Encyclopedia of Artificial Intelligence

Juan Ramón Rabuñal Dopico
University of A Coruña, Spain

Julián Dorado de la Calle
University of A Coruña, Spain

Alejandro Pazos Sierra
University of A Coruña, Spain

Information Science REFERENCE

INFORMATION SCI
Hershey · New York

Director of Editorial Content: Kristin Klinger
Managing Development Editor: Kristin Roth
Development Editorial Assistant: Julia Mosemann, Rebecca Beistline
Senior Managing Editor: Jennifer Neidig
Managing Editor: Jamie Snavely
Assistant Managing Editor: Carole Coulson
Typesetter: Jennifer Neidig, Amanda Appicello, Cindy Consonery
Cover Design: Lisa Tosheff
Printed at: Yurchak Printing Inc.

Published in the United States of America by
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue, Suite 200
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com/reference

and in the United Kingdom by
Information Science Reference (an imprint of IGI Global)
3 Henrietta Street
Covent Garden
London WC2E 8LU
Tel: 44 20 7240 0856
Fax: 44 20 7379 0609
Web site: http://www.eurospanbookstore.com

Library of Congress Cataloging-in-Publication Data

Encyclopedia of artificial intelligence / Juan Ramon Rabunal Dopico, Julian Dorado de la Calle, and Alejandro Pazos Sierra, editors.
p. cm.
Includes bibliographical references and index.
Summary: "This book is a comprehensive and in-depth reference to the most recent developments in the field covering theoretical developments, techniques, technologies, among others"--Provided by publisher.
ISBN 978-1-59904-849-9 (hardcover) -- ISBN 978-1-59904-850-5 (ebook)
1. Artificial intelligence--Encyclopedias. I. Rabunal, Juan Ramon, 1973- II. Dorado, Julian, 1970- III. Pazos Sierra, Alejandro.
Q334.2.E63 2008
006.303--dc22
2008027245

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

Multilogistic Regression by Product Units

P. A. Gutiérrez
University of Córdoba, Spain

C. Hervás
University of Córdoba, Spain

F. J. Martínez-Estudillo
INSA – ETEA, Spain

M. Carbonero
INSA – ETEA, Spain

INTRODUCTION

Multi-class pattern recognition has a wide range of applications including handwritten digit recognition (Chiang, 1998), speech tagging and recognition (Athanaselis, Bakamidis, Dologlou, Cowie, Douglas-Cowie & Cox, 2005), bioinformatics (Mahony, Benos, Smith & Golden, 2006) and text categorization (Massey, 2003). This chapter presents a comprehensive and competitive study in multi-class neural learning which combines different elements, such as multilogistic regression, neural networks and evolutionary algorithms.

The Logistic Regression model (LR) has been widely used in statistics for many years and has recently been the object of extensive study in the machine learning community. Although logistic regression is a simple and useful procedure, it poses problems when is applied to a real-problem of classification, where frequently we cannot make the stringent assumption of additive and purely linear effects of the covariates. A technique to overcome these difficulties is to augment/replace the input vector with new variables, basis functions, which are transformations of the input variables, and then to use linear models in this new space of derived input features. Methods like sigmoidal feed-forward neural networks (Bishop, 1995), generalized additive models (Hastie & Tibshirani, 1990), and PolyMARS (Kooperberg, Bose & Stone, 1997), which is a hybrid of Multivariate Adaptive Regression Splines (MARS) (Friedman, 1991) specifically designed to handle classification problems, can all be seen as different non-linear basis function models. The major drawback of these approaches is stating the typology and the optimal number of the corresponding basis functions.

Logistic regression models are usually fit by maximum likelihood, where the Newton-Raphson algorithm is the traditional way to estimate the maximum likelihood a-posteriori parameters. Typically, the algorithm converges, since the log-likelihood is concave. It is important to point out that the computation of the Newton-Raphson algorithm becomes prohibitive when the number of variables is large.

Product Unit Neural Networks, PUNN, introduced by Durbin and Rumelhart (Durbin & Rumelhart, 1989), are an alternative to standard sigmoidal neural networks and are based on multiplicative nodes instead of additive ones.

