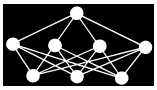

Structural Simplification of Hybrid Neuro-Logistic Regression Models in Multispectral Analysis of Remote Sensed Data

P.A. Gutiérrez[*], *C. Hervás*[*], *J.C. Fernández*[*], *M. Jurado-Expósito*[†], *J.M. Peña-Barragán*[†] *and F. López-Granados*[†]

Abstract: Logistic Regression (LR) has become a widely used and accepted method to analyze binary or multiclass outcome variables, since it is a flexible tool that can predict the probability for the state of a dichotomous variable. A recently proposed LR method is based on the hybridization of a linear model and Evolutionary Product-Unit Neural Network (EPUNN) models for binary classification. This produces a high number of coefficients, so two different methods for simplifying the structure of the final model by reducing the number of initial or PU covariates are presented in this paper, both being based on the Wald test. The first method is a Backtracking Backward Search (BBS) method and the second is similar but based on the standard Simulated Annealing process for the decision steps (SABBS). In this study, we used aerial imagery taken in mid-May to evaluate the potential of two different combinations of LR and EPUNN (LR using PUs (LRPU), as well as LR using Initial covariates and PUs (LRIPU)) and the two presented methods for structural simplification of the final models (BBS and SABBS) for discriminating *Ridolfia segetum* patches (one of the most dominant, competitive and persistent weed in sunflower crops) in one naturally infested field of southern Spain. Then, we compared the performance of these methods to six commonly used classification algorithms, our proposals obtaining a competitive performance and a lower number of coefficients.

Key words: *Logistic regression; structural simplification; product-unit neural networks; evolutionary algorithms*

Received: ??
Revised and accepted: ??

1. Introduction

Classification problems attempt to solve the task of deciding the class membership y of an unknown data item $\mathbf{x}$ based on a data set $D = \{(\mathbf{x}_i, y_i)\}$ $i = 1, ..., n$ of

[*]Dept. of Computer Science and Numerical Analysis, University of Córdoba, 14071, Córdoba, Spain, {i02gupep@uco.es,chervas@uco.es,fernandezcaballero@gmail.com}

[†]Institute for Sustainable Agriculture, CSIC, 14080, Córdoba, Spain, {montse.jurado@ias.csic.es,pa2pebaj@uco.es,flgranados@ias.csic.es}

data items $\mathbf{x}_i$ with known class membership. The $\mathbf{x}_i$ are usually k-dimensional feature vectors, whose components are called covariates or independent variables. In most problem domains, there is no functional relationship between y and $\mathbf{x}$. In this case the relationship has to be described more generally by a probability distribution $P(\mathbf{x}; y)$; one then assumes that the data set D contains independent samples from P. From statistical decision theory, it is well known that the optimal class membership decision is to choose the class label y that maximizes posteriori distribution $P(y/\mathbf{x})$. Therefore there are different approaches to data classification: one which considers only one distinction between the classes previously defined and assigns a class label to an unknown data item, and another which attempts to model $P(y/\mathbf{x})$. This latter attempt yields not only a class label for a data item, but also a probability of class membership. Logistic Regression (LR), artificial neural networks (ANNs), and decision trees are all members of the second class, although they vary considerably in building an approximation to $P(y/\mathbf{x})$ from data. However, in spite of the great number of techniques developed to solve classification problems, there is no optimum methodology or technique to solve specific problems. This point has encouraged the comparison and combination of different types of classification [1, 2].

A recently proposed LR method is based on the hybridization of a linear model and Evolutionary Product-Unit Neural Network (EPUNN) models for binary [3] and multi-class [4] classification problems. The estimation of the model coefficients is carried out in two phases. First, the number of PU basis functions and the exponents' vector are determined by means of an evolutionary neural network algorithm. Secondly, a standard maximum likelihood optimization method determines the rest of the coefficients in the new space given by the initial variables and the PU basis functions previously estimated. This model allows the generation of non-linear classification surfaces and the identification of possible strong interactions that may exist between the covariates that define the classification problem. These models are less complex (number of new covariates or number of exponents in these covariates) than the alternative higher order polynomial models. However, the models result in a high number of coefficients, so two different methods for simplifying the structure of the final model by reducing the number of initial or PU covariates are presented in this paper, both being based on the Wald test. The first method is a Backtracking Backward Search (BBS) method, that starts with the full model with all the covariates, initial and PUs, pruning variables to the model sequentially and successively, until no further pruning can be made to improve the fit. At each step, the least significant covariate is selected in the discriminant function. The selected covariate is deleted if this does not reduce the fit. If it does, the second least significant covariate is considered. The second method is similar but based on the standard Simulated Annealing process for the decision steps (SABBS).

In order to analyze the performance and robustness of the proposed methodology, it is applied in this to a real agronomical problem that involves the discrimination of *Ridolfia segetum* patches in sunflower fields, using multispectral imagery. Sunflower (*Helianthus annuus* L.) is one of the most abundant crops in Andalusia, Southern Spain, with more than 320,000 ha sown annually [5]. Sunflower sowing and harvesting times are February-March and July-August, respectively, mainly

grown under dry land conditions. *R. segetum* Moris (corn caraway) is a very frequent annual, umbelliferous weed that is abundant in clay soils in Andalusia. Its life cycle coincides with that of the sunflower, which enhances its competitive ability and results in an average crop yield reduction of about 32% when infestation is two *R. segetum* plants m^{-2} [6]. This weed is hard to control because it is not controlled by the pre-emergence and pre-plant incorporated herbicides used in sunflower. Consequently, post-emergence strategies such as tillage or hand weeding are commonly used, otherwise weed obstructs the harvester due to the fact that it still has partly green steam during the sunflower harvesting. This is a serious drawback if the harvester is equipped with yield monitor as habitually happens in precision agriculture management. Patchy distribution of broadleaf weeds in sunflower fields is well documented [7]. However, herbicides or other control strategies are not addressed to the infested zones, but are instead applied over the entire fields. The potential for overuse or application and the corresponding eco-environmental problems are evident. To overcome the possibility of minimizing the impact of inappropriate control strategy, the idea of Site-Specific Weed Management (SSWM) has been developed in the context of precision agriculture [8]. A key component of SSWM is that accurate and appropriate weed maps are required to take full advantage of site-specific herbicide applications. Mapping weed patches based on ground survey techniques on field scale is time consuming, expensive and unapproachable in field areas with difficult access. Remote sensing of weed canopies may be more efficient and suitable than field surveys and the majority of studies on discriminating weeds in cultivated systems have involved discrete broadband remote sensing (multispectral sensors) [9]. Approaches based on EPUNNs have been previously applied to remotely sensed images for agronomical objectives [10, 11].

