ELSEVIER

Pattern Recognition Letters 16 (1995) 945-954

Pattern Recognition Letters

Optimal combinations of pattern classifiers

Louisa Lam *, Ching Y. Suen

Centre for Pattern Recognition and Machine Intelligence, Concordia University, Suite GM-606, 1455 de Maisonneuve Boulevard West, Montreal, Quebec H3G 1M8, Canada

Received 10 September 1994; revised 15 February 1995

Abstract

To improve recognition results, decisions of multiple classifiers can be combined. We study the performance of combination methods that are variations of the majority vote. A Bayesian formulation and a weighted majority vote (with weights obtained through a genetic algorithm) are implemented, and the combined performances of 7 classifiers on a large set of handwritten numerals are analyzed.

Keywords: Multiple classifier systems; OCR; Majority vote; Genetic algorithm; Bayesian method

1. Introduction

In the recognition of handwritten characters and words, there has been a recent movement towards combining the decisions of several classifiers in order to arrive at improved recognition results. This is due to a number of reasons, among which are the demands imposed by real-life applications and the availability of a wide variety of algorithms. Practical applications demand highly reliable classification, which is extremely difficult for a single algorithm to achieve. Since many algorithms are available for these tasks, it is logical to consider the use of several classifiers to achieve higher reliability. The combination can be implemented using different strategies. In (Suen et al., 1990) and (Suen et al., 1992), the combined decision is obtained by a majority vote of the individual classifiers, and variations of this scheme are implemented in (Gader et al., 1990) and (Xu et al., 1992). When the individual classifiers output ranked lists of decisions, these rankings can be used to derive combined decisions (Ho et al., 1994). Further developments in obtaining a combined decision include statistical approaches (Franke and Mandler, 1992; Huang and Suen, 1993), formulations based on Bayesian and Dempster–Shafer theories of evidence (Franke and Mandler, 1992; Xu et al., 1992), and neural networks (Lee and Srihari, 1993). In all these cases, it was found that using a combination of classifiers can result in remarkable improvements in the recognition performance, and this is true regardless of whether the classifiers are independent or make use of orthogonal features. (The independence assumption is sometimes made for theoretical considerations, even though its validity is usually unknown in practice.)

Among all the combination methods, majority vote is the simplest to implement, and its simplicity has permitted theoretical analysis (Lam and Suen, 1994). In this work, we study extensions of this method to cases in which classifiers are assigned unequal weights based on their performance. These weights are obtained through the optimization of an objective func-

* Corresponding author. Email: llam@cenparmi.concordia.ca

SSDI 0167-8655(95)00050-X

tion for the combined decision. Two different, alternative methods of obtaining optimal weights are studied: the use of a Bayesian formulation and a genetic algorithm.

In Section 2 we describe the methods, the objective function, and the experimental procedure. The experimental results are presented in Section 3, and we conclude with some observations and analyses of the methods in Section 4.

2. Details of the methods

In combining decisions of pattern classifiers, the method used depends on the nature of the output produced by each recognition algorithm. If this output consists of a single assigned class, then many of the combination methods would not be applicable, and one of the most suitable means of arriving at a combined decision would be some form of voting. In addition to simple majority vote (in which all votes have equal weight), the votes can be weighted so that each classifier carries the same weight for all pattern classes, or the weights can be determined according to the performance of each classifier on each class.

We will study and compare the results of the two weighting procedures. In the former case, the assignment of weights will be based on optimizing the value of an objective function through a genetic algorithm. For the second system of weighting, a Bayesian combination rule (Xu et al., 1992) will be used, and the value of a parameter will be determined so that the same function is optimized. In this way, we can realistically compare the results of the procedures on the same sets of data.

2.1. Optimization of objective function

In pattern recognition, the recognition (correct) and substitution (error) rates are often used to measure the performance of a classifier. Ideally, one would like to maximize the recognition rate and minimize the substitution rate, but this is very difficult to achieve in practice. When there is a third option of rejecting the input sample in case of uncertainty, it is a common experience that the procedures used to reduce the error rate would also lead to higher rejection (and lower recognition) rates. On the other hand, it is impractical to eliminate the reject option (and force the classifier to decide on the identity of every sample), since the decisions made in uncertain cases would cause a disproportionate increase in the error rate.