BACKGROUND

In the classification problem, measurements x_i, $i = 1,2,...,k$, are taken on a single individual (or object), and the individuals are to be classified into one of J classes on the basis of these measurements. It is assumed that J is finite, and the measurements x_i are random observations from these classes. A training sample $D = \{(\mathbf{x}_n, \mathbf{y}_n); n = 1, 2,...,N\}$ is available, where $\mathbf{x}_n = (x_{1n},...,x_{kn})$ is the vector of measurements taking values in $\Omega \subset \mathbb{R}^k$, and $\mathbf{y}_n$ is the class level of the nth individual. In this chapter, we will adopt the common technique of representing the class levels using a "1-of-J" encoding vector $\mathbf{y} = (y^{(1)}, y^{(2)},...,y^{(J)})$, such as $y^{(l)} = 1$ if $\mathbf{x}$ corresponds to an example belonging to class l and $y^{(l)} = 0$ otherwise. Based on the training sample, we wish to find a decision function $C : \Omega \rightarrow \{1,2,...,J\}$ for classifying the individuals. In other words, C provides a partition, say $D_1,D_2,...,D_J$, of Ω, where D_l corresponds to the lth class, $l = 1,2,...,J$,

and measurements belonging to D_l will be classified as coming from the lth class. A misclassification occurs when a decision rule C assigns an individual (based on measurements vector) to a class j when it is actually coming from a class $l \neq j$.

To evaluate the performance of the classifiers we can define the Correctly Classified Rate by

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

where $I(.)$ is the zero-one loss function. A good classifier tries to achieve the highest possible CCR in a given problem.

Suppose that the conditional probability that $\mathbf{x}$ belongs to class l verifies:

$$p\left(y^{(l)} = 1 \middle| \mathbf{x}\right) > 0,\ l = 1,2,...,J,\ \mathbf{x} \in \Omega,$$

and set the function:

$$f_l(\mathbf{x}, \boldsymbol{\theta}_l) = \log \frac{p\left(y^{(l)} = 1 \middle| \mathbf{x}\right)}{p\left(y^{(J)} = 1 \middle| \mathbf{x}\right)},\ l = 1,2,...,J,\ \mathbf{x} \in \Omega$$

where $\boldsymbol{\theta}_l$ is the weight vector corresponding to class l and $f_J(\mathbf{x}, \boldsymbol{\theta}_J) \equiv 0$. Under a multinomial logistic regression, the probability that $\mathbf{x}$ belongs to class l is then given by

$$p\left(y^{(l)} = 1 \middle| \mathbf{x}, \boldsymbol{\theta}\right) = \frac{\exp f_l\left(\mathbf{x}, \boldsymbol{\theta}_l\right)}{\sum_{j=1}^{J} \exp f_j\left(\mathbf{x}, \boldsymbol{\theta}_j\right)},\ l = 1,2,...,J$$

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

The classification rule coincides with the optimal Bayes' rule. In other words, an individual should be assigned to the class which has the maximum probability, given the vector measurement $\mathbf{x}$:

$$C(\mathbf{x}) = \hat{l},$$

where

$$\hat{l} = \arg\max_l f_l(\mathbf{x}, \hat{\boldsymbol{\theta}}_l),\ \text{for } l = 1,...,J.$$

On the other hand, because of the normalization condition we have:

$$\sum_{l=1}^{J} p\left(y^{(l)} = 1 \middle| \mathbf{x}, \boldsymbol{\theta}\right) = 1,$$

and the probability for one of the classes (in the proposed case, the last) need not be estimated (observe that we have considered $f_J(\mathbf{x}, \boldsymbol{\theta}_J) \equiv 0$).

MULTILOGISTIC REGRESSION AND PRODUCT UNIT NEURAL NETWORKS

Multilogistic Regression by using Linear and Product-Unit models (MLRPU) overcomes the nonlinear effects of the covariates by proposing a multilogistic regression model based on the combination of linear and product-unit models, where the nonlinear basis functions of the model are given by the product of the inputs raised to arbitrary powers. These basis functions express the possible strong interactions between the covariates, where the exponents are not fixed and may even take real values. In fitting the proposed model, the non-linearity of the PUNN implies that the corresponding Hessian matrix is generally indefinite and the likelihood has more local maximum. This reason justifies the use of an alternative heuristic procedure to estimate the parameters of the model.

