COMPUTERS AND ELECTRONICS IN AGRICULTURE 64 (2008) 293–306

available at www.sciencedirect.com

ScienceDirect

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

Logistic regression product-unit neural networks for mapping *Ridolfia segetum* infestations in sunflower crop using multitemporal remote sensed data

P.A. Gutiérrez[a,*], *F. López-Granados*[b], *J.M. Peña-Barragán*[b], *M. Jurado-Expósito*[b], *C. Hervás-Martínez*[a]

[a] *Department of Computer Science and Numerical Analysis, University of Cordoba, Campus de Rabanales, 14071 Cordoba, Spain*
[b] *Institute for Sustainable Agriculture, CSIC, Apdo. 4084, 14080 Cordoba, Spain*

ARTICLE INFO

Article history:
Received 25 March 2008
Received in revised form
23 June 2008
Accepted 30 June 2008

Keywords:
Artificial neural networks
Evolutionary algorithms
Multi-spectral imagery
Precision agriculture
Remote sensing
Weed patch classification

ABSTRACT

Remote sensing (RS), geographic information systems (GIS), and global positioning systems (GPS) may provide the technologies needed for farmers to maximize the economic and environmental benefits of precision farming. Site-specific weed management (SSWM) is able to minimize the impact of herbicide on environmental quality and arises the necessity of more precise approaches for weed patches determination. *Ridolfia segetum* is one of the most dominant, competitive and persistent weed in sunflower crops in southern Spain. In this paper, we used aerial imagery taken in mid-May, mid-June and mid-July according to different phenological stages of *R. segetum* and sunflower to evaluate the potential of evolutionary product-unit neural networks (EPUNNs), logistic regression (LR) and two different combinations of both (logistic regression using product units (LRPU) and logistic regression using initial covariates and product units (LRIPU)) for discriminating *R. segetum* patches and mapping *R. segetum* probabilities in sunflower crops on two naturally infested fields. Afterwards, we compared the performance of these methods in every date to two recent classification models (support vector machines (SVM) and logistic model trees (LMT)). The results obtained present the models proposed as powerful tools for weed discrimination, the best performing model (LRIPU) obtaining generalization accuracies of 99.2% and 98.7% in mid-June. Our results suggest that a strategy to implement SSWM is feasible with minima omission and commission errors, and therefore, with a very low probability of not detecting *R. segetum* patches. The paper proposes the application of a new methodology that, to the best of our knowledge, has not been previously applied in RS, and which obtains better accuracy than more traditional RS classification techniques, such as vegetation indices or spectral angle mapper.

1. Introduction

Remote sensing, geographic information systems (GIS), and global positioning systems (GPS) may provide the technologies needed for farmers to maximize the economic and environmental benefits of precision farming (Seelan et al., 2003). Patchy distribution of grass (monocotyledonous) and broadleaf (dicotyledonous) weeds in fields is well documented

* *Corresponding author at*: Department of Computer Science and Numerical Analysis, University of Cordoba, Campus de Rabanales, Albert Einstein Building 3rd Floor, 14071 Cordoba, Spain. Tel.: +34 957 218 349; fax: +34 957 218 630.
E-mail address: i02gupep@uco.es (P.A. Gutiérrez).
0168-1699/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.compag.2008.06.001

(Barroso et al., 2004a; Jurado-Expósito et al., 2003, 2005a). However, herbicides are not applied to the infested zones, but they are usually broadcast over entire fields. The potential for over application and the corresponding economical and environmental problems is evident. One aspect of overcoming the possibility of minimizing the impact of herbicide on environmental quality is the development of more efficient approaches for crop production determination and for site-specific weed management (SSWM). Srinivasan (2006) reported an overall overview of the current status of site-specific weed, nutrient, crop diseases and water management. Timmermann et al. (2003) evaluated the economical and ecological benefits of SSWM in annual crops in a 4 years study, concluding that costs savings were 90% and 60% for broadleaf and grass weeds herbicides, respectively.

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 surveys techniques on field scale is time consuming, expensive and unapproachable in field areas with difficult access. The importance of remote sensing in site-specific agriculture has been widely reviewed (Plant, 2001). 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) (Brown and Noble, 2005; Thorp and Tian, 2004). In this sense, the analysis of the accuracy of weed maps is essential to make possible the implementation of models with a high performance, leading to a reduction of the overall herbicide application. Detection of late-season weed infestation has demonstrated to have tremendous possibilities when spectral differences between crops and weeds prevail (Koger et al., 2003; López-Granados et al., 2006).

Sunflower (*Helianthus annuus* L.) is one of the most abundant crops in Andalusia, Southern Spain, with more than 320,000 ha sown annually (MAPA, 2007). Sunflower sowing and harvesting times are February–March and July–August, respectively, being mainly grown under dry land conditions. *Ridolfia segetum* Moris (corn caraway) is an annual, umbelliferous weed very frequently found and abundant in clay soils in Andalusia. Its life cycle coincides with that of the sunflower, which enhances the competitive ability and results in an average crop yield reduction of about 32% when infestation is two *R. segetum* plants per m^2 (Carranza-Cañadas et al., 1995). This weed is not controlled by pre-emergence and pre-plant incorporated herbicides, as it is in sunflower, and consequently post-emergence strategies such as tillage or hand weeding are commonly used. Otherwise weed obstructs the harvester since it still has partly green stem during the sunflower harvesting. This is a serious inconvenience if the harvester is equipped with a yield monitor, as it is frequent in precision agriculture management.

Logistic regression (LR) has become a widely used and accepted method of analysis of binary or multiclass outcome variables as it is more flexible and it can predict the probability for the state of a dichotomous variable (e.g., red-attack/non-attack for mountain pine beetle red-attack damage (Wulder et al., 2006)) based on the predictor variables (e.g., vegetation indices, slope, solar radiation, etc.) and has been widely applied in forestry to estimate tree and stand survival under competition (Monserud and Sterba, 1999; Vanclay, 1995). In the field of remote sensing, LR has been used with different aims, for example, for land-cover change detection (Fraser et al., 2005), and for mapping insect tree defoliation (Magnussen et al., 2004). LR and multiple linear regression techniques have also been used for identifying spectrally sensitive regions in order to detect nitrogen deficiency in corn (Cetin et al., 2005).

Approaches based on artificial neural networks (ANNs) have been considered in agricultural remote sensing (Goel et al., 2003; Gutiérrez et al., 2008; Uno et al., 2005; Yang et al., 2000a) to associate complicated spectral information with target attributes without any constraints for sample distribution (Mather, 2000). ANNs have long been applied for the land-cover classification in remotely sensed images (Atkinson and Tatnall, 1997; Kanellopoulos and Wilkinson, 1997; Zhai et al., 2006). In ANNs, the hidden neurons are the functional units and can be considered as generators of function spaces. Most existing neuron models are based on the summing operation of the inputs, and, more particularly, on sigmoidal unit functions, resulting in what is known as the multilayer perceptron (MLP).

Product-unit neural network (PUNN) models are an alternative to MLPs and are based on multiplicative neurons instead of additive ones. They correspond to a special class of feed-forward neural network introduced by Durbin and Rumelhart (1989). While MLP network models have been very successful, networks that make use of product units (PUs) have the added advantage of increased information capacity (Durbin and Rumelhart, 1989). That is, smaller PUNNs architectures can be used than those used with MLPs (Ismail and Engelbrecht, 2002). They aim to overcome the nonlinear effects of variables by means of nonlinear basis functions, constructed with the product of the inputs raised to arbitrary powers. These basis functions express the possible strong interactions between the variables, where the exponents may even take on real values and are suitable for automatic adjustment.

Classical neural networks training algorithms assume a fixed architecture difficult to establish beforehand. Evolutionary algorithms (EAs), which are stochastic search algorithms that execute a global search in the input space preventing the fall to local optimum (Angeline et al., 1994), have demonstrated great accuracy in designing a near optimal architecture. This fact, together with the complexity of the error surface associated with a PUNN, justifies the use of an EA to design the structure and to adjust the weights of these models (Martínez-Estudillo et al., 2006). Moreover, genetic algorithms (GAs) (the most widely applied EAs) have been independently used in several remote sensing applications (Mertens et al., 2003; Zhan et al., 2003).

Evolutionary artificial neural networks (EANNs) (Yao, 1999) are the combination of ANNs and EAs. Combining these two methods, each technique complements the disadvantages of the other (Yao, 1999). For example, a contribution by ANNs is the flexibility of nonlinear function approximation, which cannot be easily implemented with prototype EAs. Furthermore, neural networks play a significant role in remote sensing since they can handle massive, complex and incomplete data sets efficiently. As such, they are candidates to produce bet-

ter results than some of the traditional classifiers, requiring in some cases less training data (Moody et al., 1996).

On the other hand, EAs have freed ANNs from simple gradient descent approaches of optimization. In fact, traditional ANNs based on backpropagation algorithms have some limitations. At first, the architecture of the artificial neural networks is fixed and a designer needs much knowledge to determine it. Besides, the error function of the learning algorithm must have a derivative. Finally, the algorithm frequently gets stuck in local optima because it is based on gradient-based search without stochastic property. The combination of EAs and ANNs can overcome these shortcomings and is particularly useful when the activation function of the neurons is non-differentiable and traditional gradient-based training algorithms cannot be used. Because an EA can treat non-differentiable and multimodal spaces, which are the typical case in the classification of remotely sensed imagery, there must be a great motivation to apply EANN to classification of remotely sensed imagery.