In view of the above, it would be natural for a measurement of classifier performance to contain some trade-off factor between the recognition/rejection and error rates. For example, at the first and second IPTP competitions in Japan (Noumi et al., 1994), the cost of an error was set at 10 times that of a rejection, so the precision index was established as

$$\text{Rejection} + 10 * \text{Error}.$$

For our present experiment, the objective function to be maximized was defined as

$$F = \text{Recognition} - \beta * \text{Error},$$

where β has values 10, 15, 20, 25, and 30. Obviously, the value of β varies with the accuracy or reliability desired for a particular application. The function F is related to the precision index defined above; for example, when $\beta = 10$, maximizing F is equivalent to minimizing Rejection + 11 * Error. These high levels of reliability are set for our experiment because they can be sustained by the classifiers considered.

2.2. Genetic algorithm

First proposed by Holland (1975), genetic algorithms have been found to be robust and practical optimization methods. A genetic algorithm begins with an initial set (also called a *population*) of randomly generated potential solutions to an optimization problem. The value of an objective function (*fitness value*) of each solution is evaluated, and the "best" solutions are selected for survival. Then the genetic algorithm manipulates these selected solutions in its search for better solutions. Each solution is encoded into a binary string (*chromosome*), so that new encoded solutions can be generated through the exchange of information among surviving solutions (*crossovers*) as well as sporadic alterations in the bit string encodings of the solutions (*mutations*).

This method is applied to our problem to obtain an optimal set of weights, one for the vote of each classifier across all pattern classes. Optimality is defined according to the value of F described above, which is also used as the fitness value. To ensure that fitter strings have proportionally higher chances of surviving in the subsequent generation, the selection mechanism is implemented by a roulette scheme (Goldberg, 1989) after the fitness values have been linearly scaled so that the maximum scaled fitness value is 1.5 times the average fitness value of the population. The scaling mechanism was implemented because the variance in string fitness values becomes small after several generations, with the result that all strings would have approximately the same number of offsprings without scaling, thereby neutralizing the propagation of fitter strings. Since a linear scaling may cause low fitness values to be mapped onto negative scaled values, the chromosomes having fitness values below a certain threshold f_t are eliminated. In this case

$$f_t = f_{\text{ave}} - 2.5\sigma$$

where f_{ave} is the average fitness of the population, and σ is the standard deviation of fitness values in the population. So if f is the fitness value of a string, its scaled value f' is given by

$$f' = \frac{f - f_t}{f_{\max} - f_t} * (1.5 * f_{\text{ave}} - f_t) + f_t, \tag{1}$$

where $f_{\max}$ is the maximum fitness of the population before scaling.

The population size used is 50, and each gene occupies 10 bits, so the weight of each classifier has range between 0 and 1.023. The probabilities of crossover and mutation are 0.9 and 0.05 respectively. Control parameters in these ranges have been proposed by several researchers to guarantee good performance on carefully chosen testbeds of objective functions (Srinivas and Patnaik, 1994). This is a robust process, in the sense that the same optimal fitnesses would result upon replications of the process starting from random weights. The distributions of weights that produce the optimal solutions may vary with each repetition, but the procedure would result in the same maximum values of the objective function F, as well as the same recognition and error rates. Fig. 1 shows an example of the behaviors of the maximum and average fitness values through the generations.

From Fig. 1, it can be observed that the fitness values do not vary greatly through the generations. This is due to the relatively large number (seven) of classifiers being combined, and the high individual performances. In other words, whenever a weighted majority of the votes are in agreement, then the group decision would be quite reliable. For this reason, varying the weights does not create dramatic improvements. On the other hand, any gain in performance is useful for practical applications.

2.3. Bayesian combination rule

The genetic algorithm implemented assigns a weight to the vote of each classifier (also called an *expert*), and this weight would be applied to all patterns regardless of the decision made by the expert. Another method of determining the weights is through the Bayesian decision rule, which takes into consideration the performance of each expert on the training samples of each class. In particular, the confusion matrix C of each classifier on a training set of data would be used as indications of its performance. For a problem with M possible classes plus the reject option, C is an $M \times (M+1)$ matrix in which the entry C_{ij} denotes the number of patterns with actual class i that is assigned class j by the classifier when $j \leqslant M$, and when $j = M+1$, it represents the number of patterns that are rejected.