Non-Linear Model Proposed

The general expression of the proposed model is given by:

$$f_l(\mathbf{x}, \boldsymbol{\theta}_l) = \alpha_0^l + \sum_{i=1}^{k} \alpha_i^l x_i + \sum_{j=1}^{m} \beta_j^l \prod_{i=1}^{k} x_i^{w_{ji}},\quad l = 1,2,...,J-1$$

where

$\boldsymbol{\theta}_l = (\boldsymbol{\alpha}^l, \boldsymbol{\beta}^l, \mathbf{W})$,

$\alpha^l = (\alpha_0^l, \alpha_1^l, ..., \alpha_k^l)$,

$\beta^l = \left(\beta_1^l, ..., \beta_m^l\right)$ and

$\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_m)$,

with $\mathbf{w}_j = (w_{j1}, w_{j2},..., w_{jk})$,

$w_{ji} \in \mathbb{R}$.

As has been stated before, the nonlinear part of $f_l(\mathbf{x},\boldsymbol{\theta}_l)$ corresponds to Product Unit Neural Networks (PUNN), introduced by Durbin and Rumelhart (Durbin & Rumelhart, 1989) and subsequently developed by other authors (Janson & Frenzel, 1993), (Leerink, Giles, Horne & Jabri, 1995), (Ismail & Engelbrecht, 2000), (Martínez-Estudillo, Hervás-Martínez, Martínez-Estudillo & García-Pedrajas, 2006), (Martínez-Estudillo, Martínez-Estudillo, Hervás-Martínez & García-Pedrajas, 2006). Advantages of product-unit based neural networks include increased information capacity and the ability to form higher-order combinations of inputs. They are universal approximators and it is possible to obtain upper bounds of the VC dimension of product unit neural networks that are similar to those obtained for sigmoidal neural networks (Schmitt, 2001). Despite these obvious advantages, product-unit based neural networks have a major drawback. Their training is more difficult than the training of standard sigmoidal based networks (Durbin & Rumelhart, 1989). The main reason for this difficulty is that small changes in the exponents can cause large changes in the total error surface. Hence, networks based on product units have more local minima and a greater probability of becoming trapped in them. It is well-known (Janson & Frenzel, 1993) that back-propagation is not efficient in training product-units. Several efforts have been made to develop learning methods for product units (Leerink, Giles, Horne & Jabri, 1995), (Martínez-Estudillo, Martínez-Estudillo, Hervás-Martínez, & García-Pedrajas, 2006), mainly in a regression context.

Estimation of the Model Coefficients

In the supervised learning context, the components of the weight vectors $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2,..., \boldsymbol{\theta}_{J-1})$ are estimated from the training dataset D. To perform the maximum likelihood (ML) estimation of $\boldsymbol{\theta}$, one can minimize the negative log-likelihood function

$$L(\boldsymbol{\theta}) = -\frac{1}{N}\sum_{n=1}^{N} \log p\left(\mathbf{y}_n \middle| \mathbf{x}_n, \boldsymbol{\theta}\right) =$$

$$= \frac{1}{N}\sum_{n=1}^{N}\left[-\sum_{l=1}^{J} y_n^{(l)} f_l(\mathbf{x}_n, \boldsymbol{\theta}_l) + \log \sum_{l=1}^{J} \exp f_l(\mathbf{x}_n, \boldsymbol{\theta}_l) \right]$$

The error surface associated with the proposed model is very convoluted with numerous local optimums. The non-linearity of the model with respect to the parameters $\boldsymbol{\theta}_l$ and the indefinite character of the associated Hessian matrix do not recommend the use of gradient-based methods to maximize the log-likelihood function. Moreover, the optimal number of basis functions of the model (i.e. the number of hidden nodes in the product-unit neural network) is unknown. Thus, the estimation of the vector parameter $\hat{\theta}$ is carried out by means a hybrid procedure described below.