Thus, the goal of this work is to assess the potential of two different combinations of LR and EPUNNs (LR using PUs, LRPU, and LR using Initial covariates and PUs, LRIPU) and the two presented methods for simplifying the structure of the final models (BBS and SABBS) for discriminating *R. segetum* patches in two different naturally infested fields. The results indicate that, with fewer restrictive assumptions, the models proposed are able to reduce the number of coefficients substantially without any significant decrease in classification accuracy.

The rest of the paper is organized as follows. Section 2 is devoted to the description of the standard binary LR model. Section 3 describes PUNNs and their evolution. Sections 4 includes the details of the LRPU and LRIPU models and Section 5 presents the different algorithms for structural simplification proposed in this work. Finally, the experiments and the comparison test carried out are included in Section 6 and Section 7 summarizes the conclusions of our work.

2. Binary Logistic Regression

The binary Logistic Regression (LR) technique considers a binary outcome variable y that is observed together with a vector $\mathbf{x}_i = (1, x_{i1}, x_{i2}, ..., x_{ik})$ of covariates for each of the n_{T} training samples (assuming that the vector of inputs includes the constant term 1 to accommodate the intercept). The two-class is coded via a 1/0 response y_i, associated with the first class. LR [12] is a widely used statistical modeling technique in which the conditional probability p of the dichotomous outcome

event is related to a set of explanatory variables $\mathbf{x}$ in the form:

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right) = \boldsymbol{\beta}^{\mathrm{T}}\mathbf{x} \tag{1}$$

where $\boldsymbol{\beta} = (\beta_0, \beta_1, ..., \beta_k)$ is the vector of coefficients of the model, $\boldsymbol{\beta}^{\mathrm{T}}$ is the transposed vector and the odd of the event is $p/(1-p)$. A simple calculation in (1) shows that the probability of occurrence of an event as a function of the covariates is nonlinear and is given by:

$$p(\mathbf{x}, \boldsymbol{\beta}) = \frac{e^{\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}}}{1 + e^{\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}}} = \frac{e^{f_{\mathrm{LR}}(\mathbf{x},\boldsymbol{\beta})}}{1 + e^{f_{\mathrm{LR}}(\mathbf{x},\boldsymbol{\beta})}}$$

The most commonly used method for obtaining the vector of coefficients $\boldsymbol{\beta}$ is the Iteratively Re-weighted Least Squares (IRLS), which is a nonlinear optimization algorithm that uses a series of weighted least squares subproblems to find LR model maximum-likelihood coefficient estimation. The implementation of IRLS applied in this work is based on that provided in [13], using the conjugate gradient method for solving the associated matricial equation.

3. Evolutionary Product Unit Neural Networks (EPUNNs)

The models we are testing are LR models based on the hybridization of the standard linear model and nonlinear terms constructed with basis functions obtained from EPUNNs. In this way, this section describes the specific details of the PUNN models and the Evolutionary Algorithm used for obtaining the PU coefficients.

3.1 Product Unit Neural Networks

PUNNs are an alternative to Multilayer Perceptrons (MLPs) and are based on multiplicative nodes instead of additive ones [14]. A multiplicative node is given by:

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

where k is the number of inputs and $\mathbf{w}_j = (w_{j1}, w_{j2}, ..., w_{jk})$. PUNNs have several advantages, including increased information capacity and the ability to express strong interactions between input variables. Furthermore, it is possible to obtain upper bounds of the Vapnik-Chervonenkis (VC) dimension of PUNNs similar to those obtained for MLPs [15]. Despite these advantages, PUNNs have a major handicap: they have more local minima and more probability of becoming trapped in them [16]. The activation function of the PUNN considered in this work is given by:

$$f_{\mathrm{PUNN}}(\mathbf{x}, \boldsymbol{\theta}) = \beta_0 + \sum_{j=1}^{m} \beta_j B_j(\mathbf{x}, \mathbf{w}_j)$$

1: **Evolutionary Algorithm**:
2: Generate a random population of size $1,000$
3: **repeat**
4: Calculate the fitness of every individual in the population
5: Rank the individuals with respect to their fitness
6: The best individual is copied into the new population
7: The best 10% of population individuals are replicated and they substitute the worst 10% of individuals
8: Apply parametric mutation to the best 10% of individuals
9: Apply structural mutation to the remaining 90% of individuals
10: **until** the stopping criterion is fulfilled

Fig. 1 *Evolutionary Algorithm (EA) framework*

with $\boldsymbol{\theta} = (\boldsymbol{\beta}, \mathbf{w}_1, ..., \mathbf{w}_m)$. The outputs of the PUNNs are interpreted from the probability point of view, so the softmax transformation is used. The softmax activation function is given by:

$$g(\mathbf{x}, \boldsymbol{\theta}) = \frac{e^{f_{\text{PUNN}}(\mathbf{x}, \boldsymbol{\beta})}}{1 + e^{f_{\text{PUNN}}(\mathbf{x}, \boldsymbol{\beta})}}$$

3.2 Evolutionary Algorithm

In the past decade, Evolutionary Algorithms (EAs) and ANNs have been combined as a key research area, providing an interesting platform for simultaneously optimizing both the weights and architecture of connectionist models [17, 18, 19] while avoiding the shortcomings of traditional BackPropagation [20]. In this way, an Evolutionary Algorithm (EA) has been selected to estimate the coefficients and the structure of the PUNNs that minimize the classification error function. The complexity of the error surface of the proposed model justifies the use of an EA as part of the process of estimation of the model coefficients. Among the different metaheuristics, the EA selected in this work has proved excellent results when evolving PUNNs [21].

The basic framework of the evolutionary algorithm is the following: the search begins with an initial population of neural networks and, in each iteration, the population is updated using a population-update algorithm which evolves both its structure and weights. The population is subject to the operations of replication and mutation. Crossover is not used due to its potential disadvantages in evolving ANNs [22, 23].

The algorithm evolves architectures and connection weights simultaneously, each individual being a fully specified PUNN. The PUNNs are represented using an object-oriented approach and the algorithm deals directly with the PUNN phenotype. Each connection is specified by a binary value indicating if the connection exists and a real value representing its weight. As the crossover is not considered, this object-oriented representation does not assume a fixed order between the different hidden nodes. The general structure of the EA has been included in Fig. 1.

The fitness function $A(\boldsymbol{\theta})$ is a strictly decreasing transformation of the cross-entropy error function $E(\boldsymbol{\theta})$ [24] given by $A(\boldsymbol{\theta}) = 1/(1 + E(\boldsymbol{\theta}))$ where $\boldsymbol{\theta}$ are the parameters of the individual and $E(\boldsymbol{\theta})$ is:

$$E(\boldsymbol{\theta}) = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log g(\mathbf{x}_i, \boldsymbol{\theta}) + (1 - y_i)\log(1 - g(\mathbf{x}_i, \boldsymbol{\theta}))\right]$$

where $g(\mathbf{x}, \boldsymbol{\theta})$ is the softmax activation function.