From the matrix C, we can obtain the total number of samples belonging to class i as the row sum $\sum_{j=1}^{M+1} C_{ij}$, while the column sum $\sum_{i=1}^{M} C_{ij}$ represents the total number of samples that are assigned class j by this expert. When there are K experts, there would be K confusion matrices $C^{(k)}$, $1 \leqslant k \leqslant K$. Consequently, the conditional probability that a pattern x actually belongs to class i, given that expert k assigns it to class j, can be estimated as

$$P(x \in C_i \mid e_k(x) = j) = C_{ij}^{(k)} \Big/ \sum_{i=1}^{M} C_{ij}^{(k)}. \tag{2}$$

For any pattern x such that the classification results by the K experts are $e_k(x) = j_k$ for $1 \leqslant k \leqslant K$, we can define a belief value that x belongs to class i as

$$bel(i) = P(x \in C_i \mid e_1(x) = j_1, \ldots, e_K(x) = j_K). \tag{3}$$

By applying the Bayes' formula and assuming independence of the expert decisions (Xu et al., 1992), $bel(i)$ can be approximated by

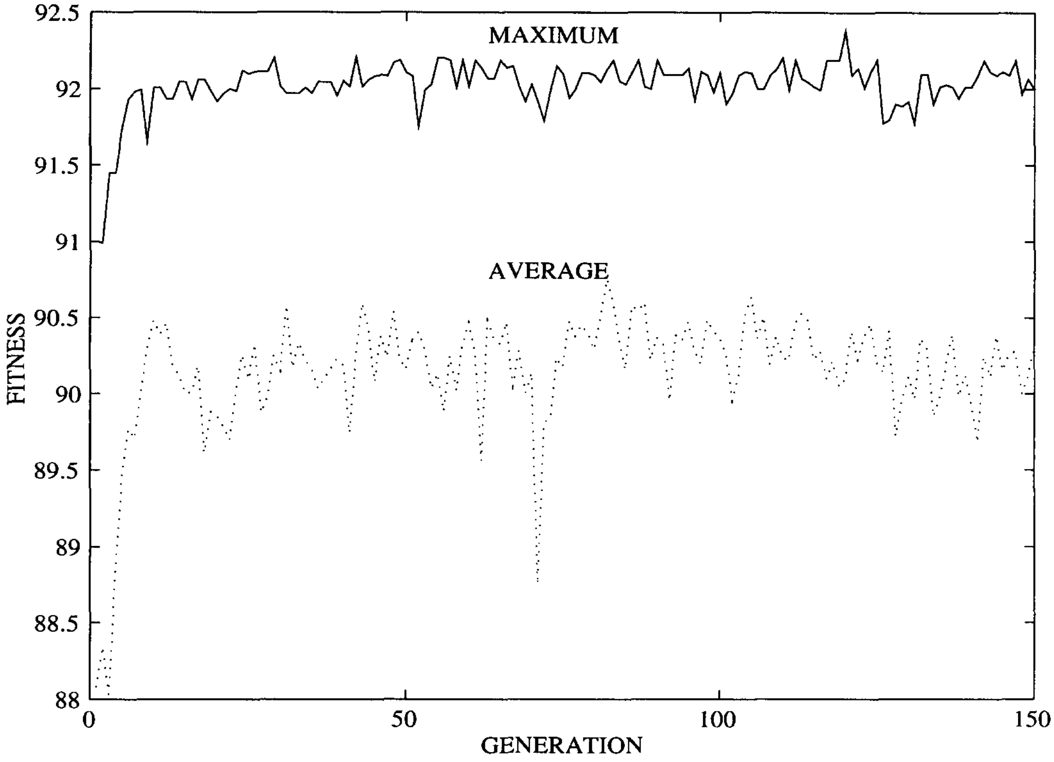

Fig. 1. Maximum and average fitness values of genetic algorithm.

$$bel(i) \doteq \frac{\prod_{k=1}^{K} P(x \in C_i \mid e_k(x) = j_k)}{\sum_{i=1}^{M} \prod_{k=1}^{K} P(x \in C_i \mid e_k(x) = j_k)} \quad (4)$$

for $1 \leqslant i \leqslant M$.

For any input pattern x, we can assign x to class j if $bel(j) \geqslant bel(i)$ for all $i \neq j$ and $bel(j) > \alpha$ for a threshold α. Otherwise x is rejected, and it is also rejected if $e_k(x) = M+1$ for all k (i.e., if x is rejected by all classifiers).

The results obtained from this method depend on the value of α chosen. As α increases, so does the degree of certainty expected of the decision; therefore the error rate would decrease, but the recognition rate would be lower also. Fig. 2 gives an example of the behavior of the recognition and error rates when α varies from 0.1 to 0.99999999.

