NRC Publications Archive
Archives des publications du CNRC

An Evolution Strategies Approach to the Simultaneous Discretization of Numeric Attributes in Data Mining
Valdés, Julio; Molina, L.C.; Peris, N.

This publication could be one of several versions: author's original, accepted manuscript or the publisher's version. /
La version de cette publication peut être l'une des suivantes : la version prépublication de l'auteur, la version acceptée du manuscrit ou la version de l'éditeur.

Publisher's version / Version de l'éditeur:

Proceedings of the World Congress on Evolutionary Computation, 2003

Questions? Contact the NRC Publications Archive team at
PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca. If you wish to email the authors directly, please see the
first page of the publication for their contact information.

Vous avez des questions? Nous pouvons vous aider. Pour communiquer directement avec un auteur, consultez la
première page de la revue dans laquelle son article a été publié afin de trouver ses coordonnées. Si vous n'arrivez
pas à les repérer, communiquez avec nous à PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca.

National Research Council Canada   Conseil national de recherches Canada

National Research
Council Canada

Conseil national
de recherches Canada

Institute for
Information Technology

Institut de technologie
de l'information

NRC·CNRC

*An Evolution Strategies Approach to the Simultaneous Discretization of Numeric Attributes in Data Mining ***

Valdés, J., Molina, L.C., and Peris, N.
December 2003

Canada

An Evolution Strategies Approach to the Simultaneous Discretization of Numeric Attributes in Data Mining

Julio J. Valdés
National Research Council of Canada
Institute for Information Technology
M50, 1200 Montreal Rd.
Ottawa, ON K1A 0R6, Canada
julio.valdes@nrc.ca

Luis Carlos Molina
Mexican Institute of Petroleum
Lázaro Cárdenas 152
07720, Mexico City
Mexico
lcmolina@imp.mx

Natán Peris
CSA Integración de Sistemas
Mare de Deu de les Neus 72
08031, Barcelona, Spain
nperisan@muniversal.net

Abstract- Many data mining and machine learning algorithms require databases in which objects are described by discrete attributes. However, it is very common that the attributes are in the ratio or interval scales. In order to apply these algorithms, the original attributes must be transformed into the nominal or ordinal scale via discretization. An appropriate transformation is crucial because of the large influence on the results obtained from data mining procedures. This paper presents a hybrid technique for the simultaneous supervised discretization of continuous attributes, based on Evolutionary Algorithms, in particular, Evolution Strategies (ES), which is combined with Rough Set Theory and Information Theory. The purpose is to construct a discretization scheme for all continuous attributes simultaneously (i.e. global) in such a way that class predictability is maximized w.r.t the discrete classes generated for the predictor variables. The ES approach is applied to 17 public data sets and the results are compared with classical discretization methods. ES-based discretization not only outperforms these methods, but leads to much simpler data models and is able to discover irrelevant attributes. These features are not present in classical discretization techniques.

1 Introduction

Many data mining and machine learning algorithms [Qui89] [CN89] [FKY96] require data in which objects are described by sets of discrete attributes. In practice, however, a great number of attributes are of a continuous nature, as they come from measurements, sensors, etc. (e.g. temperature, weight). Therefore in order to use these algorithms, the continuous attributes must be transformed into discrete, but the way in which it is done have a large impact on the results obtained by the data mining techniques.

Several techniques have been proposed for both the supervised and unsupervised case [And73] [Ker92] [FI93], [MRMC00]. In the former one, the class information of the studied objects is available and can be used for guiding the discretization process. Algorithms like k-means, ChiMerge and partition using Minimal Description Length Principle (MDLP) [FI93] belong to this family and are popular. However, they were formulated for transforming only one continuous attribute at a time. Further, the number of classes or intervals for partitioning the attribute must be set forth in ad-

vance (e.g. k-means), and in others, some significance level must be established (e.g. ChiMerge). Usually these parameters are given by the expert or found using other techniques. In the multivariate case these techniques perform the discretization in an attribute-wise manner. That is, each variable is transformed separately. However, with this approach the inter-relations within the prediction attributes is not taking into account. In real world data, attributes are usually interrelated in subtle, non-linear ways, and redundancies of different degrees are present. Therefore, the discretization of each attribute independently of the others may lead to important information losses, thus increasing the chance of missing interesting relations in the knowledge discovery process.

This paper presents a hybrid technique for the simultaneous supervised discretization of continuous attributes, based on evolutionary algorithms (in particular, Evolution Strategies (ES) [Rec73] [Bac91]). It also uses Rough Set Theory [Paw82] [Paw91] and Information Theory, as is done in inductive learning [Qui86] [Qui96]. The purpose is to generate a global discretization scheme for all continuous attributes simultaneously by exploiting the inter-attribute relations, in addition to the dependency between the class variable and each attribute. Class predictability is maximized w.r.t a given criterium by relating the discrete classes constructed for the predictor attributes with the classes of the decision attribute. A discretization of a continuous attribute is given by a crisp partition of its range by a set of real values (cut points). The cardinality of this set determines the number of classes into which the given attribute is to be partitioned, and the cut-points, the intervals defining each class. A joint (global) discretization scheme for a set of attributes is given by the number of classes in which each particular attribute is partitioned, and the set of cut points defining them.

