# Improving the Accuracy of a Two-Stage Algorithm in Evolutionary Product Unit Neural Networks for Classification by Means of Feature Selection

Antonio J. Tallón-Ballesteros[1], César Hervás-Martínez[2],
José C. Riquelme[1], and Roberto Ruiz[3]

[1] Department of Languages and Computer Systems,
University of Seville, Spain
atallon@us.es
[2] Department of Computer Science and Numerical Analysis,
University of Córdoba, Spain
[3] Area of Computer Science,
Pablo de Olavide University, Seville, Spain

**Abstract.** This paper introduces a methodology that improves the accuracy of a two-stage algorithm in evolutionary product unit neural networks for classification tasks by means of feature selection. A couple of filters have been taken into consideration to try out the proposal. The experimentation has been carried out on seven data sets from the UCI repository that report test mean accuracy error rates about twenty percent or above with reference classifiers such as C4.5 or 1-NN. The study includes an overall empirical comparison between the models obtained with and without feature selection. Also several classifiers have been tested in order to illustrate the performance of the different filters considered. The results have been contrasted with nonparametric statistical tests and show that our proposal significantly improves the test accuracy of the previous models for the considered data sets. Moreover, the current proposal is much more efficient than a previous methodology developed by us; lastly, the reduction percentage in the number of inputs is above a fifty five, on average.

## 1 Introduction

The classification problem has been dealt by several machine learning techniques. Algorithms which construct classifiers from sample data, such as neural networks, radial basis functions, and decision trees, have attracted growing attention for their wide applicability. The explosion of available information complicates this problem. Moreover, redundancy or noise may be present on data. Neural networks models play a crucial role in pattern recognition [2]. For many practical problems, the possible inputs to an Artificial Neural Network (ANN) can be huge. There may be some redundancy among different inputs. Pre-processing

J.M. Ferrández et al. (Eds.): IWINAC 2011, Part II, LNCS 6687, pp. 381–390, 2011.

is often needed to reduce the number of inputs to an ANN. The application of feature selection (FS) approaches has become a real prerequisite for model building due to the multi-dimensional nature of many modeling task in some fields. Theoretically, having more features should give us more discriminating power. However, this can cause several problems: an increased computational complexity and cost; too many redundant or irrelevant features; and degradation in the classification error estimation.

Our objective is to improve the accuracy and to reduce the complexity (measured by means of the number of inputs) in classification tasks of the models of Evolutionary ANNs (EANNs) with product units (PUs) that have been employed to date by us. The computational cost is very high if Evolutionary Algorithms (EAs) with different parameter settings are employed for the training of the above-mentioned networks. However, in this paper we use a specialization of an EA called TSEA (two-stage evolutionary algorithm) [11]. First of all, FS is applied to the data sets in order to eliminate noisy and irrelevant variables. In this way, the complexity could be reduced and the accuracy could be increased. The reduction in the number of inputs could decrement the number of nodes in the hidden-layer and, hence, also simplify the associated model.

This paper is organized as follows: Sect. 2 describes some concepts about evolutionary product unit neural networks (PUNNs), the TSEA and FS; Sect. 3 presents the description of our proposal; Sect. 4 details the experimentation process; then Sect. 5 shows and analyzes the results obtained; finally, Sect. 6 states the concluding remarks.

## 2 Methodology

### 2.1 Evolutionary Product Unit Neural Networks

The single-hidden-layer feed-forward network architecture is the most popular one. Multiplicative neural networks contain nodes that multiply their inputs instead of adding them. This class of neural networks comprises such types as sigma-pi networks and PU networks. The latter type was introduced by R. Durbin and D. Rumelhart [4]. The methodology employed here consists of the use of an EA as a tool for learning the architecture and weights of a PUNN model [8]. We have used models of PUNN [11] with a three-layer $k : m : j$ architecture, with $k$ nodes in the input layer, $m$ ones and a bias one in the hidden layer and $j$ nodes in the output layer. The transfer function of each node in the hidden and output layers is the identity function. Next, we are going to describe briefly the TSEA applied. A full explanation of it can be read in Sect. 3 of [11]. TSEA is used to design the structure and learn the weights of PUNNs in two sequential phases. In the first stage, TSEA evolves two populations for a small number of generations in order to explore the search space and to refine a bit random individuals. The best half individuals of each one are merged in a new population that follows the full evolutionary cycle in the second stage to perform exploitation. The main parameters of the TSEA are the maximum number of generations ($gen$) and the maximum number of nodes in the hidden layer ($neu$).

