Dynamic Classifier Selection based on Multiple Classifier Behaviour

Giorgio Giacinto and Fabio Roli

Dept. of Electrical and Electronic Eng. - Univ. of Cagliari, Piazza d'Armi, 09123 Cagliari, ITALY

1. Introduction

Multiple classifier systems (MCSs) based on the combination of a set of different classifiers are currently used to achieve high pattern-recognition performances [1]. For each pattern, the classification process is performed in parallel by different classifiers and the results are then combined according to some decision "fusion" method (e.g., the majority-voting rule) [1]. The majority of such combination methods are based on the assumption that different classifiers make "independent" errors [1]. However, in real pattern recognition applications, it is difficult to design a set of classifiers that should satisfy such an assumption [1-5]. In order to avoid the error-independence assumption, Huang and Suen proposed a combination method, named "Behaviour Knowledge Space" (BKS), based on the concept of multiple classifier behaviour (MCB) [2]. For each pattern, a vector whose elements are the decisions taken by the individual classifiers represents the behaviour of the MCS for such a pattern (see Section 2.2). In order to classify an unknown test pattern, all the training patterns exhibiting the same MCB of the test pattern are first identified. The classifications of such training patterns are then analysed, and the test pattern is assigned to the most frequent data class [2]. Another approach proposed to avoid the error-independence assumption is the so-called "dynamic classifier selection" (DCS) [3-5]. DCS methods are aimed to select for each test pattern the classifier that will most likely classify it correctly.

In this paper, a DCS method using MCB is proposed. It is worth remarking from the start that our paper is basically different from the work of Huang and Suen [2]. Our method exploits the concept of MCB for DCS purposes, while the BKS method is aimed at classifier combination.

The DCS method we propose is based on the concepts of "classifier's local accuracy" (CLA) and MCB. In particular, we exploit MCB information to compute CLA. The basic idea is to estimate the accuracy of each classifier in a local region of the feature space surrounding an unknown test pattern, and then to select the classifier with the highest value of this local accuracy to classify the test pattern [3-5]. In order to define such a local region and compute CLAs, the k-nearest neighbours of the test pattern are first identified in the training, or validation, data. The k-nearest neighbours characterised by MCBs "similar" to the one of the unknown test pattern are then selected to compute CLAs and perform DCS. This method is described in detail in the next Section. Experimental results and comparisons are reported in Section 3.

2. Classifier selection based on multiple classifier behaviour

2.1 Problem definition

Let us consider a classification task for M data classes $\omega_1,.., \omega_M$. Each class is assumed to represent a set of specific patterns, each pattern being characterised by a feature vector $\mathbf{X}$. Let us also assume that L different classifiers, C_j, $j = 1,..,L$, have been trained separately to solve the classification task at hand. Let $C_j(\mathbf{X})$ $\{1,.., M\}$ indicate the class label assigned to pattern $\mathbf{X}$ by classifier C_j. For each unknown test pattern, the problem addressed is the selection of the classifier out of L that is most likely to classify it correctly.

2.2 Multiple classifier behaviour

The multiple classifier behaviour for a given pattern $\mathbf{X}^1$ is defined by the vector $\mathrm{MCB}(\mathbf{X}^1) = \{C_1(\mathbf{X}^1), C_2(\mathbf{X}^1),.., C_L(\mathbf{X}^1)\}$ whose elements are the class labels assigned to pattern $\mathbf{X}^1$ by the L classifiers. As outlined in Section 1, a measure of *similarity* between MCBs is necessary in order to compute CLA. We defined the similarity between MCBs related to two patterns, $\mathbf{X}^1$ and $\mathbf{X}^2$, as follows:

Errore. Non si possono creare oggetti dalla modifica di codici di campo. (1)

where the function $T_i(\mathbf{X}^1, \mathbf{X}^2)$, i = 1,..,L, is defined as:

$$T_i\left(\mathbf{X}^1, \mathbf{X}^2\right) = \begin{cases} 1 & if\ C_i(\mathbf{X}^1) = C_i(\mathbf{X}^2) \\ 0 & if\ C_i(\mathbf{X}^1) \neq C_i(\mathbf{X}^2) \end{cases} \quad (2)$$

The similarity function $S(\mathbf{X}^1, \mathbf{X}^2)$ takes values in the range [0,..,1]. As an example, $S(\mathbf{X}^1, \mathbf{X}^2)=1$ if all the L classifiers assign the patterns $\mathbf{X}^1$ and $\mathbf{X}^2$ to the same class (it is worth noting that such a class can be different for different classifiers), while $S(\mathbf{X}^1, \mathbf{X}^2)=0$ if all the classifiers provide different class labels for the two patterns.

