting" equation are used we start with one interval and continue discretization making split operations. In merge methods we start with the

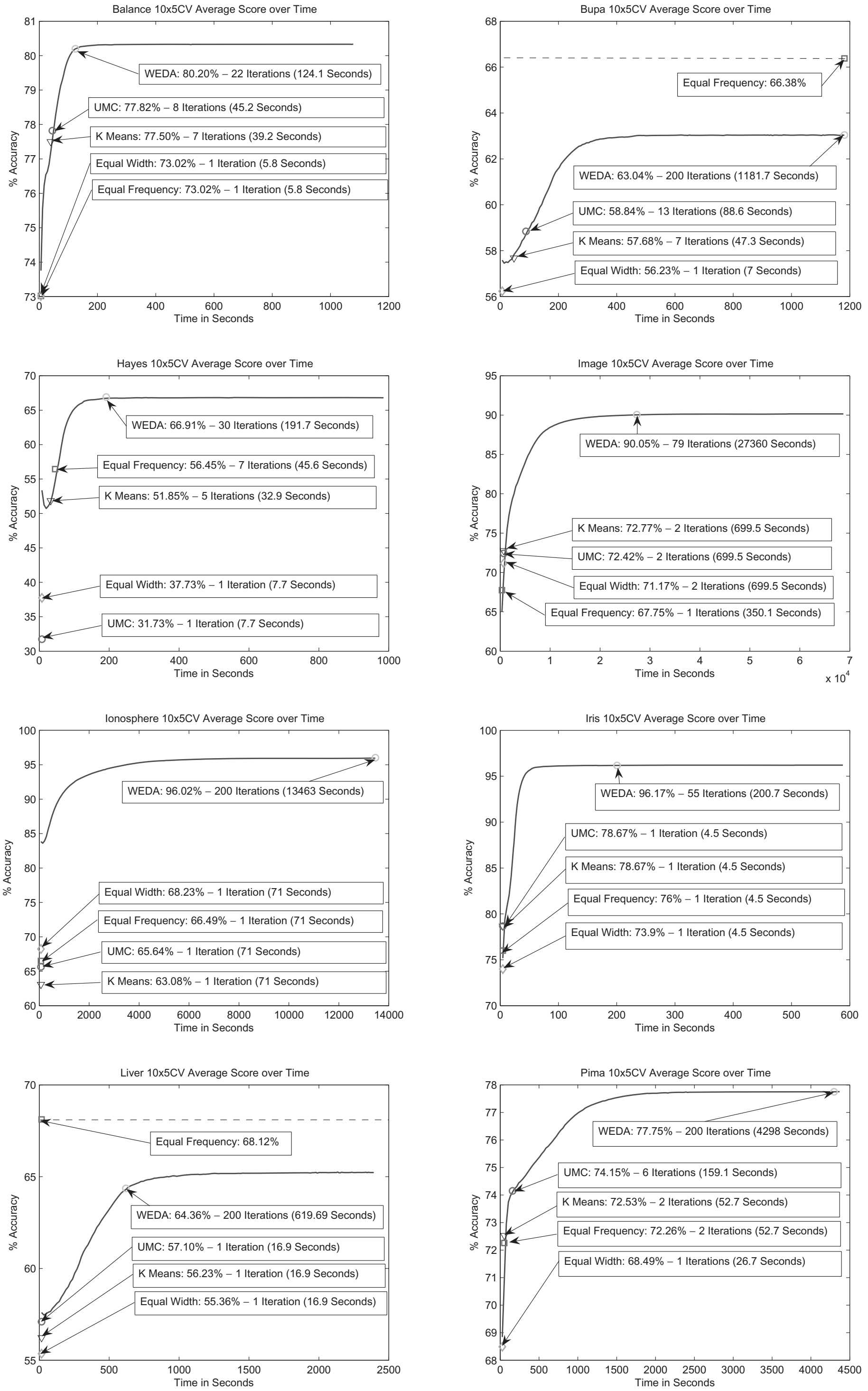


Fig. 11. The graphics show the evolution of accuracy in WEDA over time using the external cross validation with UCI databases (Part I). Each graphic shows the accuracy obtained with classical approaches, the number of iterations required by WEDA to reach the same accuracy and the time involved.

