Selection of Classifiers Based on Multiple Classifier Behaviour

Giorgio Giacinto, Fabio Roli, and Giorgio Fumera

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

Phone +39-070-6755862 Fax +39-070-6755900
e-mails {giacinto,roli,fumera}@diee.unica.it

Abstract. In the field of pattern recognition, the concept of Multiple Classifier Systems (MCSs) was proposed as a method for the development of high performance classification systems. At present, the common "operation" mechanism of MCSs is the "combination" of classifiers outputs. Recently, some researchers pointed out the potentialities of "dynamic classifier selection" (DCS) as a new operation mechanism. In this paper, a DCS algorithm based on the MCS behaviour is presented. The proposed method is aimed to exploit the behaviour of the MCS in order to select, for each test pattern, the classifier that is more likely to provide the correct classification. Reported results on the classification of different data sets show that dynamic classifier selection based on MCS behaviour is an effective operation mechanism for MCSs.

Keywords: Multiple Classifier Systems, Combination of Classifiers, Dynamic Classifier Selection, Image Classification

1. Introduction

In the field of pattern recognition, a number of multiple classifier systems (MCSs) based on the combination of outputs of a set of different classifiers have been proposed [1]. For each pattern, the classification process is performed in parallel by different classifiers and the results are then combined. Many combination methods, e.g., voting, Bayesian and Dempster-Shafer approaches, are based on "decision fusion" techniques that combine the classifications provided by different classifiers [1]. As an example, the "majority" voting rule interprets each classification result as a "vote" for one of the data classes and assigns the input pattern to the class receiving the majority of votes. Such methods are able to improve the classification accuracy of individual classifiers under the assumption that different classifiers make "independent" errors. However, in real pattern recognition applications, it is usually difficult to design a set of classifiers that exhibit an independent behaviour on the whole feature space. In order to avoid the independence assumption, Huang and Suen proposed a combination method, named "Behaviour Knowledge Space" (BKS), that exploits the behaviour of the MCS [2]. The behaviour of the MCS for each training pattern is recorded as a vector whose elements are the decisions of the classifiers of the MCS. For each unknown test pattern the MCS behaviour is considered, and the

F.J. Ferri et al. (Eds.): SSPR&SPR 2000, LNCS 1876, pp. 87-93, 2000.

training patterns that exhibit the same MCS behaviour are identified. The unknown pattern is then assigned to the class most represented among such training patterns.

In this paper, the MCS behaviour is exploited in order to perform a "dynamic classifier selection" (DCS) aimed to select, for each unknown pattern, the classifier that is more likely to classify it correctly [3-5]. The rationale behind this procedure can be explained by observing that it is easy to design an MCS where, for each pattern, there is at least one classifier that classifies it correctly. In order to select this classifier, the training patterns with the same MCS behaviour are considered and the classifier with the highest accuracy is chosen. In Section 2, the concept of MCS behaviour is defined and a selection function is presented. Experimental results and comparisons are reported in Section 3.

2. Dynamic Classifier Selection Based on MCS 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 characterized by a feature vector $\mathbf{X}$. Let us also assume that K different classifiers, C_j, $j = 1,..,K$, have been trained separately to solve the image classification task at hand. Let $C_j(\mathbf{X}) \in \{1,..,M\}$ indicate the class label assigned to pattern $\mathbf{X}$ by classifier C_j.

2.2 Multiple Classifier Behaviour

For each test pattern $\mathbf{X}^*$, a vector made up of K elements $C_j(\mathbf{X}^*)$ is available. Let us indicate with $MCB(X^*) = \{C_1(\mathbf{X}^*), C_2(\mathbf{X}^*),.., C_K(\mathbf{X}^*)\}$ the vector that represents the "Multiple Classifier Behaviour" (MCB) for pattern $\mathbf{X}^*$. $MCB(\mathbf{X}^*)$ represents the behaviour of the set of classifiers for the considered pattern.