A very recent approach (Hervás-Martínez and Martínez-Estudillo, 2007) is based on the hybridization of the LR model and evolutionary PUNNs (EPUNNs), in order to obtain binary classifiers. In a first step, an EA is used to determine the basic structure of the product-unit model. That step can be seen as a global search in the space of the model coefficients. Once the basis functions have been determined by the EA, a transformation of the input space is considered. This transformation is performed by adding the nonlinear transformations of the input variables given by the PU basis functions obtained by the EA. The final model is linear in these new variables together with the initial covariates. This hybrid model outperforms the linear part and the nonlinear part obtaining a good compromise between them and performing well compared to several other learning classification techniques. This methodology has been extended to multiclass problems (Hervás-Martínez et al., 2008).

The WEKA machine learning workbench provides a general-purpose environment for automatic classification, regression, clustering and feature selection. It contains an extensive collection of machine learning algorithms and data pre-processing methods complemented by graphical user interfaces for data exploration and the experimental comparison of different machine learning techniques on the same problem. For comparison purposes, we consider in this work two of most widely accepted WEKA methods in machine learning community: the support vector machines (SVMs) and the logistic model trees (LMTs). SVMs are a recent classification method based on the statistical learning theory of Vapnik (1995) and they have been successfully applied to very large highly nonlinear problems such as character recognition. LMT is an algorithm that combines a decision tree structure with LR models, using LR at the leaves of the tree and resulting in a single tree. LMTs have been shown to be very accurate and compact classifiers (Landwehr et al., 2005).

One of the objectives of this work was to generate a weed probability map by using logistic models approaches, obtaining in this way probability maps, capable of indicating a range of weed presence likelihood, rather than a binary indication of weed or weed-free. On the other hand, we aimed to assess the potential of different classification models based on PUNNs and LR for reaching the best approach to map *R. segetum* patches in sunflower crop at a field scale, using remote sensed information and assessing the best date for discriminating this weed. Four were the models tested: (a) the LR statistical model, (b) the EPUNN model, and two combinations of both, (c) LR using product units (LRPU), i.e., LR with PU basis functions obtained from EPUNNs, and (d) LR using initial covariates and product units (LRIPU), i.e., LRPU extending the model with the initial covariates of the problem. Moreover these models are compared with the previously described machine learning classifiers: (e) SVM, and (f) LMT.

2. Materials and methods

2.1. Study sites, materials and experimental design

The study was conducted at two fields in Andalusia, southern Spain: at Matabueyes, 42 ha (coordinates 37°8′N, 4°8′W, WGS84), and at Santa Cruz, 28 ha (coordinates 37°8′N, 4°6′W, WGS84) in 2003 and 2004, respectively. Both fields were naturally infested by *R. segetum*. Soil at both locations was classified as Typic Chromoxerert (USDA-NRCS, 1998), with approximately 60% clay. Sunflower cv. Jalisco was seeded in mid-March at 4 kg ha^{-1} in rows 0.7 m apart and harvested in mid-August using the farm's MF34 combine equipped with a calibrated Fieldstar® yield monitor and a differentially corrected global positioning system (DGPS) receiver (Massey Fergusson®, AGCO Corporation, Duluth, GA, USA). The field sites were managed using shallow tillage production methods. Glyphosate (Roundup, isopropylamine salt, 360 g a.i. L^{-1}, Montsanto) was applied pre-emergence at 180 g a.i. L^{-1} for the control of annual weed seedlings in sunflower. At this rate, this herbicide has no effect on *R. segetum* emergence or development.

The sunflower phenological stages considered were adapted to the study conditions from Peña-Barragán et al. (2006), the vegetative phase (from the emergence stage to the early reproductive stage) in mid-May, the flowering phase (from the head flowering stage to the initial desiccation stage of lower leaves) in mid-June, and the senescent phase (from the stage in which the reproductive head is partly desiccated and browning to the stage in which the plant becomes completely desiccated and darkish/black) in mid-July. The *R. segetum* phenological stages were also based on previous studies as follows: in mid-May, the vegetative phase (weed growth from seedling to the vegetative stage without the floral stem and the inflorescences (or umbellas) still closed), in mid-June, the flowering phase (inflorescences are yellowing and the plant is at its largest size), and in mid-July, the senescent phase (weed desiccates and turns brown).

Conventional-colour (CC) and colour-infrared (CIR) aerial imagery of the studied field were taken in mid-May, mid-June and mid-July (except CIR images in mid-June at Matabueyes, due to technical problems).

The photographs were taken by a turboprop twin-engine plane CESSNA 402. The photographs were taken on cloudless days between 12 and 14 h standard time and the average flight height was 1525 m to obtain photographs at a scale 1:10,000. An automatic pilot was used for managing both pho-

tographic equipment and GPS and the camera was a RMK TOP 15, with a Zeiss objective, and a focal distance of 153.76 mm. Kodak Aerocolor III 2444 and Kodak Aerochrome S0734 film was used for CC and CIR photographs, respectively. Then, the photographs were digitalized with an AGFA Horizon A3 scanner, considering a resolution of 635 dots per inch (dpi), brightness and contrast not being adjusted on the digitalized images. The next step was to orthorectify the digitised images, using the fiducial marks of the aerial calibration certificate, 40 ground control points taken with a differential GPS TRIMBLE PRO-XRS equipped with a TDC-1 unit (centimetre precision) and a 10-m resolution raster DEM. Finally, images were resampled to a pixel size representing 40 cm × 40 cm ground area.

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. The scanner produced a RGB digital image with 8-bit true colour, so pixels of the image showed digital counts within the range of 0–255 values. These digital values are considered as being directly proportional to the total light reflected from the scene (Flowers et al., 2001). All spatial and spectral data from images were grouped and saved to a unique multiband file taking into account two previous requirements: (1) the georeference error between images was less than one pixel, so similar pixels had the same coordinate, and (2) the NIR digital values of CIR images were corrected to the digital values of CC images, using the differences between the G and R bands of both original images.

To train and validate the classification models, a random ground sampling procedure was carried out at the time when aerial images were taken, ensuring that all parts of the field area had an equal chance of being sampled with no operator bias (McCoy, 2005). We georeferenced a total of 1600 pixels in each phenological stage, where 800 pixels corresponded to *R. segetum* class, 400 pixels corresponded to bare soil class and 400 corresponded to sunflower. In this way, the number of weed-free pixels was the same to the number of *R. segetum* pixels. The objective is the differentiation between *R. segetum* and all other pixels, as distinguishing between *soil* and *sunflower* is not needed for site-specific herbicide application.

The experimental design was conducted using a stratified holdout cross-validation procedure, where the size of the training set was approximately 3*n*/4 and *n*/4 for the generalization set, *n* being the size of the full dataset. Consequently, each dataset mentioned above was randomly split in two datasets. A 1120 instances dataset was used for model training and the remaining 480 instances formed the generalization dataset. The supervised classification process is composed of three steps: one is the classifier training which aims at creating a reliable input–output relationship between remotely sensed data and land-cover class membership; then the models obtained are evaluated through the classification of previously unseen data whose land-cover class membership is known, in order to assess their generalization capability; the final step is image classification which applies the relationship established in the training process to the whole image.

2.2. Methods

Different methods have been applied for training the classifiers. These include from classical statistical approaches to the hybrid approaches proposed by Hervás-Martínez and Martínez-Estudillo (2007).

2.2.1. Statistical methods: binary logistic regression

Typically, in supervised image classification, a set of n_T training samples or pixels $(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_{n_T}, \mathbf{y}_{n_T})$ is given. The inputs $\mathbf{x}_i$ (i.e., spectral bands) form a feature space X, and the output $\mathbf{y}_i$ (i.e., the target class) has a class label c, which belongs to a finite set C. A classification rule is designed based on the training data, so that, given a new input $\mathbf{x}_i$ of a pixel, a class c from C with the smallest probability of error is assigned to it.

In this paper the situation considered is the following: a binary outcome variable y (weed presence or weed-free) is observed together with a vector $\mathbf{x}_i = (1, x_{i1}, x_{i2}, \ldots, x_{ik},)$ of covariates for each of the n_T pixels (assuming that the vector of inputs includes the constant term 1 to accommodate the intercept). The two-class is coded via a 0/1 response y_i, where $y_i = 1$ for weed presence and $y_i = 0$ for weed-free pixels. Let p be the conditional probability associated with the first class. Logistic regression (Hosmer and Lemeshow, 1989) is a widely used statistical modeling technique in which the probability p of the dichotomous outcome event is related to a set of explanatory variables $\mathbf{x}$ in the form:

$$\text{logit}(p) = \ln\left(\frac{p}{1-p}\right) = f_{\text{LR}}(\mathbf{x}, \boldsymbol{\beta}) = \boldsymbol{\beta}^{\text{T}}\mathbf{x} \tag{1}$$

where $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_k)$ is the vector of the coefficients of the model and $\boldsymbol{\beta}^{\text{T}}$ the transpose vector and $f_{\text{LR}}(\mathbf{x}, \boldsymbol{\beta})$ is the LR model. We refer to $p/(1-p)$ as odds-ratio and to the expression (1) as the log-odds or logit transformation. A simple calculation in Eq. (1) shows that the probability of occurrence of an event as a function of the covariates is nonlinear and is given by

$$p(\mathbf{x}; \boldsymbol{\beta}) = \frac{e^{\boldsymbol{\beta}^{\text{T}}\mathbf{x}}}{1 + e^{\boldsymbol{\beta}^{\text{T}}\mathbf{x}}} \tag{2}$$

The complementary event probability can therefore be obtained as $(1 - p(\mathbf{x};\boldsymbol{\beta}))$. Once the conditional probability function defined in (2) is known, the Bayesian (optimal) decision rule can be constructed:

$$r(\mathbf{x}) = \text{sign}\left\{\ln\left(\frac{p(\mathbf{x};\boldsymbol{\beta})}{1 - p(\mathbf{x};\boldsymbol{\beta})}\right)\right\} \tag{3}$$

Given any test pixel $\mathbf{x}$, the probability p that the pixel belongs to the first class can be determined from (2). Similar to the maximum-likelihood classification, these class probabilities for each pixel may be outputted to reflect the actual proportion of classes within a pixel, thereby producing a soft, fuzzy or subpixel classification. The results from this paper advocate the utility of the LR as a potential approach for the soft classification similar to other recent approaches such as the MLP neural networks (Foody and Arora, 1996) or the decision tree regression (Xu et al., 2005). A hard clas-

sification can be produced by assigning the class having a maximum probability (in our case, as a binary outcome variable is considered, we can simply check if the probability p is greater or lower than the value 0.5). Observe that LR not only constructs a decision rule but it also finds a function that for any input vector defines the probability p that the vector $\mathbf{x}$ belongs to the first class, in our case *R. segetum* presence.

