ELSEVIER

Pattern Recognition Letters 17 (1996) 577–584

Pattern Recognition Letters

Fusion of handwritten word classifiers

Paul D. Gader *, Magdi A. Mohamed, James M. Keller

Department of Computer Engineering and Computer Science, University of Missouri–Columbia, Columbia, MO 65211, USA

Received 31 August 1995; revised 20 November 1995

Abstract

Methods for fusing multiple handwritten word classifiers are compared on standard data. A novel method based on data-dependent densities in a Choquet fuzzy integral is shown to outperform neural networks, Borda and weighted Borda counts, and Sugeno fuzzy integral.

Keywords: Handwriting recognition; Fuzzy integral; Borda count; Sugeno measures; Choquet integral; Sugeno integral; Classifier fusion; Neural networks

1. Introduction

Fusion of multiple classifier decisions is a powerful method for increasing classification rates in difficult pattern recognition problems. Researchers have found that in many applications it is better to fuse multiple relatively simple classifiers, generally involving different features and classification algorithms, than to try to build a single sophisticated classifier to achieve better recognition rates.

Classifier fusion methods include intersection of decision regions, voting, prediction by top choice combinations, Dempster–Shafer, fuzzy integrals, and neural networks (Ho et al., 1994; Huang and Suen, 1995; Keller et al., 1994). For classifier combination to be successful, classifiers must make different mistakes. A fusion method should emphasize the strengths of individual classifiers and subsets of classifiers, avoid weaknesses, and use dynamically available knowledge about the inputs, the outputs, the classes, and the classifiers.

Handwritten word recognition is difficult because of the large variation involved in the shape of characters, the illegible nature and ambiguity present in many handwritten characters, and the overlapping and the interconnection of the neighboring characters. In most applications the size of the lexicons (dictionaries) is large and the contents of the lexicons (the classes) are changing. The problem is more complex than traditional pattern recognition problems in that the number of classes is so large as to be practically infinite. This precludes the use of some decision combination methods that depend on knowing the number of classes and the identity of each class in advance. The existing development efforts have involved long evolutions of differing classification algorithms, usually resulting in a final design which is an engineering combination of many techniques (Gader et al., 1990; Ho et al., 1994; Huang and Suen, 1995; Hull et al., 1992; Suen et al., 1992).

One widely used fusion method in handwritten word recognition is the Borda count. It is simple to

* Corresponding author. E-mail: gader@ece.missouri.edu.

PII S0167-8655(96)00021-9

implement and requires no training. In this method, however, all classifiers are treated equally, which may not be desired when certain classifiers are more likely to be correct than others. Ultimately, more sophisticated techniques are necessary for fusion because different word recognizers do not contribute equally and do not place equivalent restrictions on the recognition results. A weighted Borda count was shown to achieve performance superior to the Borda count in (Ho et al., 1994).

The purpose of this paper is to describe a comparison between several decision combination strategies for handwritten word recognition: Borda count, weighted Borda count, fuzzy integrals, and multilayer perceptrons. Two types of fuzzy integrals, the Sugeno and the Choquet integral are used. The use of Borda counts has been reported by other researchers in machine printed word recognition (Ho et al., 1994).

Fuzzy integrals are non-linear functionals defined with respect to fuzzy measures. Fuzzy integrals differ from other paradigms in that both objective evidence supplied by various sources and the expected worth of subsets of these sources are considered in the fusion process (Keller et al., 1994). Fuzzy measures and integrals are thoroughly discussed in (Grabisch, 1994; Murofushi and Sugeno, 1991; Sugeno, 1977). Fuzzy integrals have been used in handwritten character classification (Cho, 1995; Keller et al., 1994; Lee and Srihari, 1993; Yamaoka et al., 1994)

Our application of the fuzzy integrals is novel; previous applications have used prior knowledge about the worth of each classifier for each class to define parameters of the fuzzy measures. This approach cannot be directly used in word recognition because of the huge number of classes. We have developed an approach in which the parameters are data dependent. This approach outperforms a non-data dependent approach. It also outperforms the Borda and weighted Borda count methods, even when data dependent weights are used, and multilayer perceptrons.

The paper briefly reviews main features of the word recognition techniques, fuzzy integrals, and decision combination strategies. Experimental results from the individual and combined techniques are also provided. Several neural networks were trained on the same data used by the fuzzy integral. We compare the fuzzy integral fusion to neural networks in two ways. In both cases, the neural networks were not able to approach the performance obtained with the fuzzy integral. We also provide analysis of the inability of the networks to achieve comparable recognition rates.

2. Word recognition techniques

The techniques used to perform handwritten word recognition apply the contextual knowledge provided by the lexicons in the process of recognizing individual words. The first is a segmentation-based technique that uses dynamic programming to compute the match between the word images and candidate strings. The other two techniques are segmentation-free techniques that compute features from each column of the word image and use classical and fuzzy hidden Markov models for computing the matching scores. We briefly present the three techniques. They are described in great detail in (Gader et al., 1995 a,b,c; Mohamed and Gader, 1995; Mohamed, 1995).