The minimum number of nodes is an unit lower than *neu*. The remaining parameters will be described further on. At the end of the TSEA, it returns the best PUNN model with a number of nodes between *neu* and $neu + 1$ in the hidden layer. We have considered a standard soft-max activation function, associated with the $g$ network model, given by:

$$g_j(\mathbf{x}) = \frac{exp\ f_j(\mathbf{x})}{\sum_{j=1}^{J} exp\ f_j(\mathbf{x})} \quad j = 1, ..., J \tag{1}$$

where $J$ is the number of classes in the problem, $f_j(\mathbf{x})$ is the output of node $j$ for pattern $\mathbf{x}$ and $g_j(\mathbf{x})$ is the probability that this pattern belongs to class $j$. Given a training set $D = (x_i, yi)\ i = 1, ..., N$, a function of cross-entropy error is used to evaluate a network g with the instances of a problem, which is reflected in the following expression:

$$l(g) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{J}(y_i^j ln(g_j(\mathbf{x}_i))) \tag{2}$$

and substituting $g_j$ defined in (2),

$$l(g) = -\frac{1}{N}\sum_{i=1}^{N}\left(-\sum_{j=1}^{J} y_i^j f_j(\mathbf{x_i}) + ln(\sum_{j=1}^{J} exp\ f_j(\mathbf{x}_i))\right) \tag{3}$$

where $y_i^j$ is the target value for class $j$ with pattern $\mathbf{x_i}$ ( $y_i^j = 1$ if $\mathbf{x_i} \in$ class $j$ and $y_i^j = 0$ otherwise), $f_j(\mathbf{x_i})$ is the output value of the neural network for the output neuron $j$ with pattern $\mathbf{x_i}$. Observe that soft-max transformation produces probabilities that sum to one and therefore the outputs can be interpreted as the conditional probability of class membership. On the other hand, the probability for one of the classes does not need to be estimated because of the normalization condition. Usually, one activation function is set to zero; in this work $f_J(\mathbf{x}_i) = 0$ and we reduce the number of parameters to estimate. Thus, the number of nodes in the output layer is equal to the number of classes minus one in the problem. Since the EA objective is to minimize the chosen error function, a fitness function is used in the form $A(g) = (1 + l(g))^{-1}$.

The TSEA loops are repeated until the maximum number of generations, in each case, is reached or until the best individual or the population mean fitness does not improve during gen-without-improving generations (20 in this paper).

Parametric and structural mutations have been used and follow the expressions and details given in [11]. Table 1 summarizes the main TSEA parameters.

## 2.2 Feature Selection

FS is the problem of choosing a small subset of features that ideally is necessary and sufficient to describe the target concept. There are various ways in which FS algorithms can be grouped according to the attribute evaluation measure:

**Table 1.** TSEA general parameters

| Parameter | Value |
|---|---|
| $Population\ size(N)$ | 1000 |
| $gen - without - improving$ | 20 |
| $Interval\ for\ the\ exponents\ w_{ji}/coefficients\ \beta_j^l$ | $[-5, 5]$ |
| $Initial\ value\ of\ \alpha_1$ | 0.5 |
| $Initial\ value\ of\ \alpha_2$ | 1 |
| $Normalization\ of\ the\ input\ data$ | $[1, 2]$ |
| $Number\ of\ nodes\ in\ node\ addition\ and\ node\ deletion\ operators$ | $[1, 2]$ |

depending on the type (filter or wrapper technique) or on the way that features are evaluated (individual or subset evaluation). The filter model [7] relies on general characteristics of the data (such as distance, consistency, and correlation) to evaluate and select feature subsets without involving any mining algorithm. The wrapper model requires a predetermined mining algorithm and uses its performance as evaluation criterion.

