Automatic Design of Multiple Classifier Systems by Unsupervised Learning

Giorgio Giacinto and Fabio Roli

Department of Electrical and Electronic Engineering, University of Cagliari, Italy
Piazza D'Armi, 09123, Cagliari, Italy
Phone: +39-070-6755874 Fax: +39-070-6755900
{giacinto,roli}@diee.unica.it

Abstract. In the field of pattern recognition, multiple classifier systems based on the combination of the outputs of a set of different classifiers have been proposed as a method for the development of high performance classification systems. Previous work clearly showed that multiple classifier systems are effective only if the classifiers forming them make independent errors. This achievement pointed out the fundamental need for methods aimed to design ensembles of "independent" classifiers. However, the most of the recent work focused on the development of combination methods. In this paper, an approach to the automatic design of multiple classifier systems based on unsupervised learning is proposed. Given an initial set of classifiers, such approach is aimed to identify the largest subset of "independent" classifiers. A proof of the optimality of the proposed approach is given. Reported results on the classification of remote sensing images show that this approach allows one to design effective multiple classifier systems.

1. Introduction

In the field of pattern recognition, multiple classifier systems (MCSs) based on the combination of the outputs of a set of different classifiers have been proposed as a method for the development of high performance classification systems. Typically, the combination is based on voting rules, belief functions, statistical techniques, Dempster-Shafer evidence theory, and other integration schemes [1, 2]. The most of such combination methods assume that the classifiers forming the MCS make independent classification errors. This assumption is necessary to guarantee an increase of classification accuracy with respect to the accuracies provided by classifiers forming the MCS. As an example, Hansen and Salamon showed that a MCS based on a simple "majority" combination rule can provide very good increases of accuracy if classifiers make independent errors [3]. Tumer and Ghosh pointed out that accuracy increases depend on error uncorrelation much more that on the particular combination method adopted [4].

The above-mentioned achievements pointed out the fundamental need for methods aimed to design ensembles of independent classifiers. However, in the pattern recognition field, the most of the work focused on the development of combination

methods. Some papers addressing the problem of the design of an ensemble of independent nets appeared in the neural networks literature [5]. However, the results of such work can be exploited only in part for the MCSs formed by different classifiers (e.g., statistical and neural classifiers). An overview of the work related to the design of MCSs is given in Section 2.

In this paper, an approach to the automatic design of MCSs formed by different classification algorithms is proposed (Section 3). Instead of attempting to design a set of "independent" classifiers directly, a large set, usually containing independent but also correlated classifiers, is initially created (Section 3.1). Given such a set, our approach is aimed to identify the largest subset of independent classifiers by an unsupervised learning algorithm (Section 3.2). We also point out the rationale behind the proposed approach and prove the optimality of our design method (Section 3.3). Experimental results and comparisons are reported in Section 4. Conclusions are drawn in Section 5.

2. Related Work

In the pattern recognition literature, to the best of our knowledge, no work directly addressed the problem of designing ensembles of independent classifiers. Some papers indirectly addressed it by proposing combination methods that that do not need of the assumption of independent classifiers [6-8]. It is easy to see that such methods exhibit advantages from the viewpoint of the design of the MCSs. However, the related combination functions are much more complex than the ones based on the “independence” assumption. In addition, the theory developed for ensembles of independent classifiers cannot be exploited to evaluate the performances of the MCSs based on these methods. As an example, it is not possible to assume that the error rate is monotonically decreasing in the number of the combined classifiers [3].

Research work addressing the problem of designing an ensemble of "independent" nets has been carried out in the neural networks field. Earlier studies investigated the effectiveness of different “design parameters” for creating independent neural nets [5]. In particular, Partridge quantified the relative impact of the major parameters used in the design of a neural network and he found the following “ordering”: “net type”, “training set structure”, “training set elements”, “number of hidden units”, and “weight seed” [9]. Recently, it seems to the authors that two main strategies for designing an ensemble of independent nets emerged from the neural networks literature [5]. One, that can be named "overproduce and choose" strategy, is based on the creation of an initial large set of nets and a subsequent choice of an "optimal" subset of independent nets. The other strategy attempts to generate a set of independent nets directly. Partridge and Yates described a design method for neural network ensembles based on the "overproduce and choose" strategy [10]. They introduced some interesting "diversity" measures that can be used for choosing an "optimal" subset of independent classifiers. However, they did not propose a systematic method for choosing such a set. Only an experimental investigation of three possible techniques is described. In addition, the problem of the optimality of such "choose" techniques is not addressed. Opitz and Shavlik presented and

algorithm called ADDEMUP that uses genetic algorithms to search actively for a set of independent neural networks [11]. Rosen described a method that allows one to train individual networks by backpropagation not only to reproduce a desired output, but also to have their errors linearly decorrelated with the other networks [12]. Individual networks so trained are then linearly combined.