The paper is organized as follows: Section 2 presents the discretization problem. Section 3 approaches discretization from an evolutionary algorithms perspective (focussing on evolution strategies), and presents algorithms based on three criteria. Section 4 presents three different experiments performed with different data sets and Section 5 discusses the results obtained, as well as comparisons with two classical discretization methods. The conclusions are presented in Section 6.

2 The Simultaneous Discretization of Numeric Attributes

Consider an *information system $S = < U, A >$* [Paw82] where U and A are non-empty finite sets, called the *universe* and the set of *attributes* respectively, such that each $a \in A$ has a domain V_a and an evaluation function f_a assigns to each $u \in U$ an element $f_a(u) \in V_a$ (i.e. $f_a(u) : U \to V_a$). Typical examples of are data matrices with nominal or ordinal attributes. Sometimes, A is of the form $A^p \bigcup \{d\}$, where the set A^p is called *prediction* attributes and d the *decision* attribute. A more general kind of information system is obtained if the elements of A have domains given by arbitrary sets, not necessarily finite (for example, if $V_a \subseteq \mathbb{R}$, where $\mathbb{R}$ is the set of real numbers). Data matrices with interval or ratio variables are examples of systems of this kind. Consider two information systems $S_d = < U, A_d >$ and $S_c = < U, A_c >$ with the same universe U and attributes A, but with different domains. Thus, the attributes have the same cardinality $n = card(A_d) = card(A_c)$, and the information systems are defined as: for all $a^d \in A_d$ and $u \in U, f_{a^d}(u) : U \to V_{a^d}^d \subset \mathbb{N}^+$ ($\mathbb{N}^+$ is the set of natural numbers and $V_{a^d}^d$ is finite). $S_c = < U, A_c >$ and for all $a^c \in A_c$ and $u \in U, f_{a^c}(u) : U \to V_{a^c}^c \subseteq \mathbb{R}$ ($\mathbb{R}$ is the set of real numbers). A *discretization* between information systems is a mapping $\mathcal{D} : S_c \to S_d$.

Discretizations can be defined in many ways. Here a Discretization is considered to be given by a collection of parametrized functions φ_i, $1 \leq i \leq n$ of the form: $V_{a_i}^d = \varphi_i(V_{a_i}^c, \ldots, V_{a_i}^c, \hat{p}_i)$, where $\hat{p}_i$ is a set of parameters. These functions map the sets of domains of the attributes in A_c to those in A_d and leads to a *global* discretization, in the sense that the transformation of a particular attribute depends on all of them. In the particular case in which $V_{a_i}^d = \varphi_i(V_{a_i}^c, \hat{p}_i)$ the discretization is *attribute-wise* or *local*. Global or local discretizations can be easily constructed if a collection of natural numbers $\mathcal{M} = \{m_1, \ldots, m_n\}$ ($m_i \in \mathbb{N}^+, 1 \leq i \leq n$), and a collection of vectors $\mathcal{T} = \{\vec{t_1}, \ldots, \vec{t_n}\}$ are given s.t. $\vec{t_i} \in \mathbb{R}^{m_i}$ for all $1 \leq i \leq n$. For a given attribute a_i, the corresponding vector $\vec{t_i}$ induces a partition of $V_{a_i}^c$ into $m_i + 1$ adjacent classes or categories. The elements of $\vec{t_i}$ are called cut-points.

Examples of popular supervised discretization methods are the ChiMerge [Ker92] and the one introduced in [FI93], using the the Minimum Description Length Principle (MDLP) [Ris86]. The ChiMerge method is a statistically based approach for attribute-wise discretization. At the beginning it places each numeric value into its own class and merge them according to a χ^2 test applied to neighboring classes. The hypothesis tested is that two adjacent classes are independent, which is based on the comparison between the expected and observed frequencies of values found in the corresponding classes. The merging procedure is applied until a χ^2-threshold is reached.

The MDLP was applied to the discretization problem in [FI93] within a recursive entropy minimization heuristic for controlling the generation of decision trees. A coding scheme is defined which enables the comparison of information gains obtained with different cut points of the studied attribute, in terms of their codified lengths. Then, they are accepted or rejected according to the MDLP criterium. These two methods will be used for comparating the ES-based discretizations introduced in the next section.

3 An Evolutionary Algorithm Approach to Supervised Simultaneous Discretization

The power of evolutionary algorithms (EA) in solving function optimization problems makes genetic algorithms, evolution strategies, and others, a natural choice from a computational intelligence perspective to the discretization problem. An evolutionary computation-based discretization algorithm can be expressed as $\mathcal{D} = < \mathcal{EA}, \mathcal{C}_r, \mathcal{P}ar, \mathcal{P} >$ where $\mathcal{EA}$ is an evolutionary algorithm, $\mathcal{C}_r$ is a criterium for evaluating the quality of the mapping, $\mathcal{P}ar$ a collection of parameters controlling the algorithm, and $\mathcal{P}$ is a post-processing stage. The post-processing stage used here consist on removing cut-points without affecting the value of the fitness function. It leads to important model simplifications.