Given that the results depend on the choice of α, and because we would like to compare optimal results obtained from different methods, the value of α was chosen to maximize the value of the same objective function F. In Fig. 2, the optimal value of α is 0.999952 when $\beta = 10$, and this value of α represents the threshold above which the ratio of the changes in the recognition and error rates would exceed β. In this figure, each dotted line contains the points yielding the same value of F, and it can be seen that the maximum value of F obtained is approximately 96. More detailed experimental results are presented in Table 5 of Section 3.2.

2.4. Classifiers and experimental data

The combination methods described in this section can be applied to any category of patterns or classifiers, therefore they can be tested on any type of data. Our experiment was performed on seven classifiers used to process a large collection of handwritten numerals. This database consists of 46451 numerals collected by the Industrial Technology Research Institute (ITRI) of Taiwan. Of these, 24427 were used to train the individual classifiers, and the other 22024 samples form the test set for the recognizers. Since the com-

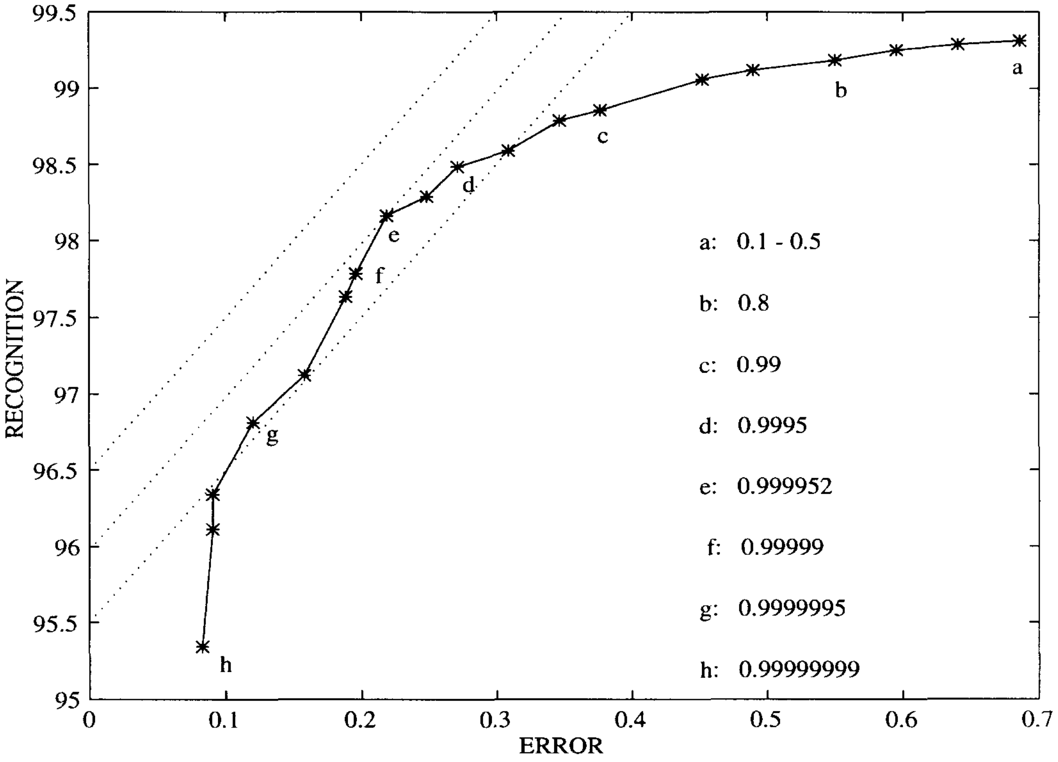

Fig. 2. Recognition results for different values of α (shown as labelled).

bination methods should first be trained on representative classification results, only the results of the test set are used in our study. This set was subdivided into 50 subsets at ITRI, with 5 subsets per class. The first 3 subsets of each class were assigned to set A and the rest to set B, with the result that 13272 samples belong to set A and 8752 to B.

The seven classifiers used to recognize this data cover a wide variety of approaches. The features used include the pixels of the pattern, contour and algebraic features, together with structural information derived from a skeleton. The classifications are based on tree classifiers, dynamic programming, relaxation and exhaustive matching, as well as a neural network. More details about these experts are contained in (Liu et al., 1994; Suen et al., 1992; Tu et al., 1991). These classifiers have not been trained to the same extent on the training set of 24427 samples, and their individual performances on sets A and B are shown in Table 1.