However, the results of the above research work can be exploited only in part for MCSs formed by different classifiers (e.g., statistical and neural classifiers). As an example, the "diversity" measures proposed by Partridge and Yates can be exploited in general, while the work of Rosen is tailored to neural network ensembles. In addition, to the best of our knowledge, no work addressed the problem of the optimality of the design method proposed.

3. Automatic Design of MCSs by Unsupervised Learning

3.1 Background and Basic Concepts of the Proposed Approach

First of all, let us formulate briefly the task of the design of a MCS. In general, such task can be subdivided into two subtasks: the design of the "members" of the MCS, and the design of the combination function.

It is worth remarking that, in this paper, we address the problem of the design for MCSs based on combination functions that assume the independence of classifiers. Therefore, the design task basically consists of finding a set of independent classifiers as large as possible. Given such a set, a simple majority rule is sufficient to design an effective MCS. (Hansen and Salamon showed that the error rate of such a MCS goes to zero in the limit of infinite set size [3]).

Among the two main design strategies recently defined in the neural networks literature, our approach follows the so called “overproduce and choose” strategy (see Section 2). The rationale behind this choice is that we think that the direct design of only independent classifiers is a very difficult problem that is beyond the current state of the theory of MCSs. In addition, the overproduce and choose strategy seems to fit well with the novel paradigm of “weak” classifiers (i.e., the creation of very large sets of classifiers which can do a little better than making random guesses [13]). Also the interesting paradigm of “reusable” classifiers recently introduced by Bollacker and Ghosh might be exploited for the creation of large sets of classifiers [14]. Finally, the overproduce and choose strategy is successfully used in other fields (e.g., in the field of the software engineering [15].).

With regard to the overproduction phase, we basically extended to MCSs the conclusions of Partridge concerning the design parameters that maximize the independence for neural network ensembles [9]. In particular, we basically create the initial set of classifiers using different classification algorithms, as the “classifier type” is the best design parameter according to Partridge.

Concerning the choose phase, first of all, let C be the set of the N classifiers generated by the overproduction phase:

$$C = \{c_1, c_2, \ldots, c_N\}. \quad (1)$$

The rationale behind our approach is based on the following assumptions on such set C (equations 2, 3, and 4).

Let us assume that C is formed by the following union of M subsets C_i:

$$C = \bigcup_{i=1}^{M} C_i \quad (2)$$

where the subsets C_i meet the following assumption:

$$\forall i, j \; i \neq j \; C_i \cap C_j = \varnothing \quad (3)$$

and the classifiers forming the above subsets satisfy the following requirements:

$$\forall c_l, c_m \in C_i, \forall c_n \in C_j, \forall i, j \; i \neq j \; \text{prob}(c_l \text{ fails}, c_m \text{ fails}) > \text{prob}(c_l \text{ fails}, c_n \text{ fails}). \quad (4)$$

In the above equation, the terms prob(c_l fails, c_m fails) and prob(c_l fails, c_n fails) state for the compound error probabilities of the related classifier couples. Such error probabilities can be estimated by the number of coincident errors made by the couples of classifiers on a validation set.

Equation 4 simply states that the compound error probability between any two classifiers belonging to the same subset is higher than the one between any two classifiers belonging to different subsets. Consequently, theoretically speaking, the M subsets forming C can be identified by any "clustering" algorithm grouping the classifiers on the basis of the compound error probability [16].

After the identification of the subsets C_i, i=1...M, our approach takes one classifier from each subset in order to create the largest subset $C^*=\{c^*_1, c^*_2, \ldots, c^*_M\}$ containing only independent classifiers. (Or the subset C* of the most independent classifiers, if the complete independence cannot be obtained).

It can be seen that the creation of the "optimal" subset C* is as much difficult as the number of the possible subsets to be considered is large. (In Section 3.3, some additional hypotheses that allows one to guarantee the creation of the optimal subset C* are given).

