Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 33 (2007) 96–109
Expert Systems with Applications
www.elsevier.com/locate/eswa

Initial breeding value prediction on Manchego sheep by using rule-based systems ☆

Luis delaOssa, M. Julia Flores, José A. Gámez *, Juan L. Mateo, José M. Puerta

Departamento de Sistemas Informáticos – SIMD (i³A), Universidad de Castilla-La Mancha, Campus Universitario s/n, Albacete 02071, Spain

Abstract

In this paper we present an application of rule-based expert systems to a farming problem. Concretely the prediction of the breeding value in Manchego ewes is studied for the early stage of their life in which the standard (BLUP) methodology cannot be applied. In this case the pedigree index (arithmetical mean between parents' breeding value) is used to make the estimation. An alternative to this method is presented here, which is based on the use of two different types of rule-based systems: regression rules and linguistic fuzzy rules. The approach proposed is data-driven in the sense that the rules are learnt from data. The results obtained show that the learnt systems are more accurate than the pedigree index, especially for the regression rules case. On the other hand, the linguistic fuzzy rules systems are more easier to understand for human experts, and this is a point to be take into account because of the nature of the problem we are dealing with.
© 2006 Elsevier Ltd. All rights reserved.

Keywords: Rule based expert systems; Regression trees; Linguistic fuzzy rules; Breeding value estimation; Fuzzy rule learning; Estimation of distribution algorithms

1. Introduction

Manchego sheep (Gallego, Torres, & Caja, 1994) are the native breed in Castilla-La Mancha (Spain), and the two main products obtained from them (Manchego cheese (CRDOQM, 2004) and Manchego lamb (CRDECM, 2004)) represent more than 50% of animal production in the region. Because of these economical implications and with the aim of enhancing Manchego sheep production, a selection scheme (called ESROM) based on the animal's genetic merit was started by the authorities fifteen years ago. The ESROM Selection Scheme (SS), which is similar to other selection schemes developed for other breeds, includes a series of activities whose joint purpose is the *genetic improvement of the breed with respect to the production of milk*, and is run by several public organizations including AGRAMA, the National Association of Manchego sheep breeders. Evidence of the success of the ESROM is the 25 extra litres produced at each lactation by the ewes obtained by artificial insemination within the ESROM program.

One of the major points in the ESROM scheme is the estimation[1] of the animal's genetic merit or breeding value (*BV*), and its use in flock replacements. In the ESROM scheme the BV is estimated by using BLUP (*Best Linear Unbiased Prediction*) animal model (Jurado, 1994), which is a complex method based on relating different traits by equations and solving them by simultaneously considering all the available information. The estimated BV is then used by the main ESROM tools, allowing us to place animals in the genealogical ranking and to take decisions about which animals will improve the flock genetic trend,

☆ This work has been partially founded by FEDER and the Junta de Comunidades de Castilla-La Mancha under projects PBC-02-002 and PBI-05-022.
* Corresponding author. Tel.: +34 967 599298; fax: +34 967 599224.
E-mail address: jgamez@info-ab.uclm.es (J.A. Gámez).

[1] We should use *Estimated Breeding Value* (EBV), but for the sake of simplicity we maintain the notation of BV, although it is clear that we are dealing with estimations.

0957-4174/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2006.04.004

which animals can be entered (or not) in the stud catalogue or market, which ewes are candidates to be used as good mothers by artificial insemination, etc. Moreover, ESROM encourages stock breeders to select their flock replacements on the basis of the animal's BV.

The BV of an animal is a numerical value that in the ESROM represents the deviation of the animal with respect to the mean BV of the Manchego ewes born in 1990 (the base year). The estimation of the BV by BLUP is a complex process that in the ESROM scheme is carried out every six months in a specialized center. Furthermore, the BV of an animal is a dynamic value, because it can change from one measurement to the next due to changes in the animal's own production data, changes in its relatives, changes in its flock, etc.

In this work, we focus on a concrete case of BV estimation: *estimating the BV of new-born ewes*. In fact, given that the ultimate goal of ESROM is to improve milk production, BLUP methodology is only applied to an ewe after its first offspring-birth and its corresponding controlled lactation. Up to this moment the estimated BV of an ewe is computed directly from its parents BVs by using the *pedigree index* (PI or pedigree in short), i.e., the arithmetical mean between the father's and mother's BV. Our *goal* in this work is to investigate the use of alternative prediction systems to the PI, with the aim of obtaining a better estimation and thus to have at our disposal better information for decision making during the early stages of Manchego ewe's life. In concrete, we study the use of rule-based systems as predictors, but with the constraint of looking for *simple* predictors, that is, predictors using a small number of variables (note that the pedigree index only uses two variables).

Concretely, the main contributions of the paper are:

- A study of the alternative techniques to the pedigree index in order to improve its breeding value estimation.
- A comparison of the usefulness of two different types of rule-based systems: regression rules and linguistic fuzzy rules. The comparison is based on the precision of the obtained system and on its ease of understanding for human experts.
- A new method based on a recent family of evolutionary algorithms (*Estimation of Distribution Algorithms*) to carry out the task of learning weighted linguistic fuzzy rules.

To achieve our goals we have structured the paper in six sections apart from this introduction. In Section 2 we describe the two types of rule-based systems used in the paper. Section 3 contains the description of the datasets on which our current research is based. The next two Sections 4 and 5 are devoted to describe the learning algorithms used to induce the rule-based systems. Section 6 contains the experiments carried out and the analysis of the results. Finally, in Section 7 we present our conclusions.

2. Rule based systems

Without any kind of doubt, rule-based systems constitute one of the widely used paradigms within the expert systems technology. One of the reasons for their success is their capability of being useful as:

- *predictive* systems, that is, the system can be used to infer the output for a target variable given an input, i.e., a set of attribute-value pairs for (some of) the predictive variables,
- *descriptive* systems, that is, the rules describe interesting relations between the problem variables.

Although *Rule-Based Systems* (RBS) are always made up of rules with the form *If antecedent Then consequent*, there are great differences (both syntactic and semantic) depending on the theory considered (first-order logic, fuzzy logic, probability theory, etc.). In this work we focus on two distinct paradigms which, in our opinion, cover two different goals: regression rules and linguistic fuzzy rules.

2.1. Regression rules

By *regression rules* (RR) we mean rules whose consequent is a linear regression model. More specifically, if Y is the target variable and $X_1, \ldots, X_n$ are the predictive attributes, then in a regression rule:

- The *antecedent* of the rule is a conjunction of conditions over (possibly some of) the predictive variables, defining a region of the input space. For example, If $X_1 \leqslant r_1$ and $X_3 \in (-\infty, r_2]$ could be a possible antecedent for a problem defined over $\{X_1, X_2, X_3, Y\}$.
- The *consequent* of the rule is a linear regression model which predicts the value of Y from the value of the predictive attributes $\{X_1, \ldots, X_n\}$, i.e., in our example we could have Then $Y = a_0 + a_1 \cdot X_1 + a_2 \cdot X_2 + a_3 \cdot X_3$ as the rule consequent.

Although the use of linear regression models implies the assumption of independence between the predictive attributes and the assumption of linearity between the target variable and the predictive ones, the advantage of using regression rules instead of a single linear regression model lies in the fact that the approximation of variable Y is carried out in a local manner (a model is defined for every region, i.e., for each rule antecedent) instead of globally (a single regression model for all the input space).

With respect to inference, when the input sub-spaces induced by the rules' antecedents are disjoint,[2] a given input instance can only fire a rule, so the inference reduces to the pattern-matching phase to identify such a rule and to

[2] This will be our case.

fire it, i.e., to return the value computed for Y by solving the local regression model encoded in the rule consequent.

2.2. Linguistic fuzzy rules

Fuzzy Rules (FRs) (Zadeh, 1973) are based on *Fuzzy Set Theory* (Zadeh, 1965) and are grounded on the use of fuzzy predicates, X is A, where X is a problem domain variable and A is a fuzzy set[3] The typical[4] structure of a fuzzy rule is as follows:

$$\text{If } X_1 \text{ is } v_1^{j_1} \& \ldots \& X_n \text{ is } v_n^{j_n} \text{ Then } Y \text{ is } v_y^{j_l} \quad (1)$$

where $\{X_1, \ldots, X_n, Y\}$ are problem domain variables and $\{v_1^{j_1}, \ldots, v_n^{j_n}, v_y^{j_l}\}$ are fuzzy sets defined over the domain of their corresponding variables.

When no restriction exists in the selection of the fuzzy sets $\{v_1^{j_1}, \ldots, v_n^{j_n}, v_y^{j_l}\}$ we talk about *approximative fuzzy rules* (Bárdossy & Duckstein, 1995). This type of system usually achieves the highest precision in the prediction task due to the fact that every fuzzy set appearing in each rule can be tuned independently of the others. However, the obtained rule system is a long way from being interpretable by human experts.