Let $D = \{(\mathbf{x}_l, y_l); 1 \leq l \leq n_T\}$ be the training data set, where the number of samples is n_T. Here it is assumed that the training sample is a realization of a set of independent and identically distributed random variables. The unknown regression coefficients β_i, which have to be estimated from the data, are directly interpretable as log-odds ratios or, in term of $\exp(\beta_i)$, as odds ratios. That log-likelihood used as the error function is

$$l(\boldsymbol{\beta}) = \sum_{l=1}^{n_T} y_l \log p(\mathbf{x}_l; \boldsymbol{\beta}) + (1 - y_l) \log(1 - p(\mathbf{x}_l; \boldsymbol{\beta})) = \sum_{l=1}^{n_T} y_l \boldsymbol{\beta}^T \mathbf{x} - \log(1 + e^{\boldsymbol{\beta}^T \mathbf{x}}) \quad (4)$$

The estimation of the coefficient $\boldsymbol{\beta}$ is usually carried out by means of an iterative procedure like the Newton–Raphson algorithm or the iteratively reweighted least squares (IRLS) (Hastie et al., 2001). Typically the algorithm converges, since the log-likelihood is concave, but overshooting can occur. In the rare cases where log-likelihood decreases, step size halving will guarantee convergence. The conditions under which a global maximum exists and the maximum likelihood estimators which do not diverge are discussed by McLachlan (1992) and references therein.

2.2.2. *Statistical methods: support vector machines and logistic model trees*

SVM is now a very popular tool in machine learning, which explores the kernel techniques. Reasons for this popularity include geometric exploration and an accurate performance in many classification applications. SVM is basically a binary classifier, which finds the maximal margin (hyperplane) between two classes. SVM can classify nonlinearly separable data sets by plotting the data into a high-dimensional feature space using kernels. For further details and recent development of the SVMs, we refer readers to Cristianini and Shawe-Taylor (2000) and Vapnik (1995).

Tree induction methods and linear models are popular techniques for the prediction of nominal classes and numeric values in supervised learning tasks. For predicting numeric quantities, some work has been carried out on combining these two schemes into 'model trees', i.e., trees that contain linear regression functions at the leaves. Model trees, like ordinary regression trees, predict a numeric value for an instance that is defined over a fixed set of numeric or nominal attributes. Unlike ordinary regression trees, model trees construct a piecewise linear (instead of a piecewise constant) approximation to the target function. The final model tree consists of a tree with linear regression functions at the leaves, and the prediction for an instance is obtained by sorting it down to a leaf and using the prediction of the linear model associated with that leaf.

LMT is an algorithm that adapts this idea for classification problems, using LR instead of linear regression. In fact, LMT can be regarded as a new scheme for selecting the attributes to be included in the logistic regression models, and introduces a way of building the logistic models at the leaves by refining logistic models that have been trained at higher levels in the tree, i.e. on larger subsets of the training data. This algorithm has been showed as producing very accurate and compact classifiers (Landwehr et al., 2005).

2.2.3. *Logistic regression using covariates obtained by product-unit neural network models*

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 evolutionary product-unit neural networks.

2.2.3.1. Product-unit neural networks. PUNNs are an alternative to MLPs, and are based on multiplicative neurons instead of additive ones. A multiplicative neuron is given by $\prod_{i=1}^{k} x_i^{w_{ji}}$, where k is the number of the inputs. When the exponents are $\{0,1\}$ a higher order unit is obtained, namely the sigma–pi unit (Lenze, 1994). In contrast to the sigma–pi unit, in the product-unit the exponents are not fixed and may even take real values.

Product-unit based neural networks 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 (Vapnik, 1995) of product-unit neural networks similar to those obtained for MLP (Schmitt, 2002).

Despite these advantages, PUNNs have a major handicap: they have more local minima and more probability of becoming trapped in them (Ismail and Engelbrecht, 2002). The main reason for this difficulty is that small changes in the exponents can cause large changes in the total error surface and therefore their training is more difficult than the training of standard MLPs. Several efforts have been made to carry out learning methods for product-units (Ismail and Engelbrecht, 2002; Janson and Frenzel, 1993). The back propagation algorithm, which is the most common algorithm for training multilayer neural networks, does not work very well with the product-units because of its complex error surface.

The structure of the neural network considered is described in Fig. 1: an input layer with k neurons, a neuron for every input variable, a hidden layer with m neurons and an output layer with one neuron.

There are no connections between the neurons of a layer and none between the input and output layers either. The activation function of the j-th neuron in the hidden layer is given by $\prod_j(\mathbf{x}, \mathbf{w}_j) = \prod_{i=1}^{k} x_i^{w_{ji}}$, where w_{ji} is the weight of the connection between input neuron i and hidden neuron j and $\mathbf{w}_j = (w_{j1}, \ldots, w_{jk})$ is the weight vector. The activation function

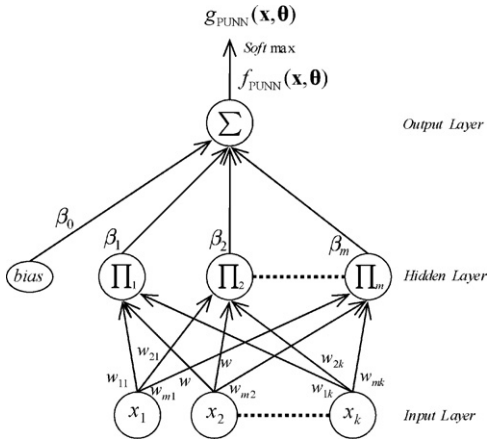

Fig. 1 – Model of a product-unit neural network.

of the output neuron is given by

$$f_{PUNN}(\mathbf{x},\boldsymbol{\theta}) = \beta_0 + \sum_{j=1}^{m} \beta_j \prod_j (\mathbf{x}, \mathbf{w}_j) \tag{5}$$

where β_j is the weight of the connection between the hidden neuron j and the output neuron. The transfer function of all hidden and output neurons is the identity function.

We consider the softmax activation function (Bishop, 1995) given by

$$g_{PUNN}(\mathbf{x},\boldsymbol{\theta}) = \frac{\exp(f_{PUNN}(\mathbf{x},\boldsymbol{\theta}))}{1+\exp(f_{PUNN}(\mathbf{x},\boldsymbol{\theta}))} \tag{6}$$

where $f_{PUNN}(\mathbf{x}, \boldsymbol{\theta})$ is the output of the output neuron for pattern $\mathbf{x}$ and $g_{PUNN}(\mathbf{x}, \boldsymbol{\theta})$ is the probability that a pattern $\mathbf{x}$ has of belonging to the "weed presence" class. With this model, the cross-entropy error function is defined by the same expression than in (4), substituting $\boldsymbol{\beta}^T\mathbf{x}$ with $f_{PUNN}(\mathbf{x}, \boldsymbol{\theta})$.

2.2.3.2. Evolutionary product-unit neural networks. In order to estimate the parameters and the structure of the PUNNs that minimizes the classification error function, an Evolutionary algorithm has been considered. The algorithm is similar to the one proposed by Martínez-Estudillo et al. (in press). The population-based evolutionary algorithm for architectural design and the estimation of real-coefficients have points in common with other evolutionary algorithms in the bibliography (Angeline et al., 1994; García-Pedrajas et al., 2002; Yao and Liu, 1997). The search begins with an initial population. This population is updated in each generation using a population-update algorithm, and is subject to the evolutionary operations of replication and mutation. Crossover is not used due to its potential disadvantages in evolving artificial networks (Angeline et al., 1994). For this reason, this EA belongs to the evolutionary programming (EP) paradigm. The general structure of the EA is detailed next:

Evolutionary programming algorithm

(1) Generate a random population of size N_P

(2) Repeat until the maximum number of generations

(2a) Apply parametric mutation to the best 10% of individuals. Apply structural mutation to the remaining 90% of individuals

(2b) Calculate the fitness of every individual in the population

(2c) Add best fitness individual of the last generation (elitist algorithm)

(2d) Rank the individuals with respect to their fitness

(2e) Best 10% of population individuals are replicated and substitute the worst 10% of individuals

(3) Select the best individual of the population in the last generation and return it as the final solution

First, the initial population is generated: the algorithm begins with the random generation of a larger number of networks than the number of neurons used during the evolutionary process. $10N_P$ networks are generated, where N_P is the number of individuals of the population to be trained during the evolutionary process. We consider $l(\boldsymbol{\theta})$ as the error function of an individual $f_{PUNN}(\mathbf{x}, \boldsymbol{\theta})$ of the population, g being a PUNN; and then, the fitness measure is a decreasing strictly transformation of the error function $l(\boldsymbol{\theta})$ given by $A(g) = 1/(1 + l(\boldsymbol{\theta}))$, where $0 < A(g) \leq 1$.