3.1 An Evolution Strategy Approach to Simultaneous Discretization

Evolution Strategies are naturally suited for building EA-based supervised global discretization algorithms because of their representation scheme (real-valued vectors), and their power in function optimization [VME00] [VMP00].

3.1.1 Classical Evolution Strategies

The elements composing an ES algorithm are: *i)* generation of the initial population, *ii)* recombination mechanisms, *iii)* mutation, *iv)* selection mechanisms, *v)* termination criteria.

An ES algorithm is usually expressed as follows:
ES = $(\mu, \lambda, l, R, \Phi, \mathcal{X}, \Delta\sigma, \Delta\theta, \tau)$
where μ is the population size, λ is the number of offsprings produced in each generation, l is the number of triplets (variables, σ, α) for each individual, R is the replacement policy $(\mu + l, \mu, \lambda)$, $\Phi : R^l \to R^+$ is the fitness function, $\mathcal{X}$ is a recombination operator, $\Delta\sigma$ is the increment/decrement value for modifying the standard deviation σ of each individual, $\Delta\theta$ is the increment/decrement value for the parameter controlling the correlation of deviations, and τ is a termination criterium.

ES are well suited for solving optimization problems in complex systems. The individuals are n-dimensional vectors $\vec{x} \in \Re^n$, with some additional parameters. Given an objective function $\mathcal{F} : \Re^n \to \Re$, having vectors as arguments (the individuals), the fitness function Φ is identified with $\mathcal{F}$, that is $\Phi(\vec{a}) = \mathcal{F}(\vec{x})$. In Evolution Strategies the individuals have the form $\vec{a} = (\vec{x}, \vec{\sigma}, \vec{\alpha}) \in I = \Re^n \times A_s$ where $\vec{x}$ is the object variable component, $\vec{\sigma}$ is the vector of standard deviations and $\vec{\alpha}$ the vector of rotation angles.

$As = \Re_+^{n_\sigma} \times [-\pi, \pi]^{n_\alpha}$, $n_\sigma \in \{1, \ldots, n\}$, and $n_\alpha \in \{0, (2n - n_\sigma)(n_\sigma - 1)/2\}$

Each individual includes a set of standard deviations σ_i as well as a set of rotation angles $\sigma_{ij} \in [-\pi, \pi]$. This parameters completely determine the n-dimensional gaussian distribution $p(\vec{z}) = \frac{\exp(-\frac{1}{2}\vec{z}^T C^{-1}\vec{z})}{\sqrt{(2\pi)^n \cdot \det(C)}}$, where C is the variance-covariance matrix.

The rotation angles are related with the variances as $\tan 2\alpha_{ij} = \frac{2C_{ij}}{\sigma_i^2 - \sigma_j^2}$.

The generation of a correlated vector $\vec{\sigma_c}$ from an incorrelated one $\vec{\sigma_u} = \vec{N}(\vec{o}, \vec{\sigma})$ is given by the multiplication of σ_u with N_σ rotation matrices $R(\alpha_{ij}) = r_{kl}$ where
$r_{ii} = r_{jj} = \cos(\alpha - ij)$
$r_{ij} = -r_{ji} = -\sin(\alpha - ij)$

The space of the individuals is $I = \Re^n \times \Re^n \times \Re^{n(n-1)/2}$

Mutation is an asexual operator $m_{\{\tau,\tau',\beta\}} : I^\lambda \to I^\lambda$ and produces a triple $(\vec{x}', \vec{\sigma}', \vec{\alpha}')$ which in compact notation is $m_{\{\tau,\tau',\beta\}} = (\vec{x}, \vec{\sigma}, \vec{\alpha}) = (\vec{x}', \vec{\sigma}', \vec{\alpha}')$

Specifically $\sigma_i' = \sigma_i \cdot \exp(\tau' \cdot N(0,1) + \tau \cdot N_i(0,1))$, $\alpha_j' = \alpha_j + \beta \cdot N_j(0,1)$, and $\vec{x}' = \vec{x} + \vec{N}(\vec{0}, A(\vec{\sigma}', \vec{\alpha}'))$, where $i \in \{1, \ldots, n\}$, and $j \in \{1, \ldots, n \cdot (n-1)/2\}$. $N(0,1)$ is the standard random gaussian variable, and $N_i(0,1)$ means that the random gaussian variable is sampled again for each possible value of i. $\vec{N}(\vec{u}, V)$ is the gaussian random vector with mean $\vec{u}$ and variance-covariance matrix V.

3.1.2 Recombination

These operator creates an individual $\vec{a}' = (\vec{x}', \vec{\sigma}', \vec{\alpha}')$ from a population $P(t) \in I^\mu$. If indices S and T denote two randomly chosen parents, the index T_i indicates that T has to be resampled for each value of i. $\gamma \in [0,1]$ is an uniform random variable, resampled for each value of i when it appears in the form γ_i. Some operators are: *(a)* no recombination $(x_{S,i})$, *(b)* discrete $(x_{S,i} \text{ or } x_{T,i})$, *(c)* discrete panmictic $(x_{S,i} \text{ or } x_{T_i,i})$, *(d)* intermediate $(x_{S,i} + (x_{T,i} - x_{S,i})/2)$, *(e)* intermediate panmictic $(x_{S,i} + (x_{T_i,i} - x_{S,i})/2)$, *(f)* generalized intermediate $(x_{S,i} + \gamma \cdot (x_{T,i} - x_{S,i}))$, *(g)* generalized intermediate panmictic $(x_{S,i} + \gamma_i \cdot (x_{T_i,i} - x_{S,i}))$, *(h)* global $(x_{S,i} \text{ or } x_{T_i,i})$.