On the other hand, in this work we focus on the so-called *Linguistic* or Mamdani fuzzy rules (Mamdani & Assilian, 1975). In this type of fuzzy rule system, the domain of each input/output variable is partitioned/covered by a fixed number of fuzzy sets, each one having associated a linguistic label. For example, the domain of the variable age can be covered by the set of linguistic labels: {baby, child, teenager, adult, ancient}. By associating a fuzzy set to each linguistic label we get a *linguistic variable* (Zadeh, 1975). In the *linguistic modeling* of a system only the linguistic labels of a variable can appear in the fuzzy predicates of the rules. Because of these restrictions in the designing of the fuzzy rule system, linguistic fuzzy rules (LFR) usually have a lower precision than approximative systems, but on the other hand their rules (e.g. If car-speed is high and distance-to-next-car is short then brake-force is intense) are fully interpretable by human experts.

From the previous description of LFRs we can deduce that the knowledge base of a LFR system has two clearly differentiated components: (1) a domain data base which contains the definition of the linguistic variables; and (2) a collection of fuzzy rules defined over the linguistic variables.

With respect to inference we can distinguish the following steps:

- *Fuzzification interface*. This receives numeric inputs for the input variables and transforms them into fuzzy sets. We use *punctual* fuzzification, that is, given a real number r for an input variable X, we produce a fuzzy set $\hat{r}$, such that the membership value to $\hat{r}$ is 1 for r $(\mu_{\hat{r}}(r) = 1)$ and 0 for each value $q \neq r$ $(\mu_{\hat{r}}(q) = 0)$.
- *Defuzzification interface*. Given a fuzzy set it produces a numerical output. We use *moment* defuzzification, which returns the center-of-gravity of the given fuzzy set.
- *Inference engine*. Given an input $\mathbf{x} = \langle x_1, \ldots, x_n \rangle$ any LFR (see Eq. 1) such that $\forall_{i=1\ldots n}\ \mu_{v_i^{j_i}}(x_i) > 0$ is fired. As the fuzzy sets defining linguistic fuzzy variables usually overlap, an input usually fires several rules. When a rule is fired a fuzzy set for the target variable (Y) is obtained. In this work, the set v'_y is obtained (by using classical operators) as

$$\mu_{v'_y}(r) = \begin{cases} \mu_{v_y^{j_l}}(r) & \text{if } \mu_{v_y^{j_l}}(r) < m \\ m & \text{if } \mu_{v_y^{j_l}}(r) \geqslant m \end{cases}$$

  m being the matching degree of $\mathbf{x}$ to the rule: $m = \min_{i=1\ldots n} \mu_{v_i^{j_i}}(x_i)$.

  If k rules are fired by a given input $\mathbf{x}$ and $v_y^1, \ldots, v_y^k$ are the obtained fuzzy sets, then we have to combine them into a single output. In this work we use the weighted FITA (*First Integrate Then Aggregate*) approach, which first defuzzifies $v_y^1, \ldots, v_y^k$ into their corresponding numerical values $r_y^1, \ldots, r_y^k$ and then aggregates them into a single value by using a weighted average:

$$\hat{y} = \frac{\sum_{i=1}^{k} r_y^i \cdot m_i}{\sum_{i=1}^{k} m_i},$$

  m_i being the matching degree of $\mathbf{x}$ with respect to the ith rule fired.

On some occasions a numeric weight $w \in [0, 1]$ is associated to each LFR. This weight can be understood as the degree of importance of a rule in the whole system and, is a simple way of increasing the precision of the system without significantly decreasing its readability. If weighted LFRs are used then the final output is computed by taking into account the rule weights (w_i):

$$\hat{y} = \frac{\sum_{i=1}^{k} r_y^i \cdot m_i \cdot w_i}{\sum_{i=1}^{k} m_i \cdot w_i}.$$

3. Data

The dataset used in this work has been obtained from AGRAMA data bases, which contain data from 1989 to 2003. After the data preparation process (described in

[3] In this work we only use triangular fuzzy sets. The membership degree of a point x with respect to a triangular function defined in the interval [a, c] and maximum/middle value in b is obtained as:

$$\mu_{\text{Triangular}}(x) = \begin{cases} \frac{x-a}{b-a}, & \text{if } a \leqslant x \leqslant b, \\ \frac{c-x}{c-b}, & \text{if } b \leqslant x \leqslant c, \\ 0, & \text{otherwise.} \end{cases}$$

[4] There are different approaches to the syntax of fuzzy rules; thus TSK fuzzy rules (Sugeno & Kang, 1985; Takagi & Sugeno, 1985) have a like-regression model in the consequent, obtaining systems more precise with respect to the prediction task, but of course, with a considerable loss of comprehensibility.

Gámez, 2005) and following AGRAMA experts advice we get a dataset with 9894 records and 21 numerical variables.

As we seek to get an estimation of ewes BV at a really early stage, all the records in the dataset correspond to primipara ewes, because it is after the first offspring-birth and its corresponding lactation when ewes are evaluated using BLUP for the first time, and this is just the value that pedigree index tries to replace. The 21 available variables can be distributed in three different groups: 12 related to BV estimation (and reliability of such estimation) of the animal, its parents and grandparents; 5 related to lactation data of the animal's mother (number of controlled lactations, average and maximum production over such controlled lactations); 2 related to the ewe's unique controlled lactation; the pedigree index and BV, i.e., the target variable (see Table 1).

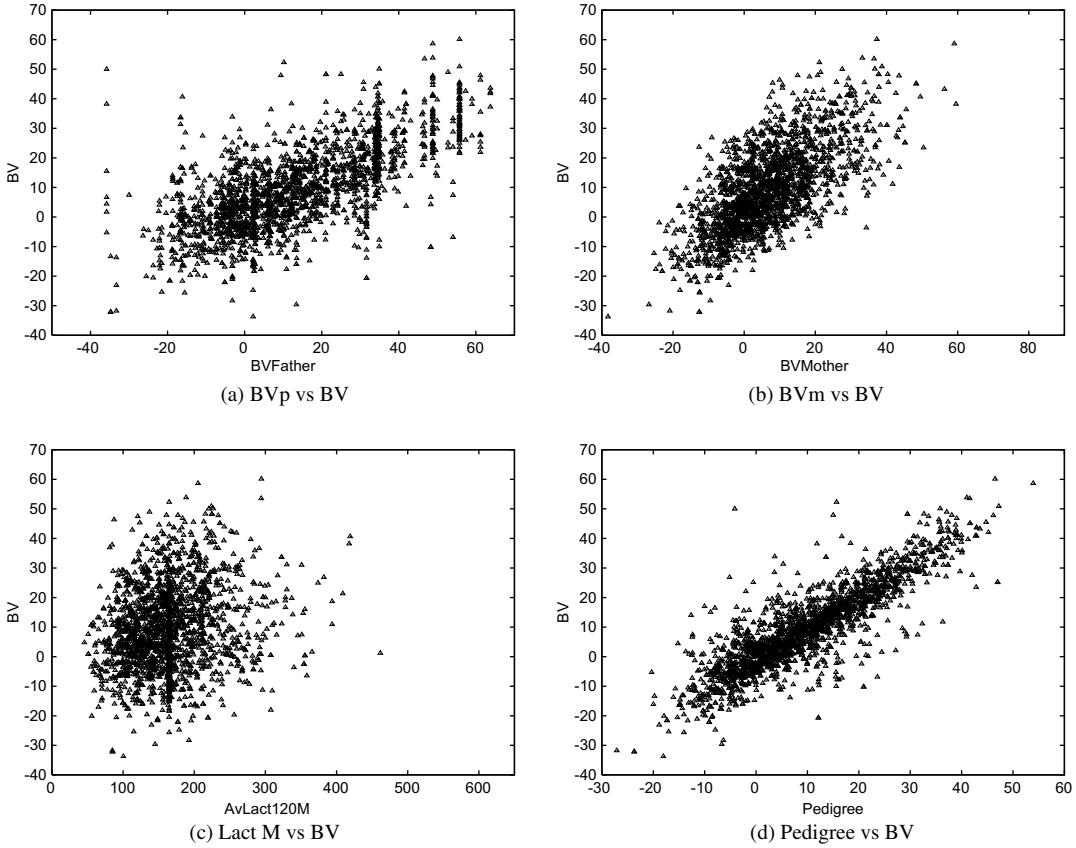
Table 1
Variables in the original data set

BV data:	Mother Lactation data:
BVFather, ReBVF, BVMother, ReBVM, BVMaternalGM, ReBVMGM, BVParentalGM, ReBVPGM, BVMaternalGF, ReBVMGF, BVParentalGF, ReBVPGF, *pedigree*, **BV**	NLactM AvLactNormM MaxLactNormM AvLact120M
Lactation data: AvLactNorm, AvLact120	MaxLact120M

BV is the target variable and *pedigree* stands for the constructed variable pedigree index.

This set of variables was selected by AGRAMA experts for a different but similar task: *BV classification* (Gámez, 2005). However, in a view of the goal we have established for this work we will only use a small subset of variables:

- First, it is clear that the lactation data of the ewes under study (AvLactNorm and AvLact120) cannot be used as predictive attributes.
- Secondly, as we aim to obtain simple predictors and given the type of rule systems we are going to induce, it is clear that we have to consider a small number of variables (the pedigree index only uses two). Thus, we consider the following two cases:
  (1) We only use variables BVfather and BVmother (BVf and BVm in short) as the pedigree index does.
  (2) As the ultimate goal of the ESROM scheme is to improve ewes' milk production, ewes' lactation data is really significant for the prediction task. Because we cannot use the ewe's own lactation data, we will investigate if the use of the ewe's mother's lactation data is of utility as predictive information. From the lactation mother data group we have selected variable AvLact120M (LactM in short) by using filter measures (correlation and mutual information). Therefore, our second predictor will use variables BVf, BVm and LactM.