The adjustment of both weights and structure of the PUNNs is performed by the complementary action of two mutation operators: parametric and structural mutation. Parametric mutation implies a modification in the coefficients (β_j) and the exponents (w_{ji}) of the model, using a self-adaptive simulated annealing algorithm (Kirkpatrick et al., 1983). Structural mutation modifies the topology of the neural nets, helping the algorithm to avoid local minima and increasing the diversity of the trained individuals. Five structural mutations are applied sequentially to each network: neuron deletion, connection deletion, neuron addition, connection addition and neuron fusion. In order to define the topology of the neural networks generated in the evolution process, three parameters are considered: m, M_E and M_I. They correspond to the minimum and the maximum number of hidden neurons in the whole evolutionary process and the maximum number of hidden neurons in the initialization process respectively. In order to obtain an initial population formed by models simpler than the most complex models possible, parameters must fulfil the condition $m \leq M_I \leq M_E$.

More details about the EA can be found in Martínez-Estudillo et al. (2006, in press).

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

given by

$$f_{\mathrm{LRPU}}(\mathbf{x}, \boldsymbol{\theta}) = \alpha_0 + \sum_{j=1}^{m} \beta_j \prod_{i=1}^{k} x_i^{w_{ji}} \tag{7}$$

where $\boldsymbol{\theta} = (\boldsymbol{\alpha}, \mathbf{W})$, $\boldsymbol{\alpha} = (\alpha_0, \beta_1, \ldots, \beta_m)$ and $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_m)$, with $\mathbf{w}_j = (w_{j1}, w_{j2}, \ldots, w_{jk})$, $w_{ji} \in R$. The coefficients $\mathbf{W}$ are given by the EA, they not being adjusted by the IRLS method. The IRLS method only optimizes the linear part of the model, i.e., the $\boldsymbol{\alpha}$ coefficients.

2.2.3.4. Logistic regression using initial covariates and product units. 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

$$f_{\mathrm{LRIPU}}(\mathbf{x}, \boldsymbol{\theta}) = \alpha_0 + \sum_{i=1}^{k} \alpha_i x_i + \sum_{j=1}^{m} \beta_j \prod_{i=1}^{k} x_i^{w_{ji}} \tag{8}$$

where $\boldsymbol{\theta} = (\boldsymbol{\alpha}, \mathbf{W})$, $\boldsymbol{\alpha} = (\alpha_0, \alpha_1, \ldots, \alpha_k, \beta_1, \ldots, \beta_m)$ and $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_m)$, with $\mathbf{w}_j = (w_{j1}, w_{j2}, \ldots, w_{jk})$, $w_{ji} \in R$. The values adjusted with IRLS correspond to the $\boldsymbol{\alpha}$ vector, the coefficients $\mathbf{W}$ being given by the EA.

Finally, in order to reduce the high amount of variables of these LRIPU models, a backward stepwise procedure is used. The method starts with the full model with all the covariates, initial and PU, 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, i.e., 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 selected covariate is deleted if this does not reduce the fit. If none of the covariates is deleted, the second least significant covariate is considered following the procedure previously described. The procedure ends when all the tests for each covariate provide p-values smaller than the fixed significance level, or when none of the two chosen covariates is deleted.

2.2.4. Model development and evaluation

To start processing data, each of the input variables was scaled in the rank [0.1, 0.9]. These bounds in the PUNN models were chosen to avoid inputs values close to 0 that can result in very large values of the function for negative exponents and to avoid dramatic changes in the outputs of the network when there are weights with large values (especially in the exponents). The new scaled variables were named R^*, G^*, B^* and NIR^*. For example, R^* is calculated as follows:

$$R^* = \frac{R - R_{\min}}{R_{\max} - R_{\min}} + 0.1 \tag{9}$$

where R is the red digital value and $R_{\min}$ and $R_{\max}$ are the minimum and maximum values in the whole dataset, respectively.

All the models were evaluated using their prediction error, defined by the correctly classified rate (CCR), over the training ($\mathrm{CCR_T}$) and the generalization ($\mathrm{CCR_G}$) datasets, which are given by the following expression:

$$\mathrm{CCR_D} = \frac{1}{N} \sum_{n=1}^{n_T} I(C(\mathbf{x}_n) = y_n) \tag{10}$$

where $I(\cdot)$ is the zero-one loss function, n_T the number of pattern of the evaluated dataset D, y the binary outcome variable (weed presence or weed-free) and $\mathbf{x}_i$ is the vector of covariates for each i-th pixel. A good classifier tries to achieve the highest possible CCR in a given problem.

Furthermore, in order to obtain a more complete evaluation of the classification task performed by the different models, the confusion matrix associated with each dataset and model was derived. A confusion matrix is a visualization tool typically used in supervised learning (Provost and Kohavi, 1998). The output is compared to the observed outcome (event or non-event, i.e., weed or weed-free pixel), and assigned one of the four possible situations: (i) true negative (TN) when negative cases are correctly identified by the classifier; (ii) false positive (FP) when the classifier incorrectly identifies a non-event case as an event; (iii) false negative (FN) when the classifier incorrectly identifies an event case as a non-event; and (iv) true positive (TP) when the positives cases are correctly identified. For a given number of cases (n_T) in a dataset D, these indexes are inserted into a 2×2 confusion or contingency matrix ($M(g)$) as

$$M(g) = \begin{pmatrix} n_{\mathrm{TN}} & n_{\mathrm{FP}} \\ n_{\mathrm{FN}} & n_{\mathrm{TP}} \end{pmatrix} \tag{11}$$

where $n_T = n_{\mathrm{TN}} + n_{\mathrm{FP}} + n_{\mathrm{FN}} + n_{\mathrm{TP}}$. The diagonal corresponds to the correctly classified patterns and the off-diagonal to the mistakes in the classification task. In this way, the $\mathrm{CCR_D}$ can also be expressed as $\mathrm{CCR_D} = (n_{\mathrm{TP}} + n_{\mathrm{TN}})/n_T$. Two other additional measures are commonly derived from a confusion matrix for evaluating the classifier: the FP rate (or Type I error) and the FN rate (or Type II error). The FP rate is the proportion of negative instances that were erroneously reported as being positive and the false negative rate is the proportion of positive instances that were erroneously reported as negative. Their expressions are the following:

$$\mathrm{FP} = \frac{n_{\mathrm{FP}}}{n_{\mathrm{TN}} + n_{\mathrm{FP}}}, \qquad \mathrm{FN} = \frac{n_{\mathrm{FN}}}{n_{\mathrm{FN}} + n_{\mathrm{TP}}} \tag{12}$$

In this way, the FP and FN rates can also be expressed as $\mathrm{FP} = 1 - \mathrm{CCR_N}$ and $\mathrm{FN} = 1 - \mathrm{CCR_P}$, where $\mathrm{CCR_N}$ and $\mathrm{CCR_P}$ are the CCR values obtained when considering only negative and positive cases of dataset D, respectively, that is:

$$\mathrm{CCR_N} = \frac{n_{\mathrm{TN}}}{n_{\mathrm{TN}} + n_{\mathrm{FP}}}, \qquad \mathrm{CCR_P} = \frac{n_{\mathrm{TP}}}{n_{\mathrm{FN}} + n_{\mathrm{TP}}} \tag{13}$$

SPSS 13.0 software for Windows (SPSS 13.0, SPSS Inc. Chicago, IL) was used for applying the IRLS algorithm in LR, LRPU and LRIPU methodologies. The different EPUNN experiments were conducted using a software package developed in JAVA by the authors, as an extension of the JCLEC framework (http://jclec.sourceforge.net/) (Ventura et al., 2008). This software package is available in the non-commercial JAVA tool

Table 1 – Non-common parameters values of the evolutionary product-unit neural network (EPUNN) algorithm for each dataset, including number of generations, and topology defining parameters

Location	Date	#Gen.	m	M_I	M_E
Matabueyes	(mid-May)	450	2	3	4
	(mid-June)	600	2	2	3
	(mid-July)	550	3	3	4
Santa Cruz	(mid-May)	400	2	2	3
	(mid-June)	300	2	2	3
	(mid-July)	400	2	2	3

#Gen., maximum number of generations; m, minimum number of hidden neurons; M_I, maximum number of hidden neurons during the initialization process; M_E, maximum number of hidden neurons.

named KEEL (http://www.keel.es) (Alcala-Fdez et al., in press). Weights are assigned using a uniform distribution defined throughout two intervals, [−5, 5] for connections between the input layer and hidden layer and, for all kinds of neurons, [−10, 10] for connections between the hidden layer and the output layer. The values of non-common parameters adjusted through a trial-and-error process are shown in Table 1.

The sequential minimal optimization (SMO) algorithm is a java implementation of the SVM methodology which was used in our experiments. This algorithm, together with an implementation of LMT, are available as part of the WEKA machine learning workbench (Witten and Frank, 2005). We considered WEKA release 3.4.0 and both methods were applied to the datasets evaluated in this paper, using their default parameter values.

