Pergamon

Neural Networks, Vol. 7, No. 5, pp. 777–781, 1994

0893-6080/94 $6.00 + .00

0893-6080(94)E0011-9

CONTRIBUTED ARTICLE

Combining the Results of Several Neural Network Classifiers

GALINA ROGOVA

Calspan Corporation

(*Received* 16 *March* 1993; *revised and accepted* 20 *January* 1994)

Abstract—*Neural networks and traditional classifiers work well for optical character recognition; however, it is advantageous to combine the results of several algorithms to improve classification accuracies. This paper presents a combination method based on the Dempster–Shafer theory of evidence, which uses statistical information about the relative classification strengths of several classifiers. Numerous experiments show the effectiveness of this approach. Our method allows* 15–30% *reduction of misclassification error compared to the best individual classifier.*

Keywords—Classifier, Neural network, Character recognition, The Dempster–Shafer theory of evidence, Evidence.

1. INTRODUCTION

Pattern recognition problems, such as classification of machine or handprinted characters, are currently solved with acceptable accuracy by using traditional classifiers or neural networks of different architectures and based on different sets of features. We may suppose that many of them tend to make recognition errors of different types; that is, they may be regarded as error independent. It is easier in many cases to apply several error-independent classifiers to the same recognition task and use their "error independence" to improve recognition performance of a combined system instead of inventing a new architecture or a feature extractor to achieve the same accuracy.

Recently, various combination techniques were proposed by different authors, where majority voting scheme, neural net, Bayesian, and the Dempster–Shafer theories were employed (Mandler & Schurmann, 1988; Xu, Krzyzak, & Suen, 1991, 1992). It appears that using the Dempster–Shafer theory of evidence is very productive, but the result depends considerably on a function that is used as a basic probability assignment. Xu, Krzyzak, and Suen (1992) applied the Dempster–Shafer theory of evidence to combine so-called "syntactic classifiers" that produce only a class label as output. They used the recognition rate and the substitution rate of each individual classifier to calculate basic probability assignments. On the other hand, neural networks as well as a number of traditional classifiers generate an output vector that can supply additional information on a "measurement level." For this type of classifier, posterior class-conditional probabilities can be calculated, providing a natural basic probability assignment. The calculation of posterior probabilities, however, demands numerous approximations that pose very difficult problems, especially in the situation when the number of classes is large.

In this paper, we present two new sets of support functions for the calculation of evidences. They permit us to obtain a considerable improvement of classification accuracy without complex computations.

Acknowledgements: The author expresses her gratitude to all members of the Recognition and Information Processing Group of Eastman Kodak, especially P. Anderson, L. Barski, T. Pawlicki, and A. Shustorovich for providing her with data produced by individual classifiers and for valuable discussions. This paper was prepared while the author was with Eastman Kodak Company.

Requests for reprints should be sent to the author at 3 Locke Drive, Pittsford, NY 14534.

2. THE DEMPSTER–SHAFER THEORY OF EVIDENCE

The Dempster–Shafer theory of evidence is a tool for representing and combining measures of evidences. This theory is a generalization of Bayesian reasoning and it is more flexible than Bayesian when our knowledge is incomplete, and we have to deal with uncertainty and ignorance. We introduce its basic concepts in this section, following Barnett (1981) and Shafer (1976).

Let Θ be a set of mutually exhaustive and exclusive atomic hypotheses, $\Theta = \{\theta_1, \ldots, \theta_K\}$. Θ is called the *frame of discernment*. Let 2^Θ denote the set of all subsets of Θ. A function $\mathbf{m}$ is called a *basic probability assignment* if:

$$\mathbf{m} : 2^\Theta \rightarrow [0, 1], \quad \mathbf{m}(\varnothing) = 0, \quad \text{and} \quad \sum_{A \subseteq \Theta} \mathbf{m}(A) = 1. \quad (1)$$

