Expert Systems with Applications 36 (2009) 12225–12235

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Multinomial logistic regression and product unit neural network models: Application of a new hybrid methodology for solving a classification problem in the livestock sector

Mercedes Torres [a,*], Cesar Hervás [b], Carlos García [a]

[a] Department of Management and Quantitative Methods, ETEA, Spain
[b] Department of Computing and Numerical Analysis, University of Córdoba, Spain

ARTICLE INFO

Keywords:
Evolutionary computation
Neural networks
Product units
Classification
Livestock application

ABSTRACT

This work presents a new approach for multi-class pattern recognition based on the hybridization of a linear and nonlinear model. We propose multinomial logistic regression where some new covariates are defined by a product unit neural network, where in turn, the nonlinear basis functions are constructed with the product of the inputs raised to arbitrary powers. The application of this methodology involves, first of all, training the coefficients and the basis structure of product unit models using techniques based on artificial neural networks and evolutionary algorithms, followed by the application of multinomial logistic regression to both the new derived features and the original ones. To evaluate the efficacy of our technique we pose a difficult problem, the classification of sheep with respect to their milk production in different lactations, using covariates that only involve the first weeks of lactation. This enables the productive capacity of the animal to be identified more rapidly and leads to a faster selection process in determining the best producers. The results obtained with our approach are compared to other classification methodologies. Although several of these methodologies offer good results, the percentage of cases correctly classified was higher with our approach, which shows how instrumental the potential use of this methodology is for decision making in livestock enterprises, a sector relatively untouched by the technological innovations in business management that have been appearing in the last few years.

1. Introduction

Classification problems are often encountered in many different fields, such as biology (Hajmeer & Basheer, 2003), medicine (Schwarzer, Vach, & Schumacher, 2000; Youngdai & Sunghoon, 2006), computer vision (Subasi, Alkan, & Koklukaya, 2005), artificial intelligence and remote sensing (Yuan-chin & Sung-Chiang, 2004). In the business world, applications of this type are becoming more and more frequent: in finance (Parag & Pendharkar, 2005; Rada, 2008; Tian-Shyug et al., 2006), marketing (Kaefer, Heilman, & Ramenofsky, 2005), and human resource management (Sexton & McMurtrey, 2005). There has been a renewed interest in this type of technique in the last few years due to the difficulties inherent in such new problems as dealing with data mining, document classification, financial forecasts, web-mining, etc. This great practical interest in classification problems has motivated researchers to develop a huge number of methods as quantitative models for classification purposes (i.e. see Bernadó & Garrell, 2003; Duda & Hart, 2001). Linear Discriminant Analysis (LDA) (Johnson & Wichern, 2002) was the first method developed to address the classification problem from a multidimensional perspective. LDA has been used for decades as the main classification technique and it is still being used, at least as a reference point, to compare the performance of new techniques. Another widely used parametric classification technique, developed to overcome LDA's restrictive assumptions (multivariate normality, equality of dispersion matrices between groups), is Quadratic Discriminant Analysis (QDA). The Logistic Regression model (LR) has also been widely used in statistics for many years and has recently been the object of extensive study in the machine learning community (Dreiseitl & Ohno-Machado, 2002; Duda & Hart, 2001; Hosmer & Lemeshow, 2000; Yuan-chin & Sung-Chiang, 2004). During the last two decades several alternative non-parametric classification techniques have also been developed, including, among others, mathematical programming techniques (Fredd & Glover, 1981), multicriteria decision aid methods (Doumpos, Zopounidis, & Pardalos, 2000), neural networks (Patuwo, Hu, & Hung, 1993; Widrow, 1962) and machine learning approaches (Kordatoff & Michlaski, 1990).

However, in spite of the great number of techniques developed to solve classification problems, there is no optimum methodology

[*] Corresponding author. Tel.: +34 957 222100; fax: +34 957 222101.
E-mail addresses: mtorres@etea.com (M. Torres), chervas@uco.es (C. Hervás), cgarcia@etea.com (C. García).

0957-4174/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2009.04.070

or technique to resolve a specific problem and this is why the comparison and combination of different types of classification is a common practice today (Lin, Lee, & Lee, 2008; Major & Ragsdale, 2001; Martínez, Hervás, et al., 2006).

As a matter of fact, the present work deals with the application of a new hybrid methodology that combines multinomial logistic regression and unit-product network models as an alternative to other well known techniques (some relatively recent and others more traditional) for solving a real classification problem in the livestock sector. It is an extension to more than two classes of a recent work which propose this method for two classes (Hervás & Martínez, 2007).

The combination of the two techniques is justified by the fact that, although LR is a simple and useful procedure, it poses problems when applied to a real problem of classification, where we frequently cannot make the stringent assumption that there are additive and purely linear effects of the covariates. These difficulties are usually overcome by augmenting or replacing the input vector with new variables, basis functions, which are transformations of the input variables, and then by using linear models in this new space of derived input features. Methods like sigmoidal feedforward neural networks (Bishop, 1995), projection pursuit learning (Friedman & Stuetzle, 1981), generalized additive models (Hastie & Tibshirani, 1990) and PolyMARS (Kooperberg, Bose, & Stone, 1997), and a hybrid of multivariate adaptive splines (Friedman, 1991; Tian-Shyug et al., 2006), specifically designed to solve classification problems, can be seen as different non-linear basis function models.

The unit-product networks are similar to standard sigmoidal neural networks but are based on multiplicative nodes instead of additive ones. These functions correspond to a special type of neural networks called product-unit neural networks (PUNN) introduced by Durbin and Rumelhart (1989), and developed by Ismail and Engelbrecht (1999) and Schmitt (2002). The nonlinear basis functions of the model are constituted by the product of the variables initially included in the problem formulation raised to arbitrary powers. We estimate the variables' exponents and determine the optimum number of product units in several steps (solving one of the main problems in the use of this type of models). In a first step an evolutionary algorithm (EA) is applied that optimizes a loss function. However, these algorithms are relatively poor at finding the precise optimum solution in the region that the algorithm converges to. So a local optimization algorithm was used, in a second step, to improve the EA's lack of precision. Once the basis functions have been determined by the EA, the model is linear in these new variables together with the initial covariates, and the fitting proceeds with the standard maximum likelihood optimization method for multinomial logistic regression. Finally, a backward-step procedure is applied, pruning variables sequentially to the model obtained previously until further pruning does not improve the fit. In this way the models obtained can be simpler and more comprehensible for researchers in the livestock sector.

The performance of the proposed methodology was evaluated in a real problem which consists of classifying a sheep flock into three classes, according to its milk production capacity, by using solely the first milk controls, and thus shortening the current evaluation process that uses the selection schema of the Manchegan breed (Montoro & Pérez-Guzmán, 1996). Three classes are established: the best productive ones (called "good"), the worst (called "bad") and the intermediate (called "normal"). With the results of the classification, the stock farmer would be able to identify the most productive animals in the flock with a minimum of necessary information and could then contribute to the genetic progress of the breed. Moreover, these models could lead to a decrease in the great differences in production that have been found in the last few years between different Spanish sheep breeds, like the Manchegan, with respect to other breeds (the French Lacaune, for example) (Buxadé, 1998; Gallego & Bernabeu, 1994; Serrano & Montoso, 1996).

So we have here an application of new computational methodologies for the management of a dairy, a sector relatively untouched by the technological innovations in business management that have been appearing in the last few years (Torres, Hervás, & Amador, 2005). In general, the greater part of operational researchers' and agrarian economists' attention has been concentrated on the area of animal feed, due more to their connection with the animal food industry than to any connection with the dairy establishments themselves.