3.1.3 Selection

Basically there are two variants: $(\mu + \lambda)$ selects the μ best individuals from the union of the parents and the offsprings in order to form the next generation. (μ, λ) selects the μ best from the λ offsprings (requires $\mu < \lambda$).

3.1.4 Termination Criteria

Typical criteria used for terminating ES algorithms are: *i)* reaching a given number of generations, *ii)* surpassing a maximum computation time, *iii)* obtaining an individual with a fitness equal or better than a given threshold, *iv)* the absolute or relative difference in fitness between the best and worst individuals is under a given threshold, and *v)* an absolute or relative difference measure between the best individuals in successive generations (it indicates the lack of significant improvement of the algorithm (stagnation), if it falls under a preset threshold).

3.2 Some Extensions of the Classical Algorithm

This paper introduces several additional features with respect to those described in the classical algorithm and they are integrated in the actual software implementation used in this research. These extensions are heuristic mechanisms oriented to improve the search robustness, cover a broader portion of the search space, improve the speed of convergence and introduce more flexibility.

- Mutation based on a Cauchy distribution: As suggested in [YL97] mutation according to a Cauchy distribution provides broader tails; increasing the mutation probability and helping to evade local extrema.

- Different approaches for generating initial populations. *(i)* generation of λ random individuals (increasing the number of elements benefits the search for a global optimum); *(ii)* uniform distribution within the search space (a more homogeneous coverage benefits the optimum search); *(iii)* placement of the initial individuals at or near to the boundaries of the search space (if the optimum is within the hypervolume defined by them, in principle it could be reached by new individuals obtained with continuous recombination operators); *(iv)* cluster of the initial population around a specific point in the search space [Sch81], (it enables a comparison with classical optimization methods starting with an initial approximation).

- Fitness based selection. Bias the selection by choosing the parents according to a probability distribution based on the individual fitness (in Nature better adapted individuals have a better chance to produce offsprings). Besides the uniform distribution (the classical), the following were introduced: *i)* linear $(P(p_i) = F(p_i)/\Sigma_{k \in \mathcal{P}} F(p_k))$; *ii)* quadratic $(P(p_i) = F(p_i)^2/\Sigma_{k \in \mathcal{P}} F(p_k)^2)$; *iii)* logarithmic $(P(p_i) = \ln(F(p_i))/\Sigma_{k \in \mathcal{P}} \ln(F(p_k)))$; and *iv)* inverse $(P(p_i) = (1/F(p_i))/\Sigma_{k \in \mathcal{P}}(1/F(p_k)))$. $P(p_i)$ is the probability of element p_i of being selected as parent, and $\mathcal{P}$ is the current population.

- $(best(\mu) + \lambda)$-selection. This is an intermediate selection between the classical $(\mu + \lambda)$ and (μ, λ). It operates as (μ, λ)-selection but allowing the best individual μ from the current population to be transferred to the next. The monotonic increase in fitness is maintained, as well as the preservation of the best solution found so far.

- Sorting the vector of variables. In some problems the fitness function is insensitive to the order of the elements in the vector of variables. In such problems sometimes sorting this vector by its values improves the convergence speed despite the effort involved in sorting.

- Secondary fitness function. It introduces a kind of coarse and refinement steps in the comparison between two individuals (if they have equal primary fit-

ness, preference is given to the one with better secondary fitness). In principle a single fitness function could be constructed covering both, however, the evaluation process is considerably faster and simplified with this two-step approach.

- Heterogeneous-variable length chromosomes. The object variable vectors of the ES individuals are allowed to be *a collection* of real vectors from subspaces of *different* dimension. The i-th object variable component of a population has the form:

$$
\begin{aligned}
c_i \;=\; &<\; < m_1{}^i, x^i_{11}, \ldots, x^i_{1m_1{}^i} >, \cdots, \\
&< m_p{}^i, x^i_{p1}, \ldots, x^i_{pm_1{}^i} > \qquad >
\end{aligned}
$$

It generalizes the classical definition and extends the range of problems in which ES can be applied. In particular, according to this extended representation, it is possible to make $c_i = \mathcal{T}$ (see Section-2), thus allowing a natural representation of discretization models within an ES framework. An ES population constructed in this way encodes a collection of different discretization models, which can be evolved according to a chosen fitness criterium.

3.3 Criteria for fitness

Once a discretized information system is obtained, the predictive capability of the set of discrete attributes over a previously existing partition of the elements of the universe can be evaluated in many different ways. In an evolutionary algorithm approach, measures associated with this concept can be used as fitness function during the discretization process. The target is to find discretization schemes with the best classification ability. In this paper, the fitness functions are based on: *i)* Rough Sets, *ii)* Joint Entropy, and *iii)* C4.5.