2.3 DCS based on MCB

As outlined in Section 1, our DCS method is based on the concepts of classifier's local accuracy and multiple classifier behaviour. For each unknown test pattern $\mathbf{X}^*$, the k-nearest neighbours in the training, or validation, data are first identified. Let $\mathbf{X}$ indicate a k-nearest neighbour of $\mathbf{X}^*$. Then, MCBs are computed for test pattern $\mathbf{X}^*$ and all the k-nearest neighbours of $\mathbf{X}^*$. After, the k-nearest neighbours that satisfy the condition $S(\mathbf{X}^*,\mathbf{X})$ > *similarity-threshold* are selected. Such patterns form the local region used to estimate CLAs. Let $N(\mathbf{X}^*)$ indicate such a local region surrounding test pattern $\mathbf{X}^*$. It is worth noting that the size of this neighbourhood can vary with the test pattern, because it depends on the degree of similarity between the MCB of the test pattern and the ones of the k-nearest neighbours. Accordingly, we can say that the number of k-nearest neighbours is dynamically adapted to the test pattern.

After the above selection of the patterns forming $N(\mathbf{X}^*)$, for each classifier, the simplest method to estimate CLA is to compute the ratio between the number of patterns in $N(\mathbf{X}^*)$ that were correctly classified by the classifier C_j, j = 1,..,L, and the number of patterns forming $N(\mathbf{X}^*)$. If the classifier outputs can be regarded as estimates of the class posterior probabilities, these probabilities can be taken into account in order to improve the estimation of CLA [5]. The classifier exhibiting the maximum value of CLA is then identified. It is selected to classify the test pattern if such a CLA value is substantially higher than the CLA values of the other classifiers (a "selection threshold" is fixed). Otherwise, the test pattern $\mathbf{X}^*$ is classified by majority voting.

It is worth noting that DCS methods previously presented in the literature defined the local region used for CLA computation only in terms of k-nearest neighbours [3-5]. The appropriate value of the "k" parameter was decided by experiments. It is clearly seen that to identify a single value of "k" that can work well for all test patterns is very difficult. Accordingly, DCS performances were usually greatly affected by the choice of the "k" parameter. On the contrary, in our DCS method, the use of MCB information allows us to dynamically adapt the value of the "k" parameter to the test pattern. In particular, the "k" value is increased or decreased depending on the degree of similarity between the MCB of the test pattern and those of the neighbouring training patterns. It may be said that the neighbourhood of the test pattern is defined using both "spatial" proximity and "classification" proximity. As reported in the next Section, with this approach DCS performances are poorly affected by the choice of the "k" parameter.

3. Experimental Results

Experiments were carried out using two data sets contained in the public domain data base named ELENA (Enhanced Learning for Evolutive Neural Architecture): the Phoneme_CR data set (French phoneme data) and the Satimage_CR data set (remote sensing images acquired by the LANDSAT satellite). In our experiments, we used the same data classes, features, and numbers of training and test patterns as in [4]. An MCS made up of three different classifiers was used (Table 1): the k nearest neighbours classifier (k-nn), the multilayer perceptron (MLP) neural network, and the quadratic Bayes classifier. For the sake of brevity, the reader is referred to [4] for further details on the design of these classifiers. According to [4], we randomly subdivided each data set into two equal partitions, keeping the class distributions similar to that of the full data set. Each partition was first used as a training set and then as test set. In Tables 1 and 2, the classification accuracies and the kappa coefficient values are reported as the averages of two such trials.

Table 1
Percentage accuracies and "kappa" coefficient values provided by the three classifiers.

