An Adaptive Weighted Majority Vote Rule for Combining Multiple Classifiers

C. De Stefano
DAEIIMI, Università di Cassino, I-03043 Cassino (FR), ITALY
destefano@unicas.it

A. Della Cioppa, A. Marcelli
DIIIE, Università di Salerno, I-84084 Fisciano (SA), ITALY
{adellacioppa,marcelli}@unisa.it

Abstract

In this paper we introduce a novel multiple classifier system that incorporates a global optimization technique based on a genetic algorithm for configuring the system. The system adopts the weighted majority vote approach to combine the decision of the experts, and obtains the weights by maximizing the performance of the whole set of experts, rather than that of each of them separately. The system has been tested on a handwritten digit recognition problem, and its performance compared with those exhibited by a system using the weights obtained during the training of each expert separately. The results of a set of experiments conducted on a 30,000 digits extracted from the NIST database shown that the proposed system exhibits better performance than those of the alternative one, and that such an improvement is due to a better estimate of the reliability of the participating classifiers.

1. Introduction

Traditional pattern recognition systems using a single feature descriptor and a single classification strategy have been widely studied, and many efforts have been made to define description methods and classification algorithms able to achieve high performance even in presence of distorted and noisy samples. Nonetheless, in many pattern recognition applications, especially those involving either a large number of classes to be discriminated, or data exhibiting a large variability and a significant amount of noise, high performing solutions are very difficult to achieve. For this reason, in the last decade there has been a considerable research activity on the problem of combining the classification results provided by a multitude of different classifiers, each adopting a different feature description method or a different classification strategy.

The interest in combining classification results derives from the observation that classifiers using features of different types complement one another in classification performance and this implies that, by using simultaneously more classifiers operating on different features spaces, the classification accuracy should be improved [4, 10, 8].

Even if many efforts have been done in this direction, the general problem of combining conflicting classification results, still remains unsolved, and is the topic of a relevant research activity. In particular, criteria for choosing the different classifiers to be included in a multi-expert system and for determining their number and their combination topology have been proposed [3, 11, 14, 15], as well as different combining rules specifically devised for solving conflicting decisions [18, 13, 6, 2]. In [7] an experimental comparison of various classifier combination schemes is presented, and it is shown that the sum rule outperforms the other combination schemes, although such a result depends on the most restrictive set of assumptions among those considered in the study.

Among the combining rules derived according to the general paradigm of the sum rule, one of the simplest and widely adopted is the majority vote rule [11, 12], according to which the input sample is assigned to the class for which a relative or absolute majority of classifiers agrees. If such an agreement cannot be found, the sample is either rejected or randomly assigned to one of the classes on which there is a partial consensus among the experts. The main drawback of this rule is that all the experts are considered equally reliable: as a consequence, even if an expert is very confident on its decision, the opinions of less reliable classifiers may modify the final decision of the multi-expert system. A possible way of overcoming this drawback is that of including a measure of the reliability of each expert in the combining rule. The weighted majority vote rule [11, 12] is mainly based on this idea: the votes of all the experts are collected and the input sample is

1051-4651/02 $17.00 © 2002 IEEE

assigned to the class for which the sum of the votes, each weighted by the estimated reliability of the corresponding expert, is the highest. A simple and widely adopted method for estimating the reliability of each classifier is that of considering the recognition rate on the training set [18]. When an expert assigns an unknown sample to a class, its decision is weighted in the combining rule by a factor that is proportional to the recognition rate for that class obtained by the expert on the training set.

A criticism to this approach can be formulated by noticing that even if the average performance of a classifier on the training set for a given class is very high, its classification may be very unreliable while dealing with samples belonging to that class, but located in proximity of the boundary between decision regions in its feature space. On the contrary, the same samples may be better located in a different feature space, and therefore another classifier, exhibiting a lower recognition rate for that class, may be more reliable in those particular cases. In other words, using the recognition rate on the training set as a reliability measure for an expert may provide unsatisfactory results for those samples that have not been adequately learned during the training phase of that expert. This represents a kind of paradox, because the aim of the multi-expert approach is that of increasing the performance of the single classifiers by correctly recognizing just those samples which have not been adequately learned during the training of the single classifiers.