Simultaneously we compare our model results with those obtained by a standard multinomial logistic regression that uses only the original input variables, to verify the advantages of our approach. Furthermore, other classification algorithms based on artificial neural networks were applied (Dreiseitl & Ohno-Machado, 2002). Specifically we use a standard multilayer perceptron model (MLP) that uses a back-propagation learning algorithm (Hayken, 1994; Williams & Minai, 1990), and another MLP model, where an evolutionary algorithm is coupled with a pruning one to eliminate non-significant model coefficients (Bebis & Georgipoulos, 1997; Honaver & Balakrishnan, 1998) (from now on we will call this model MLPEA). Thus we attempt to achieve the neuronal network architecture that will allow us to predict what sheep productive capacity will be relying on the least possible amount of information. We have also applied the second most popular choice of network in classification problems; the radial basis function network (RBF). This type of network has a very strong mathematical foundation and uses normalized Gaussian radial basis functions (Orr et al., 1996; Oyang, Hwang, Ou, Chen, & Chen, 2005).

Finally we apply other well known classification methods to our problem (some of them of statistical origin and others from the computational field) to compare their classification capacity with ours. We have used: the classical decision tree C4.5 (Quinlan, 1993) with pruning (http://www.cse.unsw.edu.au/quinlan/); three statistical algorithms: a Linear Discriminant Analysis, LDA, where hypothetically the instances within each class are normally distributed with a common covariance matrix; a quadratic discriminant analysis, QDA, where each covariance matrix is different and estimated by the corresponding sample covariance matrix; and, finally, the K-Nearest Neighbour algorithm (KNN) (Dasarathy, 1991; Hervás & Martínez, 2007; Kaefer et al., 2005).

The rest of the paper is organized as follows. Section 2 describes the proposed logistic regression model and the other methodologies applied, Section 3 explains the process to obtain the data set as well as the procedure to select the variables to include in the models. The results of the experiment are tabulated and discussed in Section 4. Finally, conclusions are presented in Section 5.

2. Methodology

2.1. Multinomial logistic regression with product unit covariates

Logistic regression methods are common statistical tools for modelling discrete response variables such as binary, categorical and ordinal responses. If the values of the response of these variables refer to different categories where a specific group of elements (called a sample), can be grouped into a class according to a series of characteristics which have previously been measured for each element, we are confronted with what is called a classification problem. So, in a classification problem we find the following elements: a number of features x_i, $i = 1, \ldots, p$ which are measured from each element in the sample; a finite number of classes K where the elements have to be classified as well as the group of elements (individuals or objects) that make up the sample Then if we have measurements of each variable x_i for each element of a group of size N we can represent the sample by $D = \{(\mathbf{x}_n,\mathbf{y}_n);\ n = 1,2,\ldots,N\}$ where $\mathbf{x}_n = (x_{1n},\ldots x_{pn})$ is the vector of input variables taking values in $\Omega \subset \mathrm{R}^k$ and $\mathbf{y}_n$ is the class level of the nth individual. The class level is represented with a "1-of-K" encoding vector $\mathbf{y} = (y^{(1)}, y^{(2)},\ldots,y^{(K)})$, such as $y^{(l)} = 1$ if $\mathbf{x}$ corresponds to an example belonging to class l and $y^{(l)} = 0$ otherwise. The problem is, based on the training sample, to find a decision function $C{:}\Omega \to \{1,2,\ldots,K\}$ for classifying the individuals. In other words, C provides a partition, say $D_1, D_2,\ldots, D_k$, of Ω, where D_l corresponds to the lth class, $l = 1,2,\ldots,K$, and measurements belonging to D_l will be classified as coming from the lth class. The objective is to find the decision function permitting the identification of the class where each element in the sample belongs, with the smallest error possible. It is usually assumed that the data composing the training sample are independent and identically distributed in an unknown probability distribution. Suppose that the conditional probability that $\mathbf{x}$ belongs to class l verifies: $p(y^{(l)} = 1|\mathbf{x}) > 0,\ l = 1,2,\ldots,K,\ \mathbf{x} \in \Omega$, and sets the function:

$$f_l(\mathbf{x},\boldsymbol{\theta}_l) = \log \frac{p(y^{(l)} = 1|\mathbf{x})}{p(y^{(K)} = 1|\mathbf{x})}, \quad l = 1,2,\ldots,K,\ \mathbf{x} \in \Omega \tag{1}$$

where $\boldsymbol{\theta}_l$ is the weight vector corresponding to class l and for identifiability $f_K(\mathbf{x},\boldsymbol{\theta}_K) \equiv 0$.

Under multinomial logistic regression, the probability that $\mathbf{x}$ belongs to class l is given by

$$p(y^{(l)} = 1|\mathbf{x},\boldsymbol{\theta}) = \frac{\exp f_l(\mathbf{x},\boldsymbol{\theta}_l)}{\sum_{l=1}^{K} \exp f_l(\mathbf{x},\boldsymbol{\theta}_l)}, \quad \text{for } l = 1,2,\ldots,K \tag{2}$$

where $\boldsymbol{\theta} = (\boldsymbol{\theta}_1,\boldsymbol{\theta}_2,\ldots,\boldsymbol{\theta}_{K-1})$.

Regression logistics (or soft-max in neural network literature) (Cox, 1970a; Cox & Snell, 1989b; Hosmer & Lemeshow, 2000) uses a classification rule which is based on the optimal Bayes rule and tries to assign each element in the sample to a class where it has the greatest possibility of belonging. In other words, an individual should be assigned to that class which has the maximum probability, given the vector measurement $\mathbf{x}$:

$$C(\mathbf{x}) = \hat{l},\ \text{where } \hat{l} = \arg\max_{l}\ f_l(\mathbf{x},\hat{\boldsymbol{\theta}}_l),\ \text{for } l = 1,\ldots,K \tag{3}$$

On the other hand, because of the normalization condition we have that, $\sum_{l=1}^{K} p(y^{(l)} = 1|\mathbf{x},\boldsymbol{\theta}) = 1$ and the probability that one of the classes (in our case the latest) need not be estimated. Observe that we have considered $f_K(\mathbf{x},\boldsymbol{\theta}_K) \equiv 0$. In our application K is equal to three, because the sheep are catalogued in three productive classes ("good", "normal" and "bad") therefore we only have to estimate the probability for two classes and the discrimination will depend only on two discriminating functions.

The usual parametric approach to a multinomial logistic regression problem is to use the linear model in the input variables, although in practice it may be desirable to model the predictor effects by using smooth, nonlinear functions. The logistic model as a nonlinear regression model is a special case of a generalized linear model (McCullagh & Nelder, 1989).

$$f_l(\mathbf{x},\boldsymbol{\theta}_l) = \boldsymbol{\theta}_{1l}(x_1) + \ldots + \boldsymbol{\theta}_{pl}(x_p), \quad l = 1,2,\ldots,k \tag{4}$$

where $\boldsymbol{\theta}_{il}(x_i)$, $i = 1,2,\ldots,p$ are one-dimensional functions.

What we propose in this work is the application of multinomial logistic regression models where the function $f_l(\mathbf{x},\boldsymbol{\theta}_l)$ is established partly linearly and partly nonlinearly. The nonlinear term is constituted by basis functions given by products of the input variables raised to real powers, which represents the possible interactions between the variables. The general expression of the model is given by:

$$f_l(\mathbf{x},\boldsymbol{\theta}_l) = \alpha_0^l + \sum_{i=1}^{p} \alpha_1^l x_i + \sum_{j=1}^{m} \beta_j^l \prod_{i=1}^{p} x_i^{w_{ji}}, \quad l = 1,2,\ldots,K-1 \tag{5}$$

where $\boldsymbol{\theta}_l = (\boldsymbol{\alpha}^l, \boldsymbol{\beta}^l, \mathbf{W})$, $\boldsymbol{\alpha}^l = \left(\alpha_0^l, \alpha_1^l, \ldots, \alpha_p^l\right)$, $\boldsymbol{\beta}^l = \left(\beta_1^l, \ldots, \beta_m^l\right)$ and $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2,\ldots,\mathbf{w}_m)$, with $\mathbf{w}_j = (w_{j1}, w_{j2},\ldots,w_{jp})$, $w_{ji} \in \mathbf{R}$.