3. Results

The models obtained with the different evolutionary and statistical methodologies were evaluated using their prediction error over the training and generalization sets. First of all, the EPUNN methodology was run and the corresponding statistical results are shown in Table 2. As EPUNN is an evolutionary methodology, it is based on randomly generated numbers and this makes it a non-deterministic method. For this reason, the method was run 30 times and the best individual of the final population in each execution was extracted. Table 2 includes the average, the standard deviation, the best and the worst values of the CCR over the training (CCR_T) and the generalization (CCR_G) sets of these 30 models, together with their number of connections. From the analysis of these results, we can conclude that the EPUNN methodology was quite stable in training and generalization accuracy terms, the standard deviation not being very high in any date or location ($S.D._T < 1.86\%$ and $S.D._G < 2.43\%$). The most stable generalization results were obtained in mid-June at Matabueyes ($S.D._G = 0.57\%$) and Santa Cruz ($S.D._G = 0.18\%$), together with the highest accuracy ($mean_G = 97.85\%$ and $mean_G = 98.37\%$). The number of connections of the models obtained was quite low ($mean_{\#Conn.} = 14.5$) compared to the number of connections obtained using other ANNs multi-spectral imagery analysis approaches. For example, Gutiérrez et al. (2008) obtained a PUNN model with 28 connections when predicting sunflower crop from multi-spectral imagery. This fact assures not only more interpretable models but also a better generalization capability.

Once the evolution process was applied, we selected the best training individual of the 30 runs in each location/date and we constructed the corresponding LRPU and LRIPU models, using SPSS statistical software. For comparison purposes, we also applied standard LR, and the confusion matrixes for training and generalization sets corresponding to each of the four models are shown in Table 3. The number of pixels of each target response that are predicted as belonging to a specified response are represented in each location/date, and the CCR is calculated independently for each class (CCR_N and CCR_P), in order to better evaluate the performance of the classifiers. An example of interpretation of the different confusion matrixes is the following: if we consider the generalization set in mid-June at Matabueyes, the LR model correctly classifies 226 ground-truth *R. segetum*-absence ($Y = 0$) pixels and misclassifies the remaining 14 ground-truth *R. segetum*-absence pixels assigning to them the *R. segetum*-presence label ($Y = 1$), and obtaining an independent $CCR_N = 94.2\%$ accuracy in the *R. segetum*-absence class. At the same location/date, the LR model correctly classifies 228 ground-truth *R. segetum*-presence pixels ($Y = 1$) and misclassifies 12 ground-truth *R. segetum*-presence pixels assigning to them the *R. segetum*-absence label ($Y = 0$), and obtaining an independent $CCR_P = 95.0\%$ accuracy in the *R. segetum*-presence class. The LR model finally results in a global $CCR_G = 94.6\%$.

Table 2 – Accuracy and number of connection statistical results over 30 runs of the evolutionary product-unit neural network (EPUNN) methodology

Location	Date	CCR_T (%)				CCR_G (%)				#Conn.	
		Mean	S.D.	Best	Worst	Mean	S.D.	Best	Worst	Mean	S.D.
Matabueyes	(mid-May)	70.81	0.91	72.41	69.20	69.86	0.70	71.67	68.33	13.77	1.50
	(mid-June)	97.67	0.36	98.21	96.96	97.85	0.57	98.75	96.67	9.23	0.97
	(mid-July)	77.96	1.71	80.18	74.46	77.28	2.43	80.83	70.83	14.50	1.36
Santa Cruz	(mid-May)	73.96	1.86	77.48	70.74	75.51	1.37	78.43	72.81	10.63	1.30
	(mid-June)	99.15	0.11	99.33	98.94	98.37	0.18	98.88	97.98	11.33	0.96
	(mid-July)	83.96	0.47	84.79	83.06	83.18	0.54	84.04	82.02	9.67	1.03

CCR_T, correctly classified rate in the training set; CCR_G, correctly classified rate in the generalization set; #Conn., number of connections of the models; S.D., standard deviation.

Table 3 – Confussion matrix obtained by the different models at each location/date, global accuracy and independent accuracy for each class (*Y* = 0, *R. segetum* absence, and *Y* = 1, *R. segetum* presence)

Location	Phen. stage (date)	Target response	Training			Generalization		
			Predicted response		CCR (%)	Predicted response		CCR (%)
			Y = 0	Y = 1		Y = 0	Y = 1	
Matabueyes	Vegetative (mid-May)	Y = 0	384 352 (383)[394]	176 208 (177)[166]	68.5 62.9 (68.4)[70.4]	164 148 (164)[168]	76 92 (76) [72]	68.3 61.7 (68.3) [70.0]
		Y = 1	133 171 (136)[141]	427 389 (424)[419]	76.2 69.5 (75.7)[74.8]	65 69 (67) [69]	175 171 (173)[171]	72.9 71.3 (72.1)[71.3]
		CCR (%)			72.4 66.2 (72.1)[72.6]			**70.6** 66.5 (70.2)[**70.6**]
	Flowering (mid-June)	Y = 0	547 529 (547)[552]	13 31 (13) [8]	97.7 94.5 (97.7)[98.6]	236 226 (237)[238]	4 14 (3)[2]	98.3 94.2 (98.8)[99.2]
		Y = 1	7 30 (9) [11]	553 530 (551)[549]	98.8 94.6 (98.4) [98.0]	2 12 (2) [2]	238 228 (238)[238]	99.2 95.0 (99.2)[99.2]
		CCR (%)			98.2 94.6 (98.0)[98.3]			98.7 94.6 (99.0)[**99.2**]
	Senescence (mid-July)	Y = 0	443 296 (443)[447]	117 264 (117)[113]	79.1 52.9 (79.1)[79.8]	195 138 (425)[189]	45 102 (55)[51]	81.2 57.5 (88.5)[78.8]
		Y = 1	105 131 (111)[117]	455 429 (449)[443]	81.2 76.6 (80.2)[79.1]	52 60 (53)[50]	188 180 (187)[190]	78.3 75.0 (77.9)[79.2]
		CCR (%)			80.1 64.7 (79.6)[79.5]			**79.8** 66.3 (79.6) [79.0]
Santa Cruz	Vegetative (mid-May)	Y = 0	362 353 (362)[361]	158 167 (158)[159]	69.6 67.9 (69.6)[69.4]	159 143 (159)[157]	63 79 (63)[65]	71.6 64.4 (71.6)[70.7]
		Y = 1	76 186 (73)[77]	443 333 (446)[442]	85.4 64.2 (85.9)[85.2]	33 80 (33)[35]	190 143 (190)[188]	85.2 64.1 (85.2)[84.3]
		CCR (%)			77.5 66.0 (77.8)[77.3]			**78.4** 64.3 (**78.4**)[77.5]
	Flowering (mid-June)	Y = 0	515 514 (515)[514]	5 6 (5)[6]	99.0 98.8 (99.0) [98.8]	219 219 (219)[219]	3 3 (3)[3]	98.6 98.6 (98.6)[98.6]
		Y = 1	2 4 (1)[2]	517 515 (518)[517]	99.6 99.2 (99.8)[99.6]	4 4 (4)[3]	219 219 (219)[220]	98.2 98.2 (98.2)[98.7]
		CCR (%)			99.3 99.0 (99.4)[99.2]			98.4 98.4 (98.4)[**98.7**]
	Senescence (mid-July)	Y = 0	390 368 (392)[391]	130 152 (128)[129]	75.0 70.8 (75.4)[75.2]	163 150 (165)[166]	59 72 (57)[56]	73.4 67.6 (74.3)[74.8]
		Y = 1	28 87 (29)[30]	491 432 (490)[489]	94.6 83.2 (94.4)[94.2]	16 37 (16)[14]	207 186 (207)[209]	92.8 83.4 (92.8)[93.7]
		CCR (%)			84.8 77.0 (84.9)[84.7]			83.1 75.5 (83.6)[**84.3**]

EPUNN results are presented in regular font, LR results in italic font, LRPU between parentheses and LRIPU between square brackets. The best methodology is presented in bold face.
Phen., phenological; EPUNN, evolutionary product-unit neural networks; LR, logistic regression; LRPU, logistic regression using product units; LRIPU, logistic regression using initial covariates and product units.

Table 4 – Expression of the probability equation associated to the different models

Location	Method	#Param.	Best models
Matabueyes	EPUNN	8	$P = 1/(1 + \exp(-(-0.424 + 75.419(G^{4.633}) + 0.322(R^{-1.888}) + 14.990(B^{3.496}\ G^{-3.415}))))$
	LR	4	$P = 1/(1 + \exp(-(-0.694 + 8.282(B) - 63.342(G) - 11.402(R))))$
	LRPU	8	$P = 1/(1 + \exp(-(-17.227 + 143.012(G^{4.633}) + 0.636(R^{-1.888}) + 23.021(B^{3.496}\ G^{-3.415}))))$
	LRIPU	9	$P = 1/(1 + \exp(-(18.027 + 130.674(B) - 133.662(G) - 29.346(R) + 353.147(G^{4.633}) - 3.396(B^{3.496}\ G^{-3.415}))))$
Santa Cruz	EPUNN	9	$P = 1/(1 + \exp(-(6.114 - 1.505(R^{-1.246}) - 25(G^{3.722}\ R^{1.867}) - 0.311(B^{2.665}\ N^{-3.875}))))$
	LR	4	$P = 1/(1 + \exp(-(-3.240 - 5.1(B) + 8.623(R) + 3.429(N))))$
	LRPU	9	$P = 1/(1 + \exp(-(6.803 - 1.682(R^{-1.246}) - 30.537(G^{3.722}\ R^{1.867}) - 0.317(B^{2.665}\ N^{-3.875}))))$
	LRIPU	11	$P = 1/(1 + \exp(-(1.436 + 5.427(G) + 3.169(N) - 1.268(R^{-1.246}) - 41.646(G^{3.722}\ R^{1.867}) - 0.239(B^{2.665}N^{-3.875}))))$

EPUNN, evolutionary product-unit neural networks; LR, logistic regression; LRPU, logistic regression using product units; LRIPU, logistic regression using initial covariates and product units; #Param., number of parameters of the models; *P*: probability of *R. segetum* presence; R^*, G^*, B^* and NIR^*: digital values of red (R), green (G), blue (B) and near infrared (NIR) bands. Scaled variables R^*, G^*, B^* and $NIR^* \in [0.1, 0.9]$.