In this paragraph we make a detailed description of the different aspects of the MLRPU methodology. The process is structured in four steps:

Step 1. We apply an evolutionary neural network algorithm to find the basis functions

$$B(\mathbf{x}, \hat{\mathbf{W}}) = \left\{B_1(\mathbf{x}, \hat{\mathbf{w}}_1), B_2(\mathbf{x}, \hat{\mathbf{w}}_2),..., B_m(\mathbf{x}, \hat{\mathbf{w}}_m)\right\}$$

corresponding to the nonlinear part of $f(\mathbf{x},\boldsymbol{\theta})$. We have to determine the number of basis functions m and the weight vector $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2,..., \mathbf{w}_m)$.

To apply evolutionary neural network techniques, we consider a PUNN with the following structure (Fig. 1): an input layer with a node for every input variable, a hidden layer with several nodes, and an output layer with one node for each category. The activation function of the j-th node in the hidden layer is given by

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

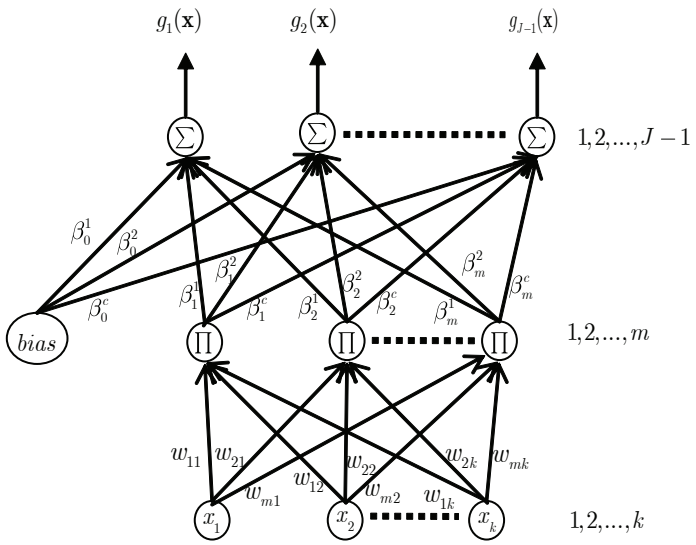
where w_{ji} is the weight of the connection between input node i and hidden node j and $\mathbf{w}_j = w_{j1},..., w_{jk})$. The activation function of the output node l is given by

$$g_l(\mathbf{x}, \beta^l, \Omega) = \beta_0^l + \sum_{j=1}^{m} \beta_j^l B_j(\xi, \omega_j),$$

where β_j^l is the weight of the connection between the hidden node j and the output node l. The transfer function of all output nodes is the identity function.

The weight vector $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2,..., \mathbf{w}_m)$ is estimated by means of an evolutionary programming algorithm detailed in (Hervás-Martínez, Martínez-Estudillo & Gutiérrez, 2006), that optimizes the error function

Figure 1. Model of a product-unit based neural network



given by the negative log-likelihood for N observations associated with the product-unit model:

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

Although in this step the evolutionary process obtains a concrete value for the $\boldsymbol{\beta}$ vector, we only consider the estimated weight vector $\hat{\mathbf{W}} = (\hat{\mathbf{w}}_1, \hat{\mathbf{w}}_2, ..., \hat{\mathbf{w}}_m)$ that builds the basis functions. The value for the $\boldsymbol{\beta}$ vector will be determined in Step 3 together with the $\boldsymbol{\alpha}$ coefficient vector.

Step 2. We consider the following transformation of the input space by adding the nonlinear basis functions obtained by the evolutionary algorithm in step 1:

$$H : \mathbb{R}^k \to \mathbb{R}^{k+m}$$
$$(x_1, x_2, ..., x_k) \to (x_1, x_2, ..., x_k, z_1, ..., z_m)$$

where $z_1 = B_1(\mathbf{x}, \hat{\mathbf{w}}_1), ..., z_m = B_m(\mathbf{x}, \hat{\mathbf{w}}_m)$.