3.4 Rough Set criterium

According to Rough Set Theory [Paw82], in order to define a set some information (knowledge) about the elements of the universe is required. This is in contrast to the classical approach where the set is uniquely defined by its elements without the need of additional information in order to define their membership. The information is represented as information systems where all evaluation functions have finite domains V_a. Vagueness and uncertainty are strongly related to indiscernibility and the approximation of sets. Accordingly, each vague concept (represented by a set), is replaced by a pair of precise sets called its lower and upper approximations. The lower approximation of a set consists of all objects which surely belong to the set, whereas the upper approximation of the concept consists of all objects which possibly belong to the set, according to the previous knowledge.

Formally, given any subset X of the universe U and an indiscernibility relation I, the lower and upper approximation of X are defined respectively as $I^*(X) = \{x \in U : I(x) \subseteq X\}$, and $I_*(X) = \{x \in U : I(x)\beta \cap X \neq \emptyset\}$, where $I(x)$ denotes the set of objects indiscernible with x.

An important issue in data analysis is the discovery of dependencies between attributes. Intuitively, a set of attributes D depends totally on a set of attributes C, denoted $C \Rightarrow D$, if all values of attributes from D are uniquely determined by values of attributes from C. In other words, D depends totally on C, if there exists a functional dependency between values of D and C.

Dependency can be defined as follows: Let D and C be subsets of A.

D *depends on* C in *degree* k ($0 \leqslant k \leqslant 1$), denoted $C \Rightarrow_k D$, if

$$
k = \gamma(C, D) = \frac{|POS_C(D)|}{|U|} \tag{1}
$$

where U is the universe of the information system, U/D are the equivalence classes induced on U by the relations determined by attribute D, and $POS_C(D) = \bigcup_{X \in U/D} C(X)$.

If $k = 1$ D *depends totally* on C, and if $k < 1$, D *depends partially* (in *degree* k) on C.

The k coefficient expresses the ratio of all elements of the universe, which can be properly classified to blocks of the partition U/D, employing attributes C and will be called the *degree of the dependency*. It can be used as a fitness measure of a discretization scheme, and the goal would be to maximize it.

3.5 Joint Entropy criterium

Let X be a set divided into k classes or categories $C_1, \ldots, C_k$, with probabilities $P(C_i)$. The standard definition of the *entropy function* (with the usual interpretation of $0 \ln(0)$ as 0) is $H = \sum_{i=1}^{k} -P(C_i) \ln P(C_i)$.

If A is a numeric attribute and $\vec{t}$ is a vector dividing the domain into n categories. Then they induce a partition $T = \{X_1, \ldots, X_n\}$ on the set of objects X. The joint entropy is given by

$$
H(A, T; X) = \sum_{i=1}^{k} \frac{|X_i|}{|X|} H(X_i) \tag{2}
$$

As in the previous case, the goal would be to find the discretization scheme maximizing this measure.

3.6 C4.5 criterium

This criterium is the one used in the C4.5 algorithm for building decision trees [Qui96], and it is based on the notion of *information gain*. The measure is the difference between the information given by the joint entropy associated with an original partition of the set of objects, and the same joint entropy, now computed for a partition induced by the values of a selected attribute.

$$
G(A, T; X) = H(X) - H(A, T; X) \tag{3}
$$

The application of this criterium, in the case of a numeric attribute A, consists of finding the m distinct values of the attribute $\{a_1, \ldots, a_m\}$, constructs the set of $m - 1$ midpoints $\{(a_1 + a_2)/2, \ldots, (a_{m-1} + a_m)/2\}$, and uses them as the landmarks for defining the partition involved in the computation of the information gain. Most interesting are those landmarks maximizing the entropy measure.

Nbr	Data Set	Attribute Type continuous	Attribute Type nominal	Missing values	Dataset Size
1	anneal	6	32	Yes	898
2	australian	6	8	No	690
3	breast	9	0	Yes	699
4	cleve	6	7	Yes	303
5	crx	6	9	No	690
6	diabetes	8	0	No	768
7	german	7	13	No	1000
8	glass	9	0	No	214
9	glass2	9	0	No	163
10	heart	13	0	No	270
11	hepatitis	6	13	Yes	155
12	horse-colic	7	15	Yes	368
13	hypothyroid	7	18	Yes	3163
14	iris	4	0	No	150
15	sick-euthyroid	7	18	Yes	3163
16	vehicle	18	0	No	846
17	wine	13	0	No	178

Table 1: Data sets from UCI used in the experiments.

4 Experiments

4.1 Data Sets

The data sets used for the experiments were selected from the repository of databases, domain theories and data generators maintained at the University of California, Irvine (http://www.ics.uci.edu/~mlearn/MLRepository.html) [BM98]. An additional data set was used (fractal), consisting of $41,616$ samples extracted from an image mosaic containing 9 different textures in a texture-based image classification problem using 6 fractal features for texture characterization [VME00].