Given that both sets A and B had been test sets for the recognizers and the division into two sets was arbitrary, the differences in results between the two sets indicate the presence of more distorted samples in set A. However, no effort was made to manipulate the data in order to obtain a more even partition.

2.5. Experimental procedure

The classification results of set A were used to train the combination methods in the following ways.

(i) To derive optimal weights (using the genetic algorithm) that should be assigned to the different classifiers for the weighted majority vote. As stated in Subsection 2.2, the search process is continued through 150 generations, with 50 chromosomes per generation. Each chromosome is decoded into a set of seven weights which are then normalized so that their sum equals one. Each weight is assigned to the vote of the corresponding classifier. The decisions of these seven classifiers on each sample of set A are combined using weighted majority vote, from which the recognition and error rates of the combined decision on set

Table 1
Performance of individual classifiers on handwritten numerals

Expert	Set A			Set B		
	Recog.	Error	Obj. F	Recog.	Error	Obj. F
E1	82.791	3.187	50.919	84.004	3.005	53.953
E2	91.132	1.982	71.316	92.207	1.874	73.469
E3	93.264	1.575	77.517	93.864	1.165	82.210
E4	87.176	1.831	68.867	88.425	1.714	71.287
E5	94.929	0.799	86.942	95.007	0.857	86.437
E6	95.999	0.821	87.786	96.023	0.697	89.054
E7	93.716	5.327	40.446	95.212	4.273	52.479

A can be obtained. The fitness value of the chromosome is the value of F derived from these rates. The optimal weights are the ones that produce the maximum value of F over the 150 generations.

(ii) For the Bayesian method, the confusion matrix obtained from set A is used to estimate the probabilities $P(x \in C_i \mid e_k(x) = j_k)$. Using these probabilities, the recognition results of set A are processed by the Bayesian combination rule for different values of the parameter α, and the α that produces the maximum value of F is approximated. The optimal α is located through a successive refinement of the scope of the search, which is a reasonable procedure because of the discrete nature of the problem.

When training is complete, the results are tested on set B in 2 ways:

(a) The optimal weights obtained from (i) are assigned to the respective classifiers, and the weighted majority rule is applied to obtain recognition results for the combination.

(b) For the Bayesian method, the probabilities and optimal value of α described in (ii) are used to combine the decisions of the seven classifiers.

3. Experimental results

The procedures described in Subsection 2.5 are applied to the classification results of 2.4, and the results are given below.

3.1. Results of the genetic algorithm

This algorithm was trained by combining the decisions of the classifiers on set A in order to obtain a set of weights that would maximize the function

$$F = \text{Recognition} - \beta * \text{Error},$$

Table 2
Optimal results from genetic algorithm

Result	Recog.	Error	Reject
1	96.9033	0.1507	2.9460
2	96.9711	0.1582	2.8707
3	96.8279	0.1507	3.0214
4	96.6772	0.1432	3.1796

where β = 10, 15, 20, 25 and 30.

For each value of β, the search begins with 50 sets of random weights and this is propagated through 150 generations. This process is performed 9 times for each value of β, with different starting weights. Consequently, 45 cycles of search had been completed (on a total of 337,500 sets of weights). For each β, the recognition and error rates that give $F_{\max}(\beta)$ among all the 9 cycles are determined. For the five values of α, this procedure produces the four sets of results shown in Table 2. Among these results, the second set appears far more frequently as optimal with the smaller β's; this is logical given that its error rate is the highest, which means that the fitness of this result would decrease as β increases.

Of the results shown in Table 2, the first set of values gives the maximum F for all values of β, and therefore they are considered optimal for our experiment. It should be noted that the set of weights producing a value of F may not be unique, due to the fact that F does not vary continuously with the weights, but is piecewise constant. In other words, F changes only when the weights have shifted sufficiently to change the majority vote for some patterns. In addition, different sets of weights may create different combined decisions in a small number of cases, but result in equal recognition and error rates. For example, the optimal results are actually obtained from the two sets of weights shown in Table 3.