The severity of both structural and parametric mutations depends on the temperature $T(\boldsymbol{\theta})$ of the PUNN model, defined by:

$$T(\boldsymbol{\theta}) = 1 - A(\boldsymbol{\theta}), \;\; 0 \leq T(\boldsymbol{\theta}) \leq 1$$

where $A(\boldsymbol{\theta})$ is the fitness of the individual with parameters $\boldsymbol{\theta}$. Parametric mutation (Fig. 1, step 8) is accomplished for each weight of the model β_j or w_{ji} adding Gaussian noise:

$$\begin{aligned} w_{ji}(t+1) = &\quad w_{ji}(t) + \xi_1(t) \\ \beta_j(t+1) = &\quad \beta_j(t) + \xi_2(t) \end{aligned}$$

where $\xi_i(t)$ represents a one dimensional normally distributed random variable, $N(0, \alpha_i(t) \cdot T(g))$. The $\alpha_i(t)$ value is updated throughout the evolutionary process, applying the simplest heuristic 1/5 success rule of Rechenberg [25]. This allows an initial coarse-grained search, and a progressively finer-grained search, as a model approaches a solution to the problem. The modifications on the coefficients β_j should be higher than the modifications on the exponents w_{ji}, what is achieved using a higher initial $\alpha_2(t)$ value, i.e. $\alpha_2(0) >> \alpha_1(0)$. The weights are sequentially mutated, hidden node after hidden node, and a standard Simulated Annealing process is applied to accept or reject the modifications in each node.

On the other hand, there are five different structural mutations (Fig. 1, step 9): node deletion, connection deletion, node addition, connection addition and node fusion. These five mutations are applied sequentially to each network. The first four are identical to the mutations in the generalized acquisition of recurrent links (GNARL) model [20]. The node fusion mutation operates randomly selecting two hidden nodes of the neural net and substituting them by a new node that is a combination of both. All the mutations are made sequentially in the given order, with probability $T(\boldsymbol{\theta})$, in the same generation on the same network. If the probability does not select any mutation, one of the mutations is chosen at random and applied to the network.

For further details about parametric and structural mutations and other characteristics of the algorithm see [3, 21, 26].

4. Hybrid Neuro-Logistic models

As previously stated, the Neuro-Logistic Regression models used in this paper include Logistic Regression using Product Units (LRPU) and Logistic Regression using Initial covariates and Product Units (LRIPU).

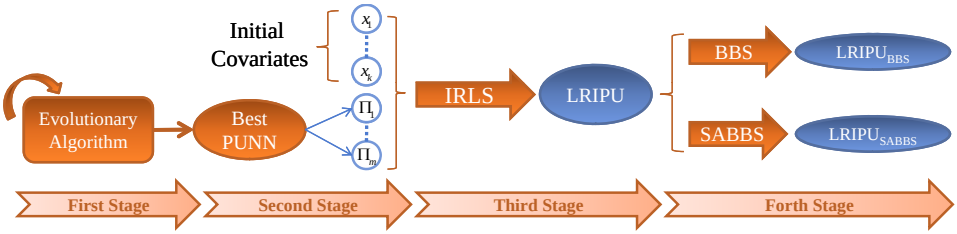

Fig. 2 *Scheme of the LRIPU coefficient optimization process*

4.1 Logistic Regression using Product Units (LRPU)

LRPU is a hybrid method that considers the EA presented in the previous section in order to obtain an EPUNN structure and hidden neuron weights that are accurate enough. When these are obtained, it applies the IRLS mechanism over the PU basis functions of the EPUNN selected. So the LRPU model composed only of PU basis function is given by:

$$f_{\text{LRPU}}(\mathbf{x}, \boldsymbol{\theta}) = \alpha_0 + \sum_{j=1}^{m} \alpha_j B_j(\mathbf{x}, \mathbf{w}_j)$$

where $\boldsymbol{\theta} = (\boldsymbol{\alpha}, \mathbf{W})$, $\boldsymbol{\alpha} = (\alpha_0, \alpha_1, ..., \alpha_m)$ and $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_m)$, with $\mathbf{w}_j = (w_{j1}, w_{j2}, ..., w_{jk})$. The coefficients $\mathbf{W}$ are given by the EA, and are not adjusted by the IRLS method. The IRLS method only optimizes the linear part of the model, i.e., the $\boldsymbol{\alpha}$ coefficients.

4.2 Logistic Regression using Initial covariates and Product Units (LRIPU)

The LRIPU model used is a hybridization of the LR model and the EPUNNs previously presented. The model extends LRPU, considering the initial covariates $\mathbf{x}$ of the problem. Its expression is given by:

$$f_{\text{LRIPU}}(\mathbf{x}, \boldsymbol{\theta}) = \alpha_0 + \sum_{j=1}^{m} \alpha_j B_j(\mathbf{x}, \mathbf{w}) + \sum_{j=1}^{k} \alpha_{(m+j)} x_j$$

where $\boldsymbol{\theta} = (\boldsymbol{\alpha}, \mathbf{W})$, $\boldsymbol{\alpha} = (\alpha_0, \alpha_1, ..., \alpha_m, \alpha_{m+1}, ..., \alpha_{m+k})$ and $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_m)$. The values adjusted with IRLS correspond to the $\boldsymbol{\alpha}$ vector, the coefficients $\mathbf{W}$ again being given by the EA.

An scheme of the different steps necessary for obtaining the model coefficients is given in Fig. 2. In the first stage, the EA is applied and, in the second stage, the PUs from the hidden layer of the best PUNN are extracted and appended to the original covariates input space. Then, the third stage consists of applying the IRLS method in order to obtain the coefficients of the LR model. The fourth stage consists of simplifying the structure of the final model and it will be presented in the next section.