It is worth noting that also the Behaviour Knowledge Space proposed by Huang and Suen tries to exploit the information contained in the behaviour vector [2]. However, while the goal of BKS is to *combine* the results of different classifiers, the proposed method is aimed to *select* the classifier out of K that is more able to correctly classify the pattern $\mathbf{X}^*$. To this end, let us consider the subset of the training patterns with the same $MCB(\mathbf{X})$ of the test pattern $\mathbf{X}^*$. In other words, we are considering all the training patterns $\mathbf{X}$ that satisfy the condition $C_j(\mathbf{X}) = C_j(\mathbf{X}^*)$, $\forall j = 1,..,K$. Let us indicate this subset of the training patterns with $S(\mathbf{X}^*)$. The goal of our procedure is to select the most accurate classifier out of K by taking into account the behaviour of the patterns in $S(\mathbf{X}^*)$. To this end, for each classifier, the classification accuracy related to the patterns in $S(\mathbf{X}^*)$ is computed. As an example, such classification accuracy can be obtained as the fraction of correctly classified patterns belonging to $S(\mathbf{X}^*)$. The K classifiers are then ranked according to the measure of

classification accuracy and the one with the highest accuracy is then chosen to classify the unknown pattern **X***.

If a given MCB(**X***) is not exhibited by any training patterns, then S(**X***) can be made up of the subset of training patterns whose MCB(**X**) differ from MCB(**X***) for a limited number of elements $c < K$. Thus S(**X***) can contain patterns whose MCS behaviour is *similar* to the one exhibited by the unknown pattern.

The above procedure is based on the assumption that, for all the patterns **X** that exhibit the same MCS behaviour, there exists at least one classifier that is able to classify them correctly. In other words, let us consider each classifier in the MCS as an "expert". When the experts disagree, we consider all the known cases where the experts exhibited the same disagreement and we select the expert who exhibit the highest accuracy for such cases.

2.3 A Measure of Classifier Accuracy

In this section a measure of classifier accuracy that takes into account the uncertainties in the classification process, is proposed. Let us assume that the classifier C_j assigns the test pattern **X*** to the data class ω_i. We indicate this by $C_j(\mathbf{X}^*) = \iota$. It is easy to see that the accuracy of classifier C_j in S(**X***) can be estimated as the fraction of patterns belonging to S(**X***) assigned to class ω_i by C_j that have been correctly classified. However, if the classifier provides estimates of the class posterior probabilities, we propose to take these probabilities into account in order to inprove the estimation of the above measure of classifier accuracy (CA). Given a pattern $\mathbf{X} \in \omega_i$, i = 1,..,M, belonging to S(**X***), the $\hat{P}_j(\omega_i \mid \mathbf{X})$ provided by the classifier C_j can be regarded as a measure of the classifier accuracy for the pattern **X**. CA can then be estimated by computing the probability that the test pattern **X*** is correctly assigned to class ω_i by the classifier C_j. According to the Bayes theorem, this probability can be estimated as follows:

$$\hat{P}\left(\mathbf{X}^* \in \omega_i \mid C_j\left(\mathbf{X}^*\right)=i\right)=\frac{\hat{P}\left(C_j(\mathbf{X}^*)=i \mid \mathbf{X}^* \in \omega_i\right)\hat{P}(\omega_i)}{\sum_{m=1}^{M}\hat{P}\left(C_j(\mathbf{X}^*)=i \mid \mathbf{X}^* \in \omega_m\right)\hat{P}(\omega_m)} \tag{1}$$

where $\hat{P}\left(C_j(\mathbf{X}^*)=i \mid \mathbf{X}^* \in \omega_i\right)$ is the probability that the classifier C_j classifies the patterns belonging to class ω_i correctly. This probability can be estimated by averaging the posterior probabilities $\hat{P}_j(\omega_i \mid \mathbf{X}_n \in \omega_i)$ provided by the classifier C_j on the training patterns $\mathbf{X}_n$ in S(**X***) that belong to the class ω_i. In other words, if N_i is the number of patterns in S(**X***) that belong to the class ω_i, then

$$\hat{P}\left(C_j(\mathbf{X}^*)=i \mid \mathbf{X}^* \in \omega_i\right)=\frac{\hat{P}_j(\omega_i \mid \mathbf{X}_n \in \omega_i)}{N_i} \tag{2}$$