Hence we shall make use of the more general basis function models that assume that the non-linear part of (5) corresponds to a special class of feed-forward neural networks, namely the product-unit neural network (PUNN), introduced by Durbin and Rumelhart (1989) and studied in several works (Engelbrecht & Ismail, 2000; Hervás & Martínez, 2007; Ismail & Engelbrecht, 1999; Martínez, Hervás, & Martínez, 2006; Martínez, Martínez, & Hervás, 2006; Schmitt, 2002). They are an alternative to the standard sigmoidal neural networks and are based on multiplicative nodes instead of additive ones. This class of multiplicative neural networks comprises such types as sigma-pi networks and product unit networks. If the exponents in (5) are {0,1} we obtain a higher-order unit, also known by the name of sigma-pi unit. In contrast to the sigma-pi units, in the product-unit the exponents are not fixed and may even take real values. In this way we get more flexible models and avoid the huge number of coefficients involved in the polynomial model. To avoid the problem that could result from networks containing product units that receive negative inputs and weights that are not integers, the values for the input variables (x_i) are limited to positive ones (because, as we know, a negative number raised to some non-integer power yields a complex number). Since neural networks with complex outputs are rarely used in applications, Durbin and Rumelhart (1989) suggest discarding the imaginary part and using only the real component for further processing. This manipulation would have disastrous consequences for the Vapnik–Chervonenkis (VC) dimension when we consider real-valued inputs. No finite dimension bounds could be derived for networks containing such units (Schmitt, 2002).

Some advantages of product-unit based neural networks (PUNNs) are their increased information capacity and the ability to form higher-order input combinations. Durbin and Rumelhart (1989) determined empirically that the information capacity of product units (measured by their capacity for learning random Boolean patterns) is approximately 3N, compared to 2N for a network with additive units for a single threshold logic function, where N denotes the number of inputs to the network. Besides that, it is possible to obtain the upper bounds of the VC dimension in product-unit neural networks similar to those obtained in sigmoidal neural networks (Ismail & Engelbrecht, 1999). It is a consequence of the Stone–Weierstrass Theorem to prove that product-unit neural networks are universal approximators (Schmitt, 2002) (observe that polynomial functions in several variables are a subset of product-unit models). A disadvantage of this type of nets with respect to standard sigmoidal ones is the greater degree of difficulty for the corresponding training process since small changes in exponent values can provoke great changes in the error surface This type of nets presents a greater number of local minimums thus increasing the possibility of getting trapped in them For this reason local back-propagation search algorithms are not very efficient for the training of product units (Martínez, Hervás, et al., 2006, 2006), To overcome this problem we use an evolutionary algorithm as part of the process for the estimation of parameters (Goldberg, 1989, 1989a, 1989b) which we explain in the following section.

2.2. Hybrid estimation methodology

The methodology proposed is based on the combination of an evolutionary algorithm (global explorer) and a local optimization procedure (local exploiters) carried out by a maximum-likelihood procedure. To perform maximum likelihood (ML) estimation of

$\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_{K-1})$, that is, the components of the vector weights, estimated in turn from the training data set, one can minimize the negative log-likelihood function:

$$L(\boldsymbol{\theta}) = -\frac{1}{N}\sum_{n=1}^{N} \log p(\mathbf{y}_n|\mathbf{x}_n, \boldsymbol{\theta}) = \frac{1}{N}\sum_{n=1}^{N}\left[-\sum_{l=1}^{K} y_n^{(l)} f_l(\mathbf{x}_n, \boldsymbol{\theta}_l) + \log\sum_{l=1}^{K} \exp f_l(\mathbf{x}_n, \boldsymbol{\theta}_l)\right] \quad (6)$$

Maximum likelihood and the Newton–Raphson algorithm is the traditional way to solve logistic regression. Typically, the algorithm converges, since log-likelihood is concave. However, in our approach, the non-linearity of the PUNN implies that the corresponding Hessian matrix is generally indefinite and the likelihood has more local maxima (so gradient-based methods are not appropriate to maximize the log-likelihood function). Moreover, it is important to point out that computation is prohibitive when the number of variables is large. Another difficulty is that the optimal number of hidden nodes in the product-unit neural network is unknown (i.e. the number of hidden nodes in the product-unit neural network). For these reasons we propose a method to estimate the parameters of the model in four steps.

(1) In a first step, an evolutionary algorithm (EA) is applied to design the structure (the number of hidden nodes and number of connections within layers) and train the weights of the exponents of the potential basis functions. That is, the evolutionary process determines the number m of potential basis functions (which represent the nonlinear part of the function $f(\mathbf{x}, \boldsymbol{\theta})$ of the model) and the corresponding vector of exponents $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_m)$.To apply evolutionary neural network techniques, we consider a product-unit neural network with the following structure: an input layer with p nodes, a node for every input variable, a hidden layer with m nodes, and an output layer with $K - 1$ nodes (K being the number of classes). There are no connections between the nodes of a layer, and none between the input and output layers either. The activation function of the jth node in the hidden layer is given by

$$B_j(\mathbf{x}, \mathbf{w}_j) = \prod_{i=1}^{p} x_i^{w_{ji}} \quad (7)$$

where p is the number of inputs, w_{ji} is the weight of the connection between input node i and hidden node j and $\mathbf{w}_j = (w_{j1}, \ldots, w_{jp})$ the weights vector. The activation function of the output node l is given by h_l:

$$h_l(\mathbf{x}, \boldsymbol{\beta}^l, \mathbf{W}) = \beta_0^l + \sum_{j=1}^{m} \beta_j^l B(\mathbf{x}, \mathbf{w_j}) \quad (8)$$

where β_j^l is the weight of the connection between the hidden node j and the output node l and β_0^l the corresponding bias. The transfer function of all hidden and output nodes is the identity function. The parameters $\mathrm{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_m)$ are estimated by means of an EA, described further on, which optimizes the error function given by the negative log-likelihood for N observations associated to the product-unit model previously explained in (6), substituting $f_l\ (\mathbf{x}_n, \boldsymbol{\theta}_l)$ for h_l:

$$L^*(\boldsymbol{\beta}, \mathbf{W}) = \frac{1}{N}\sum_{n=1}^{N}\left[-\sum_{l=1}^{K-1} y_n^{(l)} h_l(\mathbf{x}_n, \boldsymbol{\beta}^l, \mathbf{W}) + \log\sum_{l=1}^{K-1} \exp h_l(\mathbf{x}_n, \boldsymbol{\beta}^l, \mathbf{W})\right] \quad (9)$$

Although in this step the evolutionary process obtains a specific value for the $\boldsymbol{\beta}$ vector parameter, only the $\hat{\mathbf{W}} = (\hat{\mathbf{w}}_1, \hat{\mathbf{w}}_2, \ldots, \hat{\mathbf{w}}_m)$ estimated vector parameter that builds the basis functions is considered. The third step determines the value for $\boldsymbol{\beta}$ (the vector parameter) and $\boldsymbol{\alpha}$ (the coefficient vector).

(2) In a second step, a transformation of the input space is considered adding the nonlinear transformations of the input variables given by the basis functions obtained by the EA in the first step, to the initial covariates:

$$\begin{aligned} &H : \mathbf{R}^p \to \mathbf{R}^{p+m} \\ &(x_1, x_2, \ldots, x_p) \to (x_1, x_2, \ldots, x_p, z_1, \ldots, z_m) \\ &\text{where } z_1 = B_1(\mathbf{x}, \hat{\mathbf{w}}_1), \ldots, z_m = B_m(\mathbf{x}, \hat{\mathbf{w}}_m) \end{aligned} \quad (10)$$

The model is linear in these new inputs together with initial covariates.

(3) In the third step the negative log-likelihood function for N observations is minimized:

$$L(\boldsymbol{\theta}^*) = \frac{1}{N}\sum_{n=1}^{N}\left[-\sum_{l=1}^{K} y_n^{(l)}(\boldsymbol{\alpha}^l \mathrm{x}_n + \boldsymbol{\beta}^l \mathrm{z}_n) + \log\sum_{l=1}^{K} \exp(\boldsymbol{\alpha}^l \mathbf{x}_n + \boldsymbol{\beta}^l \mathbf{z}_n)\right] \quad (11)$$

where $\mathbf{z}_n = (B_1(\mathbf{x_n}, \hat{\mathbf{w}}_1), \ldots, B_m(\mathbf{x_n}, \hat{\mathbf{w}}_m))$ and $\boldsymbol{\theta}^* = (\boldsymbol{\alpha}, \boldsymbol{\beta}, \hat{\mathbf{W}})$. Now the Hessian matrix of the negative log-likelihood in the new variables $x_1, \ldots, x_p$, $z_1, \ldots, z_m$ is semi-definitely positive and the coefficient vector $\boldsymbol{\theta}^*$ is calculated with Newton's method, also known, as iteratively reweighted least squares (IRLS) (Minka, 2003).