1: **Backtracking Backward Search Algorithm**:
2: Apply IRLS over V, obtaining the $\boldsymbol{\alpha}$ coefficients and the associated CCR_{T}
3: $exit \leftarrow$ false
4: **repeat**
5: **for all** v_i in V **do**
6: Obtain Wald statistic of the variable v_i
7: $p_i \leftarrow p$-value of the Wald test with $H_0 \equiv \alpha_i = 0$
8: **end for**
9: $v_{1\mathrm{st}} \leftarrow$ variable with maximum p_i
10: $V^{'} \leftarrow V - v_{1\mathrm{st}}$
11: Apply IRLS over $V^{'}$, obtaining the $\boldsymbol{\alpha}^{'}$ coefficients and the associated $CCR^{'}_{\mathrm{T}}$
12: **if** $CCR_{\mathrm{T}} > CCR^{'}_{\mathrm{T}}$ **then**
13: $v_{2\mathrm{nd}} \leftarrow$ variable with second maximum p_i
14: $V^{'} \leftarrow V - v_{2\mathrm{nd}}$
15: Apply IRLS over $V^{'}$, obtaining the $\boldsymbol{\alpha}^{'}$ coefficients and the associated $CCR^{'}_{\mathrm{T}}$
16: **if** $CCR_{\mathrm{T}} > CCR^{'}_{\mathrm{T}}$ **then**
17: $exit \leftarrow$ true
18: **else**
19: $V \leftarrow V^{'}$
20: **end if**
21: **else**
22: $V \leftarrow V^{'}$
23: **end if**
24: **until** $exit$=true

Fig. 3 *BBS structural simplification algorithm*

5. Structural Simplification of the LRPU and LRIPU models

In order to reduce the size of LRPU and LRIPU models, we propose a forth stage of structural simplification (see Fig. 2). Two different algorithms are presented: a Backtracking Backward Search (BBS) and a Simulated Annealing Backtracking Backward Search (SABBS). Both methods make use of the Wald statistic, which is a score function commonly considered in LR. The Wald test is a statistical test, used to check whether the effect of a covariate exists or not in the odd of an event. In other words, it tests whether an independent covariate has a statistically significant effect over the dependent variable. As a result, a critical value (p-value) is obtained for each variable, where the associated coefficient equal to zero is the null hypothesis (H_0) to be contrasted. We consider both initial and PU covariates of the LRPU and LRIPU models and apply this test in order to simplify their structure.

5.1 Structural Simplification by a Backtracking Backward Search (BBS)

The first method presented starts with the full model with all the covariates, initial and PUs, pruning variables to the model sequentially and successively, until no further pruning can be made to improve the fit. It uses the Wald statistic for sorting the covariates (PU transformations or initial variables) and tries the elimination of a covariate in each step by a backtracking procedure. First the least significant covariate is selected in the discriminant function. The selected covariate is deleted if this does not reduce the fit. If it does, the second least significant covariate is considered. In this way, the algorithm is a Backtracking Backward method.

The procedure ends when none of the two chosen covariates is deleted. The pseudo-code associated with this algorithm is presented in Fig. 3, where V is the current set of covariates (initial or PUs) and CCR_{T} is the Correct Classification Rate or accuracy in the training set, which is defined by:

$$CCR = \frac{1}{N}\sum_{i=1}^{n} I(C(\mathbf{x}_n) = y_i), \tag{2}$$

where $I(\bullet)$ is the zero-one loss function, $C(\mathbf{x}_n)$ is the class predicted by the model, y_i is the expected class value and n is the number of observations.

A scheme of the BBS algorithm is given in Fig. 4.

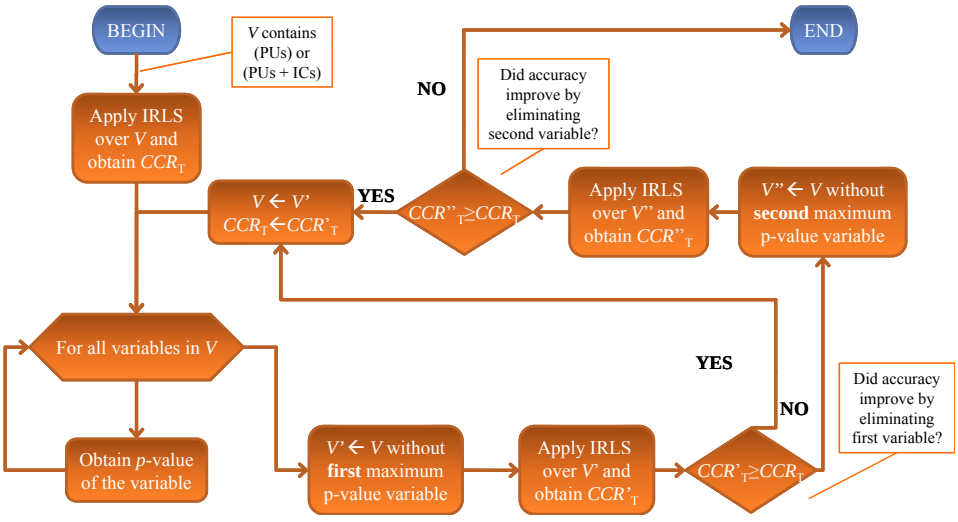

Fig. 4 *Scheme of the Backtracking Backward Search (BBS) algorithm*

5.2 Structural Simplification by a Simulated Annealing Backtracking Backward Search (SABBS)

The second method is based on the standard SA heuristic [27]. The algorithm is very similar to that presented in the previous subsection but, when the elimina-

1: **Simulated Annealing Backtracking Backward Search**:
2: Apply IRLS over V, obtaining the $\boldsymbol{\alpha}$ coefficients and the associated CCR_{T}
3: $exit \leftarrow$ false; $T \leftarrow 0.01 \cdot$ Number variables
4: **repeat**
5: **for all** v_i in V **do**
6: Obtain Wald statistic of the variable v_i
7: $p_i \leftarrow p$-value of the Wald test with $H_0 \equiv \alpha_i = 0$
8: **end for**
9: $v_{1\mathrm{st}} \leftarrow$ variable with maximum p_i
10: $V^{'} \leftarrow V - v_{1\mathrm{st}}$
11: Apply IRLS over $V^{'}$, obtaining the $\boldsymbol{\alpha}^{'}$ coefficients and the associated $CCR^{'}_{\mathrm{T}}$
12: $dif \leftarrow (CCR^{'}_{\mathrm{T}} - CCR_{\mathrm{T}})$
13: **if** $dif < 0$ **and** $U(0,1) > e^{dif/T}$ **then**
14: $v_{2\mathrm{nd}} \leftarrow$ variable with second maximum p_i
15: $V^{'} \leftarrow V - v_{2\mathrm{nd}}$
16: Apply IRLS over $V^{'}$, obtaining the $\boldsymbol{\alpha}^{'}$ coefficients and the associated $CCR^{'}_{\mathrm{T}}$
17: $dif \leftarrow (CCR^{'}_{\mathrm{T}} - CCR_{\mathrm{T}})$
18: **if** $dif < 0$ **and** $U(0,1) > e^{dif/T}$ **then**
19: $exit \leftarrow$ true
20: **else**
21: $V \leftarrow V^{'}$
22: **end if**
23: **else**
24: $V \leftarrow V^{'}$
25: **end if**
26: $T \leftarrow 0.2T$
27: **until** $exit$=true

Fig. 5 *SABBS structural simplification algorithm*

tion of a variable results in a lower training CCR (CCR_{T}), it is accepted with a probability $e^{dif/T}$, where dif is the CCR_{T} difference between the model obtained using the variable and not using it, $dif = (CCR^{'}_{\mathrm{T}} - CCR_{\mathrm{T}})$, and T is the current temperature. The initial value for the temperature is $T = 0.01N$, where N is the number of initial covariates and PUs of the model. In each iteration, the temperature is updated with a $r = 0.2$ freezing factor. The pseudo-code associated with this algorithm is presented in Fig. 5, where $U(0,1)$ is a random uniform variable in the interval $[0,1]$. Finally, a scheme of the SABBS algorithm is given in Fig. 6. Those steps different from the BBS algorithm are marked in dark grey.