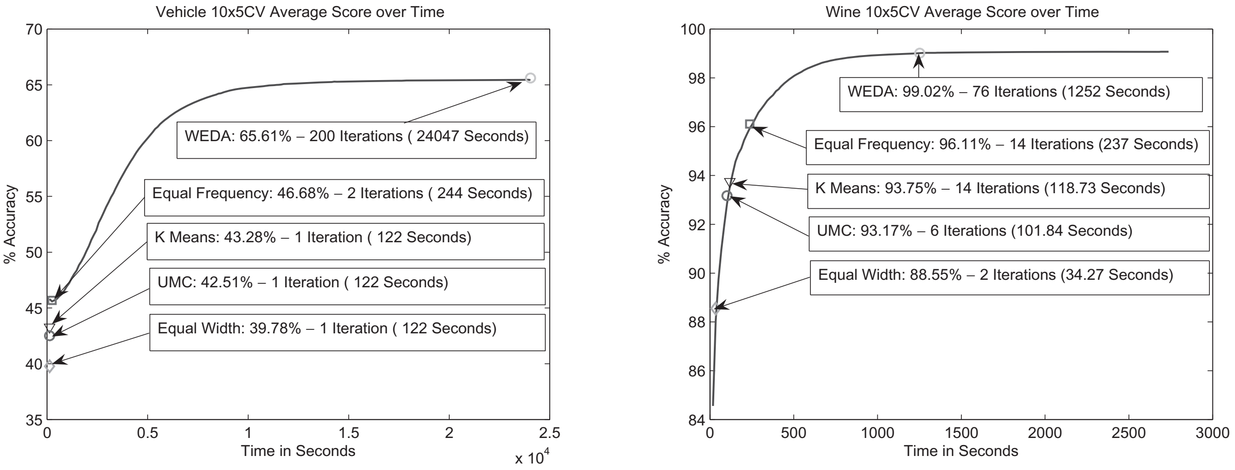


Fig. 12. The graphics show the evolution of accuracy over time in an external cross validation with UCI databases (Part II).

whole dataset as candidate intervals and, with some criterium, merge operations are made in adjacent intervals. Non-Hierarchical methods do not start with a predefined set of intervals that do not form any hierarchy. For example, these methods can scan the ordered values once and sequentially form the intervals.

- *Global vs. local* [17]. Global methods perform the discretization process using the same number of regions for the whole data, and discretization is performed once. Local methods allow different sets of intervals.
- *Eager vs. lazy* [31]. Eager methods carry out discretization before begining the classification task. Lazy methods perform discretization during the learning process, and both tasks are performed at the same time.
- *Disjoint vs. non-disjoint*. Disjoint methods perform discretization creating disjoint intervals. Non-disjoint methods can use non-disjoint intervals. Some learning models can impose conditions over the intervals whereas other learning models such as Naive-Bayes do not impose any conditions as regards the intervals [70].
- *Filter vs. wrapper*. This new axis is introduced in order to enhance the taxonomy. It takes into account new recent methods (such as methods based on genetic algorithms [63]). Filter methods do not use the clasification algorithm itself and the discretization process is independent of the employed classification procedure. Wrapper methods take classification accuracy information into account in the discretization process, this information will guide the search.

There has been an evolution in discretization methods, many of them make use of class information to improve classification but with different approaches. The class information is used to find partitions which discriminate the class distribution between groups in a better way. An optimal discrimination of the class distribution has a direct impact in classifier accuracy. The classifier accuracy captures the quality of the discrimination of the class distribution. WEDA makes use of classifier accuracy to guide the search of the best discrimination. There are some approximations that performs discretization as internal process and not as an aim. The induction of optimal rules is the aim and not a discretization proposal [26].

6. Conclusions

In this work a novel discretization algorithm with a global wrapper approach based on EDAs was presented. The algorithm makes use of classifier estimated accuracy to guide the search of the optimal discretization in a global way. The behaviour of the proposed approach was studied using artificial datasets and subsequently compared to classical approaches in ten UCI datasets by means of the estimated predictive accuracy. The results of the experiments demonstrate that the use of classifier information obtains very competitive discretization policies compared to the approaches of the traditional algorithms. On the other hand, the ability to be used with other classification paradigms is essential to take advantage of new upcoming discrete classifiers. The only drawback is a higher CPU cost than classical approaches.

A future work line consists of making use of the relationship of the discretization policy variables in WEDA.