	Phoneme_CR		**Satimage_CR**	
Classifier	**% Accuracy**	**kappa**	**% Accuracy**	**kappa**
MLP neural network	86.29	0.67	84.20	0.80
Quadratic Bayes	78.66	0.52	85.71	0.82
k-nn	87.77	0.70	87.59	0.85

Table 2 shows the performances of our selection method based on MCB and those of two combination methods, namely, the BKS method and the majority voting rule. For the sake of comparison, the performances of the best individual classifier and the "oracle" are also shown. The "oracle" is the ideal selector that always chooses the classifier, if any, with the correct classification.

It is worth noting that as proposed in [5] we computed CLAs by exploiting the class posterior probabilities provided by three considered classifiers. Concerning the initial size of the "neighbourhood" used for CLA estimates, we ran experiments with "k" values ranging from one to fifty-one using the Euclidean distance metric. This neighbourhood was defined with respect to training data. The accuracy of the proposed DCS method shown in Table 2 is the maximum accuracy obtained by varying the value of "k" within the considered range. However, it is worth noting that, thanks to the use of MCB information, performances were poorly affected by the value of the "k" parameter. The average accuracies on the Phoneme_CR and Satimage_CR data sets were 88.69% and 89.25%, respectively, with standard deviations of $2.75\ 10^{-3}$ and $4.83\ 10^{-3}$. Concerning the similarity threshold, for each test pattern, this was taken equal to the minimum similarity value exhibited by the k-nearest neighbours.

Table 2 shows that our DCS method based on MCB always outperformed the BKS combination method, thus suggesting the usefulness of exploiting MCB information for classifier selection purposes. The accuracies provided by MCB-based DCS were also better than those provided by the combination using the majority voting rule and the best individual classifier.

We also assessed the statistical significance of the results reported in Table 2. Table 3 shows the values of the Zeta test related to the statistical significance of the differences in accuracy between our selection method, the BKS combination method, the majority rule, and the best individual classifier. Such differences are very significant, since Zeta Statistics values larger than 1.75 imply a degree of significance greater than 92%.

Table 2
Percentage accuracies and "kappa" coefficient values of the proposed DCS method, the majority rule, the best classifier and the "oracle". Obviously the Kappa coefficient value of the oracle could not be computed.

	Phoneme		**Satimage**	
Classification Algorithm	**% Accuracy**	**kappa**	**% Accuracy**	**kappa**
Best classifier	87.77	0.70	87.59	0.85
Oracle	95.75	-	94.14	-
MCB-based DCS	89.23	0.74	90.09	0.88
BKS	87.98	0.71	87.43	0.84
Majority Rule	87.47	0.70	88.63	0.86

Table 3
Values of the Zeta test related to the statistical significance of the differences in accuracy between our selection method, the BKS method, the majority rule, and the best individual classifier.

Zeta test	Best classifier	BKS	Majority Rule
Phoneme_CR MCB-based DCS	1.75	1.51	1.75
Satimage_CR MCB-based DCS	3.53	3.40	1.94

References

[1] Kittler, J., Roli, F.: Proc. of the First International Workshop on Multiple Classifier Systems (MCS 2000). Springer-Verlag Pub., Lecture Notes in Computer Science, Vol. 1857, J.Kittler and F.Roli Eds., (2000) 1-404.

[2] Huang, Y.S. and Suen, C.Y.: A method of combining multiple experts for the recognition of unconstrained handwritten numerals. IEEE Trans. On Pattern Analysis and Machine Intelligence 17(1) (1995) 90-94.

[3] Giacinto, G., and Roli, F.: Adaptive Selection of Image Classifiers. Proc. of the 9th ICIAP, Lecture Notes in Computer Science 1310, Springer Verlag, (1997) 38-45.

[4] Woods, K., Kegelmeyer, W.P., and Bowyer, K.: Combination of multiple classifiers using local accuracy estimates. IEEE Trans. on Pattern Analysis and Machine Intelligence 19(4) (1997) 405-410.

[5] Giacinto, G., Roli F. and Fumera G.: Selection of Image Classifiers. Electronics Letters, 36(5) (2000) 420-422.