6. Experiments

We have tested the described methodology in a real agronomical problem of precision farming, consisting of mapping weed patches in crop fields, through remote sensed data.

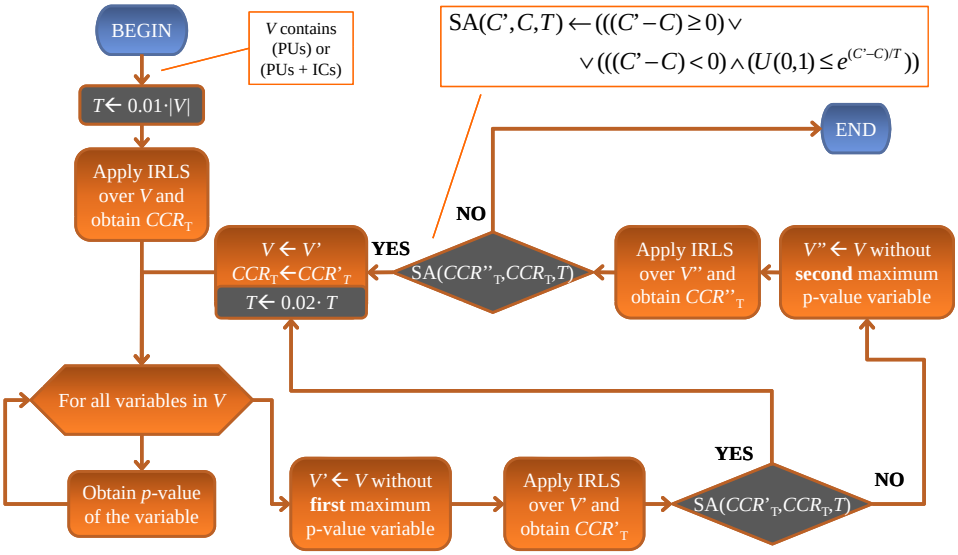

Fig. 6 *Simulated Annealing Backtracking Backward Search (SABBS) algorithm*

6.1 Study sites, materials and experimental design

The study was conducted at two fields in Andalusia, southern Spain: at Matabueyes, 42 ha (coordinates 37º8'N, 4º8'W, WGS84), and at Santa Cruz, 28 ha (coordinates 37º8'N, 4º6'W, WGS84), in 2003 and 2004, respectively. Both fields were naturally infested by *R. segetum*. Conventional-Colour (CC, 400-700 nm) and Colour-near InfraRed (CIR, 500-900 nm) aerial photographs of the field were taken in mid-May. The four input variables included the digital values of all bands in each available image, that is: CC images responded to Blue (B, 400-500 nm), Green (G, 500-600 nm), and Red (R, 600-700 nm) broad bands of the electromagnetic spectrum, and CIR images to G, R and Near-InfraRed (NIR, 700-900 nm) bands. Further information about acquisition of aerial photographs, digitization and orto-rectification is given in [10] and [28].

To train and validate the classification models, a random ground sampling procedure was carried out, ensuring that all parts of the field area had an equal chance of being sampled with no operator bias [29]. For each image, we obtained 2,400 pixels as ground-truth pixels and georeferenced a total of 1,600 pixels in each phenological stage, where 800 pixels corresponded to *R. segetum* class, 400 pixels corresponded to the bare soil class and 400 corresponded to that of sunflower. The objective is the differentiation between *R. segetum* and weed-free (bare soil and sunflower) pixels. The experimental design was conducted using a stratified 10-fold cross-validation procedure with ten runs for each fold. This paper extends the results presented in [10] (where the combination of LR and EPUNNs was applied to the same problem) by studying the influence of the structural simplification methods herein proposed (BBS and SABBS) in the final results. Furthermore, the experimental design followed in [10] consisted of a holdout cross-validation and only one execution, the results not being comparable to those presented in this paper.

The models compared in the different experiments are the following: firstly, the application of the IRLS algorithm only over the PU basis functions extracted from EPUNN model of the EA (LRPU) and over the same basic functions together with initial covariates (LRIPU). Secondly, the two different structural simplification algorithms (BBS and SABBS) are applied over both LRPU and LRIPU models (LRPU_{BBS}, $\text{LRPU}_{\text{SABBS}}$, $\text{LRIPU}_{\text{BBS}}$ and $\text{LRIPU}_{\text{SABBS}}$).

Afterwards, all these models are compared to six machine learning classifiers: LR with attribute selection (SimpleLogistic), LR with a full logistic model (MultiLogistic), Logistic Model Trees (LMT), the C4.5 classification tree inducer, the naive Bayes tree learning algorithm (NBTree) and the AdaBoost.M1 algorithm with 100 maximum number of iterations (AdaBoost100) and using C4.5 as the base classifier. The description of these algorithms can be found in [30].

The EA was implemented using the Evolutionary Computation framework JCLEC [31] (`http://jclec.sourceforge.net`) and it is available in the non commercial JAVA tool named KEEL [32] (`http://www.keel.es`). The parameters used in the EA are common for both datasets. The PUNN models have the following structure: one input layer with four input nodes (corresponding to R, G, B and NIR digital values), one hidden layer with at least one hidden node and a maximum of six nodes (the number of hidden nodes is modified by the structural mutation) and one output layer with one output node, corresponding to the probability of *R. segetum* presence. The number of nodes that can be added or removed in a structural mutation is within the $[1, 2]$ interval. The number of connections that can be added or removed in a structural mutation is within the $[1, c]$ interval, where c is the integer part of a third of the number of connections in the model. The stop criterion is reached when the following condition is fulfilled: for 20 generations there is improvement neither in the average performance of the best 10% of the population nor in the fitness of the best individual. Regarding the parameters of BBS and SABBS methods (i.e. the number of covariates analyzed in each step, 2, the value for the initial temperature, $0.01 \cdot N$ and the freezing factor, $r = 0.2$), they have been obtained as the best result of a preliminary experimental design.

The other algorithms are available as part of the WEKA machine learning workbench [33] and we applied them to the Matabueyes and Santa Cruz datasets, selecting their default parameter values.