Acknowledgements

This work has been possible thanks to the *SAIOTEK* and *ETORTEK* programs of the Basque Government, and to the project TIN2005-03824 of the Spanish Ministry of Education and Science.

References

[1] J. Bacardit and J.M. Garrell, *Evolving Multiple Discretizations with Adaptive Intervals for a Pittsburgh Rule-Based Learning Classifier System*, In Proceedings of the Genetic and Evolutionary Computation Conference, volume 2724 of Lecture Notes in Computer Science, pages 1818–1831, Chicago (Illinois,USA), 2003, Springer-Verlag.

[2] S.D. Bay, *Multivariate Discretization of Continuous Variables for Set Mining*, In Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 315–319, Boston (Massachusetts, USA), 2000. ACM Press.

[3] R. Blanco, I. Inza and P. Larrañaga, Learning Bayesian networks in the space of structures by estimation of distribution algorithms, *International Journal of Intelligent Systems* **18**(2) (2003), 205–220.

[4] J.S. De Bonet, C. Isbell and P. Viola, MIMIC: Finding optima by estimating probability densities, in: *Advances in Neural Information Processing*, (Vol. 9), M. Mozer M. Jordan and M. Perrone, eds, Denver 1996, MIT Press, Cambridge.

[5] M. Boullé, *Khiops: A Discretization Method of Continuous Attributes with Guaranteed Resistance to Noise*, In Proceedings of the 3rd International Conference on Machine Learning and Data Mining, Volume 2734 of Lecture Notes in Computer Science, Leipzig (Germany), 2003, Springer-Verlag, 50–64.

[6] M. Boullé, Khiops: A statistical discretization method of continuous attributes, *Machine Learning* **55**(1) (2004), 53–69.

[7] M. Boullé, *A Grouping Method for Categorical Attributes having Very Large Number of Values*, In Proceedings of the 4th International Conference on Machine Learning and Data Mining in Pattern Recognition, Volume 3587 of Lecture Notes in Computer Science, pages 228–242, Leipzig (Germany), 2005, Springer-Verlag.

[8] M. Boullé, Optimal bin number for equal frequency discretizations in supervised learning, *Intelligent Data Analysis* **9**(2) (2005), 175–188.

[9] R. Butterworth, D.A. Simovici, G.S. Santos and L. Ohno-Machado, A greedy algorithm for supervised discretization, *Journal of Biomedical Informatics* **37**(4) (2004), 285–292.

[10] E. Castillo, J.M. Gutierrez and A.S. Hadi, *Expert Systems and Probabilistic Network Models*, Springer-Verlag, 1996.

[11] J. Catlett, *On Changing Continuous Attributes into Ordered Discrete Attributes*, In Proceedings of the 5th European Working Session on Learning, Vol. 482 of Lecture Notes in Computer Science, pages 164–178, Porto (Portugal), 1991, Springer-Verlag.

[12] J.Y. Ching, A.C. Wong and K.C.C. Chan, Class-dependent discretization for inductive learning from continuous and mixed-mode data, *IEEE Transactions on Pattern Analysis and Machine Learning* **17**(7) (1995), 641–651.

[13] B.S. Chlebus and S.H. Nguyen, *Discretization Problem for Rough Set Methods*, In Proceedings of the 1st International Conference on Rough Sets and Current Trend in Computing, Volume 1424 of Lecture Notes in Computer Science, pages 545–552, Warsaw (Poland), 1998, Springer-Verlag.

[14] E. Consortium, *Elvira: An Environment for Creating and using Probabilistic Graphical Models*, In Proceedings of the 1st European Workshop on Probabilistic Graphical Models, Cuenca (Spain), 2002.

[15] J. Demsar, Statistical comparisons of classifiers over multiple data sets, *Journal of Machine Learning Research* **7** (2006), 1–30.

[16] P. Domingos and M. Pazzani, On the optimality of the simple Bayesian classifier under zero-one loss, *Machine Learning* **29**(2–3) (1997), 103–130.

[17] J. Dougherty, R. Kohavi and M. Sahami, *Supervised and Unsupervised Discretization of Continuous Features*, In Proceedings of the 12th International Conference on Machine Learning, Lecture Notes in Artificial Intelligence, pages 194–202, San Francisco (CA, USA), 1995.

[18] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, John Wiley and Sons, 1973.