It is instructive to note the following points:

Table 3
Weights producing optimal results

Expert	Weight 1	Weight 2
E1	0.0359	0.0642
E2	0.1882	0.1986
E3	0.0898	0.0892
E4	0.0918	0.1028
E5	0.2219	0.2122
E6	0.2532	0.2310
E7	0.1191	0.1021

Table 4
Optimal results from combining 6 experts

Combination	Recog.	Error	Reject
E1, E2, E4–E7	96.5642	0.1507	3.2851
E1, E3, E4–E7	96.4210	0.1658	3.4132

(i) E1 is usually assigned the lowest weight by this search algorithm. Given that its performance is low, the result supports the validity of the process.

(ii) Results 4 of Table 2 are obtained only when $\beta = 30$, or the cost of an error is very high (31 times that of a rejection). In this case, the optimal result produces a lower error rate, and E7 was actually assigned the lowest weight, which is consistent with its high error rate.

(iii) E3 is usually assigned a low weight (directly above that of E1), which cannot be explained by its performance alone. Since the reason can be that this classifier does not make much contribution to the combined result (in the sense that the vote of this expert may not change the combined decision of the other experts in most cases), an attempt was made to verify this hypothesis.

The decisions of the 6 experts without E3 were combined using the entire procedure described above in this subsection, as well as those of 6 experts with E3 replacing E2. In other words, the search process is propagated through the entire 45 cycles for each set of 6 experts, and the optimal results are determined as described above. These results, shown in Table 4, indicate that the combined performance is better with E2 than E3. Given that E2 definitely has weaker individual performance than E3 (as shown in Table 1), it can be inferred that E2 is more effective as a complement to the other classifiers, and this point has been identified by the genetic algorithm.

The optimal weights shown in Table 3 are then applied to set B, and the results are summarized later in Table 6.

Table 5
Results for different values of α

Value of α	Recog.	Error	F ($\beta = 10$)
0.5	99.3144	0.6856	92.4584
0.8	99.1863	0.5500	93.6864
0.95	99.0582	0.4520	94.5378
0.995	98.7870	0.3466	95.3213
0.9995	98.4856	0.2712	95.7734
0.99995	98.1617	0.2260	95.9015
0.999995	97.6343	0.1884	95.7508
0.9999995	96.8131	0.1205	95.6076

3.2. Results of Bayesian combination

Using the procedure described in Subsection 2.5, the classifier decisions are combined using the Bayesian method. It should be noted that the results are sensitive to the value of α chosen. Since higher values of α require higher levels of confidence, the error rate would be a decreasing function of α, as is the recognition rate. The function F, however, would increase to a maximum, then decrease. The decrease begins at the point where the trade-off between the recognition and error rates involve a factor larger than β. Table 5 shows some results obtained from set A for different values of α, when $\beta = 10$.

For each value of β, the α that yields the maximum F is determined. Logically, higher values of β impose higher costs on errors, from which it follows that the optimal solutions would be more reliable. In order to achieve this, α also needs to have a larger magnitude. This is in fact the case, except that for $\beta = 20$, 25 and 30, results have stabilized, so the same $\alpha = 0.9999999$ and optimal recognition results are obtained. When $\beta = 10$, $\alpha = 0.999952$. These α's were then used as thresholds for set B, and the combined recognition results are given in Table 6. In this table, the results of simple majority vote are also included for comparison.

An examination of the patterns misclassified by the combinations shows the following.

(i) As expected, all the errors of Bayesian 2 are contained in those of Bayesian 1.

(ii) Of the substitutions made by majority vote, all except one are also misclassified in the same way by weighted majority vote derived from the genetic algorithm.

(iii) Of the 14 misclassifications made by majority vote, 10 are common to all the classifiers.

The 14 misclassified samples from set B are shown in Fig. 3, where the class of each sample is indicated,

Table 6
Results of combination methods

Method	Set A				Set B			
	Recog.	Error	Rej.	Obj. F	Recog.	Error	Rej.	Obj. F
Majority vote	96.233	0.196	3.571	94.274	96.778	0.160	3.062	95.178
Bayesian 1 (α=0.999952)	98.162	0.218	1.620	95.977	97.784	0.571	1.645	92.071
Bayesian 2 (α=0.9999999)	96.338	0.090	3.572	95.434	96.550	0.366	3.084	92.655
Genetic algorithm	96.903	0.151	2.946	95.396	97.075	0.228	2.697	94.790