The prior probabilities $\hat{P}(\omega_i)$ can be estimated as the fraction of patterns in S($\mathbf{X}$*) that belong to class ω_i. If we let N be the total number of patterns belonging to S($\mathbf{X}$*), then

$$\hat{P}(\omega_i) = {}^{N_i}\!/_{N} \tag{3}$$

Therefore, substituting equations (2) and (3) in equation (1) the following estimate of CA for classifier C_j is obtained:

$$CA_j(\mathbf{X}^*) = \frac{\sum_{\mathbf{X}_n \in \omega_i} \hat{P}_j(\omega_i \mid \mathbf{X}_n) \cdot W_n}{\sum_{m=1}^{M} \sum_{\mathbf{X}_n \in \omega_m} \hat{P}_j(\omega_i \mid \mathbf{X}_n) \cdot W_n} \tag{4}$$

where, in order to handle the "uncertainty" in the size of S($\mathbf{X}$*), the class posterior probabilities can be "weighted" by a term $W_n = 1/ d_n$, where d_n is the Euclidean distance of the pattern $\mathbf{X}_n$ belonging to S($\mathbf{X}$*) from the test pattern $\mathbf{X}$*.

2.4 An Algorithm for DCS Based on MCS Behaviour

In the following, a dynamic classifier selection algorithm is described.

Input parameters: test pattern $\mathbf{X}$*, MCB($\mathbf{X}$) for the training data, the rejection threshold value, and the selection threshold value

Output: classification of the test pattern $\mathbf{X}$*

STEP 1: Compute MCB($\mathbf{X}$*). If all the classifiers assign $\mathbf{X}$* to the same data class, then the pattern is assigned to this class.

STEP 2: Identify the training patterns whose MCB($\mathbf{X}$) = MCB($\mathbf{X}$*).

STEP 3: Compute $CA_j(\mathbf{X}^*)$, j = 1,…,K

STEP 4: ***If*** $CA_j(\mathbf{X}^*) <$ rejection-threshold ***Then*** Disregard classifier C_j

STEP 5: Identify the classifier C_m exhibiting the maximum value of $CA_j(\mathbf{X}^*)$

STEP 6: For each classifier C_j, compute the following differences $d_j = \left[CA_m(\mathbf{X}^*) - CA_j(\mathbf{X}^*)\right]$

STEP 7: ***If*** $\forall\, j, j \neq m, d_j >$ selection-threshold ***Then*** Select Classifier C_m
Else Select randomly one of the classifiers for which $d_j <$ selection-threshold

Step 3 identify the training patterns that make up S($\mathbf{X}$*). If this set is empty, the training patterns with a MCB($\mathbf{X}$) that differs from MCB($\mathbf{X}$*) for $c <$ K elements can be included in S($\mathbf{X}$*).

Step 4 is aimed at excluding from the selection process the classifiers that exhibit CA values smaller than the given rejection threshold.

Step 6 computes the differences d_l in order to evaluate the "reliability" of the selection of the classifier C_m. If all the differences are higher than the given selection-threshold, then it is reasonably "reliable" that classifier C_m should correctly classify the test pattern $\mathbf{X}$*. Differently, a random selection is performed among the classifiers

for which $d_j <$ selection-threshold. Alternatively, random selection can be substituted by the combination of these classifiers.

3. Experimental Results

Experiments have been carried out using three data sets contained in the ELENA (Enhanced Learning for Evolutive Neural Architecture) data base. In particular, we used the following data sets: phoneme_CR (French phoneme data), satimage_CR (remote sensing images acquired by the LANDSAT satellite), and texture_CR (images of the Brodatz's textures). Further details on these data sets can be found via anonymous ftp at *ftp.dice.ucl.ac.be* in the directory *pub/neural-nets/ELENA/databases*. In our experiments, we used the same data classes, features, and numbers of training and test patterns used in [4].

A set made up of five different classifiers was used (Table 1): the k nearest neighbours classifier, the multilayer perceptron (MLP) neural network, the C4.5 decision tree [6], the quadratic Bayes classifier (QB) and the linear Bayes classifier (LB). For the sake of brevity, we refer the reader interested in more details on the design of these classifiers to [4]. Tables 1 and 2 show the percentage accuracies of the individual classifiers for the data sets used. We randomly partitioned each data set into two equal partitions, keeping the class distributions similar to that of the full data set. Each partition was firstly used as training set and then as test set. In Table 1, the accuracies for each trial are reported, while in Table 2 the accuracies are reported as the average of the two results.