[19] R. Etxeberria and P. Larrañaga, *Global Optimization using Bayesian Networks*, In Proceedings of the 2nd Symposium on Articial Intelligence Adaptive Systems, Volume 9, pages 332–339, La Habana (Cuba), 1999.

[20] U.M. Fayyad and K.B. Irani, On the handling of continuous-valued attributes in decision tree generation, *Machine Learning* **8**(1) (1992), 87–102.

[21] U.M. Fayyad and K.B. Irani, *Multi-Interval Discretization of Continuousvalued Attributes for Classification Learning*, In Proceedings of the 13th International Joint Conference on Artificial Intelligence, pages 1022–1027, Chambery (France), 1993. Morgan Kaufmann.

[22] S. Ferrandiz and M. Boullé, *Multivariate Discretization by Recursive Supervised Bipartition of Graph*, In Machine Learning and Data Mining, Volume 3587 of Lecture Notes in Artificial Intelligence, pages 253–264, Leipzig (Germany), 2005, Springer-Verlag.

[23] S. Ferrandiz and M. Boullé, *Supervised Evaluation of Dataset Partitions: Advantages and Practice*, In Machine Learning and Data Mining, Volume 3587 of Lecture Notes in Artificial Intelligence, pages 600–609, Leipzig (Germany), 2005, Springer-Verlag.

[24] J. Gama, *Dynamic Discretization of Continuous Attributes*, In Proceedings of the 6th Iberoamerican Conference on Artificial Intelligence, Lecture Notes in Artificial Intelligence, pages 160–169, Lisbon (Portugal), 1998.

[25] P. Geurts and L. Wehenkel, *Investigation and Reduction of Discretization Variance in Decision tree Induction*, In Proceedings of the 11th European Conference on Machine Learning, Volume 1810 of Lecture Notes in Computer Science, pages 162–170, Cataluña (Spain), 2000, Springer-Verlag.

[26] R. Giráldez, J.S. Aguilar-Ruiz and J.C. Riquelme Santos, *Natural Coding: A More Efficient Representation for Evolutionary Learning*, In Proceedings of the Genetic and Evolutionary Computation Conference, Vol. 2723 of Lecture Notes in Computer Science, pages 979–990, Chicago (Illinois,USA), 2003, Springer-Verlag.

[27] C. González, J.A. Lozano and P. Larrañaga. Mathematical modeling of UMDA algorithm with tournament selection, *International Journal of Approximate Reasoning* (31) (2002), 313–340.

[28] K. Grabczewski, *SSV Criterion Based Discretization for Naive Bayes Classifiers*, In Proceedings of the International Conference on Artificial Intelligence and Soft Computing, Volume 3070 of Lecture Notes in Computer Science, pages 574–579, Zakopane (Poland), 2004, Springer-Verlag.

[29] R.C. Holte, Very simple classification rules perform well on most commonly used datasets, *Machine Learning* **11**(1) (1993), 63–91.

[30] C.-N. Hsu, H-J. Huang and T.-T. Wong, Implications of the Dirichlet assumption for discretization of continuous variables in naive Bayesian classifiers, *Machine Learning Journal* **53**(3) (2002).

[31] H. Huang and C. Hsu, *Bayesian Classification for Set and Interval Data*, In Proceedings 2000 International Computer Symposium, ChiaYi (Taiwan), 2000.

[32] I. Inza, M. Merino, P. Larrañaga, J. Quiroga, B. Sierra and M. Girala, Feature subset selection by genetic algorithms and estimation of distribution algorithms. A case study in the survival of cirrhotic patients treated with TIPS, *Artificial Intelligence in Medicine* **23**(2) (2001), 187–205.

[33] R. Kerber, *Chimerge: Discretization of Numeric Attributes*, In Proceedings of the 10th National Conference on Artificial Intelligence, pages 123–128, San Jose (California,USA), 1992.

[34] P. Knotkanen, P. Myllymaki and H. Tirri, *A Bayesian Approach to Discretization*, In Proceedings of the European Symposium on Intelligent Techniques, pages 265–268, Bari (Italy), 1997.

[35] R. Kohavi and M. Sahami, *Error-Based and Entropy-Based Discretization of Continuous Features*, In Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, pages 114–119, Portland (Oregon,USA), 1996.