(4) Finally, we use a backward-step procedure, starting with the full model with all the covariates and sequentially pruning the covariates of the model obtained in the second step, until further pruning does not improve the fit. At each step, we delete the least significant covariate to predict the response variable, that is, the one which shows the greatest critical value (p-value) in the hypothesis test, where the associated coefficient equal to zero is the hypothesis to be contrasted. The procedure finishes when all tests provide p-values smaller than the fixed significance level, and the model selected in the previous step fits well.

2.3. General structure of the EA

A population-based evolutionary algorithm is used for the architectural design and the estimation of real-coefficients. The search begins with an initial population, and in each iteration the population is updated using a population-update algorithm. The general structure of the EA is the following:

(1) Generate a random initial population of size N_R.
(2) Repeat the following steps until the stopping criterion is fulfilled:
  (a) Calculate the fitness of every individual in the population and rank the individuals regarding their fitness.
  (b) The best individual is copied into the new population (elitism).
  (c) The best 10% of individuals of the population are replicated and substitute the worst 10% of individuals.
  (d) Apply parametric mutation to the best 10% of individuals.
  (f) Apply structural mutation to the remaining 90% of individuals.

Keeping in mind that h is a product-unit neural network and can be seen as a multivaluated function:

$$h(\mathbf{x}, \boldsymbol{\beta}, \mathbf{W}) = (h_1(\mathbf{x}, \boldsymbol{\beta}^{\mathbf{1}}, \mathbf{W}), \ldots, h_{K-1}(\mathbf{x}, \boldsymbol{\beta}^{K-1}, \mathbf{W})) \quad (12)$$

we consider $L^{*}(\boldsymbol{\beta},\mathbf{W})$ as the error function of an individual $h(\boldsymbol{\beta},\mathbf{W})$ of the population. To measure the fitness of each individual, a decreasingly strict transformation of the error function $L^{*}(\boldsymbol{\beta},\mathbf{W})$ is carried out and given by:

$$A(h) = \frac{1}{1 + L^{*}(\boldsymbol{\beta},\mathbf{W})} \quad \text{where } 0 < A(h) \leqslant 1 \tag{13}$$

The mutations carried out by the algorithm can be parametric or structural. Parametric mutations affect network weights while the structural ones influence the network topology (hidden nodes and connections). Parametric mutations consist of adding a normally distributed random variable with mean zero and standard deviation σ_1 to each exponent w_{ji} of the model, while a normally distributed one with mean zero and standard deviation σ_2 is added to the rest of the coefficients β_j^l. The standard deviation is updated throughout the evolution according to the 1/5 success rule method of Rechenberg (1975) which establishes the ratio of successful mutation as 1/5. Therefore, if the ratio of successful mutation is greater than 1/5, the mutation should increase; otherwise the deviation should decrease. The adaptation tries to avoid being trapped in local minima and to speed up the evolutionary process when searching conditions are suitable. The modification of the exponents is different from the modification of the rest of the coefficients, being $\sigma_1 < \sigma_2$, because the changes in the exponents greatly affect individual fitness. The structural mutation applied by the algorithm modifies the number of hidden nodes as well as the connections between the nodes within the input and hidden layers and those within the hidden and output layers, which affect the net topology. We have applied five types of mutations: added nodes, deleted nodes, added connections, deleted connections and found nodes.

The parameters used in the evolutionary algorithm are the following: the exponents w_{ji} are initialized in the interval $(-5, 5)$, the coefficients β_{kj} are initialized in $(-10, 10)$. The maximum number of nodes in the hidden layer is $m = 6$ which is large enough if we take the number of input variables into account. We begin with networks with two nodes. The number of nodes that can be added or removed in a structural mutation is one or two. The number of connections that can be added or removed in a structural mutation is a number from one to six. The size of the population is $N_R = 2000$. The stop criterion is reached whenever one of the following two conditions is fulfilled: (i) for five generations there is no improvement either in the average performance of the best 20% of the population or in the fitness of the best individual. (ii) The algorithm achieves 100 generations.

The input variables are normalized in the range (0.1, 0.9) to avoid the sign saturation problems found in product unit transfer functions. More details about the parametric and structural mutation of the evolutionary algorithm can be seen in Martínez, Hervás, et al. (2006, 2006).

2.4. MLP coupled with EA classifier (MLPEA)

To compare our results with those obtained using any standard neural network model, we use a multilayer perceptron model (MLP) that uses a back-propagation learning algorithm (Hayken, 1994). We have also applied a radial basis function network (RBF), a very popular and successful method in classification problems, that implements a normalized Gaussian radial basis function network and uses the K-means clustering algorithm to provide the basis functions (Orr et al., 1996; Oyang et al., 2005).

Moreover another MLP model will be applied (which will be called MLPEA), where an evolutionary algorithm is used to design its architecture, coupled with a pruning one to eliminate non-significant model coefficients (Honaver & Balakrishnan, 1998). The EA applied is the same as that applied to design the structure and train the weights of the product-unit neural network described in 2.3. The maximum and minimum number of nodes in the hidden layer is also the same.

The input variables are normalized in the range (0.1,0.9) to avoid the sign saturation problems found in the sigmoid transfer functions that are used in neuronal net models (Hayken, 1994). Our experimental studies showed that three nodes in the hidden layer gave the best results because a higher number caused overtraining.

2.5. Decision tree classifiers

We use an extended form of ID3, C4.5 (Quinlan, 1993, 1986) for building the decision tree used in our analysis. C4.5 accounts for unavailable values, continuous attribute value ranges, the pruning of decision trees and rule derivation, and it also uses the gain-ratio criterion to select attributes when partitioning the data. Unlike the entropy-based criterion used in ID3, the gain-ratio criterion does not exhibit a strong bias in favour of attributes having many outcomes for a test. Quinlan's goodness-of-split measure as compared to alternative probabilistic measures is found in Breiman (1996).

2.6. Statistics classifiers

As already mentioned, LDA needs to fulfil a series of hypotheses to be correctly applied, (for example, assuming that the covariance matrices are equal) which does not always occur. Due to this, Quadratic Discriminant Analysis (QDA) is used. In this method the decision boundary between each pair of classes is described by a quadratic equation in such a way that, although the procedure for estimation by QDA is similar to those for LDA, separate covariance matrices must be estimated for each class. But when the number of independent variables is high, this can mean a dramatic increase in parameters. Classification based on the KNN algorithm differs from the other methods considered here, as this algorithm uses the data directly for classification, without building a model first (Dasarathy, 1991). As such, no details of model construction need to be considered and the only adjustable parameter in the model is K, the number of nearest neighbours to include in the estimate of class membership: the value of $P(y/x)$ is calculated simply as the ratio of members of class "y" among the K nearest neighbours of x. By varying K, the model can be made more or less flexible (small or large values of K, respectively).

We also applied standard multinomial logistic regression (LR), with the original variables, to compare its performance to our approach (LRPU).

LDA and QDA were applied with the platform called KEEL (Knowledge Extraction based on Evolutionary Learning), fruit of a research project aimed at developing a Computational Environment for integrating the design and use of knowledge extraction models from data using evolutionary algorithms. It is available on the project's web site (http://www.keel.es). C4.5, KNN, MLP, RBF and LR models are part of the classifier algorithms employed in the WEKA machine learning environment (Witten & Frank, 2000). LRPU and MLPEA were applied with specific made-to-order programs.