According to the above hypotheses, the subset C* is the best solution for our design task, as it contains the largest subset of independent classifiers, or the subset of the most independent classifiers, contained into the initial set C.

Finally, it is worth doing the following remarks on the proposed approach:

- The above hypotheses on the set C are in agreement with real cases related to the "production" of classifier ensembles. As an example, a neural network ensemble obtained by trainings with different weight seeds is likely to meet the assumptions of equations 1-4 due to the common problem of "local minima". (We can assume that the subsets C_i are formed by nets related to the different local minima);
- The rationale behind the "clustering-based" approach to the identification of the set C* is analogous to the one behind the "region-based" approach to image segmentation. It can be convenient to group "similar" pixels in order to identify the most "independent" ones (i.e., the edge pixels);

- The identification of classifier "clusters" allows one to highlight cases of "unbalanced" ensembles where, for example, there is a majority of "correlated" classifiers that negatively influences the ensemble performances.

3.2 The Proposed Approach

As described in the previous section, the proposed approach is constituted by two main phases: the overproduction and the choose phases. In this section, we give further details on the choose phase. With regard to the overproduction phase, let us assume that the set C has been generated according to the strategy outlined in Section 3.1.

The choose phase is subdivided into the following main stages:

- Unsupervised learning for identifying the subsets C_i, i=1...M
- Creation of the subset C*

Unsupervised Learning for Subsets Identification

This stage is implemented by an unsupervised learning algorithm that, according to equation 4, basically groups the classifiers belonging to the set C on the basis of the compound error probability. In particular, a hierarchical agglomerative clustering algorithm is used [16]. Such algorithm starts assigning each of the N classifiers to an individual cluster. Then, two or more of these trivial clusters are merged, thus nesting the trivial clustering into a second partition. The process is repeated to form a sequence of nested clusters. The stop criterion is based on the analysis of the so called "dendogram". The reader interested in more details about hierarchical agglomerative clustering is referred to [16].

In order to understand better this stage of our approach, it is worth remarking the analogy with the well known problem of "data clustering" [16]. The classifiers belonging to the set C play the roles of the "data" and the subsets C_i represent the data "clusters". Analogously, the compound error probability among couples of classifiers plays the role of the distance measure used in data clustering. In particular, in order to perform such a clustering of classifiers, it is easy to see that two "distance" measures are necessary: a distance measure between two classifiers and a distance measure between two clusters of classifiers. We defined the first measure on the basis of the compound error probability:

$$\forall\, c_s,\, c_t \in C \quad d(c_s,\, c_t) = 1 - \text{prob}(c_s \text{ fails},\, c_t \text{ fails}). \tag{5}$$

According to equation 5, two classifiers are as more distant as more they do not make coincident errors. Therefore, the above distance measure groups classifiers that make coincident errors and assigns independent classifiers to different clusters.

The "distance" between two clusters was defined as the maximum "distance" between two classifiers belonging to such clusters:

$$\forall\, C_i,\, C_j \; i = 1...M,\; j = 1...M \; i \neq j \quad d(C_i,\, C_j) = \max_{c_s \in C_i,\, c_t \in C_j} \{d(c_s,\, c_t)\}. \tag{6}$$

The rationale behind equation 6 can be seen by observing that two clusters containing two independent classifiers must not be merged (even if the other classifiers belonging to such clusters are very correlated), as the subset C* is formed by extracting one classifiers from each cluster. (It is worth also noticing that the same kind of distance measure is also used for data clustering purposes [16]).

It is easy to see that equation 6 can be used also for measuring the distance between a classifier and a cluster previously formed.

Finally, it is worth also noticing that our method computes all the above distance measures with respect to a validation set in order to avoid "overfitting" problems.

Creation of the Subset C*

The subset C* is created by taking one classifier from each cluster C_i. In particular, for each classifier of a given cluster, the average distance from all the other clusters is computed. The classifier characterized by the maximum distance is chosen. The set C* is formed by repeating this procedure for each subset C_i.