4.2 Experimental Settings

There are no standard ways to evaluate the results given by discretization algorithms. The approach used here will measure the quality of a discretization model by looking at the classification error obtained, when the discretized model obtained from an original information system is used as input to a machine learning algorithm targeting the decision attribute. In order to asses both the performance of ES-based discretizations, as well as some of its properties, three kinds of experiments were conducted: 1) comparison between classification errors obtained with machine learning methods (ID3, C4.5 and C4.5-rules) applied to discretizations obtained with the ChiMerge, MDLP methods and the ES approach on selected data sets, 2) comparison between C4.5 and ES-based discretizations for a broader range of data sets, and 3) comparison between ES discretizations with fixed criterium (rough sets in this case) but with different selection operators.

In (1) classification errors on the decision attribute were evaluated according to three well known algorithms: ID3 [Qui86], C4.5 and C4.5-Rules [Qui96]. These methods required discretized data and are appropriate for comparison purposes, done in the following way: The chosen data sets were Iris, Wine, Bupa (Table 1), and Fractal. For each data set a classical discretization technique was applied to each

non-decision attribute, and the discretized data was classified with the machine learning algorithms mentioned above. The average number of classes per attribute was computed for the two models given by the ChiMerge and the MDLP methods, and that number was used as the maximal number of categories per attribute achievable during the ES-based discretizations. This approach is actually very conservative and clearly biased in favor of the classical methods used, as maybe a better solution with the ES-RS, ES-JE and ES-C4.5 algorithms could be obtained by allowing these algorithms to explore more elaborate models w.r.t. the one given obtained with ChiMerge and MDLP.

ES-based discretizations were computed after 25 generations using (μ,λ)-selection with $\mu = 50$ and $\lambda = 350$. Linear probability distribution was used for the selection of new parents, and recombination was set to fitness-based-scan [BFM97], which usually gives good performance in function optimizations. The elements of the $\vec{\sigma}$ vectors were in the $[0.001 - 0.1]$ range and no rotation angle vectors $\vec{\alpha}$ were used.

In Experiment 2), the combined discretization-classification of the C4.5 algorithm was compared with a discretization using ES with C4.5 fitness as criterium (ES-C4.5) (See Table 3), and a classification given by the C4.5 algorithm itself. In other words, the classification algorithm and the fitness criterium were according to the C4.5 algorithm, and only an ES-based discretization makes the difference. $(\mu + \lambda)$ and (μ, λ) were used, with $\mu = 150$ and $\lambda = 350$. $\vec{\sigma}$ and $\vec{\alpha}$ vectors were set as above and an average of 20 generations were used. All evaluations (5-fold cross-validation classification errors) were computed with the WEKA platform [WF99] for all the data sets described in Table 1.

In Experiment 3) the purpose was to observe the behavior of the mean number of categories/attribute resulting from ES-based discretizations when using different selection mechanisms. The rough set criterium was fixed and the RSL library [GS94] was used as the evaluation platform. ES parameters were as in Experiment 2) above.

5 Results

The results obtained for Experiment 1) are shown in Table 2 (classification errors for the training set are included as reference). In absolute terms, 5-fold cross-validation shows that for all data sets and all the classification algorithms, the smallest errors are obtained when ES-based discretization data are used. In some cases the errors between the classical and ES-based techniques are several times higher (for example, ChiMerge vs. ES-C4.5 for Wine classifying with ID3, MDLP vs. ES-C4.5 for Iris, also with ID3, or MDLP vs. ES-C4.5 for Wine, with the C4.5 rules classifier). With few exceptions, the best over-all discretizations are obtained with Evolution Strategies using the C4.5 fitness criterium (ES-C4.5 algorithm). The training set results suggests that ES-RS and ES-JE are probably more prone to overfitting the models. As explained in the previous section, all ES-based algorithms were not allowed to generate discretizations with a number of categories/attribute higher

Data Set	Discretization Method	Classification Algorithm						mean number of categories per attribute
		ID3		C4.5		C4.5rules		
		5 C.V.	Training	5 C.V.	Training	5 C.V.	Training	
iris	ChiMerge	3.33	2.00	4.00	2.67	4.67	2.67	4.25
	MDLP	4.66	2.66	4.00	2.67	4.67	2.67	4.00
	ES-RS	3.33	0.00	2.67	0.00	4.67	1.33	3.25
	ES-JE	3.33	0.00	2.67	1.33	5.33	2.00	2.75
	ES-C4.5	1.33	1.33	2.67	1.33	2.67	1.33	2.00
wine	ChiMerge	6.74	0.00	9.50	3.37	5.61	1.68	10.00
	MDLP	6.17	0.00	5.61	3.37	7.86	2.24	4.15
	ES-RS	4.49	0.00	6.18	0.00	5.05	1.68	2.07
	ES-JE	2.25	0.00	4.49	0.00	5.05	1.12	1.30 (*)
	ES-C4.5	1.12	0.00	1.68	0.00	2.24	0.56	1.30 (*)
bupa	ChiMerge	18.84	0.56	34.78	23.47	32.75	22.03	11.50
	MDLP	28.86	27.8	28.70	28.80	28.99	28.12	2.33
	ES-RS	18.55	0.00	25.78	8.11	37.39	21.74	6.00
	ES-JE	26.96	0.00	35.65	15.07	38.55	23.19	5.66
	ES-C4.5	23.47	8.98	22.89	10.72	24.63	17.97	3.83
fractal	ChiMerge	15.50	0.50	23.50	15.90	24.80	18.20	16.00
	MDLP	20.90	7.40	24.00	18.10	25.90	17.50	8.00
	ES-RS	13.02	0.00	20.20	7.80	21.80	16.60	7.33
	ES-JE	16.00	0.20	22.00	15.60	24.00	15.70	6.16
	ES-C4.5	13.60	8.90	16.00	10.30	15.40	12.00	4.33