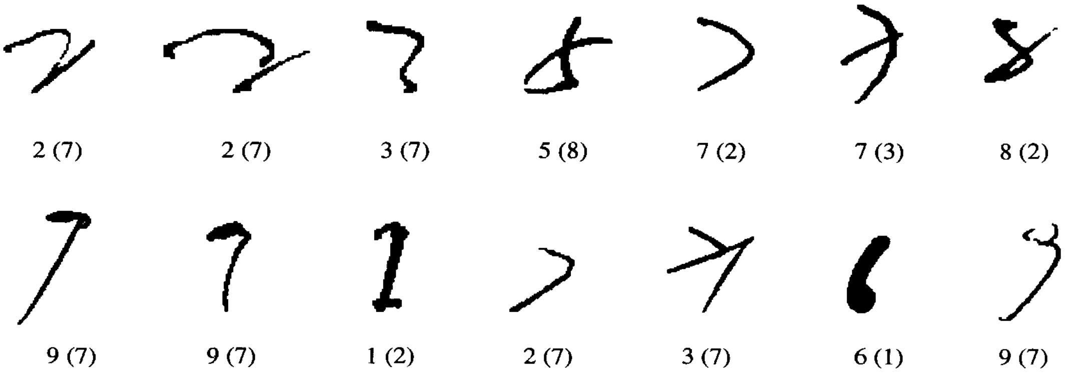

Fig. 3. Patterns misclassified by majority vote.

and the recognition result is given in parentheses.

4. Analyses and observations

From Table 6, it can be seen that the results of majority vote follow the general trend of the individual classifiers in that the results of set B are better than those of A in every aspect, which is reasonable since no training of the combination is involved. This is not the case for the other two combination methods when the error rates are considered. These methods are trained to produce optimal results on set A, while their performances on set B are lower.

These differences in performance lead to a salient point regarding the size and quality of the training data. Theoretically, if the training set is large enough as well as truly representative of the data in general, then there should be no discrepancy between the results obtained from the two sets. In reality, it is very difficult to partition a database into sets of the same quality and complexity, especially since the degree of recognizability of patterns actually depends on the features and the classifier used. This being the case, it remains a challenging problem as to how one can devise a partition of a database into sets that would be considered equal in difficulty for classifiers in general. It would be even more challenging to obtain or select a training set that could give optimal performance for unknown test data.

It is a simpler matter to increase the size of a database, provided that time and effort are available. When we examine the results of the combination methods, it is evident that a training set of 13272 samples is insufficient to establish accurate values of $bel(i)$ for $0 \leqslant i \leqslant 9$, for the Bayesian method. These values are calculated from the confusion matrices of the classifiers, each of which has 100 entries (not counting the rejections) for the classification of numerals. The off-diagonal entries, representing erroneous identifications, usually have very low magnitudes. Paradoxically, the better the classifier, the smaller would be these entries. Consequently, $bel(i)$ would be partly based on rather scant evidence. This problem is especially serious when one recognizer makes a particular misclassification only in the test set. In this case, the corresponding factor in the numerator of $bel(i)$ (obtained from the training set) would be zero, and hence the combined decision cannot be correct regardless of the decisions of the other classifiers.

The problems arising from size and uniformity of the datasets have less pronounced effects on the genetic algorithm, because only a few parameters need to be determined from the training data. In our problem, only the 7 optimal weights (one for each classifier) have to be defined. The less specific nature of the training process has also resulted in a closer proximity between the results of sets A and B.

Moreover, the genetic algorithm is capable of identifying dependencies among classifiers, by assigning lower weights to those that are less effective in influencing the group decision in the optimal direction. This is a very useful feature, since this kind of knowledge is not available when classifiers are designed and implemented. In general, it is a non-trivial problem to decide whether classifiers are independent, since there can exist many possible overlaps between feature sets and classification methods. Even after a recognizer has been trained and tested, and its individual performance is known, it is still difficult to know how significant a role it would assume as a member of a group. The genetic algorithm can help to obtain such information, which can lead to simpler and more efficient multiple classifier systems.

In terms of computing time involved, it is certain that genetic algorithms take much longer to train, in order to search for optimal weights. However, once these weights are obtained and applied to the system, then the weighted majority vote is much faster than the Bayesian combination at the recognition stage.

