Alternative OVA Proposals for Cooperative Competitive RBFN Design in Classification Tasks

Francisco Charte Ojeda[1], Antonio Jesús Rivera Rivas[2], María Dolores Pérez-Godoy[2], and María Jose del Jesus[2]

[1] Dept. of Computer Science and Artificial Inteligence
University of Granada, Spain
francisco@fcharte.com

[2] Dept. of Computer Science
University of Jaén, Spain
{arivera,lperez,mjjesus}@ujaen.es

Abstract. In the Machine Learning field when the multi-class classification problem is addressed, one possibility is to transform the data set in binary data sets using techniques such as One-Versus-All. One classifier must be trained for each binary data set and their outputs combined in order to obtain the final predicted class. The determination of the strategy used to combine the output of the binary classifiers is an interesting research area.

In this paper different OVA strategies are developed and tested using as base classifier a cooperative-competitive RBFN design algorithm, CO^2RBFN. One advantage of the obtained models is that they obtain as output for a given class a continuous value proportional to its level of confidence. Concretely three OVA strategies have been tested: the classical one, one based on the difference among outputs and another one based in a voting scheme, that has obtained the best results.

Keywords: OVA, RBFNs, Multi-class classification.

1 Introduction

A general approach to tackle several kind of classification problems is data transformation. For example, in multi-class classification, One-Versus-All (OVA) [1] is one of the most well-known.

The OVA strategy obtains a data set for each class included in the original data set. Thereby, each obtained data set contains two classes: the positive class or the class to predict, and the negative class that comprise the rest of classes. A classifier is trained for each binary data set and finally the outputs of these classifiers are combined in order to obtain the resulting class. In most of cases, this class correspond to the classifier with higher output for the positive class.

Radial Basis Function Networks (RBFNs) are one of the most important Artificial Neural Network (ANN) paradigms in the machine learning field. An

I. Rojas, G. Joya, and J. Cabestany (Eds.): IWANN 2013, Part I, LNCS 7902, pp. 331–338, 2013.

RBFN is a feed-forward ANN with a single layer of hidden units, called radial basis functions (RBFs) [2]. The overall efficiency of RBFNs has been proved in many areas [3] such as pattern classification, function approximation and time series prediction.

An important paradigm for RBFN design is Evolutionary Computation [4]. There are different proposals in this area with different scheme representations: Pittsburgh [5], where each individual is a whole RBFN, and cooperative-competitive [6], where an individual represents a single RBF.

Authors have developed an algorithm for the cooperative-competitive design of Radial Basis Functions Networks, CO^2RBFN [7], that has been successfully used in multi-class classification.

As demonstrated in [1] the use of OVA strategies can improve the results of a multi-class classifier. Thus, the aim of this paper is testing different OVA techniques with the RBFN design algorithm, CO^2RBFN. Concretely three OVA approaches have been implemented: the classical one, one based on the difference among outputs and another one based in a voting scheme.

The text is organized as follows. In Section 2, OVA methodology to multi-class classification is described as well as the concrete three methods to obtain the output class. The cooperative-competitive evolutionary model for the design of RBFNs applied to classification problems, CO^2RBFN, is described in Section 3. The analysis of the experiments and the conclusions are shown in Sections 4 and 5, respectively.

2 The OVA Approach to Multi-class Classification

There are many situations in which the class associated to a set of input attributes is not binary, but one of a set of outputs with more than two options. When it is necessary to work with a data set of this kind, a multi-class data set, there are two main methods to follow: design a classifier able to work with several classes, or split the original problem, applying the divide-and-conquer technique, by transforming the data set so that it can be processed with binary classifiers.

The decomposition of a multi-class data set in binary ones can be done using different approaches, being One-vs-All (OVA) one of the best known. The basic idea is to produce as many data sets as classes exist in the original multi-class data set, taking in each one of them a certain class as *positive* (P) and the rest as *negative* (N). Each of these data sets will be used to train a binary-independent classifier, therefore obtaining several predictions as output: one for each class.

The final predicted class could change depending on how the binary outputs obtained are combined. The kind of output generated by the binary classifiers will also influence this result; a rule based system will only indicate if the output is P or N without any additional information, on the other hand a neural network will give a weight associated to each of the two possible outputs, not a simple P or N. In the following subsections the traditional OVA approach will be exposed, along with the specific variations used within the experimentation of this proposal.