In this paper we propose a weighted majority vote approach in which the weights are obtained by maximizing the performance of the whole set of experts, rather than that of each of them separately, as it happens when the expert reliability is obtained during the training. In practice, once each classifier has been adequately trained on the training set, we consider a different and statistically independent test set, and assume that the weights to be used in the combining rule are those that lead to the multi-expert highest recognition rate. It follows that the problem of computing the weights can be reformulated as an optimization problem, where the multi- expert recognition rate is the function to maximize depending on the weights to be estimated. The proposed method adopts a Genetic Algorithm (GA) to evolve the set of weights assigned to each class for each classifier. There are in the literature a few other approaches adopting a GA in the framework of combining classifiers, but they have been mainly used for selecting either the features to be used by a single classifier [9, 16] or the actual set of classifier to be combined [10, 17].

2. The Genetic Search for the Optimal Weights

The combiner adopts the classical weighted majority vote rule:

$$\text{assign the sample } s \text{ to the class } k \text{ if}$$
$$\sum_{i=0}^{N_c} w_{ik} \cdot \delta_{ik} = \max_{j=1}^{N_c} \sum_{i=0}^{N_c} w_{ij}\delta_{ij} \qquad (1)$$

with

$$\delta_{ik} = \begin{cases} 1 \text{ if } E_i \text{ gives the class } k \\ 0 \text{ otherwise} \end{cases}$$

where E_i is the i-th expert and N_c and N_e represent, respectively, the number of classes and the number of experts to be combined.

As mentioned in the introduction, the weights w_{ik} are provided by an optimization procedure implemented by means of a Genetic Algorithm [5, 1]. GAs are an abstraction of Biological Evolution based on the concept of Adaptation. This strategy has been widely adopted on both numerical and combinatorial optimization, since classical non–probabilistic strategies are restricted to special problems, require specific knowledge and fail in the presence of landscapes with multiple local optima [1]. The features that make GA suitable for optimization problems can be summarized as follows:

- they do not require any specific knowledge about the problem at hand, but only values of the function to be optimized;
- they can explore several regions of the configuration space simultaneously and by means of the selection the search process is concentrated on the most promising regions.
- furthermore, by using probabilistic transition rules they are able to manage landscapes with a wide number of local optima.

Starting from a population of possible solutions to the problem at hand, a GA generates new solutions by means of a *selection mechanism* together with the genetics-inspired operators of *crossover* and *mutation*, hoping to evolve the population towards the most promising regions of the solution space. The solutions are encoded by means of "chromosomes" which consist of strings of "genes", e.g. bits, whose values are called "allele". The selection mechanism is aimed to choosing the chromosomes in the population that will be allowed to reproduce in such a way that better chromosomes in the population have higher chances to be chosen for reproduction and for genetic manipulation. The probability that a chromosome will reproduce is evaluated by means of a *fitness function* defined

on the solution space. Typical selection mechanisms used are the Roulette Wheel and the Tournament selection schemes [1]. As regards the genetic operators, the crossover exchanges parts of two selected chromosomes, thus generating two offspring. Such genetic operator is usually applied with a probability p_c called *crossover rate*, typically in the range $[0.6, 1.0]$. Finally, mutation works by changing randomly the allele in some location in the chromosome with a probability p_m called *mutation rate*, typically $\sim 10^2 \div 10^3$ per bit. As concern these two operators, it should be pointed out that the probabilities p_c and p_m have to be tuned depending on the particular problem we work with and represent an important feature of a GA. For example, there are problems in which the mutation rate neither depends on the number of parameter to be optimized nor on the length of the chromosome. So, a fine tuning of such rates can avoid premature convergence to less performing solutions. The termination criterion for the evolution can be a maximum number of generations (evaluations) or a specific requirement on the fitness function to be optimized.

3. Experimental Findings and Conclusion

With the aim to test the effectiveness of our approach in combining classifiers we have used as case study a handwritten digit recognition problem. The data sets used in the experiments have been extracted from the NIST database. In particular, we have extracted 3,000 samples of each class, and they have been divided into three sets: the training set TR1 used to train each classifier, the training set TR2 used to implement the environment the GA works with, and a test set TS used for performance evaluation. The three sets have been extracted in such a way to be statistically independent.