(a) BVp vs BV (b) BVm vs BV

(c) Lact M vs BV (d) Pedigree vs BV

Fig. 1. Scatter plot showing the correlation of the selected predictive variables with respect to BV (subsample 20% of the dataset).

```
BVf <= 21.045 :
|   BVm <= 5.78 :
|   |   BVm <= -6.33 : LM1
|   |   BVm >  -6.33 : LM2
|   BVm >  5.78 : LM3
BVf >  21.045 :
|   BVm <= 4.555 : LM4
|   BVm >  4.555 : LM5

LM1: BV= .005 BVf + .071 BVm - 8.236
LM2: BV= .005 BVf + .039 BVm + 0.496
LM3: BV= .005 BVf + .024 BVm + 10.40
LM4: BV= .009 BVf + .054 BVm + 11.59
LM5: BV= .009 BVf + .037 BVm + 25.63
```

Fig. 2. An example of model tree.

Fig. 1 shows a scatter plot of the selected variables (plus pedigree index) versus the target variable BV. As we can observe there are different degrees of correlation (with respect to BV) for the selected variables.

4. Learning regression rules

To learn the regression rules knowledge base we will use an indirect approach: first, we learn a *model tree* from the data set; and second, we extract the rules from the tree.

Model trees (Quinlan, 1992; Wang & Witten, 1997) (see Fig. 2) are binary decision trees with linear regression functions at the leaf nodes. They can represent any piecewise linear approximation to an unknown function and form the basis of a successful technique for predicting continuous numeric values. In this work, we use the public implementation of M5′ algorithm (Wang & Witten, 1997) available in the WEKA data mining suite (Witten & Frank, 2005). M5′ includes the use of pruning for over-fitting prediction and a smoothing process to compensate discontinuities between adjacent linear models, which substantially increases the accuracy of predictions.

Once the model tree has been learnt we extract the regression rules by simply following the different paths from the root to each leaf. Thus, from the model tree in Fig. 2 we get the following rule set:

R1. If $\mathrm{BVf} \leqslant 21.045$ and $\mathrm{BVm} \leqslant 5.78$ and $\mathrm{BVm} \leqslant -6.33$ Then LM1
R2. If $\mathrm{BVf} \leqslant 21.045$ and $\mathrm{BVm} \leqslant 5.78$ and $\mathrm{BVm} > -6.33$ Then LM2
R3. If $\mathrm{BVf} \leqslant 21.045$ and $\mathrm{BVm} > 5.78$ Then LM3
R4. If $\mathrm{BVf} > 21.045$ and $\mathrm{BVm} \leqslant 4.555$ Then LM4
R5. If $\mathrm{BVf} > 21.045$ and $\mathrm{BVm} > 4.555$ Then LM5

However, because the model trees are binary, it is quite frequent that in non-trivial trees the same variable appears many times along a branch, giving rise to complex antecedents. Therefore, it is interesting to rewrite the obtained rules in such a way that a variable only appears in one conjunct of the antecedent, making its interpretation as well as the pattern matching phase easier. For example, rules R1 and R2 can be rewritten as follows:

R1. If $\mathrm{BVf} \leqslant 21.045$ and $\mathrm{BVm} \leqslant -6.33$ Then LM1
R2. If $\mathrm{BVf} \leqslant 21.045$ and $\mathrm{BVm} \in (-6.33, 5.78]$ Then LM2

5. Learning linguistic fuzzy rules

Although in the literature we can find different proposals for learning LFRs we base this work on the so-called grid-based methods. These methods assume that the linguistic variables have been defined previously and focus on the rule generation process.

Linguistic variables can be defined by domain experts or learnt from data. Although we will come back to this topic (Section 5.2), one of the easiest and most frequently used ways of constructing linguistic variables from data is to choose the number of linguistic labels, and then to build a symmetrical fuzzy partition of the domain by using triangular fuzzy sets. Thus, if a linguistic variable has l labels, the real domain of such a variable is divided into $l-1$ equi-width. Then, the l points $\{p_1,\ldots,p_l\}$ defining the limits of the intervals are used to define the fuzzy sets associated to the linguistic labels: triangular fuzzy set (p_{i-1},p_i,p_{i+1}) is associated to the ith linguistic label, except for the left-most and the right-most labels, which will have (p_1,p_1,p_2) and (p_{l-1},p_l,p_l), respectively. Fig. 6a shows an example of symmetrical partition with $l=7$. Grid-based methods use this division of linguistic variables (concretely the support[5] associated to each label) to create an n-dimensional grid which divides the input space into (overlapped) sub-spaces, and then they learn at most one rule for each input sub-space. The following algorithms/approaches have been used in this work:

[5] The support of a fuzzy set A are the points with membership degree greater than 0. The support associated to a triangular fuzzy set (l,m,r) is the interval $[l,r]$.

- *Wang and Mendel* (WM) algorithm (Wang & Mendel, 1992) is one of the most commonly LFRs learning algorithms because of its simplicity and efficiency (linear on the number of instances in the data set). The WM algorithm behaves greedily by selecting for every non-empty sub-space the rule (consequent) with the highest degree of importance (see Wang & Mendel (1992) for details).
- *COR methodology* (Casillas, Cordón, & Herrera, 2002). The main shortcomings of the WM algorithm result from its greedy behaviour. Thus, in each sub-space it looks for the rule with the best individual performance, without considering that the interaction between all the system rules will actually define its global performance.

  Casillas et al. (2002) propose a WM-based method in which they study the cooperation between the different rules of the system. This modification, known as *COR Methodology* (from *Cooperative Rules*) is based on replacing the *greedy* behaviour of WM algorithm in the selection of each rule, by a *combinatorial search of cooperative rules* in the space of all rule candidate sets. As opposed to the *greedy* and local philosophy of the WM algorithm, the use of COR tries to accomplish a global analysis.

  Let $\{S_1,\ldots,S_t\}$ be the set of non-empty sub-spaces with respect to the defined linguistic variables and a given data-set D. Let $\mathbf{C_i}$ the set of available consequents (i.e. those that have positive examples in D) for sub-space S_i. Then, the search space defined by COR methodology is

$$\overset{t}{\underset{i=1}{\otimes}}(\mathbf{C_i}\cup\{\aleph\})$$

  that is to say, the search space is the Cartesian product of the total set of possible consequents augmented with the *null* ($\aleph$) consequent.

  Once the search space has been defined, two components have to be specified to instantiate the COR methodology in a concrete algorithm:
  - A *search method*: local search, evolutionary algorithms, etc.
  - The *score* to evaluate the goodness of the candidate solutions proposed by the search algorithm. To evaluate an individual/point/solution of the search space, it is decodified into its corresponding LFR system. To do this, we run over the candidate solution and for each position $(1,\ldots,t)$ we generate a rule with the antecedent that defines sub-space S_i and the consequent specified by the ith position of the individual/solution being decodified. Those rules whose consequent is null ($\aleph$) are removed (not generated). Then, the LFR system obtained is used to predict the value of the target variable for each instance contained in the data set and the (root) mean squared error (with respect to the actual value of the target variable) is computed as the fitness/goodness of the individual being evaluated. Of course, as we look for the individual/system with the smallest associated error, our problem is a minimization one.
- *WCOR methodology* (Alcalá, Casillas, Cordón, & Herrera, 2002). As commented in Section 2.2 a way of improving the accuracy of LFRs is to assign a weight in $[0,1]$ to each rule. Alcalá et al. (2002) propose WCOR as an extension of COR in which weighted LFRs are allowed. Therefore, in WCOR the search space is defined as follows:

$$\overset{t}{\underset{i=1}{\otimes}}(\mathbf{C_i}\cup\{\aleph\})\overset{t}{\underset{i=1}{\otimes}}[0,1].$$

Of course, the problem is more complex now because we have a larger search space and also because we have to deal with a hybrid problem: combinatorial and numerical optimization.

5.1. UMDA-COR and UMDA-WCOR

In this work we use *Estimation of Distribution Algorithms* (EDAs) (Larrañaga & Lozano, 2001) as the search engine to guide the discovery process. EDAs have been successfully applied to the problem of learning LFRs by using the COR methodology (Flores, Gámez, & Puerta, 2005) and we now extend their application to WCOR, this being one of the main contributions of the paper.

EDAs are a recent metaheuristics that have attracted a great deal of interest during the last 5 years. EDAs are evolutionary algorithms based on populations as well as genetic algorithms (Michalewicz, 1996) (GAs), but in which genetics has been replaced by the estimation/learning and sampling of a probability distribution which relates the variables or genes forming to an individual or chromosome. Fig. 3 shows the general outline of EDAs evolution process. As we can see, steps (b) and (c) replace the classical selection + crossover + mutation used in genetic algorithms. Step (b) is the key point in EDAs algorithms, because working with the joint probability distribution is intractable even in small problems, so that a simpler model has to be estimated/learnt.

