Evolutionary Learning by a Sensitivity-Accuracy Approach for Multi-class Problems

F.J. Martínez-Estudillo, P.A. Gutiérrez, C. Hervás and J.C. Fernández

Abstract—**Performance evaluation is decisive when improving classifiers. Accuracy alone is insufficient because it cannot capture the myriad of contributing factors differentiating the performances of two different classifiers and approaches based on a multi-objective perspective are hindered by the growing of the Pareto optimal front as the number of classes increases. This paper proposes a new approach to deal with multi-class problems based on the accuracy (C) and minimum sensitivity (S) given by the lowest percentage of examples correctly predicted to belong to each class. From this perspective, we compare different fitness functions (accuracy, C , entropy, E , sensitivity, S , and area, A) in an evolutionary scheme. We also present a two stage evolutionary algorithm with two sequential fitness functions, the entropy for the first step and the area for the second step. This methodology is applied to solve six benchmark classification problems. The two-stage approach obtains promising results and achieves a high classification rate level in the global dataset with an acceptable level of accuracy for each class.**

I. Introduction

One of the fundamental problems of machine learning is the classification or discrimination of unknown examples into two or more classes based on a number of examples whose correct class is known, called the "training dataset". Performance evaluation is decisive at many stages during the improvement of classifiers. The process of designing a new classification algorithm usually implies an iterative procedure where each iteration significantly alters the classifier, which then requires re-evaluation to establish its impact on performance. To evaluate a classifier, the machine learning community has traditionally used the correct classification rate or accuracy to measure its default performance. In the same way, accuracy has been frequently used as the fitness function in evolutionary algorithms when solving classification problems. However, the pitfalls of using accuracy have been pointed out by several authors [1], [2]. Actually, it is enough to simply realize that accuracy cannot capture all the different behavioral aspects found in two different classifiers. Even in the simplest case where there are only two classes, the accuracy states a one-dimensional ordering where we find two different types of errors. This problem is especially significant when we deal with classification problems that differ in their prior class probabilities (class imbalances) or where there are a great number of classes. It is well known that there is a significant failure when one class is much less common than another.

When there are two classes, an alternative to accuracy to overcome these difficulties is ROC plots [1], which measure the misclassification rate of one class and the accuracy of the other. The ROC plot is a two dimensional one, with the misclassification rate of one class ("negative") on the horizontal axis and the accuracy of the other class ("positive") on the vertical axis. The ROC plot preserves all performance-related information about a classifier and it also allows instant visual inspection of key relationships in the performances of several classifiers. The standard ROC perspective is limited to classification problems with two classes. The extension to the standard two class ROC for multi-class problems (Q -classes) considers a multiobjective optimization problem [3], where the objective is to simultaneously minimize the $Q(Q-1)$ misclassification rates. In terms of the confusion matrix, the extension considers off-diagonal elements. The main shortcoming of this approach is that unfortunately the dimension of the Pareto optimal front grows at the rate of the square of the number of classes. This behaviour of the Pareto front has several consequences. Firstly, it increases the difficulties for a graphic representation that would allow us to visualize the performance of the classifiers. Secondly, it is straightforward to prove that the density of the Pareto front decreases dramatically with respect to the number of objectives, in our case, the $Q(Q-1)$ misclassification rates. Moreover, in multiobjective optimization, it is well known that the probability of one point dominating over another point decreases dramatically as the number of objectives increases. Finally, it is important to keep in mind the computational problem associated with a multi-objective optimization problem that has a lot of objectives.

The first part of this paper is devoted to proposing and studying a two-dimensional performance measure for multi-class classifiers that could be seen as a trade-off between

Manuscript received December 10, 2007. This work has been financed in part by TIN 2005-08386-C05-02 projects of the Spanish Inter-Ministerial Commission of Science and Technology (MICYT) and FEDER funds.

F.J. Martínez-Estudillo is with the Department of Management and Quantitative Methods, ETEA, Escritor Castilla Aguayo 4, 14005, Córdoba, Spain, (corresponding author, phone +34957222120; fax +34957222107; email: fjmestud@etea.com).

P. A. Gutiérrez is with the Department of Computing and Numerical Analysis of the University of Córdoba, Campus de Rabanales, 14071, Córdoba, Spain (email: zamarck@yahoo.es).

C. Hervás is with the Department of Computing and Numerical Analysis of the University of Córdoba, Campus de Rabanales, 14071, Córdoba, Spain (email: email: chervas@uco.es).