2.7. Parameters of the evaluation of the findings

To measure the yield of the models, we use the Correct Classification Rate (CCR) which represents the percentage of sheep correctly classified out of the total number of observations in training and generalization sets, respectively. For a perfect classifier, CCR will be equal to 1. We have also determined the CCR by class, that is, the percentage of sheep correctly classified as pertaining to a given class to the total number of sheep that belong to that class. In this way we can analyse the individual difficulties posed by the different methodologies used to recognise each class in particular. We also analyse classification errors made with the different techniques applied to each lactation and by classes.

3. Experiments

The application of the proposed methodology is based on the genealogical and productive records pertaining to a 20 year period (1980–2000) on a sheep farm of Manchegan breed sheep that is located in the Spanish province of Ciudad Real. This breed is one of the most important in Spanish flocks although the production of milk obtained in the main area where this breed is found reaches only about 70 litres per animal and lactation. This figure is very far from the real potential production of this race, since the total mean production per animal on those livestock farms that carry out official milking controls is 166 litres per lactation period or 135 l in 120 days (Molina, 1987; Montoro & Pérez-Guzmán, 1996). So we try to use the new methodology as an instrument to aid in the classification of sheep according to their productive capacity using solely the first milk controls, shortening the current program for the genetic selection of this breed (Torres et al., 2005).

Although there were changes in the size of the herd during this time, there are currently still 3000 mothers in the production phase. Nonetheless, the lack of a systematic and rigorous method for the gathering of information (traditionally carried out by the shepherds themselves) made the filtering of the initial data a long and arduous task. Due to this, only those registers of original data with the most complete information were selected, thereby limiting the useful data at our disposal considerably. The lack of efficiency in the cataloguing of sheep flock data is not of particular importance in this study, however, although it is a frequent problem found by researchers in the sector and considerably curtails progress in the current selection process of the Manchegan breed (Molina, 1987).

3.1. Data

Production data collection was carried out in the following way: once the sheep had given birth, the first control was registered on the day of the week specified for this task, before the eighth day after birth, to give the controls a week-long time span. The suckling period of the lambs lasted between 35 and 50 days, according to the growth reached and taking into account whether there had been a single or double birth. During this period, the lambs were separated from their mothers for 12 h before the control (only one daily control was taken during this phase, thus doubling the production obtained in the first milking). Once the lambs had been weaned, the sheep were milked twice daily until the end of the lactation period. The production was measured by volume, taken in two daily milking controls after weaning and always without revision. In the flock analysed, all the sheep were milked from the moment of birth.

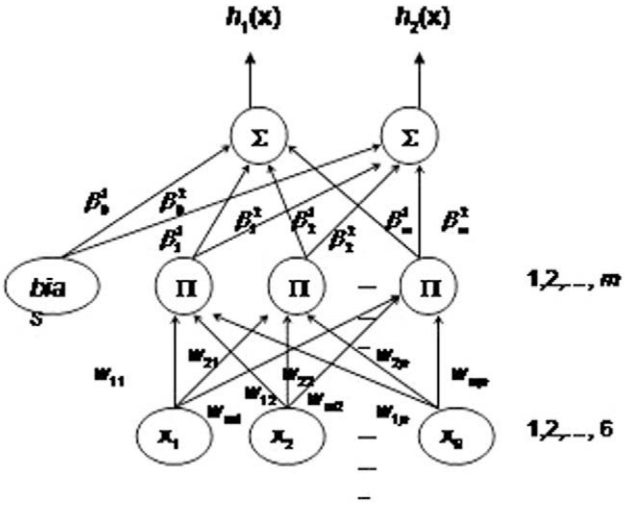

Fig. 1. Graphic representation of the protect unit based neural network applied (for nine classes E-1 = 2 output modes, and p = 6 input variables).

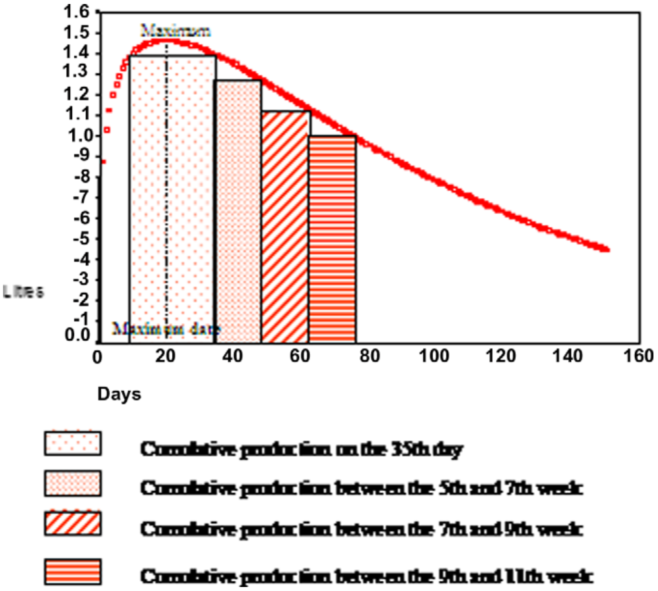

Fig. 2. Graphic representation of the variables used to recognize the sheep's production category.

Table 1
Descriptive statistics on lactation duration (in days).

Lactation	Mean	Minimum	Maximum	Standard deviation	Variation coefficient (%)	Median	Sample size
First	138.28	90	189	27.97	20.22	137.0	178
Second	144.48	91	240	33.20	22.98	142.0	133
Third	145.29	90	228	31.98	22.01	147.5	112

Table 2
Descriptive statistics on milk production during different lactations.

Lactation	Mean milk production in 150 days (l)	Standard deviation (l)	Variation coefficient (%)	25th Percentile (l)	75th Percentile (l)
First	128.26	38.69	30.16	100	148
Second	143.32	45.10	31.47	108	173
Third	153.53	56.78	36.98	110	182

Table 3
Linear correlation coefficients between the variable total production and the independent variables.

Lactation	Linear correlation maximum and total production	Linear correlation production on the 35th day and total production	Linear correlation maximum date total production
First	0.78	0.69	−0.05
Second	0.84	0.72	−0.09
Third	0.57	0.83	−0.13

3.2. Variables

The variable used to establish the productive category of the sheep flock was the production, in litres, obtained during 150 lactation days, because this is the variable used for the genetic selection program for the Manchegan breed. We considered this span of time because the mean duration of the lactations analysed was nearer to this figure (as we can see in Table 1) than the 120 days demanded by the official milking control for normalizing milk production.

We used milk production data from 178, 133 and 112 sheep in first, second and third lactation respectively, whose controls were recorded weekly, with lactation periods over 90 days and whose first controls were recorded before 35 days after birth. The number of cases in the three different lactations is not the same because of the information loss in some controls.

Therefore, the sheep productive category was established by their milk production in 150 lactation days, this being the dependent variable of the classification models applied. As mentioned in the introduction, we established three categories: the “good” one, made up of the sheep whose productions were over the 75th percentile (codified as 1,0,0 by network recognition and as 1 for the rest of the methodologies applied); the “bad”, those sheep with productions under the 25th percentile (codified as 0,0,1 by the network recognition and as 3 for the rest of the techniques used) and “normal”, which included the remaining 50% of the cases (codified as 0,1,0 or as 2). The use of percentiles to establish the productive categories in each lactation, instead of fixed production quantities, was due to the need for considering the production increase caused by the age of the sheep, which is usually measured by the birth or lactation number. Sheep milk production increases with the birth number until the third or fourth birth, according to the breed, decreasing after the fifth birth (see (Gallego & Bernabeu, 1994 & Pérez & Gracia, 1994)).

The limits for establishing the productive categories were fixed according to the livestock break-even point, which was previously calculated with all the cost and revenue stock farming data. That is why we used the information about the range of prices received by sheep stock farmers in the last few years (1990–2000) according to

Table 4
Discriminant functions obtained with LRPU model.