In this work we use the simplest EDA algorithm: UMDA *Univariate Marginal Distribution Algorithm* (Mühlenbein, 1998). The UMDA imposes the stronger assumption with respect to the underlying probability distribution: marginal independence among all the variables. In practice this assumption implies that the t-dimensional joint probability distribution is factorised as

$$P(s_1,s_2,\ldots,x_t)=\prod_{i=1}^{t}P(s_i),$$

that is, no structural learning is needed, and only marginal probabilities are required during parameter learning.

Therefore, the application of UMDA to COR comes down to the estimation of the marginal probability distribution for the set of values/consequents allowed for each sub-space, i.e., $C_i\cup\{\aleph\}$. In our implementation the estimated probabilities are smoothed out by using Laplace

(1) $D_0 \leftarrow$ Generate the initial population (m individuals)
(2) Evaluate the population D_0
(3) $k = 1$
(4) Repeat
  (a) $D_{tra} \leftarrow$ Select $n \leq m$ individuals from D_{k-1}
  (b) Estimate/learn a new model $\mathcal{M}$ from D_{tra}
  (c) $D_{aux} \leftarrow$ Sample m individuals from $\mathcal{M}$
  (d) Evaluate D_{aux}
  (e) $D_k \leftarrow$ Select m individuals from $D_{k-1} \cup D_{aux}$
  (f) $k = k + 1$
Until stop condition

Fig. 3. Description of a canonical EDA.

correction. With respect to sampling, because of the independence assumption, each position (marginal distribution) is sampled independently.

In spite of its simplicity, in general, UMDA has a good performance and, in particular, when applied to the COR methodology it behaves better than GAs (Flores et al., 2005).

In WCOR a candidate solution is a hybrid individual of length $2t$: the first t positions represent the consequent selected for each sub-space (as in COR) and positions $t+1,\ldots,2t$ are numbers in $[0,1]$ and represent the weights assigned to the rules. Thus, position i ($1 \leqslant i \leqslant t$) represents the consequent of the rule generated by subspace S_i and position $t+i$ represents the weight of that rule. Alcalá et al. (2002) use a GA that separately applies crossover over the discrete $(1,\ldots,t)$ and numerical $(t+1,\ldots,2t)$ parts of the chromosome and then combines the results obtaining eight offsprings from the combination of two parents. In this work, and encouraged by the success of UMDA when applied to COR, we extend our previous approach to cope with weights. Thus, we use the previously described UMDA algorithm to manage the discrete part of the problem and UMDA_g to cope with the numerical part. UMDA_g (Larrañaga, Etxeberria, Lozano, & Peña, 1999) is an adaptation of UMDA to the continuous case by using the normal distribution to model the density of each variable, while the joint density is factorised as the product of all the unidimensional and independent normal densities. Thus, model induction is reduced to the estimation of μ and σ^2 for each variable. Furthermore, as each variable is independently simulated, any standard method for sampling from a normal distribution can be used (e.g. Box and Muller).

Therefore, in the learning phase of our algorithm we run through the index $i = 1,\ldots,2t$ estimating a marginal discrete distribution if $i \leqslant t$ and a unidimensional normal distribution if $i > t$. Analogously, in the sampling phase we run through the index $i = 1,\ldots,2t$ sampling the appropriate distribution.

5.2. Tuning the definition of the linguistic variables

The definition of linguistic variables by choosing the number of labels (l) and then constructing a symmetrical fuzzy partition is a typical choice when using LFRs. However, a careful definition of the linguistic variables usually helps to improve the performance of the system. There are two main ways of doing this:

- Once the LFR system has been learnt a post-processing step is carried out that tunes the membership functions of the fuzzy set used in the linguistic variables in order to improve the system's precision. In (Cordón, Herrera, Hoffmann, & Magdalena, 2001, chap. 4) GAs are used to guide the tuning process.
- A different possibility is to tune the linguistic variables in combination with a rule induction mechanism. That is, different candidate definitions are created and scored by learning (and evaluating) a LFR system for each definition. In this approach it is common to tune not only the membership functions but also the number of labels used in the definition of each linguistic variable, i.e., the following parameters are tuned simultaneously: (1) the number of labels l_i for each variable X_i; and (2) the three points (left, middle, right) that describe the fuzzy set associated to each linguistic label. Again, GAs are a common choice to guide the tuning process (Cordón, Herrera, Hoffmann, et al., 2001; Cordón, Herrera, & Villar, 2001, chap. 4).

In this work we focus on the second approach in order to tune the definition of the linguistic variables involved in our problem. The major points of our algorithm are:

- In our case, as we will use the definitions obtained as the input for an evolutionary-based learning process, we deal with a simplified version of the problem described above. Concretely, we assume a fixed number of labels for the linguistic variables and this number (given by the user) is the same for all the variables. Furthermore, as in symmetrical partitions, the fuzzy set associated to the jth linguistic label only has a non-empty intersection with the $(j-1)$th and $(j+1)$th labels, intersecting them at height 0.5. This assumption implies that only the middle point of the triangular fuzzy sets has to be tuned, except for the left-most and right-most labels because in these cases the middle point is located at the minimum

and maximum value of the problem domain variable. Therefore, if l is the number of linguistic labels for each one of the n linguistic variables, then a candidate solution to our tuning problem can be codified by a vector of $n(l-2)$ doubles:

variable X_1						variable X_n				
c_1^2	c_1^3	$\cdots$	c_1^{l-2}	c_1^{l-1}	$\cdots$	c_n^2	c_n^3	$\cdots$	c_n^{l-2}	c_n^{l-1}

where c_i^j stands for the middle point of the fuzzy set associated to the jth label of the ith linguistic variable. Remember that c_i^1 and c_i^l are not included because they are fixed to min_i and max_i, respectively.

- As in Cordón, Herrera, and Villar (2001) we use the WM algorithm to score each definition. That is, for each candidate solution we run WM taking as input the definition for the linguistic variables that such a solution encodes. The (root) mean squared error associated to the system learnt by WM will be the fitness associated to that candidate solution.
- To guide the search we have used a simple generational GA with real-code representation (an array of doubles). Convex combination has been used as a crossover operator, i.e., given a value $\alpha \in [0,1]$, position c_i^j is obtained for both off-springs as: $c_i^{j'} = \alpha \cdot c_i^{ja} + (1-\alpha) \cdot c_i^{jb}$ and $c_i^{j''} = \alpha \cdot c_i^{jb} + (1-\alpha) \cdot c_i^{ja}$, where a and b denote the two parents. With respect to mutation, position c_i^j is mutated by replacing its current value with a random number uniformly generated in $[c_i^{j-1}, c_i^{j+1}]$.
- Once the genetic algorithm stops and returns a definition for the linguistic variables we perform the following process: If c_i^j and c_i^{j+1} are too close, then we remove their corresponding labels from the definition and add a

rule	BVf	BVm	BV
R1	$\leq$ 21.045	$\leq$ -3.455	0.3045·BVf + 0.6593·BVm + 0.2134
R2	$\leq$ -16.67	$\in$ (3.455,5.775]	0.0593·BVf + 0.9698·BVm - 7.8384
R3	$\in$ (-16.67,-11.815]	$\in$ (3.455,5.775]	-0.7244·BVf + 0.5616·BVm - 12.6089
R4	$\in$ (-11.815,3.095]	$\in$ (3.455,5.775]	0.1835·BVf + 0.5499·BVm + 0.5749
R5	$\in$ (3.095,21.045]	$\in$ (3.455,5.775]	0.4504·BVf + 0.594·BVm - 0.8803
R6	$\leq$ 21.045	$>$ 5.775	0.2776·BVf + 0.5632·BVm + 0.9711
R7	$\in$ (21.045,31.43]	$\leq$ -7.19	0.2533·BVf + 0.5876·BVm + 5.2531
R8	$\in$ (31.43,31.795]	$\leq$ -7.19	-0.1053·BVf + 0.39·BVm + 5.6334
R9	$\in$ (31.795,33.365]	$\leq$ -9.125	-0.1053·BVf + 0.2197·BVm + 6.1588
R10	$\in$ (31.795,33.365]	$\in$ (-9.125,-7.19]	-0.1053·BVf + 0.2197·BVm + 5.3739
R11	$\in$ (33.365,36.915]	$\leq$ -7.19	0.0392·BVf + 0.6176·BVm + 12.488
R12	$\in$ (21.045,29.66]	$\in$ (-7.19,-2.98]	0.4245·BVf + 0.0773·BVm - 2.5614
R13	$\in$ (29.66,29.875]	$\in$ (-7.19,-2.98]	17.9016·BVf + 0.0773·BVm - 526.6381
R14	$\in$ (29.875,31.54]	$\in$ (-7.19,-2.98]	1.1671·BVf + 0.0773·BVm - 26.6427
R15	$\in$ (31.54,31.795]	$\in$ (-7.19,-2.98]	0.3002·BVf + 0.8243·BVm + 0.1103
R16	$\in$ (31.795,36.915]	$\in$ (-7.19,-2.98]	0.2328·BVf - 0.6177·BVm - 0.3626
R17	$\in$ (21.045,23.995]	$\in$ (-2.98,2.675]	0.0317·BVf + 0.0559·BVm + 9.5463
R18	$\in$ (23.995,30.565]	$\in$ (-2.98,2.675]	0.3958·BVf + 0.0559·BVm + 1.3403
R19	$\in$ (30.565,31.54]	$\in$ (-2.98,2.675]	-3.4522·BVf + 0.0559·BVm + 119.2695
R20	$\in$ (31.54,31.705]	$\in$ (-2.98,2.675]	-4.2389·BVf + 0.0559·BVm + 140.0565
R21	$\in$ (21.045,28.665]	$\in$ (2.675,12.615]	0.3458·BVf + 0.3955·BVm + 3.0776
R22	$\in$ (28.665,31.35]	$\in$ (2.675,6.13]	-0.1328·BVf + 0.6809·BVm + 17.8058
R23	$\in$ (28.665,31.35]	$\in$ (6.13,7.475]	3.7877·BVf - 0.2361·BVm - 99.5954
R24	$\in$ (28.665,31.35]	$\in$ (7.475,12.615]	-0.5591·BVf + 0.8742·BVm + 28.767
R25	$\in$ (31.35,31.705]	$\in$ (2.675,12.615]	-1.6702·BVf + 0.262·BVm + 58.3904
R26	$\in$ (31.705,36.915]	$\in$ (-2.98,12.615]	0.0335·BVf + 0.5188·BVm + 14.4944
R27	$>$ 36.915	$\leq$ 12.615	0.4308·BVf + 0.6447·BVm + 1.2847
R28	$>$ 21.045	12.615	0.514·BVf + 0.5318·BVm - 1.4461