J.C. Fernández is with the Department of Computing and Numerical Analysis of the University of Córdoba, Campus de Rabanales, 14071, Córdoba, Spain (email: fernandezcaballero@gmail.com).

scalar and misclassification rate measures. Assuming that all misclassifications are equally costly and there is no penalty for a correct classification, we start from the premise that a good classifier should combine a high classification rate level in the global dataset with an acceptable level for each class. Concretely, we consider the traditionally used accuracy C and the minimum of the sensitivities of all classes S, that is, the lowest percentage of examples correctly predicted as belonging to each class with respect to the total number of examples in the corresponding class. The sensitivity versus accuracy pair (S,C) expresses two features associated with a classifier: global performance C and the rate of the worst classified class S. It is clear that two quantities can not collect all the information given by the $Q(Q-1)$ misclassification rates included in the confusion matrix. Nevertheless, the (S,C) pair tries to find an intermediate point between scalar measures and multidimensional ones based on misclassification rates. Our approach can be represented in a two dimensional space to visualize performance regardless of the number of classes in the problem and, moreover, it could be especially useful when dealing with imbalanced classification problems.

Secondly, several problems are studied from the perspective of (S,C). We consider the standard multilayer perceptron MLP to be the base classifier and an evolutionary neural networks algorithm searching the optimal weight set of the MLP, designing its architecture, finding the most adequate number of neurons in the hidden layer and the optimal number of connections. A novelty fitness function, the area function, $A(S,C)$, is built. The area fitness function tries to find a good balance between the classification rate level in the global dataset and an acceptable level for each class. The performance of this fitness function together with three fitness functions: the accuracy, C, cross-entropy, E, and sensitivity, S, are analyzed in six classification problems, obtaining diverse results from the perspective of (S,C).

Finally, we build an evolutionary algorithm in two stages using two sequential fitness functions, the entropy for the first step and the area for the second step. The approach obtains promising results and a high classification rate level in the global dataset with an acceptable level of accuracy for each class. There is a comparison with other fitness functions as well as the graphic representation in the 2-D space (S,C) of the different classifiers obtained.

The paper is structured as follows. First, we present our approach based on the sensitivity versus accuracy pair (S,C) and explain its properties in depth. The third section contains the base version and the two-stage version of the evolutionary neural network algorithm and the fitness functions used in the experimental setup. Finally, the paper concludes with an analysis of the results obtained in six benchmark classification problems and a brief discussion of issues that can be followed up in future work.

II. Accuracy and Sensitivity

We consider a classification problem with Q classes and N training or testing patterns with g as a classifier obtaining a $Q \times Q$ contingency or confusion matrix $M(g)$:

$$M(g) = \left\{ n_{ij}; \sum_{i,j=1}^{Q} n_{ij} = N \right\}$$

where 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 the correctly classified patterns and the off-diagonal to the mistakes in the classification task.

Let us denote the number of patterns associated with class i by $f_i = \sum_{j=1}^{Q} n_{ij}, i = 1,\ldots,Q$. We start by defining two scalar measures that take the elements of the confusion matrix into consideration from different points of view. 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 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 we define the sensitivity S of the classifier as the minimum value of the sensitivities for each class:

$$S = min\{S_i; i = 1,\ldots,Q\}$$

We define the correct classification rate or accuracy C

$$C = (1/N)\sum_{j=1}^{Q} n_{jj},$$

that is the rate of all the correct predictions.

Specifically, we consider the two-dimensional measure (S,C) associated with classifier g. The measure tries to evaluate two features of a classifier: global performance in the whole dataset and the performance in each class. We represent the sensitivity S on the horizontal axis and accuracy C on the vertical axis. One point in (S,C) space dominates another if it is above and to the right, i.e. it has more accuracy and greater sensitivity.

Next, we show the relationship between S and C.

Proposition 1.

Let us consider a Q-class classification problem. Let C and S be respectively the accuracy and the sensitivity associated with a classifier g, then $S \le C \le 1-(1-S)p^*$, where p^* is the minimum of the estimated prior probabilities.

Proof.