Step 3. We minimize the negative log-likelihood function for N observations:

$$L(\boldsymbol{\alpha},\boldsymbol{\beta}) = \frac{1}{N}\sum_{n=1}^{N}\left[-\sum_{l=1}^{J} y_n^{(l)}(\boldsymbol{\alpha}^l\mathbf{x}_n + \boldsymbol{\beta}^l\mathbf{z}_n) + \log\sum_{l=1}^{J}\exp(\boldsymbol{\alpha}^l\mathbf{x}_n + \boldsymbol{\beta}^l\mathbf{z}_n)\right]$$

where $\mathbf{x}_n = (1, x_{1n}, ..., x_{kn})$. Now, the Hessian matrix of the negative log-likelihood in the new variables $x_1, x_2, ..., x_k, z_1, ..., z_m$ is semidefinite positive. Then, we could apply Newton's method, also known, in this case, as Iteratively Reweighted Least Squares (IRLS). Although there are other methods for performing this optimization, none clearly outperforms IRLS (Minka, 2003). The estimated vector coefficient $\hat{\boldsymbol{\theta}} = (\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}, \hat{\mathbf{W}})$ determines the model:

$$f_l(\mathbf{x},\hat{\theta}) = \hat{\alpha}_0^l + \sum_{i=1}^{k}\hat{\alpha}_i^l x_i + \sum_{j=1}^{m}\hat{\beta}_j^l\prod_{i=1}^{k} x_i^{\hat{w}_{ji}},$$

$$l = 1, 2, ..., J-1$$

Step 4. In order to select the final model, we use a backward stepwise procedure, starting with the full model with all the covariates, initial and PU covariates, and successively prune variables sequentially to the model until further pruning does not improve the fit.

Application to *Remote Sensing*

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

Remote sensing systems can provide a large amount of continuous field information at a reasonable cost. Remote sensed imagery shows great potential in modelling different agronomic parameters for its application in precision farming. One aspect of overcoming the possibility of minimizing the impact of agriculture on environmental quality is the development of more efficient approaches for crop production determination and for site-specific weed management. Potential economic and environmental benefits of site-specific herbicide applications include reduced spray volume, herbicide costs and non-target spraying, and increased control of weeds, (Thompson, Stafford & Miller, 1991), (Medlin, Shaw, Gerard, & LaMastus, 2000).

We face a mapping weed patches problem through the analysis of aerial photographs. Images and data sets have been given by the Precision Farming and Remote Sensing Unit of the Department of Crop Protection, Institute of Sustainable Agriculture (CSIC, Spain), whose members reported previous results in predicting *Ridolfia segetum* Moris patches, (Peña-Barragán, López-Granados, Jurado-Expósito & García-Torres, 2007). The data analyzed correspond to a study conducted in 2003 at the 42 ha-farm Matabueyes, naturally infested by *R. segetum*. At a field study, the nature of 2,400 pixels was determined, being them considered as ground-truth pixels: 800 pixels were classified as *R. segetum* and 800 pixels were classified as *R. segetum* free pixels.

Input variables include the digital values of all bands in each available image, that is: Red (R), Green (G) and Blue (B), for June image, and R, G, B and Near Infrared (NIR) for May and July images. The experimental design was conducted using a stratified holdout cross-validation procedure, where the size of training set was approximately $0.7n$ (1,120 pixels) for the training set and $0.3n$ (480 pixels) for the generalization set, n being the size of the full dataset.

In all experiments, the EA has been applied with the same parameters. SPSS 13.0 software (SPSS, 2005) was used for applying IRLS algorithm and in order to select the more significant variables in the final model, through a backward stepwise procedure.

The models compared in the different experiments are the following: firstly, we extract the best PUNN model of the EA (EPUNN); secondly, we obtain standard Logistic Regression model using initial covariables (LR); finally, we apply Logistic Regression only over basic function extracted from EPUNN model (MLRPU) and over the same basic functions together with initial covariables (MLRLPU).

Results

Performance of each model has been evaluated using the Correctly Classified Ratio in the generalization set (CCR_G). In Table 1 we show the matrix results of classification over train and generalization sets for the three classification problems and the four models proposed (one problem at each date, May, June and July, and four models, EPUNN, LR, MLRPU and MLRLPU). Best CCR_G results are obtained with MLRPU and MLRLPU at May and June, although at July EPUNN model yields the best results. At all dates, differences between standard LR and hybrid LR (MLRPU and MLRLPU) are very significant. Table 2 includes models obtained at the date that leads to better classification results, that is, at June.