Table 2: Classification errors obtained with Evolution Strategies using Rough Set (ES-RS), Joint Entropy (ES-JE) and C4.5 criteria (ES-C4.5), in comparison with ChiMerge and MDLP. 5-CV is 5-fold cross-validation, and training, the whole data set. Three classification algorithms were used: ID3, C4.5 and C4.5 rules. See text for explanation about (*).

than the average between ChiMerge and MDLP, thus, constraining their search. However, ES-based results were better, thus suggesting a potential for further improving these results. On the other hand, with the exception of the Bupa data set, the number of categories/attribute of the ES-based discretizations is several times smaller than those given by ChiMerge and MDLP. This is a very remarkable feature, because not only were better classification errors obtained, but also *much simpler data models* (and consequently simpler decision rules). This is crucial when the results are interpreted by human experts (humans have difficulty handling more than 7-9 categories simultaneously). In particular, for the case of Wine data, less than 2 categories/attribute were found, indicating that some irrelevant attributes were excluded from the model. This is a very interesting feature of ES-based discretizations not present in any other method.

The results obtained for Experiment *2)* are shown in Table 3. The best classification error for each data set is marked with a (*), clearly indicating that ES-based discretization outperforms the C4.5 algorithm in 82.4% of the data sets (14 out of 17). Moreover, C4.5 errors are greater than ES-C4.5 by an average absolute difference of 3.62 when ES-C4.5 performs better, whereas when C4.5 is better (in 17.6% of the cases), the average absolute difference is only 0.35. Within the ES, in 58.82% of the cases (considering all 17), the (μ, λ) selection performs better than the $(\mu + \lambda)$. If data sets are investigated individually, the results from Table 3 can be improved (all ES discretizations were computed with the same set of parameters regardless of the data set used). For example, according to Table 3, the C4.5 algorithm performed better than the ES for sick-euthyroid data. However, an independent experiment using $\mu = 250$ and a σ range of [0.01,0.1] gave an error of 1.64, better than $C4.5$, thus showing the potential for even better ES performance.

Data Set	Classification Error				
		Evolution Strategies (C4.5 criterium)			
	C45	$(\mu + \lambda)$		(μ, λ)	
anneal	2.01	1.68		1.45	* +
australian	15.08	14.93	* +	15.08	
breast	5.30	4.01		3.72	* +
cleve	25.75	21.13	* +	21.46	
crx	14.35 *	14.64	+	15.37	
diabetes	27.61	21.23	* +	22.92	
german	27.80	25.70	* +	25.80	
glass	27.11	22.13	* +	23.84	
glass2	22.70	11.05		10.43	* +
heart	19.26	18.89		17.41	* +
hepatitis	21.94	19.36		18.71	* +
horse-colic	14.41	13.59		12.23	* +
hypothyroid	0.86 *	1.11	+	1.21	
iris	4.670	5.34		2.67	* +
sick-euthyroid	1.97 *	2.57		2.47	+
vehicle	27.90	25.42		21.87	* +
wine	6.18	7.31		3.38	* +
Nbr. of best	3	5		9	
overall best	3	14			

Table 3: Comparison of the classification accuracy in 5-fold cross-validation experiments between the C4.5 algorithm and the Evolution Strategies approach using the C4.5 criterium. The (*) indicates the best result for a given data set. The (+) indicates the best result within the ES variants. The last two rows shows the number of times in which the corresponding algorithm gives the over-all best classification error.

Data Set	Rough Sets Criterium			
	Best $(\mu + \lambda)$-selection		Best (μ, λ)-selection	
	Dependency coefficient	MC/A	Dependency coefficient	MC/A
anneal	1.00	1.50 (*)	1.00	1.50 (*)
autralian	1.00	2.16	1.00	2.33
crx	1.00	2.16	1.00	2.33
diabetes	1.00	4.25	1.00	4.50
german	1.00	1.28 (*)	1.00	1.28 (*)
glass	1.00	5.22	1.00	3.66
glass2	1.00	2.55	1.00	2.55
heart	1.00	2.15	1.00	2.15
iris	1.00	2.50	1.00	2.50
vehicle	1.00	3.27	1.00	3.27
wine	1.00	2.00	1.00	2.00

Table 4: Simultaneous discretization with Evolution Strategies using the Rough Sets criterium for $(\mu + \lambda)$ and (μ, λ) selection mechanisms. MC/A is the mean number of categories/attribute in the resulting discretization. (*) indicates values smaller than 2.