We begin by proving the upper bound. We will denote by J the class with the minimum of the prior probabilities. From the definitions of accuracy and sensitivity, and taking

into account that $\sum_{j=1}^{Q} f_j = N$ and $p^* = f_J / N$, we see that:

$$C = \sum_{j=1}^{Q} \frac{n_{jj}}{f_j}\frac{f_j}{N} = \sum_{j=1}^{Q} S_j \frac{f_j}{N} = S\frac{f_J}{N} + \sum_{j\neq J} S_j \frac{f_j}{N} \leq$$
$$\leq S\frac{f_J}{N} + \sum_{j\neq J}\frac{f_j}{N} = S\frac{f_J}{N} + 1 - \frac{f_J}{N} = 1-(1-S)\frac{f_J}{N} \quad (1)$$

On the other hand, the lower bound can be obtained

$$C = \frac{1}{N}\sum_{i=1}^{Q} n_{ii} = \sum_{i=1}^{Q} \frac{n_{ii}}{f_i}\frac{f_i}{N} = \sum_{i=1}^{Q} S_i \frac{f_i}{N} \geq S\sum_{i=1}^{Q}\frac{f_i}{N} = S \quad (2)$$

and according to (1) and (2), we conclude that $S \leq C \leq 1-(1-S)p^*$. ■

Therefore, each classifier will be represented as a point in the shaded region in Figure 1. Several points in (S,C) space are important to note. The lower left point $(0,0)$ represents the worst classifier and the optimum classifier is located at the $(1,1)$ point. Furthermore, the points on the vertical axis correspond to classifiers that are not able to predict any point in a concrete class correctly. Note that it is possible to find among them classifiers with a high level of accuracy, particularly in problems with small p^*.

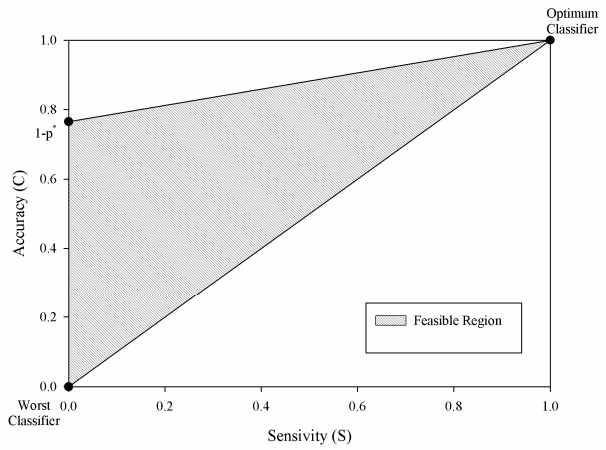

Fig. 1. Feasible region in the two-dimensional (S,C) space for a concrete classification problem.

From the concrete shape of the region we can make the following comments. First of all, observe that an increase in accuracy C does not imply an increase in sensitivity S. Reciprocally, an increase in sensitivity S does not mean an increase in accuracy C. On the other hand, it should be noted that for a fixed value of accuracy C, a classifier will be better when it corresponds to a point nearer to the diagonal of the square.

When the number of classes increases or the problem is highly imbalanced, the quantity $p^* = f_J / N \leq 1/Q$ decreases in both cases, and the straight line which defines the upper border of the feasible set in (S,C) space tends to be horizontal and so the range of S values will be large even for high values of C. In these conditions the use of C as the only measure will probably be inadequate as it hides many different possibilities for S. These comments show us that the sensitivity versus accuracy measure can be especially useful in problems concerning imbalance or when there is a great number of classes, and confirm again the inadequacy of accuracy in these situations.

Finally, it is possible to prove that each point in the shaded region in Figure 1 corresponds to a concrete classifier (confusion matrix). As a consequence, the bounds previously stated in Proposition 1 can not be improved.

III. Evolutionary Algorithms and Fitness Functions for Classification Problems

A. Base Evolutionary Algorithm

In this section we consider an evolutionary algorithm that tries to move the classifier population towards the optimum classifier located in the $(1,1)$ point in the (S,C) space. We think an evolutionary algorithm could be an adequate scheme that allows us to improve the quality of the classifiers, measured in terms of accuracy and sensitivity, directing the solutions towards the $(1,1)$ point.

The classifier chosen was the standard multilayer perceptron MLP and an evolutionary neural network algorithm was applied to estimate the structure and learn the weights of the neural network models. The basic framework of the evolutionary algorithm is the following: the search begins with an initial population of neural networks and the population is updated in each iteration using a population-update algorithm which evolves both the structure and the weights. The population is subjected to the operations of replication and mutation. Crossover is not used due to its potential disadvantages in evolving artificial networks [4]. With these features the algorithm falls into the class of evolutionary programming [5].