[36] R. Kohavi and D.H. Wolpert, *Bias Plus Variance Decomposition for Zeroone Loss Functions*, In Lorenza Saitta, editor, Proceedings of the 13th International Conference on Machine Learning, pages 275–283. Morgan Kaufmann, 1996.

[37] A.V. Kozlov and D. Koller, *Nonuniform Dynamic Discretization in Hybrid Networks*, In Proceedings of the 13th Conference on Uncertainy in Artificial Intelligence, pages 314–325, Providence (Rhode Island, USA), 1997.

[38] L. Kurgan, Discretization algorithm that uses class attribute interdependence maximization, *IEEE Transactions on Knowledge and Data Engineering* **16**(2) (February 2004), 145–163.

[39] W. Kwedlo and M. Kretowski, *An Evolutionary Algorithm using Multivariate Discretization for Decision Rule Induction*, In Proceedings of the Third European Conference on Principles of Data Mining and Knowledge Discovery, Volume 1704 of Lecture Notes in Artificial Intelligence, pages 392–397, Prague (Czech Republic), 1999, Springer-Verlag.

[40] P. Langley, W. Iba and K. Thompson, *An analysis of Bayesian Classifiers*, In Proceedings of the 10th International Conference on Artificial Intelligence, pages 223–228, 1992.

[41] P. Larrañaga, R. Etxeberria, J.A. Lozano and J.M. Peña, Combinatorial optimization by learning and simulation of Bayesian networks. In Proceedings of the 16th Annual Conference on Uncertainty in Artificial Intelligence (UAI-00), pages 343–352, San Francisco (USA), 2000. Morgan Kaufmann.

[42] P. Larrañaga, R. Etxeberria, J.A. Lozano and J.M. Peña, *Optimization in Continuous Domains by Learning and Simulation of Gaussian Networks*, In Proceedings of the 2000 Genetic and Evolutionary Computation Conference Workshop Program, pages 201–204, 2000.

[43] P. Larrañaga and J.A. Lozano, *Estimation of Distribution Algorithms*, A New Tool for Evolutionary Computation. Kluwer Academic Publisher, 2002.

[44] P. Larrañaga, J.A. Lozano and J.M. Peña, *Optimization by Learning and Simulation of Bayesian and Gaussian Networks*, Technical Report Technical Report KZZA-IK-4-99, Department of Computer Science and Artificial Intelligence, University of the Basque Country, 1999.

[45] L. Liu, A.K.C. Wong and Y. Wang, A global optimal algorithm for class-dependent discretization of continuous data, *Intelligent Data Analysis* **8**(2) (2004), 151–170.

[46] H. Mühlenbein and T. Mahnig, FDA – a scalable evolutionary algorithm for the optimization of additively decomposed functions, *Evolutionary Computation* **7**(4) (1999), 353–376.

[47] H. Mühlenbein, The equation for the response to selection and its use for prediction, *Evolutionary Computation* **5**(3) (1998), 303–346.

[48] H. Mühlenbein and G. Paass, *From Recombination of Genes to the Estimation of Distributions I. Binary Parameters*, In Proceedings of the 4th International Conference on Parallel Problem Solving, Volume 1141 of Nature Lecture Notes in Computer Science, pages 178–187, Berlin (Germany), 1996.

[49] M. Minsky, Steps toward artificial intelligence, *Transactions on Institute of Radio Engineers* **49** (1961), 8–30.

[50] M.J. Pazzani, *Beyond Independence: Conditions for the Optimality of the Simple Bayesian Classifier*, In Proceedings of the 13th International Conference on Machine Learning, Volume 29, pages 105–112, Bari (Italy), 1996.

[51] M. Pelikan and D.E. Goldberg, *Genetic Algorithms, Clustering, and the Breaking of Symmetry*, In Proceedings of the 6th International Conference on Parallel Problem Solving from Nature, Volume 1917 of Lecture Notes In Computer Science, pages 385–394, London (UK), 2000, Springer-Verlag.

[52] M. Pelikan and D.E. Goldberg, *Hierarchical Problem Solving by the Bayesian Optimization Algorithm*, In Proceedings of the Genetic and Evolutionary Computation Conference, Lecture Notes in Computer Science, pages 267–274, Las Vegas, Nevada (USA), 2000. Morgan Kauffman.