Whereas the probability theory assigns a measure of *probability* to atomic hypotheses θ_i, $\mathbf{m}(A)$ represents the *belief* in a not necessarily atomic hypothesis A. For $A \neq \theta_i$, $\mathbf{m}(A)$ reflects our ignorance because it is a belief we cannot further subdivide among the subsets of A. $\mathbf{m}(A)$ is a measure of support we are willing to assign to a composite hypothesis A at the expense of support $\mathbf{m}(\theta_i)$ of atomic hypotheses θ_i. If, for the frame of discernment Θ, we set $\mathbf{m}(\theta_i) \neq 0$ for all θ_i and $\mathbf{m}(A) = 0$ for all $A \neq \theta_i$, we find ourselves in the situation of probability theory with $\sum_i \mathbf{m}(\theta_i) = 1$ and $m(\theta_i)$ that may be regarded as a probability of θ_i.

Because $\mathbf{m}(A) + \mathbf{m}(\neg A) \leq 1$, the amount of belief committed, neither A nor compliment of A is the degree of ignorance. Therefore, the Dempster–Shafer theory of evidence allows us to represent only our actual knowledge "without being forced to overcommit when we are ignorant."

If $\mathbf{m}$ is a basic probability assignment, then a function $\mathbf{Bel}: 2^{\Theta} \rightarrow [0, 1]$ satisfying:

$$\mathbf{Bel}(B) = \sum_{A \subseteq B} \mathbf{m}(A) \tag{2}$$

is called *a belief function*. We can consider a basic probability assignment as a generalization of a probability density function whereas a belief function is a generalization of a probability function.

There is one-to-one correspondence between the belief function and the basic probability assignment. If A is an atomic hypothesis, $\mathbf{Bel}(A) = \mathbf{m}(A)$.

If $\mathbf{m}_1$ and $\mathbf{m}_2$ are basic probability assignments on Θ, their combination or *orthogonal sum*, $\mathbf{m} = \mathbf{m}_1 \oplus \mathbf{m}_2$, is defined as:

$$\mathbf{m}(A) = C^{-1} \sum_{D \cap B = A} \mathbf{m}_1(B) \cdot \mathbf{m}_2(D), \tag{3}$$

where

$$C = \sum_{D \cap B \neq \varnothing} \mathbf{m}_1(B) \cdot \mathbf{m}_2(D), \quad \mathbf{m}(\varnothing) = 0, \text{ and } A \neq \varnothing. \tag{4}$$

Obviously, the combination rule may be generalized to combine multiple evidence.

Because there is one-to-one correspondence between **Bel** and $\mathbf{m}$, the orthogonal sum of belief functions $\mathbf{Bel} = \mathbf{Bel}_1 \oplus \mathbf{Bel}_2$ is defined in the obvious way.

Special kinds of **Bel** functions are very good at representing evidence. These functions are called *simple* and *separable support* functions. **Bel** is a *simple support* function if there exists an $F \subseteq \Theta$ called the focus of **Bel**, such that $\mathbf{Bel}(\Theta) = 1$ and

$$\mathbf{Bel}(A) = \begin{cases} \mathbf{s}, & \text{if } F \subseteq A \text{ and } A \neq \Theta \\ 0, & \text{otherwise} \end{cases}, \tag{5}$$

where $\mathbf{s}$ is called **Bel**'s *degree of support*.

A *separable support function* is either a simple support function or an orthogonal sum of simple support functions. Separable support functions are very useful when we want to combine evidences from several sources. If **Bel** is a simple support function with focus $F \neq \Theta$, then $\mathbf{m}(F) = \mathbf{s}$, $\mathbf{m}(\Theta) = 1 - \mathbf{s}$, and $\mathbf{m}$ is 0 elsewhere.

Let F be a focus for two simple support functions with degrees of support $\mathbf{s}_1$ and $\mathbf{s}_2$, respectively. If $\mathbf{Bel} = \mathbf{Bel}_1 \oplus \mathbf{Bel}_2$ then $\mathbf{m}(F) = 1 - (1 - \mathbf{s}_1)(1 - \mathbf{s}_2)$, $\mathbf{m}(\Theta) = (1 - \mathbf{s}_1)(1 - \mathbf{s}_2)$, and $\mathbf{m}$ is 0 elsewhere.