The general structure of the EA is the following:

(1) Generate a random population of size N.
(2) Repeat until the stopping criterion is fulfilled
  (a) Calculate the fitness of every individual in the population.
  (b) Rank the individuals with respect to their fitness.
  (c) The best individual is copied into the new population.
  (d) The best 10% of population individuals are replicated and substitute the worst 10% of individuals.
  Over that intermediate population we:
  (e) Apply parametric mutation to the best (p_m)% of individuals.
  (f) Apply structural mutation to the remaining $(100-p_m)$% of individuals.

Parametric mutation is accomplished for each weight w_{ji} of the neural network with Gaussian noise $w_{ji}(t+1) = w_{ji}(t) + \xi(t)$, where $\xi(t)$ represents a one-dimensional normally distributed random variable. The variance of the normal distribution is updated throughout the

evolution of the algorithm applying the simplest heuristic 1/5 success rule of Rechenberg [6]. On the other hand, structural mutation implies a modification in the neural network structure and allows explorations of different regions in the search space while helping to keep up the diversity of the population. There are five different structural mutations: node deletion, connection deletion, node addition, connection addition and node fusion. These five mutations are applied sequentially to each network. For more details about the general structure of the EA and the parametric and structural mutations the readers can see [7, 8].

The parameters used in the evolutionary algorithm are common for the six problems. We have considered the maximum number of hidden nodes to be $m = 6$. The number of nodes that can be added or removed in a structural mutation is within the $[1,2]$ interval. The number of connections that can be added or removed in a structural mutation is within the $[1,c]$ interval, where c is a third of the number of connections in the model. Parametric mutation is applied to the best $(p_m = 10)\%$ of individuals. The stop criterion is reached when the following condition is fulfilled: for 20 generations there is no improvement either in the average performance of the best $(p_m)\%$ of the population or in the fitness of the best individual.

B. Fitness functions

The algorithm will be evolved with the four fitness functions described below:

- The accuracy C used as the standard fitness function in classification algorithms.
- The cross-entropy error function E or Q-class multinomial deviance [9] given by:

$$E(\boldsymbol{\theta}) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{l=1}^{Q} y_n^{(l)} \log h_l(\mathbf{x}_n, \boldsymbol{\theta}_l) \qquad (3)$$

  where $h_l(\bullet, \boldsymbol{\theta}_l)$ are the softmax activation functions, $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_Q)$ the corresponding parameters and $D = \{(\mathbf{x}_n, \mathbf{y}_n)\}$ the training dataset.
- The sensitivity S of the classifier as the minimum value of the sensitivities for each class $S = mín\{S_i; i = 1, \ldots, Q\}$.
- The area A. Given a classifier g with accuracy C and sensitivity S, we build the following scalar measure associated to g:

$$A(S,C) = (1-S)(1-C) - \frac{1}{2}\left[(1-C)^2 - p^*(1-S)^2\right]$$

  The $A(S,C)$ function corresponds to the area of the region depicted in Figure 2. Observe that the global minimum of $A(S,C)$ is reached in the (1,1) optimum classifier point. Since

$$\frac{\partial A}{\partial S} = p^*(S-1) + C - 1 < 0, \quad \frac{\partial A}{\partial C} = S - C < 0,$$

  an increase in C implies a decrease in A and also an increase in S implies a decrease in A.

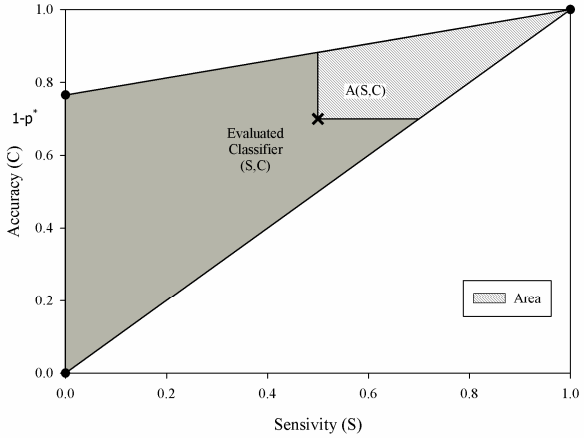

Fig. 2. Area above an evaluated classifier ($A(S,C)$ function).