Fig. 4. Regression rules-based system obtained when using BVp and BVm as predictive variables. Each column represents a conjunct of the rule, except the last one which is the consequent.

new label with the middle point at $\frac{c_i^j + c_i^{j+1}}{2}$. If the new definition has a better or similar score (using WM) than the original one, maintain the change and go on, otherwise the change is rejected and the original linguistic labels are retained. The goal of this step is to reduce the number of linguistic labels without decreasing the precision of the system. It is clear that a reduction in the number of linguistic labels yields a reduction in the number of rules of the resulting system, and usually few rules means less overfitting.

To end this section let us remark that the tuning process is carried out in a global way, so the definition obtained for a variable (e.g. BVF) will be, in general, different if the whole set of involved variables is different (e.g. $\{BVf, BVm, BV\}$ vs $\{BVf, BVm, LactM, BV\}$).

6. Experiments and results

In this section we describe the design of the experiments carried out and analyse the results obtained.

As commented in Section 3 our dataset has 9894 instances. Taking into account this (considerably large) number of instances and in order to make an honest estimation of the obtained systems accuracy/precision, we have followed a 5-fold cross validation approach in our experiments. Therefore, in each iteration of the cross validation process, about 7900 instances were used for training

rule	BVf	BVm	LactM	BV
R1	$\leq$ 21.045	$\leq$ -12.485		0.3011 · BVf + 0.5915 · BVm - 0.024 · LactM + 1.9656
R2	$\leq$ -20.945	$\in$ (-12.485,-3.455]		0.1202 · BVf + 0.2257 · BVm + 0.0859 · LactM - 20.44
R3	$\in$ (-20.945,-12.86]	$\in$ (-12.485,-3.455]		0.0736 · BVf + 0.7202 · BVm + 0.0044 · LactM - 2.3065
R4	$\in$ (-12.86,-9.505]	$\in$ (-12.485,-3.455]		0.0645 · BVf + 0.6264 · BVm + 0.004 · LactM - 4.5676
R5	$\in$ (-9.505,2.87]	$\in$ (-12.485,-3.455]		0.021 · BVf + 0.5244 · BVm - 0.0005 · LactM - 1.4177
R6	$\in$ (2.87,21.045]	$\in$ (-12.485,-3.455]		0.4088 · BVf + 0.6103 · BVm - 0.0238 · LactM + 2.0911
R7	$\leq$ -16.67	$\in$ (-3.455,-0.075]		0.0592 · BVf + 0.3201 · BVm - 0.0362 · LactM - 4.3257
R8	$\leq$ -25.265	$\in$ (-0.075,5.775]		0.0592 · BVf + 1.0863 · BVm + 0.0735 · LactM - 19.5237
R9	$\in$ (-25.265,-16.67]	$\in$ (-0.075,5.775]		0.0592 · BVf + 0.238 · BVm + 0.0092 · LactM - 7.2654
R10	$\in$ (-16.67,-11.815]	$\in$ (-3.455,5.775]		-0.7245 · BVf + 0.5587 · BVm + 0.0014 · LactM - 12.8251
R11	$\in$ (-11.815,3.905]	$\in$ (-3.455,0.795]		0.1704 · BVf + 0.5933 · BVm - 0.0005 · LactM + 0.5164
R12	$\in$ (-11.815,-10.705]	$\in$ (0.795,5.775]		3.9918 · BVf + 0.0389 · BVm - 0.0014 · LactM + 45.8813
R13	$\in$ (-10.705,-7.365]	$\in$ (0.795,5.775]		0.2607 · BVf + 0.0389 · BVm - 0.0051 · LactM + 0.9744
R14	$\in$ (-7.365,-6.425]	$\in$ (0.795,5.775]		-3.4863 · BVf + 0.0389 · BVm - 0.0367 · LactM - 18.774
R15	$\in$ (-6.425,3.905]	$\in$ (0.795,5.775]	$\leq$ 163.12	0.0231 · BVf + 0.0806 · BVm + 0.001 · LactM + 1.7215
R16	$\in$ (-6.425,3.905]	$\in$ (0.795,2.05]	$\in$ (163.12,167.54]	0.0231 · BVf + 0.3607 · BVm + 0.0032 · LactM + 1.7594
R17	$\in$ (-6.425,3.905]	$\in$ (2.05,5.775]	$\in$ (163.12,167.54]	0.0231 · BVf + 0.2537 · BVm + 0.0032 · LactM + 4.5049
R18	$\in$ (-6.425,3.905]	$\in$ (0.795,5.775]	$>$ 167.54	0.0231 · BVf + 0.0691 · BVm - 0.0017 · LactM + 1.4067
R19	$\in$ (3.905,21.045]	$\in$ (-3.455,5.775]		0.4439 · BVf + 0.6322 · BVm - 0.021 · LactM + 2.2046
R20	$\leq$ -17.675	$\in$ (5.775,16.35]		0.0593 · BVf + 0.0428 · BVm + 0.0028 · LactM - 1.5692
R21	$\in$ (-17.675,-8.04]	$\in$ (5.775,16.35]		-0.4317 · BVf + 0.0428 · BVm + 0.0011 · LactM - 2.2292
R22	$\in$ (-8.04,6.495]	$\in$ (5.775,16.35]		0.149 · BVf + 0.6579 · BVm - 0.0133 · LactM + 2.1658
R23	$\in$ (6.495,21.045]	$\in$ (5.775,16.35]		0.4017 · BVf + 0.6108 · BVm - 0.0373 · LactM + 4.9719
R24	$\leq$ 21.045	$>$ 16.35		0.2556 · BVf + 0.5788 · BVm - 0.0104 · LactM + 3.0512
R25	$\in$ (21.045,36.915]	$\leq$ -7.19	$\leq$ 121.265	0.3042 · BVf + 0.5957 · BVm - 0.0742 · LactM + 11.5822
R26	$\in$ (21.045,31.43]	$\leq$ -7.19	$>$ 121.265	-0.2093 · BVf + 0.5585 · BVm - 0.0262 · LactM + 19.2734
R27	$\in$ (31.43,36.915]	$\leq$ -7.19	$>$ 121.265	-0.2647 · BVf + 0.3529 · BVm + 0.0025 · LactM + 8.4764
R28	$\in$ (33.365,36.915]	$\leq$ -7.19	$>$ 121.265	0.0466 · BVf + 0.5695 · BVm - 0.0502 · LactM + 17.7427
R29	$\in$ (21.045,36.915]	$\in$ (-7.19,-2.98]	$\leq$ 153.025	0.3961 · BVf + 0.0862 · BVm - 0.0129 · LactM + 0.3228
R30	$\in$ (21.045,36.915]	$\in$ (-7.19,-2.98]	$>$ 153.025	0.1015 · BVf + 0.0862 · BVm - 0.1213 · LactM + 24.1899
R31	$\in$ (21.045,31.705]	$\in$ (-2.98,2.675]		0.1769 · BVf + 0.0673 · BVm - 0.0433 · LactM + 12.5495
R32	$\in$ (21.045,28.665]	$\in$ (2.675,12.615]		0.3452 · BVf + 0.4846 · BVm - 0.027 · LactM + 6.4648
R33	$\in$ (28.665,31.35]	$\in$ (2.675,12.615]		-0.3235 · BVf + 0.7807 · BVm - 0.0163 · LactM + 24.9796
R34	$\in$ (31.35,31.705]	$\in$ (2.675,12.615]		-71.835 · BVf + 0.6688 · BVm - 0.0296 · LactM + 2277.9467
R35	$\in$ (31.705,36.915]	$\in$ (-2.98,12.615]		0.0354 · BVf + 0.5673 · BVm - 0.0212 · LactM + 17.3783
R36	$>$ 36.915	$\leq$ 12.615		0.4403 · BVf + 0.6875 · BVm - 0.0255 · LactM + 4.3234
R37	$>$ 21.045	$>$ 12.615		0.5147 · BVf + 0.609 · BVm - 0.0329 · LactM + 3.0737

Fig. 5. Regression rules-based system obtained when using BVp, BVm and LactM as predictive variables. Each column represents a conjunct of the rule, except the last one which is the consequent.

the models/systems and about 1980 to test them. Besides, in order to compare the results, the same 5 partitions (train+test) have been used in all the experiments.