3.3 Optimality of the Proposed Approach

Given the above defined set C, let us assume that:

$$\forall c_i \in C,\ i = 1...N,\ \text{prob}(c_i \text{ fails}) = p, \quad p < 0.5 \tag{7}$$

$$\forall C_i, i = 1...M,\ C_i = \{c_{i1}, c_{i2}, ..., c_{in_i}\}, n_i < N \quad \text{prob}(c_{i1} \text{ fails}, c_{i2} \text{ fails}, ..., c_{in_i} \text{ fails}) = p \tag{8}$$

$$\forall c_1 \in C_1,\ \forall c_2 \in C_2, ..., \forall c_M \in C_M \quad \text{prob}(c_1 \text{ fails},\ c_2 \text{ fails}, ..., c_M \text{ fails}) = p^M. \tag{9}$$

Equation 7 assumes that all the classifiers belonging to the set C exhibit the same error probability. Equations 8 implies that the classifiers belonging to a given subset make exactly the same errors (i.e., they are completely correlated with respect to the classification errors). According to equation 9, classifiers belonging to different subsets are independent.

Given the above hypotheses, we can prove that the following equation is satisfied:

$$\forall S \subseteq C, S \neq C^*\ \text{p(MCS(S) fails)} \geq \text{p(MCS(C*) fails)} \tag{10}$$

where p(MCS(S) fails) and p(MCS(C*) fails) state for the error probabilities of the MCSs based on the sets S and C*, respectively. C* is the subset of C extracted by our design approach, that is, the set formed by one classifier for each subset C_i (Section 3.2). The majority rule combination function is assumed for such MCSs. (Hereafter the majority rule is always assumed).

The optimality of our design method is proved by equation 10, as such equation states that any subset of C different from C* exhibits a higher error probability. Consequently, C* is the largest subset of independent classifiers contained into the set C.

Proof of Equation 10

Without loosing in generality, we can assume that the subset S mentioned in equation 10 is formed according to one of the following ways:

- by subtracting some classifiers from C*;
- by adding to C* some classifiers taken from the set (C - C*);
- by using both of the two previous ways.

(It is worth noticing that any subset S can be formed according to the above strategy).

Firstly, let us consider the case that the subset S is formed by subtracting some classifiers from the set C*. In this case, the proof comes directly from the following achievement of Hansen and Salamon: the error rate of a MCS based on the majority rule is monotonically decreasing in the number of the independent classifiers combined [3]. Consequently, as the set C* is formed by a number of independent classifiers, subtracting some classifiers from C* surely increases the error rate.

Secondly, let us consider the case that the subset S is formed by adding to C* some classifiers taken from the set (C - C*).

First of all, let us point out that, according to Hansen and Salamon, the error probability of the MCS based on the set C* can be computed as follows:

$$p(MCS(C^*)\,fails) = \sum_{K>\frac{M}{2}}^{M} \binom{M}{K} p^K (1-p)^{M-K} . \qquad (11)$$

Without loosing in generality, let us assume to add some classifiers to the set C* so that the cardinality of the set S is "m", M<m≤N.

It should be remarked that the classifiers added to the set C* necessarily belong to the subsets C_i. Consequently, the set S can be regarded as formed by M clusters. It is worth noticing that the set C* can be also regarded as formed by M clusters. The basic difference with respect to the set S is that such clusters can contain only one classifier. It should be also remarked that equation 8 still holds for the clusters of the set S obtained from C* by adding classifiers.

In order to compute the value of the p(MCS(S) fails), we can still use equation 11 by observing that the set S is constituted by M clusters of classifiers completely correlated. This implies that, from the viewpoint of the error probability, any cluster can be regarded as a single classifier with a value of the error probability equal to "p". On the other hand, different clusters are independent according to equation 9. However, with respect to equation 11, it should be noticed that not all the combinations of M/2 clusters belonging to the set S contain a number of classifiers higher than m/2. In particular, the "majority", that is, m/2, can be obtained by numbers of clusters lower and higher than M/2.

Consequently, the value of the p(MCS(S) fails) can be computed as follows:

$$p(MCS(S)\,fails) = \sum_{K>\frac{M}{2}}^{M} \left(\binom{M}{K} - \alpha_K \right) p^K (1-p)^{M-K} + \sum_{K>\frac{M}{2}}^{M} \alpha_K p^{M-K} (1-p)^K \qquad (12)$$

where the terms α_K state for the number of the combinations of "K" clusters that do not contain a number of classifiers higher than m/2 (i.e., they do not contain a "majority").

It is easy to see that, for any combination of K clusters that do not contain a "majority", the remaining M-K clusters forming the set S necessarily contain a "majority". This is the reason for the term $\sum_{K>\frac{M}{2}}^{M} \alpha_K p^{M-K}(1-p)^K$ in equation 12.