3. THE CLASSIFICATION

Assume that we have an unlabeled input vector $\bar{\mathbf{x}}$. Let N be the number of different classifiers f^n, $n = 1, \ldots, N$. Also assume that each classifier produces an output vector $\bar{\mathbf{y}}^n \in \mathbf{R}^K$, $\bar{\mathbf{y}}^n = f^n(\bar{\mathbf{x}})$. Here K is the number of classes (in case of character recognition, $K = 10$ for digit classifiers and $K = 36$ for alphanumeric classifiers). For an individual classifier we assign class j to the input vector $\bar{\mathbf{x}}$ if $y_j = \max_{1 \leq k \leq K} y_k$. This decision rule does not give us a chance to say what is the measure of confidence of our classification result. For example, the output vectors $\bar{\mathbf{y}}_1 = (0, 0, 0, 1)$ and $\bar{\mathbf{y}}_2 = (0.2, 0.2, 0.2, 0.202)$ both yield the same class assignment, class 4, but the quality of this decision may be considered quite poor for $\bar{\mathbf{y}}_2$. So, if we want to combine N classifiers, this decision rule permits us to use the majority voting scheme only, which cannot take into account the quality of each vote. Suppose that for each classifier f^n and each candidate class k, we calculated the value $e_k(\bar{\mathbf{y}}^n) = e_k(f^n(\bar{\mathbf{x}}))$, which represents some measure of evidence for the proposition "$\bar{\mathbf{y}}^n$ is of class k." If we introduced these values in terms of the Dempster–Shafer theory we could combine these evidences according to this theory and choose the class with the highest evidence.

4. EXISTING METHODS FOR COMPUTATION OF EVIDENCES

To calculate evidence for a neural network output, we might consider a posterior probability of each class, given an output vector, as a basic probability assignment. To estimate class-conditional probability distribution for all K classes, we have to produce multidimensional distribution for output vector given each class. For this purpose, K histograms for K-dimensional output vectors of the training set have to be built. In these histograms, the bin size should be small enough to yield sufficient precision. If the histogram has m bins in each coordinate, the total number of bins, m^K, is too large even for a training set of substantial size. Such a histogram in practice cannot be regarded as a realization of a continuous probability density function without rather arbitrary simplifications and approximations.

Mandler and Schurmann (1988) used a combination method based on the Dempster–Shafer theory for

nearest neighbor classifiers with different distance measures. Statistical analysis of distances between learning data and a number of reference points in the input space was carried out to estimate distributions of intra- and interclass distances. These distributions were used to calculate class-conditional probabilities that were transformed into evidences and combined.

Attempts to apply a similar approach to neural network outputs brought forward questions about a choice of reference vectors and a distance measure. In addition, we would prefer to avoid approximations associated with estimation of parameters of statistical models for intra- and interclass distances. We shall present our method in the next section.

5. PROPOSED METHOD

Now we introduce a different method for calculation of evidences. We would like these values to reflect the classification abilities of each classifier. It appears that sets of outputs $\{f^n(\bar{\mathbf{x}})\}$, $n = 1, \ldots N$ computed for the training data can provide relevant information and we shall use it in our computations.

Let $\{\bar{\mathbf{x}}_k\}$ be a subset of the training data corresponding to a class k. Let $\bar{\mathbf{E}}_k^n$ be the mean vector for a set $\{f^n(\bar{\mathbf{x}}_k)\}$ for each classifier f^n and each class k. $\bar{\mathbf{E}}_k^n$ is a reference vector for each class k and $d_k^n = \phi(\bar{\mathbf{E}}_k^n, \bar{\mathbf{y}}^n)$ is a proximity measure for $\bar{\mathbf{E}}_k^n$ and $\bar{\mathbf{y}}^n$. We want the values of this function to vary between 1 and 0 with the maximum when output vector coincides with a reference vector. We shall discuss a specific form for the function ϕ later. Now we need to transform these proximity measures into evidences $e_k(\bar{\mathbf{y}}^n)$.