6.2 Evaluation of the Structural Simplification methods

Performance of each model has been evaluated using the CCR in the generalization set (CCR_{G}). In Table I, we show the mean and the standard deviation of this CCR_{G} for a total of 100 executions, and the mean and the standard deviation of the number of coefficients of the corresponding models (including α_j or w_{ji} coefficients). From the analysis of the LRPU model results, it can be concluded that the structural simplification methods considerably reduce the number of coefficients, this difference being higher for the $\text{LRPU}_{\text{SABBS}}$ method. The generalization accuracy of the LRPU models is similar or better after simplifying their structure. A very similar behaviour is observed with respect to the LRIPU model: the accuracy is similar or better when using structural simplification (especially when using the $\text{LRIPU}_{\text{BBS}}$ method) and the number of coefficients is significantly reduced (espe-

	Matabueyes		Santa Cruz	
	CCR_{G}	#Coef.	CCR_{G}	#Coef.
Method	Mean ± SD	Mean ± SD	Mean ± SD	Mean ± SD
LRPU	69.53 ± 3.58	19.66 ± 2.19	75.88 ± 2.79	25.99 ± 3.52
$\mathrm{LRPU}_{\mathrm{BBS}}$	70.07 ± 3.47	17.21 ± 3.14	76.22 ± 2.77	23.96 ± 3.14
$\mathrm{LRPU}_{\mathrm{SABBS}}$	69.49 ± 3.45	13.69 ± 2.04	75.45 ± 3.03	21.70 ± 2.83
LRIPU	69.84 ± 3.57	23.08 ± 2.46	76.10 ± 2.67	27.53 ± 3.53
$\mathrm{LRIPU}_{\mathrm{BBS}}$	70.32 ± 3.58	20.20 ± 3.45	76.26 ± 2.73	25.68 ± 3.19
$\mathrm{LRIPU}_{\mathrm{SABBS}}$	70.10 ± 3.58	15.56 ± 2.91	75.85 ± 2.78	22.80 ± 3.08

Tab. I *Statistical results (Mean and Standard Deviation, SD) of the* $CCR_{\mathbf{G}}$ *and the number of coefficients (#Coef.) obtained using the different methods proposed*

cially when using the $\mathrm{LRIPU}_{\mathrm{SABBS}}$ method). When analyzing both LRPU and LRIPU models and their structural simplification variants, the best CCR_{G} results are obtained by the $\mathrm{LRIPU}_{\mathrm{BBS}}$ method and a similar accuracy is obtained by the $\mathrm{LRIPU}_{\mathrm{SABBS}}$ methodology but with a lower number of connections.

In order to ascertain the statistical significance of the observed differences between the mean CCR_{G} and the mean #Coef. of the best models obtained for each methodology, we have applied the ANalysis Of VAriance (ANOVA) technique [34, 35, 36]. First of all, a non-parametric Kolmogorov-Smirnov test (KS-test) with a signification level $\alpha = 0.05$ was used to evaluate if the CCR_{G} and #Coef. values follow a normal distribution. As a result, a normal distribution can be assumed because all the p-values were higher than 0.05.

	Matabueyes	
	CCR_{G}	#Coef.
F-Test	0.487	$0.000(*)$
Ranking	$\mu_{\mathrm{LRIPU_B}} \geq \mu_{\mathrm{LRIPU_S}} \geq \mu_{\mathrm{LRPU_B}} \geq$	$\mu_{\mathrm{LRPU_S}} < \mu_{\mathrm{LRIPU_S}} < \mu_{\mathrm{LRPU_B}} <$
of means	$\geq \mu_{\mathrm{LRIPU}} \geq \mu_{\mathrm{LRPU}} \geq \mu_{\mathrm{LRPU_S}}$	$< \mu_{\mathrm{LRPU}} \leq \mu_{\mathrm{LRIPU_B}} < \mu_{\mathrm{LRIPU}}$
	Santa Cruz	
	CCR_{G}	#Coef.
F-Test	0.332	$0.000(*)$
Ranking	$\mu_{\mathrm{LRIPU_B}} \geq \mu_{\mathrm{LRPU_B}} \geq \mu_{\mathrm{LRIPU}} \geq$	$\mu_{\mathrm{LRPU_S}} \leq \mu_{\mathrm{LRIPU_S}} \leq \mu_{\mathrm{LRPU_B}}$;
of means	$\geq \mu_{\mathrm{LRPU}} \geq \mu_{\mathrm{LRIPU_S}} \geq \mu_{\mathrm{LRPU_S}}$	$\mu_{\mathrm{LRPU_S}} < \mu_{\mathrm{LRPU_B}}$;
		$\mu_{\mathrm{LRPU_B}} < \mu_{\mathrm{LRIPU_B}} \leq \mu_{\mathrm{LRPU}} <$
		$< \mu_{\mathrm{LRIPU}}$

$(*)$: Statistical significant different with $p-\mathrm{value} < 0.05$
B: Backtracking Backward Search (BBS)
S: Simulated Annealing Backtracking Backward Search (SABBS)

Tab. II $p-$values *of the Snedecor's F ANOVA I test and ranking of means of the Tukey statistical multiple comparison tests for the* CCR_{G} *and #Coef. using the six different methodologies*

The ANOVA test involves a linear regression model in which CCR_{G} or #Coef. are the dependent variables and the independent variable is the type of method-

ology or model used for classification. The comparison was made in terms of a critical level of the Snedecor's F value. If the significance level, p, was higher than this critical value, α, we rejected the hypothesis of identical mean CCR_{G} or #Coef. In our case, this hypothesis is accepted for mean CCR_{G} values in both locations, because the $p-$values were 0.487 and 0.332 (see Table II), they being higher than a standard significance coefficient $\alpha = 0.05$. Consequently, we can conclude that there are not significant differences in mean CCR_{G}. The same hypothesis is not accepted for #Coef., because the $p-$values are equal to 0.000, lower than $\alpha = 0.05$.

Based on these last results and accepting the hypothesis that the variances for the different levels of the #Coef. are equal, we perform a Tukey test [35] for ranking the averages of each level of the factor. Our aim is to find the level of the factor whose mean #Coef. was significantly lower than the average of the rest of the levels of the factor. Table II shows the results obtained following a post-hoc Tukey's multiple comparison test, and the ranking of the different methodologies based on these results. In these rankings, $\mu_{\mathrm{a}} \geq \mu_{\mathrm{b}}$ is used to express that, although the mean CCR_{G} or #Coef. of the "a" methodology is higher than that of "b", the differences are not significant and $\mu_{\mathrm{a}} > \mu_{\mathrm{b}}$ is used to express that the mean results from "a" methodology are significantly higher that those from "b".

From the analysis of the statistical test results, we propose $\mathrm{LRPU}_{\mathrm{SABBS}}$ methodology for both locations, because it results in very similar CCR_{G} levels but with a significantly lower number of coefficients.