C. Two-Stage Evolutionary Algorithm

As we will see in the next section, the results obtained with the different fitness functions suggest the combination of the entropy and area $A(S,C)$ as fitness functions in a two-stage evolutionary algorithm. We run the same previously explained evolutionary neural network algorithm in two stages, changing the fitness function using two sequential functions, the entropy, E, for the first step and the area, A, for the second one, and adjusting some of the parameters at each stage. This methodology is called $E + A$:

1) The first stage of the algorithm uses the cross-entropy fitness function. Exploration is favored by considering a high population size ($N = 1000$) and a very extensive structural mutation ($100 - p_m = 90$).
2) When the stop condition is fulfilled, the second stage of the algorithm starts and the fitness function used for evaluating individuals changes to the A function. We select this function because it is able to improve the sensitivity of the classifiers without losing their accuracy levels. The best 100 individuals resulting from the first stage are selected, forming the initial population of the second stage. In this manner, the population size is reduced ($N = 100$) together with the percentage of structurally mutated individuals ($100 - p_m = 10$), performing an exploitation task. The second stage of the algorithm is run for a maximum of 100 generations, instead of considering the previously mentioned stop criterion.

IV. Experiments

We consider six datasets with different features taken from the UCI repository [10] (see Table I). The experimental design for the six classification benchmark problems was conducted using a stratified holdout cross-

validation procedure, where approximately 75% of the patterns were randomly selected for the training set and the remaining 25% for the test set.

The experiments were conducted using a software package developed in JAVA by the authors, as an extension of the JCLEC framework (http://jclec.sourceforge.net/) [11]. The base evolutionary neural network algorithm is available in the non-commercial JAVA tool named KEEL (http://www.keel.es) [12]

We carry out two experiments. In the first one, we run the evolutionary algorithm with the fitness functions (accuracy, C, entropy, E, sensitivity, S, and area $A(S,C)$) and we compare the results obtained for each dataset from the perspective of accuracy and sensitivity. Our first aim is to show that different fitness functions in the evolutionary algorithm can obtain diverse results in the (S,C) space. Table II shows the statistical results (mean and standard deviation in 30 executions of the algorithm). From the analysis of the results obtained, we can conclude the following:

- Accuracy C generally guides the algorithm towards regions in the (S,C) space with high C and low sensitivity S, especially for datasets with lower p^* (Balance, Dermatology and Lymphography). The results confirm that the accuracy is not a robust fitness function to obtain classifiers with a high level of classification in each class. This fact has been already shown in highly imbalanced problems. However, we show that it is also true even for less imbalanced datasets such as Pima and German.
- The sensitivity fitness function S generally obtains classifiers with a better sensitivity level than the fitness accuracy C, but at a lower accuracy level. This fitness function can be extremely demanding in very imbalanced problems or those with a lot of classes (see Dermatology or Lymphography) and obtains an acceptable performance in two class problems (see Pima and German).
- The entropy reaches regions in the (S,C) space with high C and an acceptable sensitivity S level. Several studies prove that the entropy has a greater robustness than accuracy for classification problems [9],[13]. Our results confirm this fact from a different point of view.
- The area A generally obtains better sensitivity levels than other fitness functions, but slightly lower accuracy levels.

The second experiment is aimed to evaluate the proposed two-stage evolutionary algorithm. Table II includes the results of the two-stage algorithm ($E+A$) for the six datasets. A graphical analysis of the behaviour of the two-stage algorithm is presented in the Figures 3 and 4, where the best 100 individuals of the population in one execution are depicted in the (S,C) space for the Balance training and test sets, respectively. Observe that the entropy moves the algorithm forward the vertical direction in the first stage, obtaining models with a good global classification level without significantly reducing the sensitivity, while the second stage moves the population forward the horizontal one, improving the sensitivity levels.

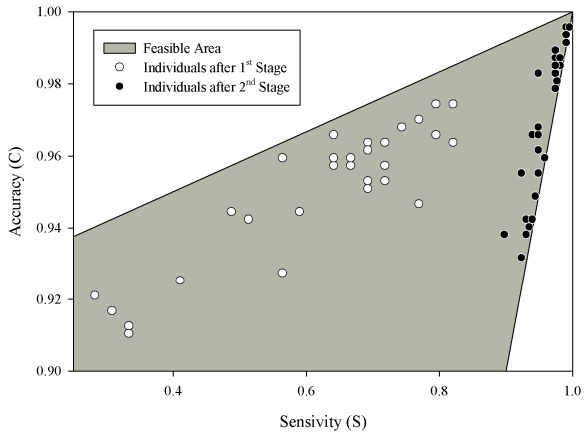

Fig. 3. Best 100 individuals before and after the second stage of the E+A algorithm for the Balance **training** set.