Variables	Function 1							
	Intercept	X_5^*	X_6^*	$\mathbf{B}_1$	$\mathbf{B}_2$			
First lactation								
Coeff.	−29.897	8.559	1.396	71.642	16.052			
Std Error	6.018	9.046	6.407	18.175	66.607			
p-Value	.000	.344	.827	.000	.810			
	Function 2							
Coeff.	−8.196	2.810	5.472	21.154	17.768			
Std Error	2.336	7.177	5.429	11.940	66.588			
p-Value	.000	.695	.314	.076	.790			
$\mathrm{B}_1 = (X_2^*)^{0.122}(X_3^*)^{0.336}(X_4^*)^{0.194}(X_6^*)^{0.350}$								
$B_2 = (X_1^*)^{-3.735}(X_2^*)^{-2.940}(X_4^*)^{10.841}(X_5^*)^{-0.558}(X_6^*)^{2.026}$								
# coefficients = 19								
	Function 1							
	Intercept	X_1^*	X_2^*	X_4^*	X_5^*	X_6^*	$\mathbf{B}_1$	$\mathbf{B}_2$
Second lactation								
Coeff.	−21.621	−14.859	−7.718	11.438	−19.376	26.961	85.551	36.748
Std Error	4.635	11.128	4.938	9.425	16.195	11.537	26.657	15.081
p-Value	.000	.182	.118	.225	.232	.019	.001	.015
	Function 2							
Coeff.	−9.758	−11.345	−2.846	5.996	−7.896	18.044	49.468	9.494
Std Error	2.937	8.298	3.500	7.813	11.838	9.145	22.050	11.129
p-Value	.001	.172	.416	.443	.505	.048	.025	.394
$\mathrm{B}_1 = (X_1^*)^{0.203}(X_3^*)^{0.552}(X_5^*)^{0.609}$								
$B_2 = (X_2^*)^{3.748}(X_3^*)^{-1.290}(X_4^*)^{1.593}(X_5^*)^{0.982}$								
# coefficients = 23								
	Function 1							
	Intercept	X_1^*	X_2^*	X_3^*	X_4^*	X_5^*	X_6^*	$\mathbf{B}_1$
Third lactation								
Coeff.	−253.616	−44.502	8.579	−263.454	-33.809	37.357	−168.638	728.146
Std Error	170.219	45.257	18.146	278.478	71.720	61.809	147.172	538.416
p-Value	0.136	0.325	0.636	0.344	0.637	0.546	0.252	0.176
	Function 2							
Coeff.	−93.491	−19.127	−8.552	−130.337	−53.123	32.499	−66.998	329.500
Std Error	57.119	33.628	6.165	180.137	39.957	37.531	91.773	255.442
p-Value	0.102	0.569	0.165	0.469	0.184	0.387	0.465	0.197
$\mathrm{B}_1 = (X_3^*)^{0.320}(X_4^*)^{0.084}(X_5^*)^{-0.077}(X_6^*)^{0.212}$								
# coefficients = 20								

the Spanish Agricultural, Fish and Food Ministry statistical yearbook. This range oscillated between 0.75 and 0.80 euros, approximately, per milk litre, and thus the sheep farm break-even point varied between 100 and 110 l per sheep and lactation. So if we consider our flock production quantities (see Table 2), the 25th percentile was lower than the break-even point, therefore we considered as "bad" those sheep whose productions did not reach the 25th percentile. The 75th percentile approximated the potential productivity of the race, which is estimated at 166 l per sheep and lactation (Pérez & Gracia, 1994), and was established as the minimum production in the "good" class. The remaining 50% of the cases were classified as the "normal" class. However, if some cost or revenue factor (like the price of milk, for example) changed, other quantities could be considered to establish the productive categories.

The input variables used to estimate the sheep productive category were: Maximum production quantity, in litres (X_1); maximum production date, in number of days elapsed since birth (X_2); quantity of milk produced, in litres, from birth to the 5th week after birth (X_3); quantity of milk produced, in litres, between the 5th and 7th week after birth (X_4); quantity of milk produced, in litres, between the 7th and 9th week after birth (X_5); quantity of milk produced, in litres, between the 9th and 11th week after birth (X_6).

Figs. 1 and 2 show, respectively, a graphic representation of the product unit based neural network used for $K - 1 = 2$ classes and $p = 6$ input variables; and an example of a sheep milk production curve.

The variable-selection process was carried out according to the following criteria: the maximum production quantity, as well as cumulative production until the 5th lactation week, were selected because of their strong linear correlation with the dependent variable (both linear correlation coefficients were significant to a 99% confidence level). The maximum production date was included because it is, according to numerous authors (Purroy (1982) & Serrano & Montoso (1996)), one of the most influential variables in total milk production, although in our case the linear correlation between this variable and total production was not significant (see Table 3). The remaining independent variables considered (production quantities obtained at different lactation weeks) were selected to reach the goal of estimating the sheep productive category as soon as possible, without waiting until the end of their lactations. We considered cumulative productions until the 11th week after checking that the classification results were worse if we eliminated the input variables referred to during the production between the 7th to 9th, and 9th to 11th weeks. We used cumulative production quantities from several periods instead of daily production because the milk control dates were not the same for all sheep.

Table 5
CCR obtained in each productive category and lactation in the generalization set (MLPEA and LRPU are average results of 30 runs).

Methodology	Lactation			CCR weighted mean
	First	Second	Third	
Good sheep category				
LDA	50.0	**90.9**	30.7	54.6
QDA	63.2	72.7	61.5	65.1
C4.5	68.4	90.9	46.1	67.4
KNN	73.7	63.6	61.5	67.4
MLP	78.9	63.6	53.8	67.4
MLPEA	68.1	86.4	**69.2**	73.1
LR	73.7	72.7	38.5	62.8
LRPU	**88.9**	81.8	46.2	**74.2**
RBF	73.7	72.7	38.5	62.8
Normal sheep category				
LDA	**88.2**	**81.5**	**95.4**	**87.9**
QDA	82.3	66.7	86.4	78.3
C4.5	64.7	74.0	81.2	72.1
KNN	52.6	62.9	70.0	60.6
MLP	76.5	74.1	86.4	78.3
MLPEA	85.7	76.8	81.8	81.8
LR	73.5	70.4	86.4	75.9
LRPU	73.5	**81.5**	75.0	76.5
RBF	67.6	63.0	86.4	71.1
Bad sheep category				
LDA	65.0	41.2	70.0	57.3
QDA	52.6	58.8	90.0	63.0
C4.5	**68.4**	41.2	60.0	56.5
KNN	64.7	47.1	77.3	60.9
MLP	57.9	**64.7**	**90.0**	67.4
MLPEA	61.9	54.4	70.0	60.9
LR	63.2	58.8	80.0	65.2
LRPU	65.0	**64.7**	**90.0**	**70.3**
RBF	52.6	47.1	80.0	56.5

Table 6
CCR obtained in the total generalization set for each lactation (MLPEA and LRPU are average results of 30 runs).

Methodology	Lactation			CCR weighted mean
	First	Second	Third	
Total generalization set				
LDA	69.4	69.1	71.1	70.9
QDA	69.4	65.4	**80.0**	70.9
C4.5	66.6	67.3	66.7	66.9
KNN	63.9	58.2	71.1	64.0
MLP	72.2	69.1	77.8	72.7
MLPEA	74.6	71.8	75.6	74.0
LR	70.8	67.3	71.1	69.8
LRPU	**75.0**	**76.4**	73.3	**75.0**
RBF	65.3	60.0	71.1	65.1

The cases analysed in each lactation were randomly separated into two sets: 60% for the training set, and the remaining 40% for the generalization set (we decided not to use a validation set because of the reduced number of cases available for the second and third lactations). Thus we worked with 106 cases for training and 72 for generalization in the first lactation; 70 and 55, respectively, in the second lactation, and finally, 67 and 45 in the third lactation.

What must be emphasized here is the enormous difficulty posed by the extraction of data in these circumstances, since it is necessary for the shepherd himself to record the data out in the country.

4. Results

We go on to comment on the results obtained with the different methodologies applied (in terms of CCR) as well as the repercussion of the misclassification of livestock.

4.1. Comparison of CCR obtained with the methodologies applied

Table 4 shows the discriminant functions obtained with our approach, logistic regression with product units (LRPU). In these functions the X^* variables are the original inputs normalized in the range (0.1, 0.9). In Table 5 we can see the CCR obtained in each productive category in the generalization set with LRPU, MLPEA and other methodologies. We have included the average results obtained with MLPEA and LRPU in 30 runs. (The right-hand column shows the weighted arithmetic mean of the CCR obtained in the three lactations considered with each method).

