Expert Systems with Applications 38 (2011) 12483–12490

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Determination of relative agrarian technical efficiency by a dynamic over-sampling procedure guided by minimum sensitivity

Francisco Fernández-Navarro [a,*], César Hervás-Martínez [a], C. García-Alonso [b], M. Torres-Jimenez [b]

[a] Department of Computer Science and Numerical Analysis, University of Córdoba, Campus de Rabanales, Albert Einstein Building, 3rd Floor, 14071 Córdoba, Spain
[b] Department of Management and Quantitative Methods, ETEA, Escritor Castilla Aguayo 4, 14004 Córdoba, Spain

ARTICLE INFO

Keywords:
Neural networks
Multi-classification
Sensitivity
Accuracy
DEA-Montecarlo
Hybrid algorithm
Imbalanced datasets
Oversampling method
SMOTE
APS

ABSTRACT

In this paper, a dynamic over-sampling procedure is proposed to improve the classification of imbalanced datasets with more than two classes. This procedure is incorporated into a Hybrid algorithm (HA) that optimizes Multi Layer Perceptron Neural Networks (MLPs). To handle class imbalance, the training dataset is resampled in two stages. In the first stage, an over-sampling procedure is applied to the minority class to partially balance the size of the classes. In the second, the HA is run and the dataset is oversampled in different generations of the evolution, generating new patterns in the minimum sensitivity class (the class with the worst accuracy for the best MLP of the population). To evaluate the efficiency of our technique, we pose a complex problem, the classification of 1617 real farms into three classes (efficient, intermediate and inefficient) according to the Relative Technical Efficiency (RTE) obtained by the Monte Carlo Data Envelopment Analysis (MC-DEA). The multi-classification model, named Dynamic Smote Hybrid Multi Layer Perceptron (DSHMLP) is compared to other standard classification methods with an over-sampling procedure in the preprocessing stage and to the threshold-moving method where the output threshold is moved toward inexpensive classes. The results show that our proposal is able to improve minimum sensitivity in the generalization set (35.00%) and obtains a high accuracy level (72.63%).

1. Introduction

Classification problems based on imbalanced training datasets often occur in applications where there are rarely events of interest. That is, the size of interesting minority groups is usually in a rather small proportion in the training dataset (Chawla, Japlowicz, & Kotcz, 2006; Zhao & Huang, 2007). Imbalanced training datasets often results in low classification accuracies for minority classes (He & Garcia, 2009; Sun, Wong, & Kamel, 2009; Torres, Hervás, & García, 2009).

Many techniques are proposed to solve this kind of classification problem through either data (Kubat & Matwin, 1997) or algorithmic levels (Pazzani et al., 1994). In this paper, a dynamic over-sampling procedure (hybrid approach between data and algorithmic solutions) is proposed to improve the classification of imbalanced datasets that have more than two classes. The base over-sampling procedure is the Synthetic Minority Over-sampling Technique (SMOTE) (Chawla, Bowyer, Hall, & Kegelmeyer, 2002). This procedure has been applied in several research fields, for example in predictive microbiology (Fernández-Navarro et al., 2010; Fernández-Navarro, Hervás-Martínez, Cruz, Gutierrez, & Valero, 2011).

This procedure is incorporated into a Hybrid algorithm (HA) (Moscato & Cotta, 2003) that optimizes Multi Layer Perceptron Neural Networks (MLPs). The HA combines an Evolutionary algorithm (EA) (Back, 1996), a clustering process, and a Local Search (LS) procedure. The main objective of this research is, due to the unbalanced class structure (Fernández, Del Jesus, & Herrera, 2009; Sun et al., 2009), to check dynamic oversampling methods, where the class that increases its size is the one that has minimum sensitivity (*MS*) during the evolutive process. The base algorithm was proposed in Fernández-Navarro, Hervás-Martínez, and Gutíerrez (2011).

In recent years, several research projects related to DEA models have been developed, in the area of data mining, of which we highlight the papers by Toloo, Sohrabi, and Nalchigar (2009) and Yeh, Chi, and Hsu (2009). In the research works of Wu (2009) and Tsai, Lin, Cheng, and Lin (2009), the combination of neural networks and DEA models have already been applied successfully.

The performance of the proposed methodology was evaluated in a real problem which consists of classifying 1617 farms into three classes (efficient, intermediate and inefficient) according to Relative Technical Efficiency (RTE) obtained by use of the Monte

* Corresponding author. Tel.: +34 957 21 83 49; fax: +34 957 21 83 60.
E-mail address: i22fenaf@uco.es (F. Fernández-Navarro).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.04.031

Carlo Data Envelopment Analysis (MC-DEA) model on the 65 Agrarian Productive Strategies (APS) or typologies identified in the original database. The classification problem is very complex due to unbalanced class structure and the way in which this has determined the class each farms belongs. (see Section 3.1.1).

This paper is organized as follows: Section 2 describes the base classifier, the learning algorithm and over-sampling approaches; Section 3 explains the experiments carried out and a brief analysis of the database; Section 4 reports on the results obtained with the proposed methods and the results with methodologies used for comparative purposes and, finally, Section 5 summarizes the conclusions of our work.

2. Classification method

2.1. Base classifier

In this paper, we consider standard feed forward MLP with one input layer with independent variables or features, one hidden layer with sigmoidal hidden nodes and one output layer.

Let a coded "1-of-J" outcome variable y, (that is the outcome has the form $\mathbf{y} = (y^{(1)}, y^{(2)}, \ldots, y^{(J)})$, where $y^{(j)} = 1$ if the pattern belongs to the class j, and $y^{(j)} = 0$, otherwise); and a vector $\mathbf{x} = (1, x_1, x_2, \ldots, x_K)$ of input variables, where K is the number of input (we assume that the vector of inputs includes the constant term to accommodate the intercept or bias).