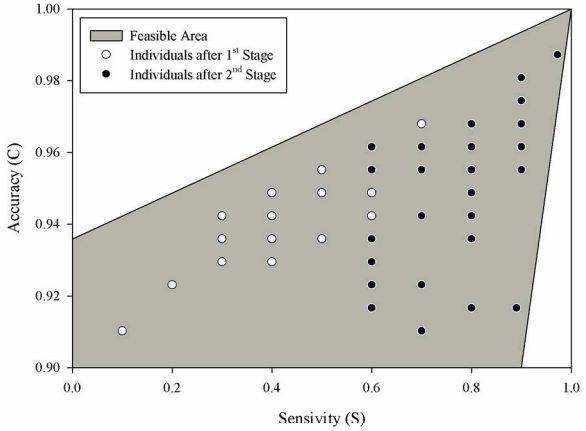

Fig. 4. Best 100 individuals before and after the second stage of the E+A algorithm for the Balance **test** set.

In order to determine the best fitness function of our evolutionary algorithm function (in the sense of its influence on the accuracy C and sensitivity S in the test dataset), the ANalysis Of the VAriance (ANOVA) statistical method or the non parametric Kruskal-Wallis test were used, depending on the satisfaction of the normality hypothesis of C and S values. Based on the hypothesis of normality, ANOVA examines the effects of some quantitative or qualitative variables (called factors) on one quantitative response. For example, in our case, the linear model for C is given by:

$$C_{ij} = \mu + F_i + e_{ij}$$

for $i = 1,2,3,4,5$ and $j = 1,2,...,30$. The factor F_i analyzes the effect over the C of the i-th level of this factor, where F_i represents the fitness function used in the algorithm, with levels: $(i=1)$ for C fitness function, $(i=2)$ for A, $(i=3)$ for S, $(i=4)$ for E and $(i=5)$ for the $E+A$

methodology. The term μ is the fixed effect that is common to all the populations. The term e_{ijk} is the influence of everything that could not be assigned on the result, or the effect of random factors. In this way, 150 simulations were carried out, corresponding to all the possible application combinations of the five levels for the first factor. The results of the ANOVA analysis for test C values show that for all datasets the fitness function effect is statistically significant at a 5% level of significance (see the first row in table III). For the Newthyroid dataset, the result of the Kruskal-Wallis test presents the same conclusion ($p-value = 0.013$). The results of the ANOVA analysis for S show that for Balance, German and Pima datasets, the fitness function effect is statistically significant at a 5% level of significance (see the third row in table III). For Dermatology and Newthyroid datasets, the result of the Kruskal-Wallis test presents the same conclusion (with $p-value = 0.000$ and $p-value = 0.004$, respectively).

In order to determine whether there are significant differences among the various fitness functions used in the EA and under the normality hypothesis, we perform a post hoc multiple comparison test of the average C and S obtained with the different levels of each factor. First, we carry out a Levene test ([14], [15]) to evaluate the equality of variances. If the hypothesis of the equality of variances is accepted, we perform a Tukey test [15] to rank the means of each level in the factor. Our aim is to find the level of each factor whose average fitness is significantly better than the average fitness of the rest of the levels in the factor. If the results of the test of Levene reject the equality of covariance matrixes, we perform a Tamhane test [16] instead of a Tukey test. Table III shows the results obtained (in the second row for G and the fourth row for S) following the above methodology.

If we analyze the average results for accuracy C, we can observe that $E+A$ methodology obtains results that are better than or similar to those obtained with the entropy E fitness function in Dermatology and Lymphography datasets, and higher results than those obtained with the other fitness functions for all datasets except German. On the other hand, the results of average sensitivity S show that the $E+A$ methodology obtains a performance that is better than or similar to the performance obtained with the A and S fitness functions, and a performance that is greater than that obtained with the E and with C fitness functions.

Under the no-normality hypothesis, we use the Mann-Witney´s pair-wise test, where we compare the $E+A$ methodology against the fitness function A using the average values of S and concluding that significant differences exist between the two methodologies in favor of $E+A$ for Dermatology dataset ($p-value = 0.000$) and in favor of A for Newthyroid dataset ($p-value = 0.001$).