Table 7
Consequences of classification errors.

Error (observed-estimated class)	Repercussion in the genetic progress goal	Repercussion in diet cost
Good–normal	A good sheep is not selected to contribute to genetic progress	The diet quality is worse than this sheep should receive and can provoke lower productivity
Good–bad	A good sheep is not selected to contribute to genetic progress and moreover it might be replaced unnecessarily	The diet quality is worse than this sheep should receive and can provoke lower productivity
Normal–good	A normal sheep is incorrectly selected to contribute to genetic progress	Diet with a higher quality and cost than the sheep would receive otherwise
Normal–bad	A normal sheep can be incorrectly replaced	The diet quality is worse than this sheep should receive and can provoke lower productivity
Bad–good	Totally erroneous selection of a bad sheep to contribute to genetic progress	Diet with a higher quality and cost than the sheep needs
Bad–normal	A bad sheep can be incorrectly maintained	Diet with a higher quality and cost than the sheep needs

Table 8
Mean uncorrected classification rate (in percentage) of the six models in the three lactations analysed (generalization set).

Model	Good as normal	Normal as good	Normal as bad	Bad as normal	Good as bad	Bad as good
LDA	40.5	9.7	14.5	**28.3**	**0**	**0**
QDA	32.6	8.4	13.2	32.6	2.3	**0**
C4.5	30.2	10.8	16.9	41.3	2.3	2.2
KNN	30.2	16.9	32.5	45.6	**0**	**0**
MLP	32.7	8.9	12.1	29.1	1.8	**0**
MLPEA	**21.7**	8.9	**9.6**	39.3	**0**	**0**
LR	36.6	11.3	11.9	32.7	1.8	**0**
LRPU	25.6	**7.2**	14.5	30.4	**0**	**0**
RBF	36.6	14.8	12.9	41.8	1.8	**0**

If we look at the results shown in Table 5 we can verify that LRPU followed by MLPEA are the best approaches in "good" class recognition.

In general, the "normal" category is the one which all the methods applied (especially LDA) find to be the best (the KNN application is the worst).

The results show us that the "bad" class is the least recognized one by the group of techniques used, although the LRPU model reached the highest CCR (70.3%) in the generalization set.

Table 6 shows the value of the CCR obtained in the total generalization set for each lactation (without distinguishing between classes). In this case the LRPU model is the one that offers the best results, followed by the MLPEA application. Therefore, if we compare the mean results obtained with the LRPU model to those of the other methods applied, we can see that our approach gets the best results in the total generalization set, considering the three productive categories simultaneously. Moreover LRPU (after the C4.5) was the technique that showed the least variability in the CCR obtained in the three lactations analysed, therefore its CCR mean is one of the most representative.

The LRPU also gets the best results in the recognition of both the extreme "good" and "bad" classes. However, the highest CCR obtained for the "normal" class is with the LDA application.

4.2. Misclassification costs

We have to take misclassification costs into account in order to obtain a model with the lowest possible cost (Johnson & Wichern, 2002). In our case, if we consider the different types of mistakes that we can commit, we will understand that the most costly ones are those that confuse the extreme classes. This is because the "good" sheep receive a more nutritious and therefore more expensive diet than their flock companions and, furthermore, will be selected for reproduction, thus contributing to the genetic progress of the race. "Bad" sheep receive a cheaper and worse diet and could be those selected for replacement in the flock. Because of this, the confusion between the extreme classes could provoke considerable cost both in quantity and quality for the farm stock. The consequences of each possible type of misclassification are summarized in Table 7. Table 8 studies the misclassification percentages committed in each class in the three lactations. We can see that several methods such as: C4.5, QDA, MLP, and RBF confuse the extreme classes (although in a reduced number of cases). However, the MLPEA, LDA, KNN and LRPU models do not confuse these categories. The rest of the possible mistakes in the classification task (the confusion between "good" and "normal" or "bad" and "normal") could result in the same cost for the farmer. So, is it a bad idea to relegate more productive capacity to a sheep than it really has, thus investing more than necessary in its feed and care, or even selecting it to improve the breed? Or, on the contrary, is the alternative worse, considering a sheep to have a lower productive capacity than it really does, thus dedicating fewer resources to its maintenance, or even selling it at a lower price than its real value and replacing it unnecessarily? If we observe Table 8, we can check how the LRPU and MLPEA methodologies commit the lowest misclassification rates in the majority of cases. The most frequent classification mistakes in general are considering either good or bad sheep as normal with all the methods applied.

5. Conclusions

In spite of the fact that new methodologies, like artificial neural networks and evolutionary algorithms, are becoming more and more frequent in the business world, the agricultural or stock breeding sectors are not among the most common application areas (like finance, accounting, engineering, productive process, marketing). With this research we have attempted to demonstrate that these methodologies could improve the techniques and economic management of livestock enterprises, exemplified by the sheep farm here studied.

The LRPU and, afterwards, MLPEA applications have offered the best results for the recognition of sheep productive categories, and therefore constitute the best approaches. If the goal is to maximize the CCR for the total generalization set, without distinction between classes, the LRPU and MLPEA techniques are the best options. If the goal is to increase the precision in extreme class recognition ("good" and "bad"), these techniques achieve the best results again. Although the LDA application is the best in "normal" class recognition, we think that the recognition of extreme classes is the most interesting one for the livestock farmer because early identification of the most productive sheep (found solely through the first milk controls) will permit him to select the best for reproduction, thus contributing to the genetic progress of the flock, and shortening the current evaluation process used by the selection schema of the Manchegan breed. Moreover the stock farmer could use the information about the productive capacity of his sheep to design the flock feeding strategy (since the most productive sheep would receive a more nutritious and expensive diet).

Moreover the identification of the least productive sheep could permit their exclusion from the selection program and even their replacement (because their maintenance implies a high opportunity cost for the enterprise as an unachieved profit). So early sheep classification could lead to a decrease in the great differences in production that have been found in the last few years between different Spanish sheep breeds, like the Manchegan, with respect to other breeds (the French Lacaune, for example).

This research has attempted to demonstrate how the proposed LRPU model as well as the artificial neural network model are able to improve other standard multivariate statistical techniques (such as standard LR, LDA, QDA, KNN and C4.5) in classification problems, even when the data quality is reduced. Therefore we can affirm that these techniques constitute a new and useful tool for decision making in the technical and economic management of livestock enterprises.

Acknowledgements

The authors gratefully acknowledge the financial support provided by the Spanish Inter-Ministerial Commission of Science and Technology (MICYT) (TIN 2008-06681-C), FEDER funds and the P08-TIC-3745 project of the "Junta de Andalucía" (Spain).

References

Bernadó, E., & Garrell, J. M. (2003). Accuracy-based learning classifier systems: Models, analysis and applications to classification tasks. *Evolutionary Computation, 11*(3), 209–238.

Bebis, M., & Georgipoulos, M. (1997). Coupling weight elimination with genetic algorithms to reduce network size and preserve generalization. *Neurocomputing, 17*, 167–194.

Bishop, M. (1995). *Neural networks for pattern recognition*. New York: Oxford University Press.

Breiman, L. (1996). Technical note: Some properties of splitting criteria. *Machine Learning, 24*(1), 41–47.

Buxadé, C. (1998). *Ovino de leche: aspectos clave.*. Madrid: Mundi Prensa.

Cox, D. R. (1970a). *Analysis of binary data*. London: Methuen & G.

Cox, D. R., & Snell, E. J. (1989b). *Analysis of binary data* (2nd ed.). London: Chapman & Hall.

Dasarathy, B. (1991). *Nearest neighbour pattern classification techniques*. Silver Spring, MD: IEEE Computer Society Press.

Doumpos, M., Zopounidis, C., & Pardalos, P. (2000). Multicriteria sorting methodology: Application to financial decision problems. *Parallel Algorithms and Applications, 15*(1–2), 113–129.