In feature subset selection, it is a fact that two types of features are generally perceived as being unnecessary: features that are irrelevant to the target concept, and features that are redundant given other features. In a previous work, we proposed BIRS (Best Incremental Ranked Subset) [10] method to obtain relevant features and to remove redundancy. Now, we combine BIRS with TSEA. The features selected are considered as input variables to the classifier. BIRS belongs to a hybrid category where the selection process is divided into two stages: in the first one, features are evaluated individually, providing a ranking based on a criterion; in stage two, a feature subset evaluator is applied to a certain number of features in the previous ranking following a search strategy. BIRS can use any evaluator in the two phases. In the cited work, BIRS uses as a subset evaluator CFS (Correlation-based Feature Selection) [5] and CNS (consistency based measure) [7] -that are established on correlation and consistency concepts- at the second phase, and SOAP (Selection Of Attributes by Projection) [9] measure and the own subset evaluator at the first phase as a ranking evaluator.

Therefore, in the experiments, spBI_CFS indicates that SOAP is employed as an individual measure in the first part of BIRS, and CFS is employed as a subset evaluator in the second part. In the same way, cnBI_CNS denotes that CNS evaluator will be used in both parts of the BIRS algorithm.

## 3 Proposal Description

Our attention is focused on evolutionary PUNNs for classification problems. The current paper presents TSEAFS methodology, a mixture of two FS methods with TSEA. First of all, some feature selectors are applied independently to the training set of all datasets in order to obtain a list of attributes, for each of them, considered for training and test phases. In this way, two reduced sets (reduced training and test sets) are generated, where only most relevant features are included. It is important to point out that the FS is performed only with training data; the reduced test set has the same features as the reduced training set. These reduced sets are taken as input to TSEA. TSEAFS operates with two filters as independent feature selectors. As a result of the FS stage, a list of relevant features is obtained with each of the FS methods for each data set. There are two different configurations in TSEA, named $1*$ and $2*$. The TSEAFS features are the following: a) PUNN have been employed, with a number of neurons in the input layer equal to the number of variables in the problem after FS; a hidden layer with a number of nodes that depends on the data set to be classified and the number of selected features; and the number of nodes in the output layer equal to the number of classes minus one because a softmax-type probabilistic approach has been used; b) two experiments have been performed for each problem with two different values for $\alpha_2$, that is associated with the residual of the updating expression of the output-layer coefficients, controls the diversity of the individuals and has a great impact over the performance [11]; c) two different configurations ($1 * \sharp$ and $2 * \sharp$) are applied to subsets obtained with each of the selectors, for each dataset. The parameters of each configuration are $neu\sharp$, $gen\sharp$ and $\alpha_2$. The first two ones take specific values depending on the dataset and the last one depends on the configuration number ($1 * \sharp$,...). Table 2 shows the main aspects of TSEA/TSEAFS configurations.

**Table 2.** Description of the TSEA/TSEAFS configurations

| Methodology | Config. | Num. of Neurons in each pop. | Size of each pop. | Num. Gener. in each pop. | $\alpha_2$ |
|---|---|---|---|---|---|
| $TSEA$ | $1*$ | $neu$ *and* $neu + 1$ | 1000 | $0.1 * gen$ | 1 |
| $TSEA$ | $2*$ | $neu$ *and* $neu + 1$ | 1000 | $0.1 * gen$ | 1.5 |
| $TSEAFS$ | $1 * \sharp$ | $neu\sharp$ *and* $neu\sharp + 1$ | 1000 | $0.1 * gen\sharp$ | 1 |
| $TSEAFS$ | $2 * \sharp$ | $neu\sharp$ *and* $neu\sharp + 1$ | 1000 | $0.1 * gen\sharp$ | 1.5 |