Results for Experiment *3)* are shown in Table 4. All dependency coefficients were 1 (i.e. complete description of the classes of the decision attribute), and with the exception of data set (glass), the mean number of categories/attribute created are either the same of very close, regardless of the selection mechanism. In particular, redundant attributes were discovered for the (anneal) and (german) data sets. These preliminary results suggest that the ES-discretization is robust w.r.t the choice of the selection operator and also that it does not hamper the ability of discovering irrelevant attributes.

6 Conclusions

The results, although preliminary, show that supervised, global discretization algorithms based on Evolution Strategies are very effective, robust, and capable of outperforming classical discretization techniques in the data mining field. Moreover, this increased performance is obtained with discretizations having a much smaller number of categories/attribute, therefore, with much simpler models. More accurate and easily interpretable models are highly pursued in Data Mining, thus, the property of ES-based discretization algorithms of discovering models of precisely this kind is a very remarkable one. In addition, it was found that ES-based discretization algorithms can detect irrelevant attributes *as part* of the discretization process (also it seems that this capability is not affected by the selection mechanism chosen). This is a feature which is not present in classical methods. The best results are obtained when the evolution strategies-based algorithm uses as fitness a C4.5 criterium, and in general (μ, λ)-selection performs better than $(\mu + \lambda)$. Additional experiments targeting individual data sets show that there is still potential for increased ES performance. Further investigations are necessary in order to study the properties and possibilities of this approach, considering that only a small number of ES operators were used, thus leaving a potential for possibly better results.

Bibliography

[And73] M. Anderberg, *Cluster Analysis for Applications*, John Wiley & Sons, London UK, 1973.

[Bac91] T. Back, *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, Oxford, 1991.

[BFM97] T. Back, D. Fogel, and Z. Michalewicz, *Handbook of Evolutionary Computation*, Oxford University Press, Oxford, 1997.

[BM98] C. L. Blake and C.J. Merz, *UCI Repository of Machine Learning Databases*, 1998.

[CN89] P. Clark and T. Niblett, *The CN2 Induction Algorithm*, Machine Learning **9** (1989), no. 1, 57–94.

[FI93] U. M. Fayyad and K. B. Irani, *Multi-interval Discretization of continuous-valued Attributes for Classification Learning*, In Proc. 13th International Conference on Machine Learning, Morgan Kaufmann Publishers, 1993, pp. 1022–1027.

[FKY96] J. Friedman, R. Kohavi, and Y. Yun, *Lazy Decision Trees*, In Proc. 13th National Conference on Artificial Intelligence, AAAI Press and the MIT Press, 1996.

[GS94] M. Gawrys and J. Sienkiewicz, *Rough Sets Library version 2.0*, Tech. Report ICS TUW Research Report 27/94, Warsaw University of Technology, September 1994.

[Ker92] R. Kerber, *ChiMerge: Discretization of Numeric Attributes*, In Proc. 10th National Conference on Artificial Intelligence, MIT Press, 1992, pp. 123–128.

[MRMC00] L. C. Molina, S. Rezende, M. C. Monard, and C. Caulkins, *Transforming a Regression Problem into a Classification Problem using Hybrid Discretization*, Revista Computación y Sistemas CIC-IPN **4** (2000), no. 1, 44–52.

[Paw82] Z. Pawlak, *Rough Sets*, International Journal of Information and Computer Sciences **11** (1982), 341–356.

[Paw91] ———, *Rough Sets, Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, New York, 1991.

[Qui86] J. R. Quinlan, *Induction of Decision Trees*, Machine Learning **1** (1986), 81–106.

[Qui89] ———, *Inferring Decision Trees using the Minimum Description Length Principle*, Information and Computation **80** (1989), no. 3, 227–248.

[Qui96] ———, *Learning First-Order Definitions of Functions*, Journal of Artificial Intelligence Research **5** (1996), 139–161.

[Rec73] I. Rechenberg, *Evolutionsstrategien: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*, Frommann-Holzboog, Stuttgart, 1973.

[Ris86] J. Rissanen, *Stochastic Complexity and Modelling.*, Ann. Statist **14** (1986), no. 1, 1080–1100.

[Sch81] H. P. Schwefel, *Numerical Optimization of Computer Models*, John Wiley & Sons, Chichester, 1981.

[VME00] J. J. Valdés, L. C. Molina, and S. Espinosa, *Characterizing Fractal features for Texture Description in Digital Images: an Experimental Study*, Proc. 15th International Conference on Pattern Recognition, Barcelona, Spain, vol. 3, IEEE Computer Society, 2000, pp. 917–920.

[VMP00] J. J. Valdés, L. C. Molina, and N. Peris, *Simultaneous Supervised Discretization of Numeric Attributes: a Soft Computing Approach (draft)*, Tech. report, LSI-UPC, Barcelona, Spain, 2000.

[WF99] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann Publishers, 1999.

[YL97] X. Yao and Y. Liu, *Fast Evolution Strategies*, Evolutionary Programming VI (Berlin) (P. J. Angeline, R. G. Reynolds, J. R. McDonnell, and R. Eberhart, eds.), Springer Verlag, 1997, pp. 151–161.