Using these models we can obtain the probability of *R.segetum* presence at all pixels of the image, including non ground-truth pixels. Figure 1, 2, 3 and 4 represents weed maps obtained using the four proposed models at June, EPUNN, LR, MLRPU and MLRLPU, respectively. Weed presence probability has been represented using a scale between white (minimum probability, nearly 0) and dark green (maximum probability, nearly 1). From these maps, the agronomical expert can decide what threshold probability value consider to apply herbicide. MLRLPU and MLRPU models clearly differentiate better between high weed density zones and weed free zones, so they have a higher interest from the point of view of intelligent site-specific herbicide application.

Table 1. Classification matrixes (Y=0, R.segetum absence; Y=1, R.segetum presence) at all dates, using best Evolutionary Product Unit Neural Network, EPUNN, Logistic Regression, LR (in italic), Logistic Regression only with Product Units, MLRPU, (in brackets), and Logistic Regression with initial covariables and Product Units, MLRLPU (in squared brackets)

Phenological Stage (Date)	Ground Truth Response	Training			Generalization		
		Predicted Response			Predicted Response		
		Y=0	Y=1	CCR (%)	Y=0	Y=1	CCR (%)
Vegetative	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]
(mid-May)	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	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]
(mid-June)	Y=1	7 *30* (9) [11]	553 *530* (551)[549]	98.8 *94.6* (98.4) [98]	2 *12* (2) [2]	238*228* (238)[238]	99.2 *95* (99.2)[99.2]
	CCR (%)			98.2 *94.6* (98)[98.3]			98.7 *94.6* (99)**[99.2]**
Senescence	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]
(mid-July)	Y=1	105 *131* (111) [117]	455 *429* (449) [443]	81.2 *76.6* (80.2) [79.l]	52 *60* (53) [50]	188 *180* (187) [190]	78.3 *75* (77.9) [79.2]
	CCR (%)			80.1 *64.7* (79.6) [79.5]			**79.8** *66.3* (79.6) [79]

Table 2. Obtained models at June for the determination of R.segetum presence probability (P) in order to obtain weed patches maps

Methodology	#coef.	Model
EPUNN	8	$P = 1/(1+\exp(-(-0.424+75.419(V^{4.633})+0.322(R^{-1.888}) +14.990(A^{3.496} V^{-3.415}))))$
LR	4	$P = 1/(1+\exp(-(-0.694+8.282(A)-63.342(V)-11.402(R))))$
MLRPU	7	$P = 1/(1+\exp(-(-17.227+143.012(V^{4.633}) +0.636(R^{-1.888})+23.021(A^{3.496} V^{-3.415}))))$
MLRLPU	9	$P = 1/(1+\exp(-(18.027+130.674(A)-133.662(V) -29.346(R)+353.147(V^{4.633}) -3.396(B^{3.496} G^{-3.415}))))$

FUTURE TRENDS

Concepts exposed in this chapter offer the possibility of developing new models of multi-class generalized linear regression, by means of considering different types of basis functions (Sigmoidal Units, Radial Basis Functions and Product Units) for the non-linear part of the proposed model. Moreover, future research could include ordinal logistic regression models with different basis functions or probit models with different basis functions.