Equation 12 can be rewritten as follows:

$$p(MCS(S)\,fails) = \sum_{K>\frac{M}{2}}^{M} \binom{M}{K} p^K (1-p)^{M-K} + \\ + \sum_{K>\frac{M}{2}}^{M} \alpha_K p^{M-K}(1-p)^{M-K}\left((1-p)^{2K-M} - p^{2K-M}\right) \quad (\mathbf{13})$$

As p<0.5, then p<1-p. In addition, 2K-M>0, as K>M/2. Consequently, the error probability in equation 12 is higher, or equal, than the one in equation 10.

Finally, let us consider the last possible way of forming the set S. The proof of equation 10 for this case is straightforward. We have already proved that the p(MCS(S) fails) increases by subtracting classifiers from C*. According to equation 13, it is also proved that such error probability further increases if classifiers taken from the set (C - C*) are then added.

This completes the proof of equation 10.

It is worth noticing that the assumptions of equations 7-9 are likely to be completely or partially met in many real cases of classifier ensembles. As an example, in neural network ensembles obtained by trainings with different weight seeds (see the remarks at the end of Section 3.1). The same holds for the ensembles of k-nearest neighbour classifiers, as subsets of very correlated classifiers are related to different ranges of the "k" parameter.

4. Experimental Results

4.1 The Data Set

The data set used for our experiments consists of a set of multisensor remote-sensing images related to an agricultural area near the village of Feltwell (UK). The images (each of 250 x 350 pixels) were acquired by two imaging sensors installed on an airplane: a multi-band optical sensor (an Airborne Thematic Mapper sensor) and a multi-channel radar sensor (a Synthetic Aperture Radar). More details about the selected data set can be found in [17, 18]. For our experiments, six bands of the optical sensors and nine channels of the radar sensor were selected. Therefore, we used a set of fifteen images. As the image classification process was carried out on a "pixel basis", each pixel was characterised by a fifteen-element "feature vector" containing the brightness values in the six optical bands and over the nine radar channels considered. For our experiments, we selected 10944 pixels belonging to five

agricultural classes (i.e., sugar beets, stubble, bare soil, potatoes, carrots) and randomly subdivided them into a training set (5124 pixels), a validation set (582 pixels), and a test set (5238 pixels). We used a small validation set in order to simulate real cases where validation data are difficult to be obtained. (Validation data are extracted from the training sets. Consequently, strong reductions of training sets are necessary to obtain large validation sets).

4.2 Experimentation Planning

Our experiments were mainly aimed to:

- evaluate the effectiveness of the proposed design approach;
- compare our approach with other design approaches proposed in the literature.

Concerning the first aim, we performed different "overproduction" phases, so creating different sets C. Such sets were formed using the following classification algorithms: a k-nearest neighbour (k-nn) classifier, a multilayer perceptron (MLP) neural network, a Radial Basis Functions (RBF) neural network, and a Probabilistic Neural Network (PNN). For each algorithm, a set of classifiers was created by varying the related design parameters (e.g., the network architecture, the "weight seed", the value of the "k" parameter for the k-nn classifier, and so on). In the following, for the sake of brevity, we report the results related to some of such sets C (here referred as C^1, C^2, C^3, and C^4):

- the set C^1 was formed by fifty MLPs. Five architectures with one or two hidden layers and various numbers of neurons per layer were used. For each architecture, ten trainings with different weight seeds were performed. All the networks had fifteen input units and five output units as the numbers of input features and data classes, respectively (Section 4.1);
- the set C^2 was formed by the same MLPs belonging to C^1 and by fourteen k-nn classifiers. The k-nn classifiers were obtained by varying the value of the "k" parameter in the following two ranges: (15, 17, 19, 21, 23, 25, 27) and (75, 77, 79, 81, 83, 85, 87);
- the set C^3 was formed by nineteen MLPS and one PNN. Two different architectures were used for the MLPSs (15-7-7-5 and 15-30-15-5). For the PNN, an a priori fixed value of the smoothing parameter equal to 0.1 was selected [19].
- the set C^4 was formed by the same MLPs belonging to C^3, three RBF neural networks, and one PNN.

With regard to the second aim of our experimentation, we compared our design method with two methods proposed by Partridge and Yates [10]. One is the "choose the best" method that, given an a priori fixed size of the set C*, choose the classifiers with the highest accuracies in order to form C*. The other is the so called "choose from subspaces" method that, for each classification algorithm, choose the classifier with the highest accuracy. (The term "subspace" is therefore referred to the subset of classifiers related to a given classification algorithm).