In Section 2.1 we describe the method used to induce the regression rules-based system. Concretely, we have run its Weka (Witten & Frank, 2005) implementation (weka.classifiers.trees.M5P) by using its default parameter setting (pruned = yes, smoothing = yes, min number of instances per leaf = 4). The systems obtained for our two different predictors [BVp ∧ BVm → BV] and [BVp ∧ BVm ∧ LactM → BV] are in Figs. 4 and 5, respectively.

With respect to the precision of this approach the results are in the row denoted as M5′ in Tables 2 and 3. For this approach we show (distinguishing between train and test) the mean ± deviation (computed over the 5 folds of the cross validation) of two standard measures for numerical prediction: root mean squared error (rmse) and correlation coefficient (corr). Also the number of rules in the resulting systems is shown.

In the case of linguistic fuzzy rules we have developed our own software, which is written in Java and uses the API provided by FuzzyJess (IRG-NCR, 2005; Orchand, 2001) for fuzzy sets representation and operations. First, we will describe the details of the process followed to tune the definition of the linguistic variables. As we remark on Section 5.2 our goal with the tuning process is to get a better starting point for the LFR discovering process than the one provided by symmetrical partitions. Because of this we have dealt with a simplified version of the problem and we have given few resources to the genetic algorithm: the training set is a sample (2000 instances) of the original one; population size has been fixed to 50 individuals; and the maximum number of generations before stopping is set to 50. The rest of parameters are: 5 or 7 labels per linguistic

Table 2
Results obtained by the different approaches when BVp and BVm are used as predictive attributes

Method	gran.	nr.	$rmse_{tra}$	$rmse_{tst}$	$corr_{tra}$	$corr_{tst}$
Pedigree	–	–	7.0727 ± 8.5e−4	7.0720 ± 0.0138	0.8618 ± 5.9e−6	0.8617 ± 4.1e−7
M5′	–	28	6.4789 ± 0.0013	6.5444 ± 0.0056	0.8839 ± 1.1e−6	0.8815 ± 4.6e−6
WM	5, 5, 5	22	7.3951 ± 0.0292	7.4607 ± 0.0488	0.8559 ± 7.1e−5	0.8531 ± 1.9e−4
	5, 5, 5′	20	7.3010 ± 0.2672	7.3078 ± 0.3182	0.8631 ± 4.1e−5	0.8616 ± 8.9e−5
	7, 7, 7	30	7.3416 ± 0.0856	7.3557 ± 0.1122	0.8601 ± 4.7e−5	0.8589 ± 1.0e−4
	7, 7, 7′	40	7.2508 ± 0.0065	7.3105 ± 0.0102	0.8607 ± 4.2e−6	0.8583 ± 3.1e−5
	4, 6, 7′	19	7.5063 ± 0.3164	7.5229 ± 0.4079	0.8482 ± 6.0e−4	0.8457 ± 0.0011
COR	5, 5, 5′	17	6.8574 ± 0.0022	6.8696 ± 0.0066	0.8696 ± 8.9e−5	0.8692 ± 0.0002
	7, 7, 7′	28	6.7627 ± 0.0013	6.8190 ± 0.0061	0.8736 ± 3.4e−5	0.8715 ± 0.0002
	4, 6, 7′	17	6.7586 ± 0.0025	6.7941 ± 0.0099	0.8731 ± 8.7e−5	0.8718 ± 0.0004
WCOR	5, 5, 5′	19	6.8458 ± 0.0392	6.8646 ± 0.0400	0.8701 ± 0.0017	0.8694 ± 0.0016
	7, 7, 7′	29	6.7204 ± 0.0084	6.7873 ± 0.0148	0.8747 ± 0.0003	0.8722 ± 0.0006
	4, 6, 7′	19	6.7652 ± 0.0089	6.7876 ± 0.0132	0.8728 ± 0.0003	0.8720 ± 0.0004

Table 3
Results obtained by the different approaches when BVp, BVm and LactM are used as predictive attributes

Method	gran.	nr.	$rmse_{tra}$	$rmse_{tst}$	$corr_{tra}$	$corr_{tst}$
Pedigree	–	–	7.0727 ± 8.5e−4	7.0720 ± 0.0138	0.8618 ± 5.9e−6	0.8617 ± 4.1e−7
M5′	–	37	6.4581 ± 0.0087	6.5171 ± .0061	0.8847 ± 1.0e−5	0.8825 ± 2.3e−6
WM	5, 5, 5, 5	59	7.4293 ± 0.0282	7.4941 ± 0.0439	0.8444 ± 6.1e−5	0.8414 ± 4.5e−5
	5, 5, 5, 5′	48	7.3826 ± 0.0841	7.4235 ± 0.0785	0.8603 ± 2.1e−5	0.8585 ± 2.4e−5
	7, 7, 7, 7	138	7.2487 ± 0.0375	7.3369 ± 0.0238	0.8585 ± 5.3e−5	0.8545 ± 2.5e−5
	7, 7, 7, 7′	105	7.0436 ± 0.0165	7.1214 ± 0.0563	0.8619 ± 3.3e−5	0.8585 ± 7.7e−5
	7, 7, 6, 7′	99	7.0479 ± 0.0162	7.1091 ± 0.0560	0.8617 ± 3.2e−5	0.8591 ± 7.8e−5
	4, 6, 6, 7′	55	7.3884 ± 0.0177	7.4460 ± 0.02047	0.8578 ± 4.3e−6	0.8548 ± 3.9e−5
COR	5, 5, 5, 5′	39	6.7432 ± 0.0066	6.7963 ± 0.0133	0.8739 ± 0.0002	0.8718 ± 0.0005
	7, 7, 7, 7′	83	6.7322 ± 0.0113	6.8462 ± 0.0135	0.8755 ± 0.0005	0.8710 ± 0.0006
	7, 7, 6, 7′	79	6.7288 ± 0.0098	6.8167 ± 0.0154	0.8758 ± 0.0004	0.8723 ± 0.0006
	4, 6, 6, 7′	43	6.8273 ± 0.0075	6.8873 ± 0.02543	0.8710 ± 0.0004	0.8685 ± 0.0012
WCOR	5, 5, 5, 5′	39	6.7161 ± 0.0114	6.7854 ± 0.0222	0.8751 ± 0.0005	0.8723 ± 0.0009
	7, 7, 7, 7′	89	6.6743 ± 0.0126	6.7899 ± 0.0270	0.8771 ± 0.0005	0.8725 ± 0.0011
	7, 7, 6, 7′	83	6.6643 ± 0.0092	6.7581 ± 0.0290	0.8773 ± 0.0004	0.8736 ± 0.0011
	4, 6, 6, 7′	42	6.7085 ± 0.0144	6.7734 ± 0.0246	0.8753 ± 0.0006	0.8727 ± 0.0010

variable, $\alpha = 0.35$ in the convex combination based crossover and probability of mutation 0.2. Fig. 6 shows the linguistic variables returned by the GA for BVf (c), BVm (e) and BV (b) when using 7 labels as input and tuning the predictor (BVf, BVm). Part (g) shows the output of the GA for variable LactM when tuning the predictor that also uses this variable as input. As we can see in all the cases there are fuzzy sets that have their middle point quite close. By applying the process described in Section 5.2 to the output of the GA we get the partitions shown in parts (d), (f) and (h) of Fig. 6. Notice that variable BV is not modified by this editing process, while BVf drops from 7 to 4 labels and BVm and LactM drop from 7 to 6 labels.

Now, we describe the experiments carried out to learn LFR systems:

- The first method we have used is the WM algorithm. We have run WM by using as input the following definition for the linguistic variables:
  - The definitions obtained by constructing symmetrical partitions with 5 and 7 labels. Denoted by $(5,5,5)$, $(7,7,7)$, $(5,5,5,5)$ and $(7,7,7,7)$, the order of the variables being BVf, BVm, [LactM,] BV.
  - The definitions obtained by the GA-based tuning process with 5 and 7 labels. Denoted by $(5,5,5)^t$, ...
  - The definitions used by editing the GA output. In this case the editing process does not have any effect on the 5-labels definitions, so only the 7-labels edited definitions are shown. By $(4,6,7)^t$ we denote the case in which BVf, BVm and BV are reduced to 4, 6 and 7 labels. Analogously, by $(7,7,6,7)$ we denote the case