Consider a frame of discernment $\Theta = \{\theta_1, \ldots, \theta_K\}$, where θ_k is the hypothesis that "$\bar{\mathbf{y}}^n$ is of class k." For any classifier f^n and each class k, a proximity measure d_k^n can represent evidence *pro*-hypothesis θ_k, and all d_i^n, with $i \neq k$, can represent evidences *pro* $\neg\theta_k$ or *contra* θ_k. We can use d_k^n as a degree of support for a simple support function with focus θ_k. This yields the basic probability assignment

$$\mathbf{m}_k(\theta_k) = d_k^n \quad \text{and} \quad \mathbf{m}_k(\Theta) = 1 - d_k^n. \tag{6}$$

In a similar manner, d_i^n are degrees of support for simple support functions with a common focus $\neg\theta_k$, if $i \neq k$. The combination of these simple support function with focus $\neg\theta_k$ is a separable support function with the degree of support $1 - \prod_{i\neq k}(1 - d_k^n)$. The corresponding basic probability assignment is

$$\mathbf{m}_{\neg k}(\neg\theta_k) = 1 - \prod_{i\neq k}(1 - d_k^n) \quad \text{and}$$

$$\mathbf{m}_{\neg k}(\Theta) = 1 - \mathbf{m}_{\neg k}(\neg\theta_k) = \prod_{i\neq k}(1 - d_k^n). \tag{7}$$

Combining our knowledge about θ_k we obtain the evidence $e_k(\bar{\mathbf{y}}^n) = \mathbf{m}_k \oplus \mathbf{m}_{\neg k}$ *pro* θ_k for class k and classifier n:

$$e_k(\bar{\mathbf{y}}^n) = \frac{d_k^n \prod_{i\neq k}(1 - d_i^n)}{1 - d_k^n[1 - \prod_{i\neq k}(1 - d_i^n)]}. \tag{8}$$

Finally, evidences for all classifiers may be combined according to the Dempster–Shafer rule to obtain a measure of confidence for each class k for the input vector $\bar{\mathbf{x}}$: $e_k(\bar{\mathbf{x}}) = e_k(\bar{\mathbf{y}}^1) \oplus \ldots \oplus e_k(\bar{\mathbf{y}}^N)$. $e_k(\bar{\mathbf{y}}^n)$, after an appropriate normalization, can be considered as Bayesian evidence function with nonzero basic probability assignments only on atomic hypotheses. Hence $e_k(\bar{\mathbf{x}}) = C\prod_n e_k(\bar{\mathbf{y}}^n)$, where C is the normalizing constant. Now we assign class j to the input vector $\bar{\mathbf{x}}$ if $e_j = \max_{1\leq k\leq K} e_k(\bar{\mathbf{x}})$.

The major problem now is to find the most effective form of the function ϕ. Several candidate functions for a proximity measure d_k^n for $\bar{\mathbf{E}}_k^n$ and $\bar{\mathbf{y}}^n$ were considered:

$$d_k^n = 1 - |E_{kK}^n - y_K^n|;$$

$$d_k^n = 1 - \|\bar{\mathbf{E}}_k^n - \bar{\mathbf{y}}^n\|;$$

$$d_k^n = \exp(-\|\bar{\mathbf{E}}_k^n - \bar{\mathbf{y}}^n\|^m);$$

$$d_k^n = \frac{(1 + \|\bar{\mathbf{E}}_k^n - \bar{\mathbf{y}}^n\|^m)^{-1}}{\sum_{1\leq i\leq K}(1 + \|\bar{\mathbf{E}}_i^n - \bar{\mathbf{y}}^n\|^m)^{-1}};$$

$$d_k^n = \frac{1}{1 + \|\bar{\mathbf{E}}_k^n - \bar{\mathbf{y}}^n\|^m}; \quad d_k^n = \cos^m(\alpha_k^n),$$

where α_k^n is the angle between $\bar{\mathbf{E}}_k^n$ and $\bar{\mathbf{y}}^n$.