2.1 How Is Predicted the Output in Traditional OVA

Assuming that the underlying binary classifier B produces as output a value expressing a weight or likelihood associated to the positive class, and being X an instance and C the total number of classes, equation 1 will give as result the index of the class to predict following the traditional OVA method. This method is denoted as Classic OVA in the experimentation section.

$$I(X) = argmax(B_i(X)) \quad i = 1 \dots C; \tag{1}$$

It is as simple as taking the class associated to the binary classifier which has generated the maximum output. For this method to work it is necessary that the values given by the binary classifiers are comparable, applying previously a normalization process if is it required. Usually it is accepted a range between 0 and 1.

2.2 Global and Local Normalization of Outputs from Binary Classifiers

In order to normalize the values obtained from CO^2RBFN, as they are not normalized internally, two different methods has been used. The influence of the normalization method in the final results is important enough to warrant special attention.

The first method explores the outputs obtained for all the instances, gets the maximum and the minimum values, and uses this information to adjust these outputs before entering the final OVA prediction process. Therefore, it is a global normalization. In contrast, the second method does a local normalization using only the values associated to each sample. In both cases the final values will be in the range 0 to 1, as has been said above.

In the experimentation the traditional OVA approach explained before has been used in two variations, global and local, which only differ in the normalization method used.

2.3 Alternative Methods to OVA Prediction

Aiming to improve the prediction made by the Classic OVA approach, always working with the same set of output values obtained from the binary classification, we have defined and tested two alternative interpretations of these values once they have been normalized.

In the first alternative, Difference OVA, each classified instance has two values incoming from each binary classifier: the weight associated to the positive class and the one which belongs to the negative class. Instead of looking for the maximum positive value, as it is done in the traditional OVA, it is possible to calculate the difference between these two weights in order to obtain a unique value. The class predicted will be that which has the maximum difference, discarding those cases in which the positive and negative weights are very near, even though the positive could be the absolute maximum.

The second alternative method proposed to do the OVA prediction, Voting OVA, is based in the idea of a majority-voting system. Given that there are several individual predictions for each instance, coming from the use of two normalization techniques and the repetitions made in the execution over the partitioned data sets, we have taken each of those predictions as a vote for a class. The votes are summarized and the class with the higher count is the final prediction.

In the experimentation, Difference OVA are used in combination with the two normalization methods described above, giving as result two final predictions. Voting OVA approach incorporates one more prediction in the set of results to analyze.

3 CO^{2}RBFN: An Evolutionary Cooperative-Competitive Hybrid Algorithm for RBFN Design

CO^{2}RBFN [7] is an evolutionary cooperative-competitive hybrid algorithm for the design of RBFNs. In this algorithm each individual of the population represents, with a real representation, an RBF and the entire population is responsible for the final solution.

The individuals cooperate towards a definitive solution, but they must also compete for survival. In this environment, in which the solution depends on the behavior of many components, the fitness of each individual is known as credit assignment. In order to measure the credit assignment of an individual, three factors have been proposed: the RBF contribution to the network output, the error in the basis function radius, and the degree of overlapping among RBFs.

The application of the operators is determined by a Fuzzy Rule-Based System. The inputs of this system are the three parameters used for credit assignment and the outputs are the operators' application probability.

The main steps of CO^{2}RBFN, explained in the following subsections, are shown in the pseudocode, in Algorithm 1. For a wider explanation of the algorithm see reference [7].

Algorithm 1. Main steps of CO^{2}RBFN

```
1. Initialize RBFN
2. Train RBFN
3. Evaluate RBFs
4. Apply operators to RBFs
5. Substitute the eliminated RBFs
6. Select the best RBFs
7. If the stop condition is not verified go to step 2
```

RBFN Initialization. To define the initial network a specified number m of neurons (i.e. the size of population) is considered. The center of each RBF is randomly allocated to a different pattern of the training set. The RBF widths,

d_i, will be set to half the average distance between the centres. Finally, the RBF weights, w_{ij}, are set to zero.

RBFN Training. The Least Mean Square algorithm [8] is used to calculate the RBF weights.