# 4 Experimentation

Table 3 describes the data sets employed. All of them are publicly available at the UCI repository [1]. The following seven have been used: *Breast* Cancer, Statlog (*Heart*), *Hepatitis*, Molecular Biology (*Promoter* Gene Sequences), *Waveform* database generator (version 2), Wine Quality ($Winequality - red$) and *Yeast*. The size of the data sets ranges from over one hundred to five thousand. The number of features depends on the problem and varies between eight and fifty eight, while the number of classes is between two and ten. Since we are using neural networks, all nominal variables have been converted to binary ones; due to this, sometimes the number of inputs is greater than the number of features. Regards the number of inputs (*In.*) it ranges between eight and one hundred fourteen. Also, the missing values have been replaced in the case of nominal variables by the mode or, when concerning continuous variables, by the mean, taking into account the full data set. These data sets have in common that present error rates in test accuracy about 20% or above with reference classifiers such as

**Table 3.** Summary of the data sets used and parameter values for TSEA and TSEAFS methodologies

| *Data set* | *Size* | *Train* | *Test* | *Feat.* | *In.* | *Cl.* | *Neu*; *Gen* | *Neu*♯; *Gen*♯ |
|---|---|---|---|---|---|---|---|---|
| *Breast* | 286 | 215 | 71 | 9 | 15 | 2 | 9; 500 | 9; 300 |
| *Heart* | 270 | 202 | 68 | 13 | 13 | 2 | 6; 500 | 4; 20 |
| *Hepatitis* | 155 | 117 | 38 | 19 | 19 | 2 | 3; 300 | 3; 300 |
| *Promoter* | 106 | 80 | 26 | 58 | 114 | 2 | 11; 500 | 6; 300 |
| *Waveform* | 5000 | 3750 | 1250 | 40 | 40 | 3 | 3; 500 | 3; 500 |
| *Winequality* − *red* | 1599 | 1196 | 403 | 11 | 11 | 6 | 6; 300 | 4; 300 |
| *Yeast* | 1484 | 1112 | 372 | 8 | 8 | 10 | 11; 1000 | 11; 1000 |

C4.5 or 1-NN.The experimental design uses the cross validation technique called stratified hold-out that consists of splitting the data into two sets: training and test set, maintaining the class distribution of the samples in each set approximately equal as in the original data set. Their sizes are approximately 3N/4 and N/4, where N is the number of patterns in the problem.

Regards to TSEA methodology, the concrete values of neu and gen parameters depend on the data set and are shown in the eighth column of Table 3. With respect to the number of generations, we have defined three kinds of values: small (300), medium (500) and large (1000). We have given, in some cases, values of our choice to the two parameters depending on the complexity of the data set (number of classes, inputs, instances,...). Other times the values are based on a previous work [11]. In TSEAFS, again there are two parameters, *neu*♯ and *gen*♯, whose value is defined for each data set. The last column of the Table 3 presents the values of them along with the ones of TSEA to have a general view of the differences. In TSEAFS the number of neurons is upper bounded by TSEA value. It is important to note that aforementioned values of the parameters concern to the base configuration $(1 * /1 * \sharp)$. The *gen*♯ parameter takes values similar to *gen* with the exception of Heart in whose case is very small (20) since the search converges quickly. The values of the remaining configurations are presented further on.

Table 4 depicts the methods used in the experimentation. There are two ones with and one without feature selection that belong respectively to TSEAFS (the current proposal) and TSEA methodologies. The feature selectors are filters. Last column defines an abbreviated name for each of them that is employed in next sections.