Dreiseitl, S., & Ohno-Machado, L. (2002). Logistic regression and artificial neural network classification models: A methodology review. *Journal of Biomedical Informatics, 35*, 352–359.

Duda, R., & Hart, P. (2001). *Pattern classification*. John Wiley.

Durbin, R., & Rumelhart, D. (1989). Product units: A computationally powerful and biologically plausible extension to backpropagation networks. *Neural Computation, 1*, 133–142.

Engelbrecht, A. P., & Ismail, A. (2000). Global optimization algorithms for training product units neural networks. In *Presented at international joint conference on neural networks IJCNN'2000*, Italy.

Fredd, N., & Glover, F. (1981). Simple but powerful goal programming models for discrimination problems. *European Journal of Operational Research, 7*, 44–60.

Friedman, J., & Stuetzle, W. (1981). Projection pursuit regression. *American Statistical Association, 76*, 817–823.

Friedman, J. (1991). Multivariate adaptive regression splines. *Annals of Statistics, 19*(1), 1–67.

Gallego, L., & Bernabeu, R. (1994). *Producción de leche: factores de variación en ganado ovino de raza Manchega* (pp. 162–173). Madrid: Mundi Prensa.

Goldberg, D. E. (1989). *Genetic algorithms in search optimization and machine learning*. Addison-Wesley.

Goldberg, D. E. (1989a). Genetic algorithms and Walsh functions. Part 2: Deception and its analysis. *Complex Systems, 3*, 153–171.

Goldberg, D. E. (1989b). Genetic algorithms and Walsh functions. Part 1: A gentle introduction. *Complex Systems, 3*, 129–152.

Hajmeer, M., & Basheer, I. (2003). Comparison of logistic regression and neural network-based classifiers for bacterial growth. *Food Microbiology, 20*, 43–55.

Hastie, T. J., & Tibshirani, R. J. (1990). *Generalized additive models*. London: Chapman & Hall.

Hayken, S. (1994). *A comprehensive foundation. Neural networks*. Nueva York: Macmillan.

Hervás, C., & Martínez, F. (2007). Logistic regression using covariates obtained by product-unit neural network models. *Pattern Recognition, 40*(1), 52–64.

Honaver, V., & Balakrishnan, K. (1998). Evolutionary design of neural architectures. A preliminary taxonomy and guide to literature.

Hosmer, D., & Lemeshow, S. (2000). *Applied logistic regression* (2nd ed.). New York: Wiley.

Ismail, A., & Engelbrecht, A. P. (1999). Training product units in feedforward neural networks using particle swarm optimisation. Durban, South Africa.

Johnson, R. A., & Wichern, D. W. (2002). *Applied multivariate statistical analysis* (5th ed.). Upper Saddle Rover, NJ: Prentice Hall.

Kaefer, F., Heilman, C. M., & Ramenofsky, S. D. (2005). A neural network application to consumer classification to improve the timing of direct marketing activities. *Computers and Operations Research, 32*(10), 2595–2615.

Kooperberg, C., Bose, S., & Stone, C. J. (1997). Polychotomous regression. *Journal of the American Statistical Association, 92*, 117–127.

Kordatoff, Y., & Michalski, R. S. 1990. *Machine learning: An artificial intelligence approach* (Vol. II).

Lin, C., Lee, J.-H., & Lee, C.-J. (2008). A novel hybrid learning algorithm for parametric fuzzy CMAC networks and its classification applications. *Expert Systems with Applications, 35*(4), 1711–1720.

Major, R., & Ragsdale, C. (2001). Aggregating expert predictions in a networked environment. *Computers and Operations Research, 28*, 1231–1244.

Martínez, A. C., Hervás, C., Martínez, F. J., et al. (2006). Hybridizationof evolutionary algorithms and local search by means of a clustering method. *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics, 36*(3), 534–546.

Martínez, A. C., Martínez, F. J., Hervás, C., et al. (2006). Evolutionary product unit based neural networks for regression. *Neural Networks, 19*, 477–486.

McCullagh, P., & Nelder, J. A. (1989). *Generalized linear models* (2nd ed.). Norwell, MA: Chapman and Hall.

Molina, M. P. (1987). Composición y factores de variación de la leche de ovejas de raza Manchega. Tesis doctoral Universidad de Valencia, España.

Montoro, V., & Pérez-Guzmán, M. D. (1996). La selección de la raza ovina Manchega. Junta de Comunidades de Castilla la Mancha, No 9.

Minka, T. (2003). *A comparison of numerical optimizers for logistic regression*. Department of Statistics, Carnegie Mellon University.

Orr, M. J. L. (1996). Introduction to radial basis function networks. Technical report. Center for Cognitive Science, University of Edinburgh, UK.

Oyang, Y.-J., Hwang, S.-C., Ou, Y.-Y., Chen, C.-Y., & Chen, Z.-W. (2005). Data classification with radial basis function networks based on a novel kernel density estimation algorithm. *Transactions on Neural Networks*(January).

Parag, C., & Pendharkar, A. (2005). A threshold-varying artificial neural network approach for classification and its application to bankruptcy prediction problem. *Computers and Operations Research, 32*(10), 2561–2582.

Patuwo, E., Hu, M., & Hung, M. S. (1993). Two group classification using neural networks. *Decision Sciences, 24*, 825–845.

Pérez, S., & Gracia, O. (1994). *Factores ambientales que influyen en la producción lechera en la raza ovina manchega*. España: Universidad de Castilla-La Mancha.

Purroy, A. (1982). Producción de leche de oveja. Monografías I.N.I.A., 36, Madrid.

Quinlan, J. R. (1993). *C4.5: programs for machine learning*. San Mateo, CA: Morgan Kaufmann.

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

Rada, R. (2008). Expert system and evolutionary computing for financial investing: A review. *Expert Systems with Applications, 34*, 2232–2240.

Rechenberg, I. (1975). *Evolutionstrategie: Optimierung technischer Systeme nach Prinzipien der Biologischen Evolution*. Stuttgart: Framman-Holzboog Verlag.

Schwarzer, G., Vach, W., & Schumacher, M. (2000). On the misuses of artificial neural networks for prognostic and diagnostic classification in oncology. *Statistics in Medicine, 19*, 541–561.

Serrano, M., & Montoso, V. (1996). Cálculo de los factores de extensión de la lactación a 120 días en ganado ovino Manchego. *Investigación agraria: Producción y Sanidad Animales, 11*(1), 69–83.

Sexton, R., & McMurtrey, S. (2005). Employee turnover: A neural network solution. *Computers and Operation Research, 32*(10), 2635–2651.

Schmitt, M. (2002). On the complexity of computing and learning with multiplicative neural networks. *Neural Computation, 14*, 241–301.

Subasi, A., Alkan, A., & Koklukaya, E. (2005). Wavelt neural network classification of EEG signals by using AR model with MLE preprocessing. *Neural Networks, 18*, 985–997.

Tian-Shyug, L., Chih-Chou, C., et al. (2006). Mining the customer credit using classification and regression tree and multivariate adaptive regression splines. *Computational Statistics and Data Analysis, 50*, 1113–1130.

Torres, M., Hervás, C., & Amador, F. (2005). Approximating the sheep milk production curve through the use of artificial neural networks and genetic algorithms. *Computers and Operation Research, 32*, 2653–2670.

Widrow, B. (1962). Layered neural nets for pattern recognition. *IEEE, 36*, 1109–1118.

Williams, R. J., & Minai, A. A. (1990). Back-propagation heuristics: A study of the extended delta-bar-delta algorithms. In *IEEE international joint conference on neural networks* (Vol. 1, pp. 595–600).

Witten, I. H., & Frank, E. (2000). Weka machine learning. Release 3.4.0, 2000.

Youngdai, K., & Sunghoon, K. (2006). Multiclass sparse logistic regression for classification of multiple cancer types using gene expression data. *Computational Statistics and Data Analysis, 51*(3), 1643–1655.

Yuan-chin, I., & Sung-Chiang, L. (2004). Synergy of logistic regression and support vector machine in multiple-class classification. IDEAL. LNCS (Vol. 3177, pp. 132–141).