RBF Evaluation.A credit assignment mechanism is required in order to evaluate the role of each RBF ϕ_i in the cooperative-competitive environment. For an RBF, three parameters, a_i ,e_i ,o_i are defined:

- The contribution, a_i, of the RBF ϕ_i, is determined by considering the weight, w_i, and the number of patterns of the training set inside its width, pi_i:

$$a_i = \begin{cases} |w_i| & if \quad pi_i > q \\ |w_i| * (pi_i/q) & otherwise \end{cases} \tag{2}$$

  where q is the average of the pi_i values minus the standard deviation of the pi_i values.
- The error measure, e_i, for each RBF ϕ_i, is obtained by counting the wrongly classified patterns inside its radius:

$$e_i = \frac{pibc_i}{pi_i} \tag{3}$$

  where $pibc_i$ and pi_i are the number of wrongly classified patterns and the number of all patterns inside the RBF width respectively.
- The overlapping of the RBF ϕ_i and the other RBFs is quantified by using the parameter o_i. This parameter is computed by taking into account the fitness sharing methodology [4], whose aim is to maintain the diversity in the population.

Applying Operators to RBFs. In CO^{2}RBFN four operators have been defined in order to be applied to the RBFs:

- Operator Remove: eliminates an RBF.
- Operator Random Mutation: modifies the centre and width of an RBF in a random quantity.
- Operator Biased Mutation: modifies, using local information, the RBF trying to locate it in the centre of the cluster of the represented class.
- Operator Null: in this case all the parameters of the RBF are maintained.

The operators are applied to the whole population of RBFs. The probability for choosing an operator is determined by means of a Mandani-type fuzzy rule based system [9]. The inputs of this system are parameters a_i, e_i and o_i used for defining the credit assignment of the RBF ϕ_i. These inputs are considered as linguistic variables va_i, ve_i and vo_i. The outputs, p_{remove}, p_{rm}, p_{bm} and p_{null}, represent the probability of applying Remove, Random Mutation, Biased

Table 1. Fuzzy rule base representing expert knowledge in the design of RBFNs

Antecedents				Consequents				Antecedents				Consequents			
	v_a	v_e	v_o	p_{remove}	p_{rm}	p_{bm}	p_{null}		v_a	v_e	v_o	p_{remove}	p_{rm}	p_{bm}	p_{null}
R1	L			M-H	M-H	L	L	R6		H		M-H	M-H	L	L
R2	M			M-L	M-H	M-L	M-L	R7			L	L	M-H	M-H	M-H
R3	H			L	M-H	M-H	M-H	R8			M	M-L	M-H	M-L	M-L
R4		L		L	M-H	M-H	M-H	R9			H	M-H	M-H	L	L
R5		M		M-L	M-H	M-L	M-L								

Mutation and Null operators, respectively. Table 1 shows the rule base used to relate the antecedents and consequents described.

Introduction of New RBFs. In this step, the eliminated RBFs are substituted by new RBFs. The new RBF is located in the centre of the area with maximum error or in a randomly chosen pattern with a probability of 0.5 respectively.

Replacement Strategy. The role of the mutated RBF in the network is compared with the original one to determine the RBF with the best behavior in order to include it in the population.

4 Experimentation

In order to test in a multi-class classification scenario the different OVA approaches developed and using as classifier our cooperative-competitive algorithm for RBFN design, CO^2RBFN, ten different data sets have been chosen from KEEL data set repository [10]. The properties of these data sets are shown in table 2. With these data sets, a typical experimental framework has been established with ten-fold cross validation (90% for training data set, 10% for test data set) and three repetitions for obtaining the results.

Table 2. Data set properties

Data-set	Instances	Attributes	Classes
Balance	625	4	3
Cleveland	467	13	5
Dermatology	358	33	6
Ecoli	336	7	8
Glass	214	9	6
Hayes-Roth	160	4	3
New-thyroid	215	5	3
Lymphography	148	18	4
Wine	178	13	3
Yeast	1484	8	10

The same configuration parameters are set up for all the CO^2RBFN versions: 200 iterations are established for the main loop and the number of individuals or RBFs are set to the twice of the number of classes existing in the processed data set.

In table 3 the average correct classification rate for test data sets of the different proposals are shown. Specifically the Base column shows the results obtained for the multi-class version of CO^2RBFN, without preprocessing the data set. In the following columns the results of different OVA strategies (Classic, Difference and Voting) are shown. For the Classic and Difference techniques two normalization alternatives are exhibited. All the OVA strategies are described in the section 2. For a given data set the best result is in bold.