The samples belonging to each data set have been described by means of two different feature sets, namely the Central Geometrical Moments (CGM) of the binary images up to the 7-th order, and the mean of the pixels belonging to the 8×8 disjoint windows that can be extracted from the binary image (MBI). Thus, each sample is described by means of 33 real variables in the first case, and by means of at most 64 real values in the second one. It is worth noticing that these feature sets have been chosen because they exhibit a certain degree of complementarity, rather than for their distinctiveness. There are many other features sets proposed in the literature for the specific problem that are more robust with respect to the variations found in large data sets, as those included in the NIST database.

As with respect to the classifiers, we have used three different schemes: a Back-Propagation neural network (BP), a Learning Vector Quantization neural network (LVQ) and a 10-Nearest Neighbour (10NN) . Thus, we have a pool of six experts to be combined, namely: E_1 (BP using MBI), E_2 (BP using CGM), E_3 (LVQ using MBI), E_4 (LVQ using CGM), E_5 (10NN using MBI), E_6 (10NN using CGM). During a training phase each classifier was separately trained on TR1.

Table 1. The results on the test set TS.

E_1	E_2	E_3	E_4	E_5	E_6
97.09	97.02	96.01	90.33	97.01	85.42

ME_{GA}	ME_{R1}	ME_{R2}
97.91	96.75	97.01

As with respect to the GA, the genes in the chromosome of an individual in the population encode the weights of each class for each classifier of the pool. According to this genetic code, each chromosome is made of 60 genes, and since each gene is encoded by means of 8 bits, each chromosome is made of 480 bits. The resulting solution space contains 2^{480} different configurations, which is fairly complex for a genetic search. The environment the GA interacts with consists of as many specimen as the number of samples in TR2, each specimen being an array of six integers, representing the classes assigned to the corresponding sample by the six experts. Therefore, the fitness ϕ of each individual is evaluated by using the weights encoded in the chromosome in eq. (1) to assign a label to each sample in TR2 and eventually by computing the recognition rate. In formula: $\phi(\sigma_i) = \frac{n_c}{n_t}$, where σ_i is the i-th chromosome in the population, n_c is the number of samples correctly classified and n_t is the total number of samples.

A total number of $500,000$ evaluations have been allowed for each evolution. Namely, we have used a population size of 500 and a maximum number of generations of $1,000$. All the experiments have been performed with tournament selection mechanism whose size has been chosen equal to 10% of the population. As regards the genetic operators, the crossover rate has been set equal to 1.0, while the mutation rate to $1/480$. Finally, the GA has been executed for 30 times with different initial populations in order to reduce the effects of the stochastic fluctuations due to the randomness of the search. The values obtained for the recognition rate on TR2 by the multi-expert which uses the weights found by the GA ranges from 98.53%, in the worst case, to 98.55% in the best one.

In Table 1 we show the recognition rates on TS of each single expert, and the recognition rate of the multi-expert ME_{GA} using as weights the best solution found by the GA. For comparison, we also report the recognition rates obtained by the multi-experts ME_{R1} and ME_{R2}, whose weights are the recognition rates of each expert on TR1 and TR2, respectively. The results reported in Table 1 show that the experts E_1, E_2 and E_5 exhibit similar performance,

Table 2. The variation of the recognition rate of each single expert on each class due to the multi-expert.

Class	E_1	E_2	E_3	E_4	E_5	E_6
0	1.7	1.4	0.7	7.4	1.0	11.7
1	2.7	1.3	3.9	6.9	0.6	12.3
2	1.1	1.8	1.9	6.1	1.2	10.9
3	1.0	-0.1	1.9	7.0	0.3	12.0
4	-0.6	0.8	-0.3	5.5	0.7	9.4
5	1.0	1.2	1.1	12.8	1.0	22.7
6	1.1	0.2	1.2	2.4	0.1	2.2
7	0.9	0.3	2.4	4.5	1.6	9.5
8	4.8	2.6	5.1	18.6	5.7	35.8
9	-0.5	-0.6	1.1	4.6	-0.4	7.2

while the recognition rates of the remaining ones ranges from 85.42 to 96.01. As expected, the recognition rate exhibited by the multi-expert ME_{GA} outperforms the best expert (E_1), as well as both the ME_{R1} and ME_{R2}. This improvement is better revealed by Table 2, which shows the relative recognition rate variations of each expert on each class obtained by adopting the ME_{GA}. More specifically, the element (i, j) in the Table is the difference between the errors on the samples of i-th class made by the j-th expert and corrected by the multi-expert, and the errors introduced by the multi-expert itself, expressed as percentage of the number of samples in each class.