6.3 Comparison to other machine learning algorithms

In Table III, the most accurate results ($\mathrm{LRIPU}_{\mathrm{BBS}}$ and $\mathrm{LRIPU}_{\mathrm{SABBS}}$) and the most interpretable results ($\mathrm{LRPU}_{\mathrm{SABBS}}$) have been included, together with the results obtained by using the different WEKA algorithms. As the algorithms considered are all deterministic, we performed ten runs of a ten-fold stratified cross-validation (using the same splits into training/test set for every method), which also gives a hundred data points. In order to obtain the number of coefficients used in each WEKA model, we had to modify the source code of the algorithms for including these values into their outputs, taking into account the special characteristics of each model. These values have also been included in Table III.

Following the same methodology than in the previous subsection, statistical tests have been applied in order to ascertain the statistical significance of the observed differences between the mean CCR_{G} and the results are included in Table IV. For the mean #Coef., the differences are very high and obvious and, in our opinion, a statistical is not needed.

In Matabueyes, the accuracy of the models proposed in this paper is significantly higher than that obtained by NBTree, MultiLogistic and SimpleLogistic and similar to that obtained by the rest of algorithms. In Santa Cruz, a significantly higher accuracy is obtained using LMT, C4.5 and AdaBoost methods. In any case, the number of coefficients of the proposed models (specially that of the $\mathrm{LRPU}_{\mathrm{SABBS}}$ model) is lower than that of those models which obtain a similar or better accuracy (LMT, C4.5 and AdaBoost100), resulting in more interpretable and efficient models.

	Matabueyes		Santa Cruz	
	CCR_{G}	#Links	CCR_{G}	#Links
Method	Mean ± SD	Mean ± SD	Mean ± SD	Mean ± SD
SLogistic	65.86 ± 3.30	4.46 ± 0.50	65.22 ± 3.89	3.54 ± 0.63
MLogistic	66.14 ± 3.27	5.00	65.23 ± 4.20	5.00
LMT	71.68 ± 3.61	181.04 ± 91.23	80.36 ± 3.05	263.51 ± 76.59
C4.5	70.74 ± 3.54	33.63 ± 8.39	79.44 ± 3.29	44.68 ± 6.90
NBTree	66.78 ± 4.01	20.72 ± 9.08	75.96 ± 4.30	43.06 ± 16.56
Ada100	70.89 ± 3.56	108.42 ± 29.96	80.03 ± 2.96	277.66 ± 122.22
$\mathrm{LRIPU}_{\mathrm{BBS}}$	70.32 ± 3.58	20.20 ± 3.45	76.26 ± 2.73	25.68 ± 3.19
$\mathrm{LRIPU}_{\mathrm{SABBS}}$	70.10 ± 3.58	15.56 ± 2.91	75.85 ± 2.78	22.80 ± 3.08
$\mathrm{LRPU}_{\mathrm{SABBS}}$	69.49 ± 3.45	13.69 ± 2.04	75.45 ± 3.03	21.70 ± 2.83

Tab. III *Statistical results obtained with the proposed methodologies compared to other machine learning algorithms*

	Matabueyes
	CCR_{G}
F-Test	$0.000(*)$
Ranking of means	$\mu_{\mathrm{LMT}} \geq \mu_{\mathrm{Ada100}} \geq \mu_{\mathrm{C4.5}} \geq \mu_{\mathrm{LRIPU_B}} \geq \mu_{\mathrm{LRIPU_S}} \geq \mu_{\mathrm{LRPU_S}}$; $\mu_{\mathrm{LMT}} > \mu_{\mathrm{LRIPU_S}}$; $\mu_{\mathrm{LRPU_S}} > \mu_{\mathrm{NBTree}} \geq \mu_{\mathrm{MLogistic}} \geq \mu_{\mathrm{SLogistic}}$
	Santa Cruz
	CCR_{G}
F-Test	$0.000(*)$
Ranking of means	$\mu_{\mathrm{LMT}} \geq \mu_{\mathrm{Ada100}} \geq \mu_{\mathrm{C4.5}} > \mu_{\mathrm{LRIPU_B}} \geq \mu_{\mathrm{NBTree}} \geq \mu_{\mathrm{LRIPU_S}} \geq \mu_{\mathrm{LRPU_S}}$; $\mu_{\mathrm{LRPU_S}} > \mu_{\mathrm{MLogistic}} \geq \mu_{\mathrm{SLogistic}}$

$(*)$: Statistical significant different with $p-$value < 0.05
B: Backtracking Backward Search (BBS)
S: Simulated Annealing Backtracking Backward Search (SABBS)

Tab. IV $p-$values *of the Snedecor's F ANOVA I test and ranking of means of the Tukey statistical multiple comparison tests for the* CCR_{G} *using the different methodologies proposed and the different machine learning algorithms*

7. Conclusions

The structural simplification methods (BBS and SABBS) presented in this paper have demonstrated an important coefficient reduction for both the hybrid neurologistic models proposed in [3] (LRPU and LRIPU), yielding to a better or similar accuracy. In this way, more interpretable models that can lead to a better understanding of the classification problem tackled have been obtained. These models can provide information to program the suitable wavelengths of further Compact Airborne Spectral Imager (CASI) images for Site-Specific Weed Management (SSWM). The next investigation could address the acquisition of specific satellite imagery for mapping *R. Segetum* in larger areas, using only the more relevant channels and reducing the cost of the images. Moreover, the comparison of

these models to six different commonly used machine learning methods has shown the $\text{LRIPU}_{\text{BBS}}$ and $\text{LRIPU}_{\text{SABBS}}$ methods as very competitive models with a lower number of coefficients and has demonstrated their capability to analyze multispectral imagery for predicting *R. segetum* presence probability in the field of study, providing a useful tool for early SSWM.

Acknowledgements

This work has been partially financed by the CICYT-TIN2005-08386-C05-02 and CICYT-AGL-2008-04689-CO3-03 projects (Spanish Inter-Ministerial Commission of Science and Technology), FEDER funds and the P08-TIC-3745 project (Junta de Andalucía). The research done by P.A. Gutiérrez and J.C. Fernández has been financed respectively by the FPU (grant reference AP-2006-01746) and the FPI (grant reference BES-2006-12543) Predoctoral Programs (Spanish Ministry of Education and Science).

References

[1] R. L. Major and C. T. Ragsdale, "Aggregating expert predictions in a networked environment," *Computers & Operations Research*, vol. 28, no. 12, pp. 1231–1244, 2001.

[2] R. Beghdad, "Applying fisher's filter to select kdd connections' features and using neural networks to classify and detect attacks," *Neural Network World*, vol. 17, no. 1, pp. 1–16, 2007.