Table 3. Average correct classification rate of different OVA strategies against the base version

Datasets	Base	Classic OVA		Difference OVA		Voting OVA
		Global	Local	Global	Local	
Balance	0.8907	0.6525	0.8810	0.9018	0.8864	**0.9071**
Cleveland	**0.5766**	0.5701	0.4940	0.5095	0.5547	0.5546
Dermatology	**0.9524**	0.6428	0.6401	0.9265	0.9247	0.9443
Ecoli	0.8167	0.5724	0.7930	0.7703	0.7781	**0.8200**
Glass	**0.6669**	0.4549	0.5703	0.5594	0.6244	0.6399
Hayes-Roth	0.6688	0.5396	0.6625	0.6938	0.7375	**0.7750**
New-thyroid	0.9511	0.8206	0.9584	0.9509	0.9556	**0.9677**
Lymphography	0.7298	0.3235	0.3374	0.6910	0.7173	**0.8165**
Wine	**0.9616**	0.6671	0.9328	0.9366	0.9385	0.9556
Yeast	**0.5780**	0.1787	0.4230	0.4569	0.5095	0.5377

From the results obtained we can conclude that OVA strategies as Classic OVA or Difference OVA do not achieve any best result with respect to the base version of CO^2RBFN (without OVA preprocessing). This fact underpins the good behavior of the base CO^2RBFN algorithm, correctly designing RBFNs for multi-class data sets.

Nevertheless, this trend changes when the more innovative OVA strategy, Voting, is applied. In fact, Voting outperforms the base version of the CO^2RBFN in five of the ten data sets. It must be also highlighted that for certain data sets, such as Hayes-Roth or Lymphography, Voting OVA has obtained significantly better results than CO^2RBFN with differences around ten points. Besides this, Voting OVA can outperforms in data sets with interesting properties, such as a moderate number of instances (Balance), attributes (Lymphography) or classes (Ecoli).

Thus, although there is tie between base CO^2RBFN and Voting OVA, the results obtained leads to carry out a more deep research about the OVA Voting strategy.

5 Conclusions

With the aim of improving the performance obtained in the classification of multi-class data sets OVA transformations can be used. The resulting binary data sets are processed by binary classifiers and the output of these ones must be combined in order to obtain the final predicted class.

In this paper different combination OVA strategies are tested using CO^2RBFN, a cooperative-competitive evolutionary algorithm for the design of RBFNs, as base classifier.

The results show that while most classic OVA strategies do not improve the performance of the base version of CO^2RBFN, the developed voting strategy outperforms this base version in certain data sets. These results encourage us to carry out a more in-deep research over the last strategy.

Acknowledgments. F. Charte is supported by the Spanish Ministry of Education under the F.P.U. National Program (Ref. AP2010-0068). This paper is partially supported by the Spanish Ministry of Science and Technology under the Project TIN 2012-33856, FEDER founds, and the Andalusian Research Plan TIC-3928.

References

1. Rifkin, R., Klautau, A.: In defense of one-vs-all classification. Journal of Machine Learning Research 5, 101–141 (2004)
2. Broomhead, D., Lowe, D.: Multivariable functional interpolation and adaptive networks. Complex Systems 2, 321–355 (1988)
3. Buchtala, O., Klimek, M., Sick, B.: Evolutionary optimization of radial basis function classifiers for data mining applications. IEEE Transactions on System, Man, and Cybernetics, B 35(5), 928–947 (2005)
4. Goldberg, D.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading (1989)
5. Harpham, C., Dawson, C., Brown, M.: A review of genetic algorithms applied to training radial basis function networks. Neural Computing and Applications 13, 193–201 (2004)
6. Whitehead, B., Choate, T.: Cooperative-competitive genetic evolution of radial basis function centers and widths for time series prediction. IEEE Transactions on Neural Networks 7(4), 869–880 (1996)
7. Pérez-Godoy, M., Rivera, A., del Jesus, M., Berlanga, F.: co^2rbfn: An evolutionary cooperative-competitive RBFN design algorithm for classification problems. Soft Computing 14(9), 953–971 (2010)
8. Widrow, B., Lehr, M.: 30 years of adaptive neural networks: perceptron, madaline and backpropagation. Proceedings of the IEEE 78(9), 1415–1442 (1990)
9. Mandani, E., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. International Journal of Man-Machine Studies 7(1), 1–13 (1975)
10. Alcalá-Fdez, J., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F.: Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. Journal of Multiple-Valued Logic and Soft Computing 17(2-3), 255–287 (2011)