Of the functions tried, two were found to have the best performance on validation sets. One of them is $\cos^2(\alpha_k^n)$:

$$\phi_1(\bar{\mathbf{E}}_k^n, \bar{\mathbf{y}}^n) = \frac{(\sum_{1\leq i\leq K} E_{ik}^n y_i^n)^2}{\|\bar{\mathbf{E}}_k^n\|^2 \|\bar{\mathbf{y}}_k^n\|^2}. \tag{9}$$

The second one is a function based on the Euclidian distance between $\bar{\mathbf{E}}_k^n$ and $\bar{\mathbf{y}}^n$:

$$\phi_2(\bar{\mathbf{E}}_k^n, \bar{\mathbf{y}}^n) = \frac{(1 + \|\bar{\mathbf{E}}_k^n - \bar{\mathbf{y}}^n\|^2)^{-1}}{\sum_{1\leq i\leq K}(1 + \|\bar{\mathbf{E}}_i^n - \bar{\mathbf{y}}^n\|^2)^{-1}}. \tag{10}$$

Our approach has a useful property of punishing overconfident, overtrained classifiers: their averages of output activations over the training set will be close to zero or 1, and this automatically means that both our proximity measures will be smaller for "fuzzier" activation vectors corresponding to the test data.

6. EXPERIMENTS AND RESULTS

Our first experiment was conducted with handprinted digits from a private data base. The training set con-

TABLE 1
Performance of Individual Classifiers for Digits

Classifier	Reject 0%	Reject 5%
Gabor-LRF	95.7%	97.9%
Bitmap-LRF	94.7%	98.0%
Gabor-GRF	93.4%	96.0%

TABLE 2
Performance of Combinations of Classifiers for Digits

Classifiers	Proximity Measure ϕ_1		Proximity Measure ϕ_2	
	Reject 0%	Reject 5%	Reject 0%	Reject 5%
Bitmap-LRF	96.4%	98.7%	97.0%	99.1%
Bitmap-LRF and Gabor-GRF	95.7%	98.2%	96.4%	98.5%
Gabor-LRF and Gabor-GRF	95.8%	98.2%	95.8%	98.5%

tained 25,000 characters and the testing set contained 4000. Output activations of three classifiers were used. The first was a two-hidden-layer neural network trained by back propagation with local receptive fields (LRF) using direct bitmap input of 20 × 30 pixels (Pawlicki, 1991). The two other neural networks used as input a set of units corresponding to features extracted by projecting the original pixel input onto a basis of Gabor wavelets (Shustorovich, 1994). One of these was a two-hidden-layer neural network with 144 input units trained by back propagation with LRF, the other was a one-hidden-layer neural network with 113 input units trained by back propagation with global receptive fields (GRF). We refer to these classifiers as Bitmap-LRF, Gabor-LRF, and Gabor-GRF, respectively. The individual and combination results for digits are given in Tables 1 and 2. The proximity measures ϕ_1 and ϕ_2 are defined as in eqns (9) and (10), respectively. The combination of these three classifiers was not any better than the combination of the best pair (Bitmap-LRF and Gabor-LRF), which means that the third one could not add anything new to the combination of the two.

Alphanumeric classifiers were trained using a data base contained 27,720 characters. The testing set contained 12,960 characters. We used output activations of three classifiers: the above-mentioned Bitmap-LRF and two polynomial classifiers, namely, a polynomial classifier with simple quadratic features (SQF) and a polynomial classifier with "fuzzy" features (FF) (Anderson & Gaborski, 1993). In both cases, the polynomial classifier is a combination of the classical least square method and a neural network-type supervised training algorithm. Characters are converted, nonlinearly, to feature vectors using different quadratic polynomials of the pixels. We refer to these classifiers as Poly-SQF and Poly-FF, respectively. The individual and combination results for alphanumerics are shown in Tables 3 and 4. The proximity measures ϕ_1 and ϕ_2 are the same as those in Table 2. As we can see in the tables, the best combination of classifiers allowed 30% reduction in error rates for digits and 25% for alphanumerics compared to the best individual classifier. These results can be favorably compared with those of the majority voting scheme. When applied to the outputs of all three digits, it decreased misclassification error by 10% for corresponding testing set. The same result was obtained when the scheme was used for all three alphanumeric classifiers.