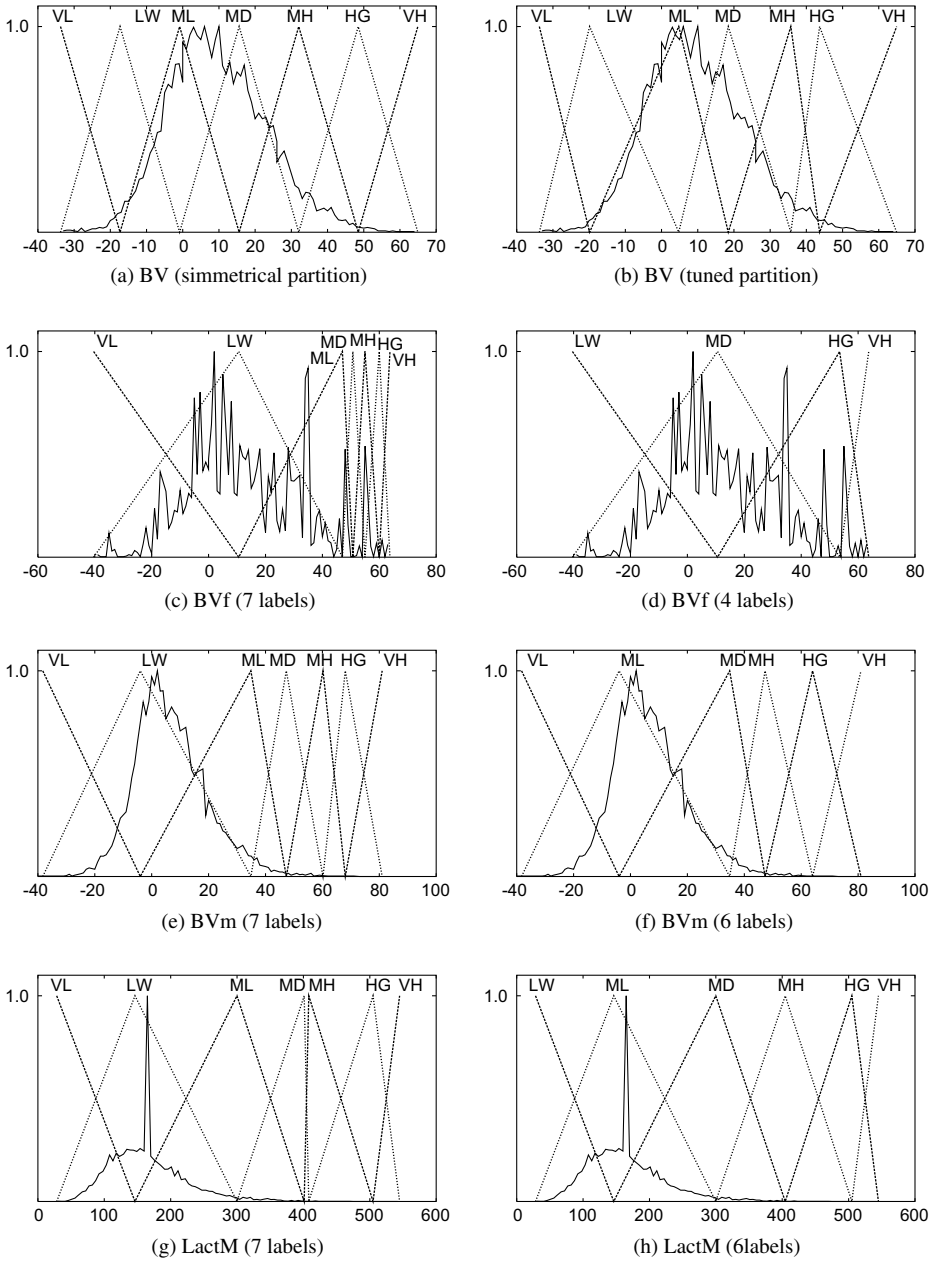


Fig. 6. Variables involved in the problem and their associated linguistic variables.

in which BVf, BVm, LactM and BV are reduced to 7, 7, 6 and 7 labels, respectively. Notice that, as the tuning process is global, different definitions are obtained for the same variable (e.g. BVf) when different sets of variables are involved in the predictor.
– Finally, in the case of the more complex predictor, namely the one that uses LactM, we have try an additional definition: $(4,6,6,7)^t$. This definition is made up by using $(4,6,7)^t$ for (BVf, BVm, LactM) and the 6-labels based definition taken from $(7,7,6,7)^t$ for LactM. The idea is to study the effect of this simpler partition over the obtained system: precision and number or rules.

The results are shown in Tables 2 and 3, where we show the same data as for the RR-based systems.

- From the results obtained by applying WM algorithm we select the tuned partitions as the definitions to be used as input for COR and WCOR. In COR and WCOR the following design decisions have been taken: the probabilistic model induced at each generation is learnt from the best 50% individuals of the current population; the current population and the sampled one are combined by truncation, i.e., they are merged, sorted by fitness and the best population-size individuals are retained; the population size is 200 and the maximum number of allowed iterations is 100.
  In this case due to the stochastic nature of the search algorithms, each (5 cross validated) experiment has been repeated ten times, and the results (Tables 2 and 3) show the mean and deviation over the 10 independent runs.

Figs. 7 and 8 show, in form of decision table, the systems obtained by COR and WCOR when using the definition of linguistic variables denoted by $(4,6,7)^t$. Analogously, Figs. 9 and 10 show the systems obtained when using the definition of linguistic variables denoted by $(4,6,6,7)^t$.

BVF ↓ \ BVM ↓	VL	ML	MD	MH	HG	VH
LW	LW	LW	MD	MD	MH	
MD	VL	ML	MD	MH		HG
HG	MD	MD	HG	VH		
VH		MH	HG		VH	

Fig. 7. LFR-based system obtained when using the definition $(4,6,7)^t$ as input for COR.

BVF ↓ \ BVM ↓	VL	ML	MD	MH	HG	VH
LW	LW 0.92	LW 0.63	MD 0.29	MD 1.00	MH 0.70	
MD	VL 0.45	ML 0.61	MD 0.70	MH 0.57	MD 0.29	HG 1.00
HG	MD 0.16	MD 0.62	HG 0.69	VH 0.42	VH 0.88	
VH		MH 0.28	HG 0.23		VH 0.72	

Fig. 8. LFR-based system obtained when using the definition $(4,6,7)^t$ as input for WCOR.

6.1. Analysis

In this section we analyze the results from two different perspectives: precision and complexity/comprehensibility of the obtained systems:

- Precision. We based our analysis about precision on the results obtained over the test.
  – With respect to the precision of the obtained systems it is clear that regression rules get the best result. This is not an unexpected result, because the *actual* value we are using in the supervised learning task was obtained by BLUP, which relies on defining and simultaneously solving a set of linear equations. With respect to the prediction carried out by the pedigree index, the systems obtained with RRs reduce the error in more than 0.5 units and improve the correlation in more than two points. Furthermore, these systems always have less error (greater correlation) than all the systems obtained by using LFRs.
  – Between the LFR-based systems it is clear that the two following comments hold: (1) The systems learnt by WCOR improve on those learnt by COR, which are better than the ones learnt by using WM; (2) it is better to use 7 labels than 5. The best results are obtained by WCOR when using definition $(7,7,6,7)^t$, which reduces error of pedigree index in more than 0.3 units and improves the correlation in more than 1.2 points.
  – With respect to the use or otherwise of the LactM variable, it can be observed that there is only a slight improvement on the precision of the obtained systems when this variable is used. That is, we should

		LACT ↓					
BVF ↓	BVM ↓	LW	ML	MD	MH	HG	VH
LW	VL		LW				
	ML	LW	LW	ML			
	MD	MD	ML	MD	MD		
	MH			MD	MH		
	HG				HG		
	VH						
MD	VL		VL				
	ML	ML	ML	LW	ML		
	MD		MH	MD	MD	ML	
	MH		MH	MD	MD		
	HG		HG				
	VH					MH	HG
HG	VL		ML	ML			
	ML	MH	MD				
	MD		HG	HG		VH	
	MH		VH	VH	HG		
	HG		HG	VH			
	VH						
VH	VL						
	ML	MD	MH	MD			
	MD		HG	MH			
	MH						
	HG						
	VH						

Fig. 9. LFR-based system obtained when using the definition $(4,6,6,7)^t$ as input for COR.

BVF ↓	BVM ↓	LACT ↓ LW	ML	MD	MH	HG	VH
LW	VL		VL .53	LW .92			
	ML		VL .52	ML .53			
	MD		ML .07	MD .84		ML .62	
	MH			ML .67	HG .84		
	HG						
	VH						
MD	VL		VL .58				
	ML	ML .34	ML 1.0	LW .39	ML .64		
	MD	MH .19	MH .58	MD .54	MD .49		
	MH		MD .34	MH .54	MD .58		
	HG		HG .52	VH .41	MD .73		
	VH					MH .66	HG .17
HG	VL		MD .01	ML .63			
	ML	MH .47	MD .81				
	MD		VH .67	HG .34	MH .55		
	MH			VH .61			
	HG		VH 1.0	HG .06			
	VH						
VH	VL						
	ML	MD .52	MH .25	MD .99			
	MD		HG .47	MH .30			
	MH						
	HG			VH .83			
	VH						

Fig. 10. LFR-based system obtained when using the definition $(4,6,6,7)^t$ as input for WCOR.