[3] C. Hervas-Martinez and F. Martinez-Estudillo, "Logistic regression using covariates obtained by product-unit neural network models," *Pattern Recognition*, vol. 40, no. 1, pp. 52–64, 2007.

[4] C. Hervás-Martínez, F. J. Martínez-Estudillo, and M. Carbonero-Ruz, "Multilogistic regression by means of evolutionary product-unit neural networks," *Neural Networks*, vol. 21, no. 7, pp. 951–961, 2008.

[5] "MAPA," Spanish Ministry of Agriculture, Fisheries and Food, 2007. [Online]. Available: http://www.mapa.es/es/estadistica/infoestad.html

[6] P. Carranza-Cañadas, M. Saavedra, and L. García-Torres, "Competition of ridolfia segetum and sunflower," *Weed Research*, vol. 35, no. 5, pp. 369–375, 1995.

[7] M. Jurado-Expósito, F. López-Granados, J. L. González-Andújar, and L. García-Torres, "Characterizing population growth rate of convolvulus arvensis in no-tillage systems," *Crop Science*, vol. 45, no. 21, pp. 2106–2112, 2005.

[8] A. Srinivasan, *Handbook of Precision Agriculture*. Haworth Press Inc, 2006.

[9] R. B. Brown and S. D. Noble, "Site-specific weed management: sensing requirements - what do we need to see?" *Weed Science*, vol. 53, no. 2, pp. 252–258, 2005.

[10] P. A. Gutiérrez, F. López-Granados, J. M. Peña-Barragán, M. Jurado-Expósito, and C. Hervás-Martínez, "Logistic regression product-unit neural networks for mapping *Ridolfia segetum* infestations in sunflower crop using multitemporal remote sensed data," *Computers and Electronics in Agriculture*, vol. 62, no. 2, pp. 293–306, 2008.

[11] P. A. Gutiérrez, F. López-Granados, J. M. Peña-Barragán, M. Jurado-Expósito, M. T. Gómez-Casero, and C. Hervás, "Mapping sunflower yield as affected by *Ridolfia segetum* patches and elevation by applying evolutionary product unit neural networks to remote sensed data," *Computers and Electronics in Agriculture*, vol. 60, no. 2, pp. 122–132, 2008.

[12] D. W. Hosmer and S. Lemeshow, *Applied logistic regression (Wiley Series in probability and statistics)*. New York; Chichester: Wiley Interscience, September 2000.

[13] P. Komarek and A. W. Moore, "Making logistic regression a core data mining tool with tr-irls," in *Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM'05)*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 685–688.

[14] R. Durbin and D. Rumelhart, "Products units: A computationally powerful and biologically plausible extension to backpropagation networks," *Neural Computation*, vol. 1, no. 1, pp. 133–142, 1989.

[15] M. Schmitt, "On the complexity of computing and learning with multiplicative neural networks," *Neural Computation*, vol. 14, pp. 241–301, 2002.

[16] A. Ismail and A. P. Engelbrecht, "Pruning product unit neural networks," in *Proceedings of the 2002 International Joint Conference on Neural Networks*, 2002, pp. 257–262.

[17] M. Köppen and K. Yoshida, "Many-objective training of a multi-layer perceptron," *Neural Network World*, vol. 17, no. 6, pp. 627–637, 2007.

[18] S. Michal and K. Stefan, "Automatic generation of neural network structures using genetic algorithm," *Neural Network World*, vol. 15, no. 5, pp. 381–394, 2005.

[19] J. T. Tsai, J. H. Chou, and T. K. Liu, "Tuning the structure and parameters of a neural network by using hybrid taguchi-genetic algorithm," *IEEE Transactions on Neural Networks*, vol. 17, no. 1, pp. 69–80, 2006.

[20] P. J. Angeline, G. M. Sauders, and J. B. Pollack, "An evolutionary algorithm that constructs recurren neural networks," *IEEE Transactions on Neural Networks*, vol. 5, pp. 54–65, 1994.

[21] F. J. Martínez-Estudillo, C. Hervás-Martínez, P. A. Gutiérrez, and A. C. Martínez-Estudillo, "Evolutionary product-unit neural networks classifiers," *Neurocomputing*, vol. 72, no. 1-2, pp. 548–561, 2008.

[22] X. Yao, "Evolving artificial neural networks," *Proceedings of the IEEE*, vol. 87, no. 9, pp. 1423–1447, 1999.

[23] X. Yao and Y. Liu, "A new evolutionary system for evolving artificial neural networks," *IEEE Transactions on Neural Networks*, vol. 8, no. 3, pp. 694–713, 1997.

[24] C. M. Bishop, *Neural networks for pattern recognition*. Oxford, UK: Oxford University Press, 1996.

[25] I. Rechenberg, *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Stuttgart: Frommann-Holzboog, 1973.

[26] A. C. Martínez-Estudillo, F. J. Martínez-Estudillo, C. Hervás-Martínez, and N. García, "Evolutionary product unit based neural networks for regression," *Neural Networks*, vol. 19, no. 4, pp. 477–486, 2006.

[27] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[28] J. M. Peña-Barragán, F. López-Granados, M. Jurado-Expósito, and L. García-Torres, "Mapping *Ridolfia segetum* moris patches in sunflower (Helianthus annuus L.) crop using remote sensing," *Weed Research*, vol. 47, no. 2, pp. 164–172, 2007.

[29] R. M. McCoy, *Field Methods in Remote Sensing*. The Guilford Press, 2004.

[30] N. Landwehr, M. Hall, and E. Frank, "Logistic model trees," *Machine Learning*, vol. 59, no. 1-2, pp. 161–205, 2005.

[31] S. Ventura, C. Romero, A. Zafra, J. A. Delgado, and C. Hervas, "JCLEC: a Java framework for evolutionary computation," *Soft Computing*, vol. 12, no. 4, pp. 381–392, 2008.

[32] J. Alcala-Fdez, L. Sánchez, S. García, M. J. del Jesús, S. Ventura, J. M. Garrell, J. Otero, C. Romero, J. Bacardit, V. M. Rivas, J. C. Fernández, and F. Herrera, "KEEL: A software tool to assess evolutionary algorithms for data mining problems," *Soft Computing*, vol. 13, no. 3, pp. 307–318, 2008.

[33] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed., ser. Data Management Systems. Morgan Kaufmann (Elsevier), 2005.

[34] R. M. Mickey, O. J. Dunn, and V. A. Clark, *Applied Statistics: Analysis of Variance and Regression, 3rd Edition*, 3rd ed. Willey, 2004.

[35] R. G. Miller, *Simultaneous Statistical Inference*, 2nd ed. New York, USA: Wiley, 1981.

[36] G. W. Snedecor and W. G. Cochran, *Statistical Methods*, 6th ed. Iowa State University Press, 1989.