TABLE 3
Performance of Individual Classifiers for Alphanumeric Characters

Classifier	Reject 0%	Reject 5%
Poly-SQF	83.7%	85.9%
Bitmap-LRF	86.1%	88.3%
Poly-FF	86.3%	88.5%

There is a very important question related to a problem of combination of several classifiers. Suppose we have a set of classifiers of different architectures and based on different sets of features. All these classifiers have different recognition power. The question is, which subset of these classifiers is the most advantageous for the combination. Experiments with all our classifiers showed that a better result is not necessarily achieved on the combination of classifiers with better individual performance. In some cases it turns out that it is more important to combine more "independent" classifiers than those with better performance. For example, Table 2 shows that the combination of the results of Gabor-GRF and Bitmap-LRF is the same as the combination of the results of Gabor-GRF and Gabor-LRF in spite of the fact that the individual performance of the latter is better. Apparently, different feature extractors used during the preprocessing stage provide more independent results than different architectures of neural networks.

More experiments were conducted for the U.S. Census Bureau/NIST First OCR Systems Competition (1992). There were three categories of isolated handprinted characters: digits, and lowercase and uppercase letters. Three-quarters of the NIST data base were used for training individual classifiers, and the last quarter was divided between a validation set and an internal test. We entered the competition with combined algorithms in all three categories. For digits and lowercase letters, we integrated the results of Bitmap-LRF, Gabor-LRF, and Poly-FF classifiers. Gabor-LRF and Poly-FF were used for uppercase letters. The combination of the algorithms decreases misclassification error by 23% for digits, by 15% for uppercase, and by 25% for lowercase letters (on our designated test) compared to the best individual algorithm used in the combinations.

TABLE 4
Performance of Combinations of Classifiers for Alphanumeric Characters

Classifiers	Proximity Measure ϕ_1		Proximity Measure ϕ_2	
	Reject 0%	Reject 5%	Reject 0%	Reject 5%
Poly-SQF and Poly-FF	87.4%	89.7%	87.4%	89.5%
Poly-SQF and Bitmap-LRF	88.9%	91.3%	88.8%	90.7%
Poly-FF and Bitmap-LRF	89.7%	92.1%	89.6%	90.4%
Poly-SQF and Poly-FF and Bitmap-LRF	90.1%	92.4%	89.5%	91.4%

The performance of the algorithms allowed Eastman Kodak Company to finish the competition among the tight group of leaders.

REFERENCES

Anderson, P. G., & Gaborski, R. S. (1993). The polynomial method augmented by supervised training for handprinted character recognition. *Proceedings of the International Conference on Neural Networks and Genetic Algorithms* (pp. 417–422). Innsbruck, Austria: Springer-Verlag.

Barnett, J. A. (1981). Computational methods for mathematical theory of evidence. *Proceedings of Seventh International Joint Conference on Artificial Intelligence* (pp. 868–875). Vancouver, BC.

Mandler, E. J., & Schurmann, J. (1988). Combining the classification results of independent classifiers based on the Dempster/Shafer theory of evidence. *Pattern Recognition and Artificial Intelligence,* **X,** 381–393.

Pawlicki, T. F. (1991). Personal communication.

Shafer, G. (1976). *A mathematical theory of evidence.* Princeton: MIT Press.

Shustorovich, A. (in press). A subspace projection approach to feature extraction: the two-dimensional Gabor transform for character recognition. *Neural Networks.*

U.S. Census Bureau/NIST First OCR Systems Conference, May 27–28, 1992, Gaithersburg, MD.

Xu, L., Krzyzak, A., & Suen, C. Y. (1991). Associative switch for combining multiple classifiers. *Proceedings of the International Joint Conference on Neural Networks* (pp. I-43–48). Seattle, WA: IEEE Press.

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