As shown in Table 3, LRIPU and EPUNN models were the better performing ones, achieving a very high accuracy in the generalization set in mid-June with 99.2% and 98.7% for the LRIPU model in Matabueyes and Santa Cruz, respectively. Moreover, LRIPU Types I and II errors are very low with values of $FP = (1 - CCR_N) = 0.08\%$ and $FN = (1 - CCR_P) = 0.08\%$ for Matabueyes and $FP = (1 - CCR_N) = 0.14\%$ and $FN = (1 - CCR_P) = 0.13\%$ for Santa Cruz. The LR model is not able to reflect the nonlinear relationships between input variables, necessary for performing a realistic classification task. The mathematical expressions of the different models are presented in Table 4, all of them being relatively simple, especially if we compare these expressions with the expressions that could be obtained using more traditional MLP Sigmoidal Units.

In Table 5, we show the performance obtained by the models developed in this work as compared to the performance obtained by SVM and LMT methodologies, two of the more recent and accurate classifier proposals in the machine learning community. As we can see in these results, SVM did not achieve a good performance in the different datasets resulting in very low generalization ability, and LMT obtains an accuracy very close to that obtained by the methodologies presented in this paper. However, EPUNN and LRIPU outperformed both SVM and LMT in five out of the six databases considered. Moreover, it is important to note that SVM produces a dichotomous classifier, which only gives a binary prediction in each pixel. The rest of models (including LMT) predict a probability, differentiating in this way pixels with a similar prediction (*R. segetum* presence or absence).

Finally, the classification task was generalized to the complete fields of study and the weed probabilities predicted by the best performing model (LRIPU) and the worst performing model (LR) were assigned to the corresponding pixels of a new map using ENVI (ENVI 4.0 software, Research Systems Inc.). We selected mid-June at Matabueyes because the best *R. segetum* discrimination results were obtained using LRIPU in this date. In order to visualize the differences in accuracy on *R. segetum* discrimination at Santa Cruz, we selected mid-July because the performances of both LR and LRIPU methods in mid-June were very similar to each other (98.4% and 98.7%, respectively, Table 5). In Figs. 2 and 3, the corresponding maps are represented. We can conclude from the analysis of the maps that LRIPU model was more precise in the prediction of the probability of *R. segetum* presence, discriminating more clearly the different patches zones of the study fields.

4. Discussion

Classification accuracy of *R. segetum* patches in the sunflower crop was consistently affected by the dates when aerial images were taken, LRIPU being the most accurate method in both locations. All the algorithms for classification studied provided higher accuracies in images taken in this order:

Table 5 – Comparative performance of the probability equations for the different approaches proposed and for two other recent state-of-the-art methodologies: logistic model trees and support vector machines

Location	Date	CCR_T (%)						CCR_G (%)					
		EPUNN	LR	LRPU	LRIPU	SVM	LMT	EPUNN	LR	LRPU	LRIPU	SVM	LMT
Matabueyes	(mid-May)	72.4	66.2	72.1	72.6	67.2	76.4	70.6	66.5	70.2	70.6	65.6	70.1
	(mid-June)	98.2	94.6	98.0	98.3	93.6	98.8	98.7	94.6	99.0	99.2	93.7	98.7
	(mid-July)	80.1	64.7	79.6	79.5	59.8	91.3	79.8	66.3	79.6	79.0	59.0	82.0
Santa Cruz	(mid-May)	77.5	66.0	77.8	77.3	67.0	89.4	78.4	64.3	78.4	77.5	65.2	77.3
	(mid-June)	99.3	99.0	99.4	99.2	97.6	99.2	98.4	98.4	98.4	98.7	97.0	98.2
	(mid-July)	84.8	77.0	84.9	84.7	78.4	86.2	83.1	75.5	83.6	84.3	77.5	83.8

CCR_T, correctly classified rate in the training set; CCR_G, correctly classified rate in the generalization set; EPUNN, evolutionary product-unit neural networks; LR, logistic regression; LRPU, logistic regression using product units; LRIPU, logistic regression using initial covariates and product units; SVM, support vector machines; LMT, logistic model trees.

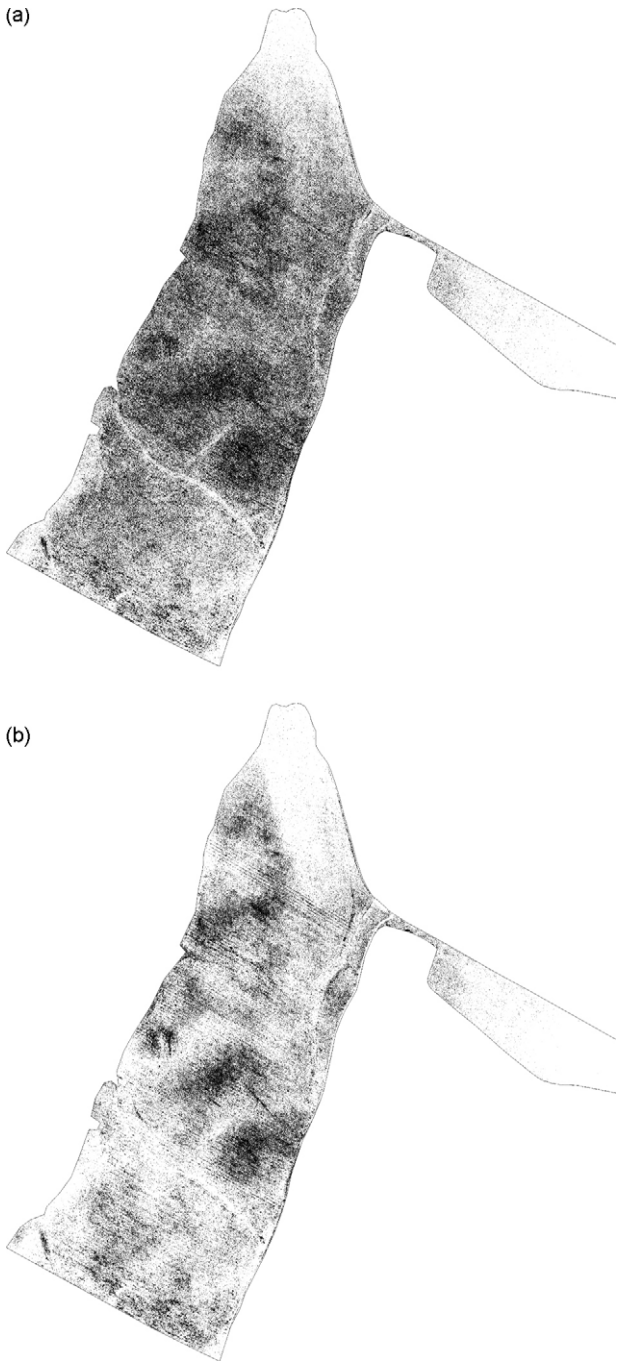

Fig. 2 – *R. segetum* probability maps for Matabueyes in mid-June. The probability ranges from $p = 0$ (*R. segetum* free pixels), represented in white colour, to $p = 1$ (*R. segetum* presence pixels), represented in dark green. (a) LR: *R. segetum* probability map. (b) LRIPU: *R. segetum* probability map.

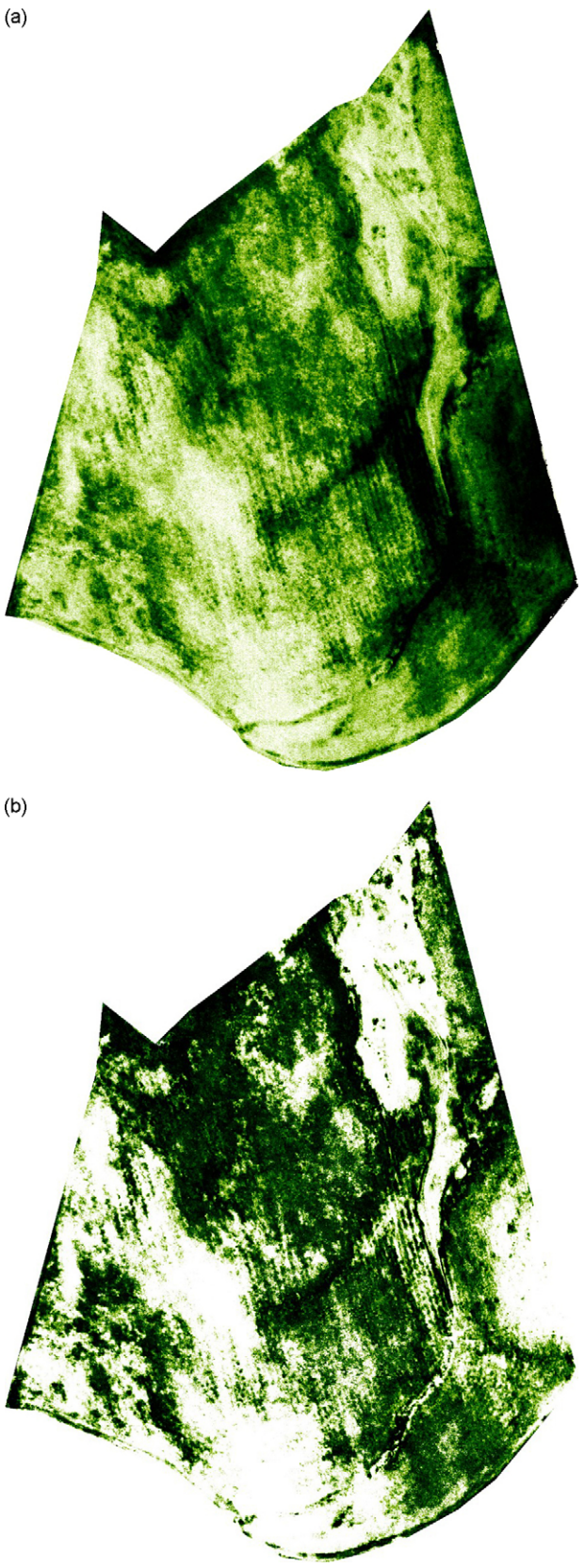

Fig. 3 – *R. segetum* probability maps for Santa Cruz in mid-July. The probability ranges from $p = 0$ (*R. segetum* free pixels, represented in white colour) to $p = 1$ (*R. segetum* presence pixels, represented in dark green). (a) LR: *R. segetum* probability map. (b) LRIPU: *R. segetum* probability map.