4.3 Results and Comparisons

Experimentation with the Set C^1

The main aim of this experiment was to evaluate the effectiveness of our approach for the design of neural network ensembles formed by a single kind of net. It is worth noticing that this is a difficult design task, as nets of the same type are poorly independent according to the Partridge results [9]. Our algorithm created a set C* formed by 7 MLPs belonging to three different architectures. This is a not obvious result, as the most obvious set C* should be formed by taking one net from each of the five architectures. Table 1 reports the performances of the MCS based on the set C* designed by our algorithm. For comparison purposes, the performances of the MCSs based on the initial set C^1 and on the other two design methods are also reported. A size of the set C* equal to five was fixed for the "choose the best" method. The performances are measured in terms of the percentage of classification accuracy, the rejection percentage, and the difference between accuracy and rejection. All the values are referred to the test set. The performances of all the three design methods are similar. (Our method is slightly better, but the difference is very small). This can be seen by observing that the initial set C^1 also provides similar performances. This means that the set C^1 does not contain classifiers very "uncorrelated" that can be extracted by a design method in order to improve performances.

Table 1. Results provided by different design methods applied to the set C^1.

MCS based on	%Accuracy	%Rejection	%(Accuracy-Rejection)
Our design method	90.52	0.82	89.70
C^1	89.83	1.20	88.63
Choose the best	90.10	0.49	89.60
Choose from subspaces	89.98	0.49	89.48

Experimentation with the Set C^2

Our algorithm extracted a set C* formed by 5 MLPs and two k-nn classifiers. The five MLPs belonged to the same three architectures of the previous experiment. The two k-nn classifiers corresponded to values of the "k" parameter equal to 21 and 77, respectively. It is worth noticing that such values are quite distant. (This result is in agreement with the expected correlation of the k-nn classifiers for close values of the "k" parameter). Table 2 reports the performances of the MCS based on the set C* designed by our algorithm. For comparison purposes, the performances of the MCSs based on the initial set C^2 and on the other two design methods are also reported. All the values are referred to the test set. A size of the set C* equal to seven was fixed for the "choose the best" method. The performances of all the three design methods are similar. Therefore, conclusions similar to the ones of the previous experiment can be drawn.

Table 2. Results provided by different design methods applied to the set C^2.

MCS based on	%Accuracy	%Rejection	%(Accuracy-Rejection)
Our design method	91.59	1.14	90.45
C^2	90.49	1.01	89.48
Choose the best	90.27	0.11	90.16
Choose from subspaces	91.32	0.97	90.35

Experimentation with the Set C^3

This experiment was aimed to evaluate the capability of our design method of exploiting the "uncorrelation" of a set of "weak" classifiers in order to improve the performances of the initial set C^3. Therefore, we created a set of nineteen MLPs whose performances were not good (ranging from 80.41% to 85.05%). However, such MLPs were based on two different architectures and different weight seeds in order to assure a reasonable degree of error uncorrelation. In addition, we used a PNN that can be expected to be "independent" from the MLPs. Our algorithm extracted a set C* formed by two MLPs, characterized by two different architectures, and the PNN.

Table 3 reports the performances of the MCSs based on the different classifier sets. All the values are referred to the test set. A size of the set C* equal to three was fixed for the "choose the best" method. The results show that our design method is able to choose classifiers more independent than the ones selected by the other methods. This achievement can be explained by observing that a detailed analysis of error uncorrelation is necessary in order to choose effective classifiers from a set of weak classifiers. This kind of analysis is not carried out by the other methods.

Table 3. Results provided by different design methods applied to the set C^3.

MCS based on	%Accuracy	%Rejection	%(Accuracy-Rejection)
Our design method	91.31	2.20	89.11
C^3	87.87	2.37	85.50
Choose the best	88.78	1.65	87.13
Choose from subspaces	89.35	1.57	87.78

Experimentation with the Set C^4

The aim of this experiment is basically similar to the previous one. Our algorithm extracted a set C* formed by one MLPs, one RBF neural network, and the PNN. Table 4 reports the performances of the MCSs based on the different classifier sets. All the values are referred to the test set. A size of the set C* equal to three was fixed for the "choose the best" method. It is easy to see that conclusions similar to the ones of the previous experiment can be drawn.