2.1. Segmentation-based model (SBM)

The inputs are a binary image of a word and a lexicon. Dynamic programming is used to efficiently compute the best matches between segmentations of the word image and each string in the lexicon. The SBM technique produces a confidence value between 0 and 1 for each string in the lexicon for which a segmentation of the appropriate length can be found. Otherwise, the system produces a negative confidence value. This confidence measure is used in the data dependent combination strategies.

2.2. Hidden Markov model (HMM)

This segmentation-free technique constructs a continuous density hidden Markov model for each string in the lexicon by concatenating character models and performs classification according to the matching scores for the optimal state sequence. The description of a word binary image as an ordered list of observation vectors is accomplished by encoding a combination of features computed from each column in the given image. A separate model is trained for each upper and lower character using the transition and gradient features computed inside word images. This technique is appropriate for word images when segmentation is ambiguous and prone to failure.

2.3. Fuzzy hidden Markov model (FHMM)

This segmentation-free technique is similar to the second technique (HMM). The differences are that in the fuzzy HMM we deal with continuous fuzzy set memberships instead of continuous densities for the symbol probabilities, conditional fuzzy measures instead of the conditional probability measures for the transitional probabilities, and the computation of the forward and backward variables is performed using the Choquet integral assuming a possibility measure. Although this technique uses the same features as the previous one, fusing the information using a non-linear integration results in different character model parameters when training, and also different matching scores when testing. The HMM and FHMM trained with the same training data and structures make different mistakes and therefore give better results when fused.

3. Fuzzy measures and integrals

3.1. Fuzzy measures

Let X be an arbitrary set and Ω a sigma-algebra of subsets of X. A set function $g: \Omega \to [0, 1]$ defined on Ω which has the following properties is called a fuzzy measure.

(1) $g(\emptyset) = 0$, $g(X) = 1$.
(2) If A, $B \subset \Omega$ and $A \subset B$, then $g(\mathrm{A}) \leqslant g(B)$.
(3) If $F_n \in \Omega$ for $1 \leqslant n < \infty$ and the sequence $\{F_n\}$ is monotone (in the sense of inclusion), then

$$\lim_{n\to\infty} g(F_n) = g\Big(\lim_{n\to\infty} F_n\Big).$$

In general, the fuzzy measure of the union of two disjoint subsets cannot be directly computed from the measures of the subsets. In light of this, Sugeno introduced the so called λ-fuzzy measure satisfying the following additional property: for all $A, B \subset X$ with $A \cap B = \emptyset$,

$$g(A \cup B) = g(A) + g(B) + \lambda g(A) g(B),$$

for some $\lambda > -1$.

Let $X = \{x_1, x_2, \ldots, x_n\}$ be a finite set and let $g^i = g(\{x_i\})$. The values g^i are called the densities of the measure. The value of λ can be found from the equation: $g(X) = 1$, which is equivalent to solving the equation

$$\lambda + 1 = \prod_{i=1}^{n} (1 + \lambda g^i).$$

3.2. Sugeno integral

Sugeno fuzzy integrals combine objective evidence for a hypothesis with the prior expectation of the importance of that evidence to the hypothesis. Let (X, Ω) be a measurable space and let $h: X \to [0, 1]$ be an Ω-measurable function. The fuzzy integral over $A \subseteq X$ of the function h with respect to a fuzzy measure g is defined by

$$\int_A h(x) \circ g(\cdot) = \sup_{E \subseteq X} \Big[\min\Big(\min_{x \in E} h(x), g(A \cap E)\Big)\Big] = \sup_{\alpha \in [0,1]} \big[\min(\alpha, g(A \cap F_\alpha))\big],$$

where $F_\alpha = \{x \mid h(x) \geqslant \alpha\}$.

The calculation of the Sugeno integral when X is a finite set is easily given. Suppose

$$h(x_1) \geqslant h(x_2) \geqslant \cdots \geqslant h(x_n)$$

(if not, X is rearranged so that this relation holds). A Sugeno fuzzy integral, e, with respect to a fuzzy measure g over X can be computed by

$$e = \max_{i=1,\ldots,n} \big[\min(h(x_i), g(A_i))\big],$$

where $A_i = \{x_1, x_2, \ldots, x_i\}$.

When g is the λ-fuzzy measure, the values of $g(A_i)$ can be computed recursively as

$$g(A_1) = g(\{x_1\}) = g^1,$$

$$g(A_i) = g^i + g(A_{i-1}) + \lambda g^i g(A_{i-1}),$$

for $1 < i \leqslant n$.

Calculating the Sugeno integral with respect to a λ-fuzzy measure requires the fuzzy densities only.