mid-June (corresponding to the flowering phase) > mid-July (corresponding to the senescent phase) > mid-May (corresponding to the vegetative phase). None of the classification methodologies used over the mid-May and mid-July imagery yielded accuracies higher than 84.3%, and, therefore it should be recommended not taking any images in the corresponding

phenological stages to discriminate *R. segetum* patches in sunflower. By contrast, all the classification algorithms studied in mid-June discriminated the weed with accuracies higher than 98%, except for LR with 94.5% of accuracy, being LRIPU the most accurate. This differential accuracy is in agreement with previous results of Peña-Barragán et al. (2007), where they already discussed the influence of *R. segetum* and sunflower phenological stages in these results.

One of the most interesting results from this study is that our methods minimized the false detection of presence (Type I error) or absence (Type II error) of weed in both locations, finding that the LRIPU method selected as optimal provided classification accuracies that were very high and similar in both locations. By contrast, pixel-based image classification using traditional vegetation indices approaches, previously reported by Peña-Barragán et al. (2007), produced maximum accuracies of 85% and 99% with R/B index at Matabueyes and Santa Cruz, respectively. The low classification accuracy obtained at Matabueyes by using conventional vegetation indices approaches, or those herein obtained in mid-May and in mid-July for image classification, may both underestimate the presence of *R. segetum* ($Y = 1$) patches and thus misclassify the total surface of weed infestation (omission error or Type II error), or overestimate the presence of *R. segetum* and therefore provide a conservative estimation of weed infestation (commission error or Type I error). From an agronomic point of view, this has important implications because any control strategy should not assume the risk of applying an unnecessary herbicide on soil or crop or of allowing *R. segetum* to go untreated. In this last case, two *R. segetum* uncontrolled plants per m^2 would result in an average sunflower yield reduction of about 32% as previously reported by Carranza-Cañadas et al. (1995).

Every algorithm, excepting SVM, estimates the probability of occurrence of *R. segetum*. This is beneficial for control strategies since the models not only predict a dichotomous outcome, i.e. presence or absence of weed, but also the probability associated with the occurrence of this event. In our case, we simply checked when this probability was greater or lower than a probability of 0.5. We applied this general assumption so that pixels ranging from 0.5 to 1 were classified as presence of *R. segetum*, and the remaining ones as absence of weed. However, when considering a wider probability threshold for presence of *R. segetum*, e.g. from 0.3 to 1, a higher amount of pixels would be classified as presence of weed and a more conservative weed map would be achieved. So, if the probability range associated to the binary outcome is changed or adjusted according to spatial distribution of *R. segetum*, a harder classification could be obtained in dates with worst classification results. For example, spatial pattern of *R. segetum* patches have shown positive correlation with field elevation, which means that this weed is dominant in areas with high elevation values (Jurado-Expósito et al., 2005b). Taking into account this information, the probability threshold associated to the first class (i.e., *R. segetum* presence) could be adjusted by decreasing it (<0.5) in higher parts of the fields and increasing it (>0.5) in the lower ones. Consequently, this more adjusted probability interval than the one selected in our study could improve our results in mid-May and mid-July and possibly, these dates could also be recommended for generating weed maps.

A key component of population dynamics of weeds is that grass and broadleaf weed infestations are often persistent and relatively stable in location year to year (Barroso et al., 2004b; Jurado-Expósito et al., 2004). Thus, late-season weed detection maps can be used to design site-specific control in subsequent years. However, to take full advantage of our results, next investigations could explore the potential of high resolution satellite imagery such as QuickBird and the coming images of WorldView II for mapping *R. segetum* patches in sunflower in larger areas (of at least over 64 km^2). Jurado-Expósito et al. (2005b), applying geostatistical techniques, demonstrated that the extension of *R. segetum* patches in sunflower was at least 9 m. According to these results, the classification accuracy herein presented, and taking into consideration the QuickBird imagery provides four channels (B, G, R and NIR) of multispectral wavebands with 2.4 m or 2.8 m of spatial resolution, it would be possible to successfully map this weed on a large surface, provided that QuickBird has been proved to be a useful data source for mapping invasive plant species (Tsai and Chou, 2006).

5. Conclusions

This study demonstrated the capability of LR and PUNN combination models to analyze multispectral imagery for predicting *R. segetum* presence probability and mapping *R. segetum* patches in the different fields of study. LRIPU and EPUNN models provided better accuracy than linear LR models both in training sets and generalization sets. Excellent generalization accuracies were obtained through the application of the best performing model (LRIPU) in mid-June at both locations. Our study corroborated that the phenological stages/dates when aerial images were taken significantly affect the accuracy in discriminating *R. segetum* patches in sunflower crop, decreasing in efficiency in the two fields considered in the following order: mid-June (corresponding to the flowering phase) > mid-July (corresponding to the senescent phase) > mid-May (corresponding to the vegetative phase). Therefore, reliable mapping of *R. segetum* in sunflower should be generated using images around mid-June, in order to apply a more efficient site-specific control in subsequent years. The maps generated using LRIPU models were more precise than those generated by the linear LR model. Moreover, two advanced methodologies (SVM and LMT) were compared to the methodologies herein presented, resulting in lower accuracy for SVM and LMT, in the six locations/dates evaluated and in five datasets, respectively. Granted that, computational requirements for EPUNN were much higher than for LR, SVM, LTM or traditional vegetation indices approaches, those necessary for LRIPU were nearly insignificant. Thus, considering that precision agriculture management requires a great accuracy, we suggest that the criteria for selecting our algorithms or vegetation indices classification should not be based on decreasing computational requirements or complexity, but on the accuracy of discrimination.

Acknowledgements

This work has been partially financed by TIN2005-08386-C05-02 and AGL-2008-04670-CO3-03, projects of the Spanish Inter-

Ministerial Commission of Science and Technology and FEDER funds. The research of P.A. Gutiérrez has been subsidized by the FPU Predoctoral Program (Spanish Ministry of Education and Science), grant reference AP2006-01746.

REFERENCES

Alcala-Fdez, J., Sánchez, L., García, S., del Jesús, M.J., Ventura, S., Garrell, J.M., Otero, J., Romero, C., Bacardit, J., Rivas, V.M., Fernández, J.C., Herrera, F., in press. KEEL: a software tool to assess evolutionary algorithms for data mining problems. Soft Computing: A Fusion of Foundations, Methodologies and Applications, doi:10.1007/s00500-008-0323-y.

Angeline, P.J., Saunders, G.M., Pollack, J.B., 1994. An evolutionary algorithm that constructs recurrent neural networks. IEEE Transaction on Neural Networks 5 (1), 54–65.

Atkinson, P.M., Tatnall, A.R.L., 1997. Neural networks in remote sensing. International Journal of Remote Sensing 18 (4), 699–709.

Barroso, J., Fernández-Quintanilla, C., Maxwell, B.D., Rew, L., 2004a. Simulating the effects of spatial weed pattern and resolution of mapping and spraying on economics of site-specific management. Weed Research 44 (3), 413–488.

Barroso, J., Fernández-Quintanilla, C., Ruiz, D., Hernaiz, P., Rew, L., 2004b. Spatial stability of *Avena sterilis* spp. ludoviciana under annual applications of low rates of imazamethabenz. Weed Research 44 (3), 178–186.

Bishop, C.M., 1995. Neural Networks for Pattern Recognition. Oxford University Press, UK.

Brown, R.B., Noble, S.D., 2005. Site-specific weed management: sensing requirements-what do we need to see? Weed Science 53 (2), 252–258.

Carranza-Cañadas, P., Saavedra, M., García-Torres, L., 1995. Competition of *Ridolfia segetum* and sunflower. Weed Research 35 (5), 369–375.

Cetin, H., Pafford, J.T., Mueller, T.G., 2005. Precision agriculture using hyperspectral remote sensing and GIS. In: Proceedings of the 2nd International Conference on Recent Advances in Space Technologies (RAST2005), pp. 70–77.

Cristianini, N., Shawe-Taylor, J., 2000. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, UK.

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

Flowers, M., Weisz, R., Heiniger, R., 2001. Remote sensing of winter wheat tiller density for early nitrogen application decisions. Agronomy Journal 93 (4), 783–789.

Foody, M., Arora, M.K., 1996. Incorporating mixed pixel in the training, allocation and testing stages of supervised classification. Pattern Recognition Letters 17 (13), 1389–1398.