**Table 4.** List of methods employed in experimentation with and without feature selection

| Feature Selector | Ranking Method | Subset Evaluation | Methodology | *Abb.Name* |
|---|---|---|---|---|
| − | *None* | *None* | *TSEA* | *FS*0 |
| *spBI_CFS* | *spBI* | *CFS* | *TSEAFS* | *FS*1 |
| *cnBI_CNS* | *cnBI* | *CNS* | *TSEAFS* | *FS*2 |

## 5 Results

This section details the results obtained, measured in Correct Classification Ratio (CCR) in the test set or in the test subset depending on that FS has been considered or not. First of all, we present the results obtained with TSEA and TSEAFS. After that, a statistical analysis compares them to determine whether there are significant differences between applying or not FS. Later, the number of inputs is analysed.

### 5.1 Results Applying TSEA and TSEAFS

The results obtained by applying TSEA methodology [11] are presented, along with those obtained with TSEAFS. Table 5 shows the mean and standard deviation (SD) of the test accuracies for each data set for a total of 30 runs. The best results without and with FS appear in boldface for each data set. From the analysis of the data, it can be concluded, from a purely descriptive point of view, that the TSEAFS methodology obtains best results for all data sets. In most of cases, the SD reduction with TSEAFS is clear and it expresses more homogeneous results compared to TSEA.

**Table 5.** Results obtained in seven data sets applying TSEA and TSEAFS

| Data set | Method | Topology | $Mean \pm SD$ | |
|---|---|---|---|---|
| | | | $Config\ 1*/1*\sharp$ | $Config\ 2*/2*\sharp$ |
| $Breast$ | $FS0$ | 15 : [9, 10] : 1 | **65.96 ± 2.89** | 62.76 ± 3.08 |
| | $FS1$ | 4 : [9, 10] : 1 | **69.85 ± 1.50** | 68.21 ± 1.08 |
| | $FS2$ | 2 : [9, 10] : 1 | 69.01 ± 0.00 | 69.01 ± 0.00 |
| $Heart$ | $FS0$ | 13 : [6, 7] : 1 | 76.62 ± 2.33 | **77.45 ± 3.09** |
| | $FS1$ | 7 : [4, 5] : 1 | 77.45 ± 2.16 | 77.69 ± 2.28 |
| | $FS2$ | 9 : [4, 5] : 1 | **78.57 ± 1.99** | 77.79 ± 1.60 |
| $Hepatitis$ | $FS0$ | 19 : [3, 4] : 1 | 82.10 ± 4.44 | **87.01 ± 3.78** |
| | $FS1$ | 10 : [3, 4] : 1 | **90.78 ± 1.79** | 89.29 ± 1.53 |
| | $FS2$ | 5 : [3, 4] : 1 | 86.14 ± 1.81 | 87.45 ± 1.49 |
| $Promoter$ | $FS0$ | 114 : [11, 12] : 1 | 65.76 ± 8.99 | **68.20 ± 9.52** |
| | $FS1$ | 7 : [6, 7] : 1 | 83.84 ± 3.83 | **85.64 ± 4.03** |
| | $FS2$ | 7 : [6, 7] : 1 | 80.00 ± 2.74 | 76.30 ± 4.10 |
| $Waveform$ | $FS0$ | 40 : [3, 4] : 2 | **84.46 ± 0.92** | 82.01 ± 1.48 |
| | $FS1$ | 14 : [3, 4] : 2 | 86.35 ± 0.85 | **86.89 ± 0.71** |
| | $FS2$ | 15 : [3, 4] : 2 | 86.02 ± 2.16 | 85.67 ± 0.96 |
| $Winequality - red$ | $FS0$ | 11 : [6, 7] : 5 | 60.95 ± 1.58 | **61.11 ± 1.02** |
| | $FS1$ | 5 : [4, 5] : 5 | **61.63 ± 1.09** | 61.25 ± 1.62 |
| | $FS2$ | 8 : [4, 5] : 5 | 61.47 ± 0.95 | 60.87 ± 1.29 |
| $Yeast$ | $FS0$ | 8 : [11, 12] : 9 | 60.05 ± 1.21 | **60.16 ± 1.10** |
| | $FS1$ | 5 : [11, 12] : 9 | 59.25 ± 1.44 | 60.06 ± 1.09 |
| | $FS2$ | 7 : [11, 12] : 9 | **60.78 ± 1.29** | 59.43 ± 1.29 |

### 5.2 Statistical-Analysis

We follow the recommendations pointed out by J. Demšar [3] to perform non-parametric statistical tests. To determine the statistical significance of the differences in rank observed for each method with all data sets, a non-parametric test might be used. There are two methods, Friedman and Iman-Davenport tests. If the null-hypothesis is rejected, we can proceed with a post-hoc test. Bonferroni-Dunn has been performed.