The Lympography dataset deserves a special mention. This is a very hard classification problem for all methodologies. Although the accuracy rate is acceptable, the sensitivity level is very low due to the minority class with only 2 patterns. This case suggests undertaking, as future work, the integration of resampling techniques, such as SMOTE [17], into our methodology to deal with highly imbalanced problems. Finally, it is worthwhile to highlight that the (S,C) approach is independent of the evolutionary algorithm and the base classifier used.

V. CONCLUSIONS

We propose a new approach to deal with multi-class classification problems. Assuming that a good classifier should combine a high classification rate level in the global dataset with an acceptable level for each class, we consider the traditionally used accuracy C and the minimum of the sensitivities of all classes S. The sensitivity versus accuracy pair (S,C) expresses two features associated with a classifier: global performance C and the rate of the worst classified class S. From this perspective, we observe the behaviour of different fitness functions such as the accuracy, entropy, sensitivity and area in an evolutionary neural network scheme in classification problems. From this analysis, we present a two-stage evolutionary algorithm with the entropy and area fitness functions, which is applied to solve six benchmark classification problems. The two-stage algorithm obtains promising results, achieving a high classification rate level in the global dataset with an acceptable level of accuracy for each class.

In our opinion, the (S,C) approach reveals a new point of view for dealing with multi-class classification problems. For instance, some suggestions for future research are the following: to study other fitness functions based on the (S,C) measures, to propose a multi-objective approach considering both S and C functions, to use other types of base classifiers, or to incorporate resampling techniques for highly imbalanced problems.

ACKNOWLEDGMENT

This work has been partially subsidized by TIN2005-08386-C05-02 projects of the Spanish Inter-Ministerial Commission of Science and Technology (MICYT) 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

[1] F. Provost and T. Fawcett, "Analysis and visualization of the classifier performance: comparison under imprecise class and cost distribution," presented at Proceedings of the Third International Conference on Knowledge Discovery (KDD97) and Data Mining, 1997.

[2] F. Provost and T. Fawcett, "Robust classification system for imprecise environments," presented at Proccedings of the Fifhteenth National Conference on Artificial Intelligence, 1998.

[3] R. M. Everson and J. E. Fieldsend, "Multi-class ROC analysis from a multi-objetive optimisation perspective," Pattern Recognition Letters, vol. 27, pp. 918-927, 2006.
[4] X. Yao, "Evolving artificial neural network," Proceedings of the IEEE, vol. 9 (87), pp. 1423-1447, 1999.
[5] D. B. Fogel, Evolutionary Computation: Toward a New Philosophy of Machine Intelligence. New York: IEEE Press, 1995.
[6] I. Rechenberg, Evolutionstrategie: Optimierung technischer Systeme nach Prinzipien der Biologischen Evolution. Stuttgart: Framman-Holzboog Verlag, 1973.
[7] C. Hervás and F. J. Martínez-Estudillo, "Logistic regression using covariates obtained by product-unit neural network models," Pattern Recognition, vol. 40, pp. 52-64, 2007.
[8] A. Martinez-Estudillo, F. Martinez-Estudillo, C. Hervas-Martinez, et al., "Evolutionary product unit based neural networks for regression," Neural Networks, vol. 19, pp. 477-486, 2006.
[9] M. Bishop, Neural Networks for Pattern Recognition: Oxford University Press, 1995.
[10] C. Blake and C. J. Merz, " UCI repository of machine learning data bases," www.ics.uci.edu/ mlearn/MLRepository.thml, 1998.
[11] S. Ventura, C. Romero, A. Zafra, et al., "JCLEC: a JAVA framework for evolutionary computation," Soft Computing, 2007.
[12] J. Alcala-Fdez, L. Sánchez, S. García, et al., "KEEL: A Software Tool to Assess Evolutionary Algorithms for Data Mining Problems," Soft Computing, 2007.
[13] C. M. Bishop, Pattern Recognition and Machine Learning: Springer, 2006.
[14] H. Levene, "In Contributions to Probability and Statistics, chap. Essays in Honor of Harold Hotelling ", pp. 278-292, 1960.
[15] R. G. Miller, Beyond ANOVA, Basics of Applied Statistics, 2 ed. London: Chapman & Hall, 1996.
[16] A. C. Tamhane and D. D. Dunlop, Statistics and Data Analysis.: Prentice Hall, 2000.
[17] N. V. Chawla, K. W. Bowyer, L. O. Hall, et al., "SMOTE: synthetic minority over-sampling technique," Journal of Artificial Intelligence Research, vol. 16, pp. 321-357, 2002.