Then, the output layer is interpreted from a point of view of probability which considers the softmax activation function. The activation function of the lth node in the hidden layer is given by:

$$g_l(\mathbf{x}, \boldsymbol{\theta}_l) = \frac{\exp f_l(\mathbf{x}, \boldsymbol{\theta}_l)}{\sum_{l=1}^{J} \exp f_l(\mathbf{x}, \boldsymbol{\theta}_l)}, \quad l = 1, 2, \ldots, J \tag{1}$$

where $g_l(\mathbf{x}, \boldsymbol{\theta}_l)$ is the probability a pattern $\mathbf{x}$ has of belonging to class l, $\boldsymbol{\theta}_l = \left(\beta_0^l, \ldots, \beta_M^l, \mathbf{w_1}, \ldots, \mathbf{w_M}\right)$ is the vector of weights of the output node, M is the number of hidden nodes, $\mathbf{w_j} = \left\{w_0^j, \ldots, w_K^j\right\}$, for $j = 1, \ldots, M$, is the vector of inputs weights of the hidden node j, and $f_l(\mathbf{x}, \boldsymbol{\theta}_l)$ is the output of the output node for pattern $\mathbf{x}$ given by:

$$f_l(\mathbf{x}, \boldsymbol{\theta}_l) = \beta_0^l + \sum_{j=1}^{M} \beta_j^l \sigma\left(w_0^j + \sum_{i=1}^{K} w_i^j x_i\right), \quad \text{for } l = 1, \ldots, J \tag{2}$$

where $\sigma(\cdot)$ is the sigmoidal activation function.

The classification rule $C(\mathbf{x})$ of the MLP model is $C(\mathbf{x}) = \arg\max\{g_l(\mathbf{x}, \boldsymbol{\theta}_l)\}$, this classification rule coinciding with the optimal Bayes' rule.

The best MLP is determined by means of a Hybrid algorithm (HA) that optimizes the error function given by the negative log-likelihood for N observations associated with the MLP model:

$$L^*(\boldsymbol{\theta}) = \frac{1}{N} \sum_{n=1}^{N} \left[-\sum_{l=1}^{J-1} y_n^{(l)} f_l(\mathbf{x}_n, \boldsymbol{\theta}_l) + \log \sum_{l=1}^{J-1} \exp f_l(\mathbf{x}_n, \boldsymbol{\theta}_l) \right] \tag{3}$$

where $y_n^{(l)}$ is equal to 1 if the pattern $\mathbf{x_n}$ belongs to the lth class and equal to 0 otherwise.

2.2. Performance measures: correct classification rate and minimum sensitivity

Minimum sensitivity (MS) and the correct classification rate or accuracy (C) measures associated with a given classifier g are considered to be the performance measures in this work.

Firstly, we have to define the MS and C measurements which are derived from the contingency or confusion matrix M.

$$M = \left\{ n_{ij};\ \sum_{i,j=1}^{J} n_{ij} = N \right\} \tag{4}$$

where J is the number of classes, N is the number of training or testing patterns and n_{ij} represents the number of times the patterns are predicted by classifier g to be in class j when they really belong to class i. The diagonal corresponds to correctly classified patterns and the off-diagonal to mistakes in the classification task.

Let us denote the number of patterns associated with class i by $f_i = \sum_{j=1}^{J} n_{ij}$, $i = 1, \ldots, J$. Let $S_i = n_{ii}/f_i$ be the number of patterns correctly predicted to be in class i with respect to the total number of patterns in class i (sensitivity for class i). Therefore, the sensitivity for class i estimates the probability of correctly predicting a class i example.

From the above quantities the minimum sensitivity (MS) of a classifier g is the minimum value of the sensitivities for each class:

$$MS = \min\{S_i; i = 1, \ldots, J\} \tag{5}$$

The correct classification rate or accuracy (C) is defined as:

$$C = (1/N) \sum_{j=1}^{J} n_{jj} \tag{6}$$

that is, the rate of all correct predictions.

Minimum sensitivity and accuracy measures express two features associated with a classifier: global performance C and the accuracy for the worst classified class S. These measures have been simultaneously taken into account in previous studies (Martínez-Estudillo, Gutiárrez, Hervás-Martínez, & Fernández, 2008), achieving good performance for the classification of imbalanced data. In this paper, the application of dynamic over-sampling techniques improves the sensitivity of the classifier population, without drastically decreasing global accuracy.

2.3. Base evolutionary algorithm

An evolutionary algorithm is applied to estimate the structure and learn the weights of standard MLP neural networks models. The basic framework of the evolutionary algorithm is the following: the search begins with an initial population of neural networks and, in each iteration, the population is updated using a population-update algorithm which evolves both its structure and weights. The population is subject to the operations of replication and mutation. Crossover is not used due to its potential disadvantages in evolving artificial networks (Angeline, Sauders, & Pollack, 1994; Fernández-Navarro, Hervás-Martínez, Gutierrez, & Carboreno, in press; Yao & Liu, 1997).

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

We considered $L^*(\boldsymbol{\theta})$ defined in (3) as the error function of an individual g in the population. The fitness measure needed to evaluate the individuals is a strictly decreasing transformation of the error function $L^*(\boldsymbol{\theta})$ given by

$$A(g) = \frac{1}{1 + L^*(\boldsymbol{\theta})}; \quad 0 < A(g) \leqslant 1 \tag{7}$$

The severity of both structural and parametric mutations depends on the temperature $T(g)$ of the neural network model, defined by:

$$T(g) = 1 - A(g), \quad 0 \leqslant T(g) \leqslant 1 \tag{8}$$

where $A(g)$ is the fitness of the individual or model g.