Fraser, R.H., Abuelgasim, T.A., Latifovic, R., 2005. A method for detecting large-scale forest cover change using coarse spatial resolution imagery. Remote Sensing of Environment 95 (4), 414–427.

García-Pedrajas, N., Hervás-Martínez, C., Muñoz-Pérez, J., 2002. Multi-objective cooperative coevolution of artificial neural networks. Neural Networks 15 (10), 1255–1274.

Goel, P.K., Prasher, S.O., Patel, R.M., Landry, J.A., Bonnell, R.B., Viau, A.A., 2003. Classification of hyperspectral data by decision trees and artificial neural networks to identify weed stress and nitrogen status of corn. Computers and Electronics in Agriculture 39 (2), 67–93.

Gutiérrez, P.A., López-Granados, F., Peña-Barragán, J.M., Jurado-Expósito, M., Gómez-Casero, M.T., Hervás-Martínez, C., 2008. 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 60 (2), 122–132.

Hastie, T., Tibshirani, R.J., Friedman, J., 2001. The elements of statistical learning. In: Data Mining, Inference and Prediction. Springer, Berlin.

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

Hervás-Martínez, C., Martínez-Estudillo, F.J., Carbonero-Ruz, M., 2008. Multilogistic regression by means of evolutionary product-unit neural networks. Neural Networks 21, 951–961.

Hosmer, D.W., Lemeshow, S., 1989. Applied Logistic Regression. Wiley, New York.

Ismail, A., Engelbrecht, A.P., 2002. Pruning product unit neural networks. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN'2002), Honolulu, USA, pp. 257–262.

Janson, D.J., Frenzel, J.F., 1993. Training product unit neural networks with genetic algorithms. IEEE Expert 8 (5), 26–33.

Jurado-Expósito, M., López-Granados, F., García-Torres, L., García-Ferrer, A., Sánchez de la Orden, M., Atenciano, S., 2003. Multi-species weed spatial variability and site-specific management maps in cultivated sunflower. Weed Science 51 (3), 319–328.

Jurado-Expósito, M., López-Granados, F., González-Andújar, J.L., García-Torres, L., 2004. Spatial and temporal analysis of *Convolvulus arvensis* L. populations over four growing seasons. European Journal of Agronomy 21 (3), 287–296.

Jurado-Expósito, M., López-Granados, F., González-Andújar, J.L., García-Torres, L., 2005a. Characterizing population growth rate of *Convolvulus arvensis* in wheat-sunflower no tillage systems. Crop Science 45, 2106–2112.

Jurado-Expósito, M., López-Granados, F., Peña-Barragán, J.M., García-Torres, L., 2005b. Weed density predicition with secondary input of DEM information. In: Stafford, J. (Ed.), Proceedings of the Fifth European Conference on Precision Agriculture. Uppsala, Sweden, pp. 115–122.

Kanellopoulos, I., Wilkinson, G.G., 1997. Strategies and best practice for neural network image classification. International Journal of Remote Sensing 18 (4), 711–725.

Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M.P., 1983. Optimization by simulated annealing. Science 220, 671–680.

Koger, H.K., Sahw, R.R., Watson, C.E., Reddy, K.N., 2003. Detecting late-season weed infestation in soybean (Glycine max). Weed Technology 17 (4), 696–704.

Landwehr, N., Hall, M., Frank, E., 2005. Logistic model trees. Machine Learning 59 (1/2), 161–205.

Lenze, B., 1994. How to make sigma–pi neural networks perform perfectly on regular training sets. Neural Networks 7 (8), 1285–1293.

López-Granados, F., Peña-Barragán, J.M., Jurado-Expósito, M., García-Torres, L., 2006. Using remote sensing for identification of late-season grassy weeds patches in wheat (*Triticum durum*) for precision agriculture. Weed Science 54 (2), 346–353.

Magnussen, S., Boudewyn, P., Alfaro, R., 2004. Spatial prediction of the onset of spruce budworm defoliation. The Forestry Chronicle 80 (4), 485–494.

MAPA, 2007. Spanish Ministry of Agriculture, Fisheries and Food. www.mapa.es/es/estadistica/infoestad.html.

Martínez-Estudillo, A.C., Mártínez-Estudillo, F.J., Hervás-Martínez, C., García-Pedrajas, N., 2006. Evolutionary product unit based neural networks for regression. Neural Networks 19 (4), 477–486.

Martínez-Estudillo, F.J., Hervás-Martínez, C., Gutiérrez, P.A., Mártínez-Estudillo, A.C., in press. Evolutionary product-unit

neural networks classifiers. Neurocomputing, available online January 15, 2008, doi:10.1016/j.neucom.2007.11.019.
Mather, P.M., 2000. Computer Processing of Remotely-sensed Images: An Introduction. Wiley, UK.
McCoy, R.M., 2005. Fields Methods in Remote Sensing. The Guilford Press, New York.
McLachlan, G., 1992. Discriminant Analysis and Statistical Pattern Recognition. Wiley, New York.
Mertens, K.C., Verbeke, L.P.C., Ducheyne, E.I., De Wulf, R.R., 2003. Using genetic algorithms in sub-pixel mapping. International Journal of Remote Sensing 24 (21), 4241–4247.
Monserud, R.A., Sterba, H., 1999. Modeling individual tree mortality for Austrian forest species. Forest Ecology and Management 113 (2/3), 109–123.
Moody, A., Gopal, S., Strahler, A.H., 1996. Artificial neural network response to mixed pixels in coarse-resolution satellite data. Remote Sensing of Environment 58 (3), 329–343.
Peña-Barragán, J.M., López-Granados, F., Jurado-Expósito, M., García-Torres, L., 2006. Spectral discrimination of *Ridolfia segetum* and sunflower as affected by phenological stage. Weed Research 46 (1), 10–21.
Peña-Barragán, J.M., López-Granados, F., Jurado-Expósito, M., García-Torres, L., 2007. Mapping *Ridolfia segetum* Moris patches in sunflower (*Helianthus annuus* L.) crop using remote sensing. Weed Research 47 (2), 164–172.
Plant, R.E., 2001. Site-specific management: the application of information technology to crop production. Computers and Electronics in Agriculture 30 (1), 9–29.
Provost, F., Kohavi, R., 1998. On applied research in machine learning. Machine Learning 30 (2/3), 127–132.
Schmitt, M., 2002. On the complexity of computing and learning with multiplicative neural networks. Neural Computation 14 (2), 241–301.
Seelan, S.K., Laguette, S., Casady, G.M., Seielstad, G.A., 2003. Remote sensing applications for precision agriculture: a learning community approach. Remote Sensing of Environment 88 (1/2), 157–169.
Srinivasan, A., 2006. Handbook of Precision Agriculture: Principles and Applications. The Haworth Press, New York.
Thorp, K.R., Tian, L.F., 2004. A review of remote sensing of weeds in agriculture. Precision Agriculture 5 (5), 477–508.
Timmermann, C., Gerhards, R., Kuehbauch, W., 2003. The economic impact of the site specific weed control. Precision Agriculture 4 (3), 249–260.
Tsai, F., Chou, M.J., 2006. Texture augmented analysis of high resolution satellite imagery in detecting invasive plants species. Journal of Chinese Institute of Engineering 29 (4), 581–592.
Uno, Y., Prasher, S.O., Lacroix, R., Goel, P.K., Karimi, Y., Viau, A., Patel, R.M., 2005. Artificial neural networks to predict corn yield from compact airborne spectrographic imager data. Computers and Electronics in Agriculture 47 (2), 149–161.
USDA-NRCS, 1998. Keys to Soil Taxonomy, 8th ed. USDA-NRCS, Washington, DC, USA.
Vanclay, J.K., 1995. Growth models for tropical forests: a synthesis of models and methods. Forest Science 41 (1), 7–42.
Vapnik, V., 1995. The Nature of Statistical Learning Theory. Springer, New York.
Ventura, S., Romero, C., Zafra, A., Osuna, J.A., Hervás-Martínez, C., 2008. JCLEC: a JAVA framework for evolutionary computation. Soft Computing: A Fusion of Foundations, Methodologies and Applications 12 (4), 381–392.
Witten, I.H., Frank, E., 2005. Data Mining: Practical Machine Learning Tools and Techniques, 2nd ed. Morgan Kaufmann, San Francisco.
Wulder, M.A., White, J.C., Bentz, B., Alvarez, M.F., Coops, N.C., 2006. Estimating the probability of mountain pine beetle red-attack damage. Remote Sensing of Environment 101 (2), 150–166.
Xu, M., Watanachaturaporn, P., Varshney, P.K., Arora, M.K., 2005. Decision tree regression for soft classification of remote sensing data. Remote Sensing of Environment 97 (3), 322–336.
Yang, C., Prasher, S.O., Landry, J.A., Ramaswamy, H.S., Ditommaso, A., 2000a. Application of artificial neural networks in image recognition and classification of crop and weeds. Canadian Agriculture Engineering 42 (3), 147–152.
Yao, X., 1999. Evolving artificial neural networks. Proceedings of the IEEE 87 (9), 1423–1447.
Yao, X., Liu, Y., 1997. A new evolutionary system for evolving artificial neural networks. IEEE Transaction on Neural Networks 8 (3), 694–713.
Zhai, Y., Thomasson, J.A., Boggess III, J.E., Sui, R., 2006. Soil texture classification with artificial neural networks operating on remote sensing data. Computers and Electronics in Agriculture 54 (2), 53–68.
Zhan, H.G., Lee, Z.P., Shi, P., Chen, C.Q., Carder, K.L., 2003. Retrieval of water optical properties for optically deep waters using genetic algorithms. IEEE Transactions on Geoscience and Remote Sensing 41 (5), 1123–1128.