The average ranks of all methods, without (FS0) and with FS (FS1-2) are respectively 2.86, 1.43 and 1.71. According to Iman-Davenport test results, since the statistic $F_F = 8.0$ is higher than the critical value at $(F(2, 12) = 3.89)$ the null-hypothesis is rejected. Therefore, we apply a post-hoc Bonferroni-Dunn test that compares a number of methods with a control method, by determining whether the average ranks differ by at least the CD. In our case, we make a comparison of the methods that employ FS (FS1-2) versus the control method (FS0) that does not use FS. CDs obtained by Bonferroni-Dunn test are 1.20 (at $\alpha = 0.05$) and 1.05 (at $\alpha = 0.10$). The ranking difference with FS0 are 1.43 for FS1 and 1.15 for FS2. Thus, there are significant differences between TSEA applying each of the FS methods and without FS. The statistical tests points out that PUNN performance improves significantly pre-processing the dataset with any of the FS methods employed in this paper. However, FS1 is better regarding to statistical significance level.

**Analysis of the number of inputs.** As previously mentioned, two FS methods implemented as filters have been applied to each dataset. Table 6 summarizes the average number of inputs of the test bed (see column labelled FS0) and those that have been obtained with the different feature selectors (see columns labelled FS1-2) along with the reduction percentage in the inputs of each selector compared to the original datasets. The maximum reduction percentage appears in boldface. The concrete value of the inputs for each case can be found in the third column of Table 5. The reduction percentage of the number of inputs is defined as:

$$Reduction\_Inputs(\%) = \left(1 - \frac{Inputs(FSi)}{Inputs(FS0)}\right) 100 \quad i = 1, 2 \tag{4}$$

where $i$ is the FS methods index $Inputs(j)$ represents the number of inputs of a given dataset with method $j$. In all cases, FS methods successfully decreased the data dimensionality by selecting, in mean, less than the quarter of the original features. Certainly, the number of selected features fluctuates between a quarter

**Table 6.** Number of inputs and reduction percentage for the test bed with and without feature selection

| Data set | Inputs | | | Reduction (%) | |
|---|---|---|---|---|---|
| | FS0 | FS1 | FS2 | FS1 | FS2 |
| *Average* | 31.43 | 7.43 | 7.57 | **59.68** | 55.32 |

and a third of the original features. FS1 method achieves a reduction percentage, on average, of 59.68% (from 31.43 to 7.43 features in average), which is the highest overall average value obtained.

### 5.3 Results Obtained with a Variety of Classifiers

Now, a comparison, applying the best filter (FS1), is performed between TSEA and other machine learning algorithms. These methods are C4.5, k-nearest neighbours (k-NN), -where k is 1-, SVM [12] and LMT [6]. Since, these methods are implemented in Weka tool [13], we have conducted the experiments and used the same cross-validation, thus the same instances in each of the partitions, that in the first experiment. Regarding the parameters, the algorithms have been run with the Weka default values. We have reported in Table 7 the results with FS1 for each dataset and algorithm. Due to we have used filters for the FS, the same reduced features set is applied to all classifiers. For TSEA is reported the best mean value of the two configurations, and for the remaining algorithms the mean. In each row, the best result appears in boldface. From an analysis of the results, we can assert the following. The TSEA method obtains the best result for four out of seven data sets; and SVM two times. Furthermore, the TSEA reports the highest mean accuracy (76.08%) followed by the SVM method (75.59%).