Finally, if we consider the results of Table 6, it is clear that for any $\beta \geqslant 4$, or for any reasonable tradeoffs between the recognition and error rates, simple majority vote produces the best results on set B. This is another facet of the comment made in this section about the size and representative capability of the training data. From the results on set A, it is evident that using more parameters in the combination process can produce more optimal results, provided the parameters are accurate reflections of the data set. For this reason, the optimal values of $F(\beta = 10)$ are 94.274, 95.396 and 95.977 for majority vote, genetic algorithm, and Bayesian 1 respectively, and this ordering agrees with the degrees to which the combination methods are modeled after set A. Consequently, for datasets with characteristics close to those of A, the same pattern of behavior would be expected. However, when these characteristics are not the same, adapting the parameters to set A can result in "overfitting", and this effect is observed in the results of set B, even though the 2 sets of data were collected together under the same conditions. For this second set, the optimal values of F are in reverse order to those of A, showing that it is more difficult to generalize as the number of parameters increases. Therefore, in the absence of a truly representative training set, simple majority vote remains the easiest and most reliable solution among the ones studied here.

Acknowledgements

The authors wish to thank Raymond Legault, Ke Liu, and Christine Nadal of CENPARMI, Lawrence Pang of ICSA, together with Kai-Hsiang Chou and Lo-Ting Tu of ITRI, for the use of their recognition results. The detailed comments of the reviewers have helped to improve and clarify this paper, and they are much appreciated.

This research was supported by the Natural Sciences and Engineering Research Council of Canada, the National Networks of Centres of Excellence program of Canada, and the FCAR program of the Ministry of Education of the province of Québec.

References

Franke, J. and E. Mandler (1992). A comparison of two approaches for combining the votes of cooperating classifiers. *Proc. 11th Internat. Conf. on Pattern Recognition*, 611-614.

Gader, P.D., D. Hepp, B. Forester and T. Peurach (1990). Pipelined systems for recognition of handwritten digits in USPS ZIP codes. *Proc. U.S. Postal Service Advanced Technology Conf.*, 539-548.

Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA.

Ho, T.K., J.J. Hull and S.N. Srihari (1994). Decision combination in multiple classifier systems. *IEEE Trans. Pattern Anal. Mach. Intell.* 16 (1), 66-75.

Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*. Univ. of Michigan Press, Ann Arbor, MI.

Huang, Y.S. and C.Y. Suen (1993). An optimal method of combining multiple classifiers for unconstrained handwritten numeral recognition. *Pre-Proc. 3rd Internat. Workshop on Frontiers in Handwriting Recognition*, 11-20.

Lam, L. and C.Y. Suen (1994). A theoretical analysis of the application of majority voting to pattern recognition. *Proc. 12th Internat. Conf. on Pattern Recognition II*, 418–420.

Lam, L. and C.Y. Suen (1994). Increasing experts for majority vote in OCR: Theoretical considerations and strategies. *Proc. 4th Internat. Workshop on Frontiers in Handwriting Recognition*, 245–254.

Lee, D.-S. and S.N. Srihari (1993). Handprinted digit recognition: a comparison of algorithms. *Pre-Proc. 3rd Internat. Workshop on Frontiers in Handwriting Recognition*, 153–162.

Liu, K., Y.-S. Huang and C.Y. Suen (1994). Image classification by classifier combining technique. *Proc. SPIE Conf. on Neural and Stochastic Methods in Image and Signal Processing III*, 210–217.

Noumi, T., T. Matsui, I. Yamashita, T. Wakahara and T. Tsutsumida (1994). Results of the Second IPTP Character Recognition Competition and studies on multi-expert handwritten numeral recognition. *Proc. 4th Internat. Workshop on Frontiers in Handwriting Recognition*, 338–346.

Srinivas, M. and L.M. Patnaik (1994). Genetic algorithms: A survey. *Computer*, 17–26.

Suen, C.Y., C. Nadal, T.A. Mai, R. Legault and L. Lam (1992). Computer recognition of unconstrained handwritten numerals. *Proc. IEEE* 80 (7), 1162–1180.

Suen, C.Y., C. Nadal, T. A.Mai, R. Legault and L. Lam (1990). Recognition of totally unconstrained handwritten numerals based on the concept of multiple experts. *Proc. 1st Internat. Workshop on Frontiers in Handwriting Recognition*, 131–143.

Tu, L.-T., Y.-S. Lin, C.-P. Yeh, I.-S. Shyu, J. L. Wang, K.-H. Joe and W.-W. Lin (1991). Recognition of handprinted Chinese characters by feature matching. *Proc. 1991 Internat. Conf. on Computer Processing of Chinese and Oriental Languages*, 154–157.

Xu, L., A. Krzyzak and C.Y. Suen (1992). Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Trans. Syst. Man Cybernet.* 22 (3), 418–435.