Figure 1. EPUNN R.segetum presence probability map

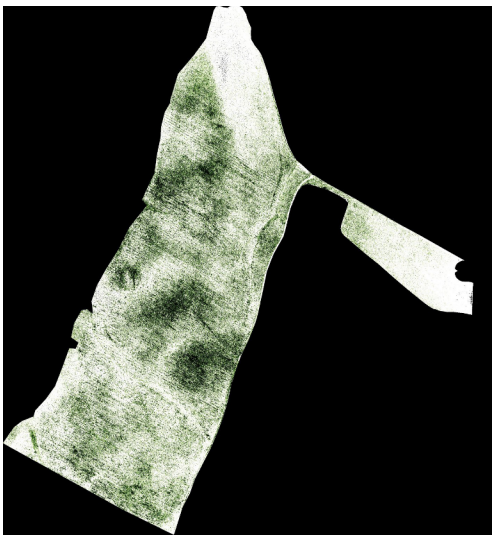


Figure 2. LR R.segetum presence probability map

Figure 3. MLRPU R.segetum presence probability map

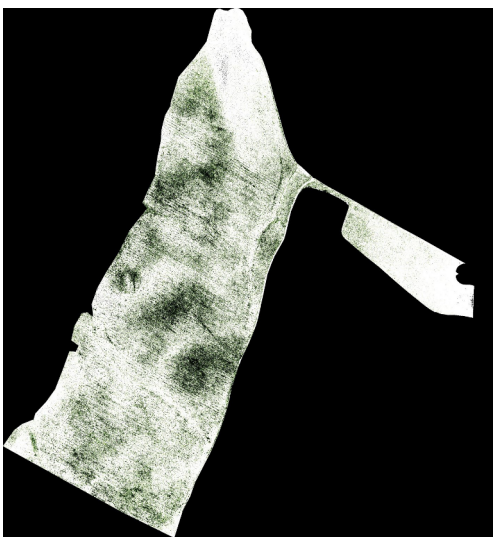
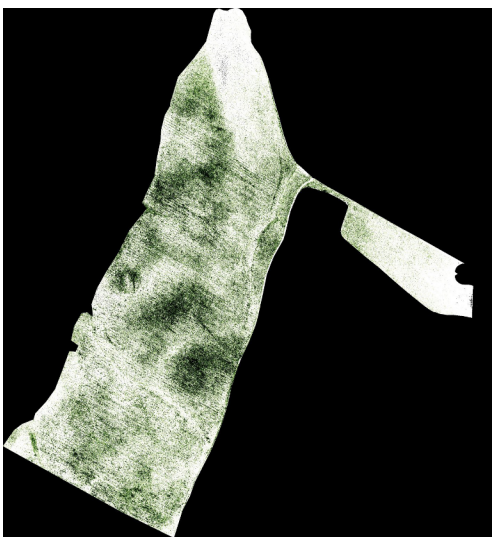


Figure 4. MLRLPU R.segetum presence probability map



CONCLUSION

To the best of our knowledge, the approach presented in this paper is a study in multi-class neural learning which combines three tools used in machine learning research: the logistic regression, the product-unit neural network model and the evolutionary neural network paradigm. Logistic regression is a well-tested statistical approach that performs well in two-class classification and can naturally be generalized to the multi-class case. On the other hand, product-unit neural network models are an alternative to standard sigmoidal neural networks with the ability to capture non-linear interaction between the input variables. Finally, evolutionary artificial neural networks present an interesting platform for optimizing both network performance and architecture simultaneously. The adequate combination of these three elements carried out in several steps in our proposal, provides a competitive methodology to solve classification problems.

REFERENCES

Athanaselis, T., Bakamidis, S., Dologlou, I., Cowie, R., Douglas-Cowie, E., & Cox, C. (2005). ASR for emotional speech: Clarifying the issues and enhancing performance. *Neural Networks*, 18(4), 437-444.

Bishop, M. (1995). *Neural Networks for Pattern Recognition*, Oxford University Press.

Chiang, J.H., (1998). A Hybrid Neural Model in Handwritten Word Recognition. *Neural Networks*, 11(2), 337-346.

Durbin, R., & Rumelhart, D. (1989). Products Units: A computationally powerful and biologically plausible extension to back propagation networks. *Neural Computation*, 1, 133-142.

Friedman, J. (1991). Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1), 1-141.

Hastie, T., Tibshirani, R.J., & Friedman, J. (2001). *The Elements of Statistical Learning. Data mining, Inference and Prediction*, Springer-Verlag.

Hervás-Martínez, C., Martínez-Estudillo, F.J., Gutiérrez, P.A. (2006). Classification by means of evolutionary product-unit neural networks. *IEEE International Joint Conference on Neural Networks (IJCNN06, Vancouver)*, 2834-2841.

Ismail, A., & Engelbrecht, A.P. (2000). Global Optimization Algorithms for Training Product Unit Neural Networks. *International Joint Conference on Neural Networks*, 1, 132-137.