TABLE I
DATASETS USED FOR THE EXPERIMENTS

Dataset	Size	#Inputs	#Classes	Distribution	p^*
Balance	625	4	3	(288,49,288)	0.0784
Dermatology	366	34	6	(112,61,72,49,52,20)	0.0546
German	1000	61	2	(700,300)	0.3000
Lymphography	148	38	4	(2,81,61,4)	0.0135
Newthyroid	215	5	3	(150,35,30)	0.1395
Pima	768	8	2	(500,268)	0.3490

TABLE II
STATISTICAL RESULTS FOR THE DIFFERENT APPROACHES CONSIDERED IN THE SIX DATABASES

Dataset	Fitness	C(%)	S(%)	Dataset	Fitness	C(%)	S(%)
Balance	C	91.79±1.03	8.33±14.64	Lymphography	C	80.45±5.25	0.00±0.00
	A	86.18±6.44	72.34±19.62		A	77.12±5.80	4.67±17.81
	S	80.19±9.31	66.53±19.18		S	68.83±8.86	1.67±9.13
	E	94.53±1.51	48.00±14.00		E	**82.88±4.51**	2.67±14.61
	$E+A$	**94.87±2.22**	**77.54±11.74**		$E+A$	82.16±5.62	**5.50±20.94**
Dermatology	C	82.34±4.25	4.33±13.31	Newthyroid	C	95.86±3.97	79.47±19.67
	A	81.47±6.28	17.87±28.27		A	**96.42±3.22**	**84.67±13.31**
	S	20.66±12.07	2.48±5.52		S	93.02±5.95	78.51±13.06
	E	97.11±1.95	80.56±10.38		E	94.57±2.23	74.66±9.72
	$E+A$	**97.25±1.57**	**84.82±9.40**		$E+A$	95.49±1.73	73.33±9.94
German	C	71.11±1.96	31.11±6.91	Pima	C	76.46±2.37	49.45±8.05
	A	67.67±2.30	63.40±3.92		A	72.15±1.88	67.56±3.08
	S	68.11±2.23	63.86±3.87		S	72.24±2.62	67.01±3.77
	E	**72.91±1.89**	45.78±7.13		E	**78.63±1.39**	58.21±2.99
	$E+A$	69.31±2.12	**65.51±3.26**		$E+A$	76.74±1.81	**72.69±3.26**

Different fitness functions determine the different methodologies: Accuracy (C), Area (A), Sensitivity (S), Entropy (E), and Two Stages algorithm ($E+A$).

TABLE III

P-VALUES OF THE *SNEDECOR'S* F ANOVA I TEST AND RANKING OF AVERAGES OF THE *TUKEY, TAMHANE OR K-W* STATISTICAL MULTIPLE COMPARISON TESTS FOR THE ACCURACY (C) AND SENSITIVITY (S) IN THE TEST SETS USING THE FIVE DIFFERENT FITNESS FUNCTIONS IN THE EVOLUTIONARY ALGORITHM

	Balance	Dermatology	German	Lymphography	Newthyroid	Pima
			C			
F or K-W test ($p-value$)	0.000 (*)	0.000 (*)	0.000 (*)	0.000 (*)	0.013 (*) K-W	0.000 (*)
Ranking of averages	$\mu_{E+A} \geq \mu_E > \mu_C > \mu_A \geq \mu_S$	$\mu_{E+A} \geq \mu_E > \mu_C \geq \mu_A > \mu_S$	$\mu_E > \mu_C > \mu_{E+A} > \mu_S \geq \mu_A$	$\mu_E \geq \mu_{E+A} \geq \mu_C \geq \mu_A > \mu_S$ $\mu_E > \mu_A$, $\mu_{E+A} > \mu_A$	–	$\mu_E > \mu_{E+A} \geq \mu_C > \mu_S \geq \mu_A$
			S			
F or K-W test ($p-value$)	0.000 (*)	0.000 (*) K-W	0.000 (*)	0.653 K-W	0.004 (*) K-W	0.000 (*)
Ranking of averages	$\mu_{E+A} \geq \mu_A \geq \mu_S > \mu_E > \mu_C$	–	$\mu_{E+A} \geq \mu_S \geq \mu_A > \mu_E > \mu_C$	–	–	$\mu_{E+A} > \mu_A \geq \mu_S > \mu_E > \mu_C$

(*) Statistically significant differences