Table 1. Percentage accuracies provided by the five classifiers applied to the ELENA data sets. Results obtained on each of the two partitions of the data sets are reported.

	Phoneme		**Satimage**		**Texture**	
Classifier	**Trial 1**	**Trial 2**	**Trial 1**	**Trial 2**	**Trial 1**	**Trial 2**
k-nn	86.38	89.16	88.11	87.06	97.75	97.75
MLP	86.79	85.79	85.62	82.77	98.51	98.51
C4.5	83.72	85.83	85.78	85.54	91.38	90.51
QB	78.91	78.42	85.93	85.48	98.87	99.20
LB	77.13	75.42	83.35	81.97	97.56	97.27

Table 2. Average percentage accuracies provided by the five classifiers applied to the ELENA data sets.

Classifier	**Phoneme**	**Satimage**	**Texture**
k-nn	87.77%	87.59%	97.75%
MLP	86.29%	84.20%	98.51%
C4.5	84.78%	85.66%	90.95%
QB	75.41%	85.78%	99.04%
LB	73.00%	83.31%	97.42%

Tables 3 and 4 show the performances of the proposed selection method (DCS-MCB) and the performances of the combination method based on the majority voting rule. For comparison purposes, the performances of the best individual classifier and the "oracle" are also shown. The "oracle" is the ideal selector which always chooses the classifier providing the correct classification if any of the individual classifier does so.

Table 3. Percentage accuracies provided by the proposed DCS method (DCS-MCB), the combination by majority voting rule, the best classifier of the ensemble, and the oracle. Results obtained on each of the two partitions of the data sets are reported.

	Phoneme		**Satimage**		**Texture**	
Classifier	**Trial 1**	**Trial 2**	**Trial 1**	**Trial 2**	**Trial 1**	**Trial 2**
Oracle	97.52	97.08	95.99	95.71	99.93	99.93
Best classifier	86.79	89.16	88.11	87.06	98.87	99.20
DCS-MCB	87.75	93.34	88.39	89.49	98.89	99.67
Majority rule	86.16	92.23	88.31	90.32	99.24	99.20

The DCS-MCB method always outperformed the best classifier of the ensemble, so pointing out that dynamic classifier selection is a method for improving the accuracies of individual classifiers. Accuracies provided by combination-based MCSs are sometimes better than the ones of selection-based MCSs. This result is very reasonable, as classifiers very "different", and, therefore, very "independent" were used in these experiments. However, our method outperformed the majority rule combination method in the most of experiments.

Table 4. Average percentage accuracies provided by the proposed DCS method (DCS-MCB), the combination by majority voting rule, the best classifier of the ensemble, and the oracle.

Classifier	**Phoneme**	**Satimage**	**Texture**
Oracle	97.30	95.85	99.93
Best classifier	87.77	87.59	99.04
DCS-MCB	90.55	88.94	99.28
Majority rule	89.20	89.32	99.22

4. Conclusions

In this paper, we have addressed the "open" research topic of selection-based MCSs. In particular, we presented a dynamic classifier selection method aimed at selecting, for each unknown pattern, the most accurate classifier of the MCS on the basis of the MCS behaviour on the unknown test pattern.

Reported results showed that dynamic classifier selection based on MCS behaviour always outperforms the best classifier in the ensemble. In addition, our selector exhibited performances that are close or better than the ones exhibited by the majority voting combination.

Acknowledgements

The authors wish to thank K.Woods, W.P. Kegelmeyer, and K.Bowyer which provided them with detailed information on the data set used in [4].

References

1. Xu, L., Krzyzak, A., and Suen, C.Y.: Methods for combining multiple classifiers and their applications to handwriting recognition. IEEE Trans. on Systems, Man, and Cyb., 22(3) (1992) 418-435
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 International Conference on Image Analysis and Processing, Lecture Notes in Computer Science 1310, Springer Verlag Ed. (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. and Roli F.: Methods for Dynamic Classifier Selection. 10th International Conference on Image Analysis and Processing, Venice, Italy (1999), 659-664
6. Quinlan, J.R.: C4.5 Programs for Machine Learning. Morgan Kaufmann, (1992)