Janson, D.J., & Frenzel, J.F. (1993). Training product unit neural networks with genetic algorithms. *IEEE Expert: Intelligent Systems and Their Applications*, 8(5), 26-33.

Kooperberg, C., Bose, S., & Stone, C.J. (2006). Polychotomous regression. *Journal of the American Statistical Association*, 92, 117-127.

Leerink, L., Giles, C., Horne, B., & Jabri, M. (1995). Learning with product units. *Advances in Neural Information Processing Systems*, 7, 537-544.

Mahony, S., Benos, P.V., Smith, T.J., & Golden, A. (2006). Self-organizing neural networks to support the discovery of DNA-binding motifs. *Neural Networks*, 19(6-7), 950-962.

Martínez-Estudillo, A.C., Martínez-Estudillo, F.J., Hervás-Martínez, C., & García-Pedrajas, N. (2006). Evolutionary Product Unit based Neural Networks for Regression. *Neural Networks*, 19(1), 477-486.

Martínez-Estudillo, A.C., Hervás-Martínez, C., Martínez-Estudillo, F.J., & García-Pedrajas, N. (2006). Hybridation of evolutionary algorithms and local search by means of a clustering method. *IEEE Transaction on Systems, Man and Cybernetics, Part. B: Cybernetics*, 36(3), 534-546.

Medlin, C.R., Shaw, D.R., Gerard, P.D., & LaMastus, F.E. (2000). Using remote sensing to detect weed infestations in Glycine max. *Weed Science*, 48(3), 393-398.

Minka, T.P., (2003). A comparison of numerical optimizers for logistic regression. http://research.microsoft.com/~minka/papers/logreg/

Peña-Barragán, J.M., López-Granados, F., Jurado-Expósito, M., & García-Torres L. (2007). Mapping Ridolfia segetum patches in sunflower crop using remote sensing. Weed Research, 47(2), 164-172.

Schmitt, M. (2001). On the Complexity of Computing and Learning with Multiplicative Neural Networks. *Neural Computation*, 14(24), 241-301.

SPSS (2005). Advanced Models. Copyright 13.0 SPSS Inc. Chicago, IL.

Thompson, J.F., Stafford, J.V. & Miller P.C.H. (1991). Potential for automatic weed detection and selective herbicide application. *Crop Protection*, 10, 254-259.

KEY TERMS

Artificial Neural Networks: A network of many simple processors ("units" or "neurons") that imitates a biological neural network. The units are connected by unidirectional communication channels, which carry numeric data. Neural networks can be trained to find nonlinear relationships in data, and are used in applications such as robotics, speech recognition, signal processing or medical diagnosis.

Evolutionary Computation: Computation based on iterative progress, such as growth or development in a population. This population is selected in a guided random search using parallel processing to achieve the desired solution. Such processes are often inspired by biological mechanisms of evolution.

Evolutionary Programming: One of the four major evolutionary algorithm paradigms, with no fixed structure or representation, in contrast with some of the other evolutionary paradigm. Its main variation operator is the mutation.

Iteratively Reweighted Least Squares (IRLS): Numerical algorithm for minimizing any specified objective function using a standard weighted least squares method such as Gaussian elimination. It is widely applied in Logistic Regression.

Logistic Regression: Statistical regression model for Bernoulli-distributed dependent variables. It is a generalized linear model that uses the logit as its link function. Logistic regression applies maximum likelihood estimation after transforming the dependent into a logit variable (the natural log of the odds of the dependent occurring or not).

Precision Farming: Use of new technologies, such as global positioning (GPS), sensors, satellites or aerial images, and information management tools (GIS) to assess and understand in-field variability in agriculture. Collected information may be used to more precisely evaluate optimum sowing density, estimate fertilizers and other inputs needs, and to more accurately predict crop yields.

Product Unit Neural Networks: Alternative to standard sigmoidal neural networks, based on multiplicative nodes instead of additive ones. Concretely, the output of each hidden node is the product of all its inputs raised to a real exponent.

Remote Sensing: Short or large-scale acquisition of information of an object or phenomenon, by the use of either recording or real-time sensing devices that is not in physical or intimate contact with the object (such as by way of aircraft, spacecraft, satellite, or ship).