Table 4. Results provided by different design methods applied to the set C^4.

MCS based on	%Accuracy	%Rejection	%(Accuracy-Rejection)
Our design method	94.83	4.71	90.11
C^4	90.46	3.05	87.41
Choose the best	88.78	1.65	87.13
Choose from subspaces	89.35	1.57	87.78

5. Conclusions

In this paper, an approach to the automatic design of MCSs formed by different classification algorithms has been described. To the best of our knowledge, in the pattern recognition field, no previous work directly addressed such a problem. Some work was carried out by neural network researchers. However, the results of such research work can be exploited only in part for MCSs formed by different classifiers. The experimental results reported in this paper showed the effectiveness of the proposed design approach. In addition, a proof of the optimality of our approach has been provided. It is worth noticing that the assumptions required by such proof are completely or partially met in many real cases of classifier ensembles.

References

1. L. Xu, A. Krzyzak, and C.Y. Suen, “Methods for combining multiple classifiers and their applications to handwriting recognition”, IEEE Trans. on Systems, Man, and Cyb., Vol. 22, No. 3, May/June 1992, pp. 418-435
2. J. Kittler, M. Hatef, R.P.W. Duin and J. Matas “On Combining Classifiers”, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.20, No.3, March 1998, pp. 226-239
3. L. K. Hansen, and P. Salamon, “Neural network ensembles”, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 12, No. 10, October 1990, pp. 993-1001
4. K. Tumer and J. Ghosh, “Error correlation and error reduction in ensemble classifiers”, Connection Science 8, December 1996, pp. 385-404
5. A. J. C. Sharkey (Ed.), Special Issue: Combining Artificial Neural Nets: Ensemble Approaches. Connection Science Vol. 8, No. 3 & 4, Dec. 1996
6. Y.S. Huang, and C.Y. Suen, “A method of combining multiple experts for the recognition of unconstrained handwritten numerals”, IEEE Trans. on on Pattern Analysis and Machine Intelligence, Vol.17, No.1, January 1995, pp.90-94
7. Y.S. Huang, K. Liu and C. Y. Suen,, “The combination of multiple classifiers by a neural network approach”, Int. Journal of Pattern Recognition and Artificial Intelligence, Vol. 9, no.3, 1995, pp.579-597
8. G. Giacinto and F. Roli, “Ensembles of Neural Networks for Soft Classification of Remote Sensing Images”, Proc. of the European Symposium on Intelligent Techniques, Bari, Italy, pp. 166-170
9. D.Partridge, “Network generalization differences quantified”, Neural Networks, Vol.9, No.2, 1996, pp.263-271

10. D.Partridge and W.B.Yates, “Engineering multiversion neural-net systems”, Neural Computation, 8, 1996, pp. 869-893
11. D.W.Opitz and J.W.Shavlik, “Actively searching for an effective neural network ensemble”, Connection Science Vol. 8, No. 3 & 4, Dec. 1996, pp. 337-353
12. B.E.Rosen, “Ensemble learning using decorrelated neural networks”, Connection Science Vol. 8, No. 3 & 4, Dec. 1996, pp. 373-383
13. C.Ji and S.Ma, “Combination of weak classifiers”, IEEE Trans. On Neural Networks, Vol.8, No.1, Jan. 1997, pp. 32-42
14. K.D.Bollacker, and J.Ghosh, “Knowledge reuse in multiple classifier systems”, Pattern Recognition Letters, 18, 1997, pp. 1385-1390
15. B.Littlewood and D.R.Miller, “Conceptual modelling of coincident failures in multiversion software”, IEEE Trans. On Software Engineering, 15(12), 1989, pp; 1569-1614
16. A.K.Jain and R.C.Dubes, Algorithms for clustering data, Prentice Hall, 1988
17. F. Roli, “Multisensor image recognition by neural networks with understandable behaviour” International Journal of Pattern Recognition and Artificial Intelligence Vol. 10, No. 8, 1996, pp. 887-917
18. S. B. Serpico, and F. Roli, “Classification of multi-sensor remote-sensing images by structured neural networks”, IEEE Trans. Geoscience Remote Sensing 33, 1995, pp. 562-578.
19. S.B.Serpico., L. Bruzzone and F.Roli, “An experimental comparison of neural and statistical non-parametric algorithms for supervised classification of remote-sensing images” Pattern Recognition Letters 17, 1996, 1331-1341.