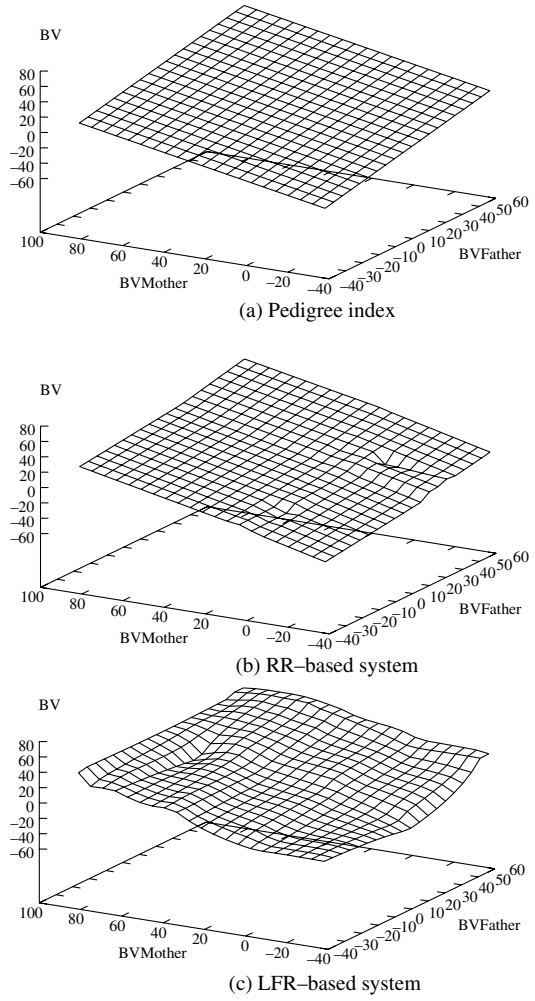


Fig. 11. Modeling of $[\mathsf{BVf} \wedge \mathsf{BVm} \rightarrow \mathsf{BV}]$ carried out by (a) pedigree index, (b) regression rules and (c) linguistic fuzzy rules.

suppose that although this variable is relevant to our problem, it is not determinant once we know the value of BVm.

– Fig. 11 shows the modeling surfaces for the predictor $[\mathsf{BVf} \wedge \mathsf{BVm} \rightarrow \mathsf{BV}]$ by running pedigree index, the RR-based system, and the WCOR $(4,6,7)^t$ LFR-based system, over a data set artificially created by systematically generating points in a grid defined over $\mathsf{BVf} \times \mathsf{BVm}$. In the plots we can observe the nature (linearity and non-linearity) of each method.

- Complexity/Comprehensibility. We base our analysis about the complexity of a system on both the number of rules and the type of rules.
  – With respect to RR-based systems the number of rules when using or otherwise variable LactM is 37 and 28, respectively. With respect to the complexity of each rule, we can see how the post-processing of the rules contributes to simplify the system. As an example, when LactM is used, the antecedent of the originally tree-extracted 17th rule looks as follows:

If $\mathsf{BVf} \leqslant 21.045$ and $\mathsf{BVm} \leqslant 5.775$ and $\mathsf{BVm} > -3.455$ and $\mathsf{BVf} \leqslant 3.905$ and $\mathsf{BVf} > -11.815$ and $\mathsf{BVm} > 0.795$ and $\mathsf{BVf} > -6.425$ and $\mathsf{LactM} \leqslant 167.54$ and $\mathsf{LactM} > 163.12$ and $\mathsf{BVm} > 2.05$ Then ...

which is far more complex than its post-processed version shows in Fig. 5. However, even with this new writing, regression rules are quite difficult to understand because the consequent is a regression equation.

  – In the case of LFR-based systems it is clear that the number of rules largely depends on the cardinality of the linguistic variables used. However, due to the tuning process carried out, we can observe how the systems learnt by COR/WCOR using definitions $(4,6,7)^t$ and $(4,6,6,7)^t$ have 17/19 and 43/42 rules, respectively, which are good figures when compared with RR-based systems. Furthermore, the consequent

used in this case is far more interpretable than a regression equation, and so we can conclude that a human expert will prefer to deal with linguistic fuzzy rules than with regression rules.

7. Concluding remarks

In this paper an application of rule-based systems to a farming problem has been studied. The problem here considered is the breeding value estimation of Manchego ewes in their early stages of life, that is, before they become mothers. After the first offspring-birth and its corresponding controlled lactation BLUP methodology can be used, but until that moment a simple predictor is considered: the pedigree index.

The goal of the paper was threefold: firstly, we have studied if the estimation made by the pedigree index can be improved by using rule-based systems; secondly, we have studied the pros and cons of solving the problem by using two different types of systems: regression rules and linguistic fuzzy rules; and finally, we have presented an EDA-based approach to the problem of learning weighted linguistic fuzzy rules.

The results obtained show that the pedigree index based estimations are significantly improved when using the rule-based predictors, attaining two points of increase in the correlation coefficient when regression rules are used. On the other hand, we have found that EDAs are a suitable technique to approach the problem of learning weighted linguistic fuzzy rules. Finally, with respect to the comparison between the two types of rules used, it is clear that regression rules achieve the highest precision, but linguistic fuzzy rules are more easier to understand for human experts.

For the future we plan to continue working on this problem and other related farming problems by identifying tasks in which intelligent/expert systems can be applied. In addition, encouraged by the results obtained by our EDA-based approach to WCOR, we plan to develop new models in which some dependencies are allowed (e.g., direct relations between consequents and their weights).

References

Alcalá, R., Casillas, J., Cordón, O., & Herrera, F. (2002). Improving simple linguistic fuzzy models by means of the weighted COR methodology. In F. J. Garijo, J. C. Riquelme, & M. Toro (Eds.), *Advances in artificial intelligence—IBERAMIA 2002* (pp. 294–302). Springer-Verlag.

Bárdossy, A., & Duckstein, B. (1995). Fuzzy rule-based modeling with application to geophysical, biological and engineering systems. CRC Press.

Casillas, J., Cordón, O., & Herrera, F. (2002). COR: A methodology to improve ad hoc data-driven linguistic rule learning methods by inducing cooperation among rules. *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics, 32*, 526–537.

Cordón, O., Herrera, F., Hoffmann, F., & Magdalena, L. (2001). Genetic fuzzy systems: Evolutionary tuning and learning of fuzzy knowledge bases. World Scientific.

Cordón, O., Herrera, F., & Villar, P. (2001). Generating the knowledge base of a fuzzy rule-based system by the genetic learning of data base. *IEEE Transactions on Fuzzy Systems, 9*(4), 667–674.

CRDECM (2004). Manchego lamb Specific Guarantee (in Spanish). Available from http://www.corderomanchego.org.

CRDOQM (2004). Manchego cheese Guarantee of Origin (in Spanish). Available from http://www.mapya.es/es/alimentacion/pags/Denominacion/htm/queso_manchego.htm.

Flores, J., Gámez, J. A., & Puerta, J. M. (2005). Learning linguistic fuzzy rules by using estimation of distribution algorithm as the search engine in the COR methodology. In *Towards a new evolutionary computation. Advances in estimation of distribution algorithms. Studies in fuzziness and soft computing* (vol. 192, pp. 259–280). Springer-Verlag.

Gallego, L., Torres, A., & Caja, G. (Eds.) (1994). Flock sheep: Manghega breed (in Spanish). Ediciones Mundi-Prensa.

Gámez, J. (2005). Mining the ESROM: a study of breeding value prediction in Manchego sheep by means of classification techniques plus attribute selection and construction. Tech. Rep. DIAB-05-01-3, Computer Systems Department. University of Castilla-La Mancha.

IRG-NCR (2005). FuzzyJ toolKit & Fuzzy Jess URL. http://www.iit.nrc.ca/IR_public/fuzzy/fuzzyJToolkit2.html.

Jurado, J. (1994). Genetic evaluation of reproductives in Manchego sheep (in Spanish). In Gallego et al. (1994), pp. 369–388.

Larrañaga, P., Etxeberria, R., Lozano, J., & Peña, J. (1999). Optimization by learning and simulation of Bayesian and Gaussian networks. Tech. Rep. EHU-KZAA-IK-4-99, University of the Basque Country.

Larrañaga, P., & Lozano, J. (2001). *Estimation of distribution algorithms. A new tool for evolutionary computation*. Kluwer Academic Publishers.

Mamdani, E., & Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man–Machine Studies, 7*, 1–13.

Michalewicz, Z. (1996). *Genetic algorithms + data structures = evolution programs*. Springer-Verlag.

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

Orchand, R. (2001). Fuzzy reasoning in Jess: The FuzzyJ toolkit and FuzzyJess. In *Proceedings of the 3rd international conference on enterprise information systems*, pp. 533–542.

Quinlan, J., (1992). Learning with continuous classes. In *Proceedings of the 5th Australian joint conference on AI*, pp. 343–348.

Sugeno, M., & Kang, G. (1985). Structure identification of fuzzy models. *Fuzzy Sets and Systems, 28*(1), 15–33.

Takagi, T., & Sugeno, M. (1985). Fuzzy identification of systems and its application to modelling and control. *IEEE Transactions on Systems, Man and Cybernetics, 15*(1), 116–132.

Wang, L., & Mendel, J. (1992). Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man, and Cybernetics, 22*(6), 1414–1427.

Wang, Y., & Witten, I. (1997). Induction of model trees for predicting with continuous classes. In *Proceedings of the poster papers of the European conference on machine learning*. University of Economics, Faculty of Informatics and Statistics, Prague.

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

Zadeh, L. (1965). Fuzzy sets. *Information and Control, 8*, 338–353.

Zadeh, L. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics, 3*(1), 28–44.

Zadeh, L. (1975). The concept of a linguistic variable and its application to approximate reasoning. *Information Science, 8*, 199–249.