[53] M. Pelikan and D.E. Goldberg, *Research on the Bayesian Optimization Algorithm*, In Proceedings of the Genetic and Evolutionary Computation Conference, Volume 1 of Lecture Notes in Computer Science, pages 212–215. Morgan Kauffman, 2000.

[54] M. Pelikan, D.E. Goldberg and F.G. Lobo, A survey of optimization by building and using probabilistic models, *Computational Optimization and Applications* **21**(1) (January 2002), 5–20.

[55] B. Pfahringer, *Compression-Based Discretization of Continuous Attributes*, In International Conference on Machine Learning, pages 456¤C 463, Tahoe City (California, USA), 1995.

[56] J.R. Quinlan, Induction of decision trees, *Machine Learning* **1**(1) (1986), 81–106.

[57] K. Revoredo and G. Zaverucha, *Search-Based Class Discretization for Hidden Markov Model for Regression*, In Proceedings of the Brazilian Symposium on Artificial Intelligence, Volume 3171 of Lecture Notes in Computer Science, pages 317–325, São Luis (Maranhão, Brazil), 2004.

[58] H. Scheffe, *The Analysis of Variance*, John Wiley, 1959.

[59] M. Sebag and A. Ducoulombier, *Extending Population-Based Incremental Learning to Continuous Search Spaces*, In Proceedings of the 5th International Conference on Parallel Problem Solving from Nature, Volume 1498 of Lecture Notes in Computer Science, pages 418–427, London (UK), 1998, Springer-Verlag.

[60] R. Setiono and H. Liu, *Chi2: Feature Selection and Discretization of Numeric Attributes*, In Proceedings of the 7th International Conference on Tool with Artificial Intelligence, pages 388¤C391, Whashington DC (USA), 1995.

[61] R. Shachter and C.R. Kenley, Gaussian in uence diagrams, *Management Science* **5**(35) (1989), 527–550.

[62] L. Torgo and J. Gama, *Search-Based Class Discretization*, In European Conference on Machine Learning, Volume 1224 of Lecture Notes in Computer Science, pages 266–273, Prague (Czech Republic), 1997, Springer-Verlag.

[63] J.J. Valdes, L.C. Molina and N. Peris, *An Evolution Strategies Approach to the Simultaneous Discretization of Numeric Attributes in Data Mining*, In Proceedings of the World Congress on Evolutionary Computation, pages 1957–1964, Canberra (Australia), 2003.

[64] J.J. Valdes, L.C. Molina and N. Peris, *An Evolution Strategies Approach to the Simultaneous Discretization of Numeric Attributes in Data Mining*, In Proceedings of the Congress on Evolutionary Computation, 2003.

[65] M. Wang and S. Geisser, Optimal dichotomization of screening test variables, *Journal of Statistical Planning and Inference* **131**(1) (2005), 191–206.

[66] I.H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, (Second Edition), Morgan Kauffman, 2005.

[67] Q. Wu, G. Prasad, T.M. McGinnity, D. Bell, S. Zhong and J. Guan, *A Novel Discretizer for Knowledge Discovery Based on Multiknowledge Approaches*, In Proceedings of the International Conference on Intelligent Computing, Volume 344 of Lecture Notes in Computer Science, pages 778–783. Morgan Kauffman, August 2006.

[68] Y. Yang, *Discretization for Naive-Bayes Learning*, Master's thesis, Monash University, 2003.

[69] Y. Yang and G.I. Webb, Discretization for naive-Bayes learning: managing discretization bias and variance, Technical report, School of Computer Science and Software Engineering Monash University, Melbourne, VIC 3800, Australia, 2003.

[70] Y. Yang and G.I. Webb, *On Why Discretization Works for Naive-Bayes Classifiers*, In Australian Conference on Artificial Intelligence 2003, Volume 2903 of Lecture Notes in Computer Science, pages 440–452, Perth (Australia), 2003, Springer-Verlag.

[71] Q. Zhang, On stability of fixed points of limit models of univariate marginal distribution algorithm and factorized distribution algorithm, *IEEE Transactions on Evolutionary Computation* **8**(1) (February 2004), 80–93.

[72] Q. Zhang, On the convergence of a class of estimation of distribution algorithms, *IEEE Transactions on Evolutionary Computation* **8**(2) (April 2004), 127–136.

[73] A.A. Zhigljavsky, *Theory of Global Random Search*, Kluwer Academic Publisher, 1991.