**Table 7.** Results obtained in seven data sets for several classifiers with FS1

| Data set | C4.5 | 1-NN | SVM | LMT | TSEA |
|---|---|---|---|---|---|
| *Breast* | 69.01 | **70.42** | 66.20 | 69.01 | 69.85 |
| *Heart* | 73.53 | 73.53 | 76.47 | 76.47 | **77.69** |
| *Hepatitis* | 84.21 | 89.47 | 86.84 | 89.47 | **90.78** |
| *Promoters* | 73.08 | 57.69 | 84.62 | 84.62 | **85.64** |
| *Waveform* | 74.40 | 75.36 | 86.88 | **87.04** | 86.89 |
| $Winequality - red$ | 50.87 | 48.88 | 59.80 | 48.64 | **61.63** |
| *Yeast* | 53.49 | 48.92 | 54.03 | **60.22** | 60.06 |
| *Average* | 68.37 | 66.33 | 73.55 | 73.64 | **76.08** |

## 6 Conclusions

This paper presented a methodology that combines FS with Evolutionary Artificial PUNN in classification problems. Specifically, a mixture of our previous TSEA methodology and FS, called TSEAFS, has been introduced. FS is performed by means of filters. The models obtained with the proposal has the advantages that are more accurate and less complex, taking into consideration the number of inputs and/or the number of nodes in the hidden-layer. Also, the current proposal is much more efficient than the previous one. An empirical study on seven UCI classification problems, that present test accuracy error rates about a twenty percent or above with C4.5 or 1-NN classifiers, has been performed to compare TSEAFS and TSEA methodologies, both of them based on

Evolutionary Artificial PUNN. Also other state-of-the-art classifiers have been tested in order to get an overall outlook.

## Acknowledgments

This work has been partially subsidized by TIN2007-68084-C02-02 and TIN2008-06681-C06-03 projects of the Spanish Inter-Ministerial Commission of Science and Technology (MICYT), FEDER funds and the P08-TIC-3745 project of the "Junta de Andalucía" (Spain).

## References

1. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository (2007)
2. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Heidelberg (2006)
3. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. 7, 1–30 (2006)
4. Durbin, R., Rumelhart, D.: Products units: A computationally powerful and biologically plausible extension to backpropagation networks. Neural Computation 1(1), 133–142 (1989)
5. Hall, M.A.: Correlation-based feature selection for discrete and numeric class machine learning. In: Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), pp. 359–366. Morgan Kaufmann, San Francisco (2000)
6. Landwehr, N., Hall, M., Frank, E.: Logistic model trees. Machine Learning 59(1-2), 161–205 (2005)
7. Liu, H., Setiono, R.: A probabilistic approach to feature selection - a filter solution. In: Proceedings of the Thirteenth International Conference on Machine Learning (ICML 1996), pp. 319–327. Morgan Kaufmann, San Francisco (1996)
8. Martínez-Estudillo F.J., Hervás-Martínez, C., Gutiérrez, P.A., Martínez-Estudillo, A. C.: Evolutionary product-unit neural networks classifiers. Neurocomputing 72(1-2), 548–561 (2008)
9. Ruiz, R., Riquelme, J.C., Aguilar-Ruiz, J.S.: Projection-based measure for efficient feature selection. J. Intell. Fuzzy Syst. 12(3-4), 175–183 (2002)
10. Ruiz, R., Riquelme, J.C., Aguilar-Ruiz, J.S.: Incremental wrapper-based gene selection from microarray expression data for cancer classification. Pattern Recognition 39(12), 2383–2392 (2006)
11. Tallón-Ballesteros, A.J., Hervás-Martínez, C.: A two-stage algorithm in evolutionary product unit neural networks for classification. Expert Systems with Applications 38(1), 743–754 (2011)
12. Vapnik, V.N.: The nature of Statistical Learning Theory. Springer, Heidelberg (1995)
13. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Data Management Systems, 2nd edn., Morgan Kaufmann, San Francisco (2005)