The findings reported above confirm our criticism to the use of the recognition rates of the single experts as weights in the majority vote rule, and support our basic idea that the reliability of the experts should be evaluated simultaneously by considering the combined effect of the outputs provided by each expert in the pool. Eventually, it may be argued that the proposed approach requires a very high computational cost (500,000 evaluations repeated 30 times) for providing only a slight improvement in the performance (less than 1less than 100,000 evaluations and that the weights provided at the end of each evolution were almost identical. In a further experiment done by using the weights obtained by averaging the weights provided after 100,000 evaluation in 3 different evolutions we observed no change in the recognition rate of the multiexpert. This simple experiment suggests that further investigations may help in reducing the computational cost of the method.

References

[1] T. Bäck. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, New York NY, 1996.

[2] S.-B. Cho and J. H. Kim. Combining multiple neural networks by fuzzy integral for robust classification. *IEEE Transactions on Systems, Man and Cybernetics*, 25(2):380–384, 1995.

[3] D. L. Hall. *Mathematical Techniques in Multi-Sensor Data Fusion*. Artech House, Inc., 1992.

[4] T. K. Ho, J. J. Hull, and S. N. Srihari. Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1):66–75, 1994.

[5] J. H. Holland. *Adaptation in Natural and Artificial Systems*. MIT Press, 1975.

[6] Y. S. Huang and C. Y. Suen. A method of combining multiple experts for the recognition of unconstrained handwritten numerals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(1):90–94, 1995.

[7] M. H. J. Kittler, R. P. W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[8] J. Kittler. Improving recognition rates by classifier combination: A theoretical framework. In D. A. C. and I. S., editors, *Progress in Handwriting Recognition*, pages 231–248, Singapore, 1997. World Scientific Publishing.

[9] L. I. Kuncheva and L. C. Jain. Designing classifier fusion systems by genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 4(4):327–337, 2000.

[10] L. Lam and C. Suen. Optimal combination of pattern classifiers. *Pattern Recognition Letters*, 16:945–954, 1995.

[11] L. Lam and C. Y. Suen. Increasing experts for majority vote in ocr: Theoretical considerations and strategies. In *Proceedings of the 4th International Workshop on Frontiers in Handwriting Recognition*, pages 245–254, 1994.

[12] L. Lam and C. Y. Suen. A theoretical analysis of the application of majority voting to pattern recognition. In *Proceedings of the 12th International Conference on Pattern Recognition*, volume 2, pages 418–420. IEEE Computer Society Press, 1994.

[13] D.-S. Lee and S. N. Srihari. A theory of classifier combination: The neural network approach. In *Proceedings of the 3th International Conference on Document Analysis and Recognition*, pages 42–45, 1995.

[14] R. K. Powalka, N. Sherkat, and R. J. Whitrow. Multiple recognizer combination topologies. In T. A. Simner M.L., Leedham C.G., editor, *Handwriting and Drawing Research: Basic and Applied Issues*, pages 329–342. IOS Press, 1996.

[15] A. Rahman and M. Fairhurst. Introducing new multiple expert decision combination topologies: A case study using recognition of handwritten characters. In *Proceedings of the 4th International Conference on Document Analysis and Recognition*, pages 886–891, 1997.

[16] V. E. Ramesh and M. N. Murty. Off-line signature verification using genetically optimized weighted features. *Pattern Recognition*, 32(2):217–233, 1999.

[17] K. Sirlantzis and M. C. Fairhurst. Optimisation of multiple classifier systems using genetic algorithm. In *Proceedings of the International Conference on Image Processing*, 2001.

[18] L. Xu, A. Krzyzak, and C. Y. Suen. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(3):418–435, 1992.