3.3. Choquet integral

The original definition given by Sugeno (1977) for the fuzzy integral is not a proper extension of the Lebesgue integral, in the sense that the Lebesgue integral is not recovered when the measure is additive. To avoid this drawback, Murofushi and Sugeno (1991) proposed the Choquet integral, referring to a functional defined by Choquet in a different context. Let h be a function on X with values in [0,1] and g be a fuzzy measure. The Choquet integral is

$$\int_X h(x) \circ g(\cdot) = \int_0^{+\infty} g(A_\alpha)\, \mathrm{d}\alpha,$$

where $A_\alpha = \{x \mid h(x) > \alpha\}$.

If X is a discrete set, the Choquet integral can be computed as follows:

$$e = \sum_{i=1}^{n} [h(x_i) - h(x_{i-1})]\, g_i^n,$$

where

$$h(x_1) \leqslant h(x_2) \leqslant \cdots \leqslant h(x_n),$$

$$h(x_0) = 0, \quad \text{and}$$

$$g_i^j = \begin{cases} g(\{x_i, x_{i+1}, \ldots, x_j\}), & i \leqslant j, \\ 0, & \text{otherwise}. \end{cases}$$

The Choquet integral reduces to the Lebesgue integral for probability measures. As for the Sugeno integral, calculating the Choquet integral for a λ-fuzzy measure requires the densities only. Assigning densities appropriately is crucial for successful application of the fuzzy integral to pattern recognition.

3.4. Use of fuzzy integrals in classifier fusion

Fuzzy integrals have been used for classifier fusion in several applications. In these cases, there is a fixed number of classifiers and a fixed number of classes. The classifiers are represented by the set X above. Each classifier produces a confidence value for each class. These confidence values are represented by the function h. A fuzzy measure on X is constructed for each class. In the case that a λ-fuzzy measure is used, each density represents the worth of the corresponding classifier for that class. The overall confidence for the class is the fuzzy integral value. The class with the largest integral value can be taken as the final decision if a crisp decision is needed.

This traditional usage is not possible here because the classes in handwritten word recognition are represented by the lexicon and are therefore dynamic. Therefore, we cannot assign density values to all possible strings. In the next section, we describe our novel solution to this dilemma.

4. Decision fusion strategies

The decision fusion strategies all use the ranks of the strings provided by each word recognizer. The HMM and FHMM do not produce output confidence values that can be compared to each other or to those produced by the SBM. The use of ranks provides a measurement which is comparable across recognizers. Moreover, we use only the top n strings in the lexicon for each recognizer. For the experiments reported below, n was chosen to be 5. In general, if a correct string is not among the top 5 choices by at least one of the classifiers, it will be extremely difficult to classify it correctly. For a given word image and lexicon, each classifier produces an ordering of the lexicon. The kth string in each ordering of the lexicon is assigned the rank $1 - k/n$. If $k > n$, the rank is defined to be 0.

The Borda count associated with a string in a lexicon is defined as the sum of the number of strings that are "below" the string in the different orderings of the lexicons produced by the classifiers. We use the equivalent definition that the Borda count is the sum of the ranks. The weighted Borda count is the weighted sum of the ranks. The weights can be fixed for every classifier or they can be a function of the match confidence between the image and the lexicon string. The same is true for the density values for the fuzzy integrals; we describe how we choose the weights/density values below.

The fuzzy integrals use the ranks for the values of the function $h(x)$. The density values are generated using two methods. In the first method, we assign

Table 1(a)
Top five choices for three classifiers

Rank	HMM	FHMM	SBM
1.0	"grant"	"stpaul"	"island"
0.8	"island"	"grant"	"grant"
0.6	"granada"	"island"	"salem"
0.4	"burwell"	"oneill"	"nehawka"
0.2	"nehawka"	"o'neill"	"roseland"

Table 1(b)
The non-data dependent measures for some of the strings among the top five choices for the three classifiers using fixed weights/densities of 0.65 for SBM, 0.25 for HMM, and 0.05 for FHMM

String	Borda count	Weighted Borda count	Sugeno integral	Choquet integral
"grant"	2.60	0.81	0.8	0.85
"island"	2.40	0.88	0.8	0.92
"nehawka"	0.60	0.31	0.4	0.32
"salem"	0.60	0.39	0.6	0.26
"granada"	0.60	0.15	0.25	0.15

each classifier a fixed density value. That density value is used for every string in every lexicon. It is considered to reflect the worth of each classifier. Examples of these non-data dependent methods for combining word classifiers are shown in Table 1. This example is difficult. The reader may have difficulty choosing the correct string.

The second method uses data dependent densities. The confidence value produced by the SBM is used to define the density value for the segmentation based technique. The density values for the HMM and the FHMM are then determined by a heuristic formula involving the SBM confidence and the agreement between the classifiers concerning the rank of each string. More precisely:

For each string in a lexicon, let

C_s = confidence value returned by the segmentation-based classifier,
g_s = density of the segmentation-based classifier,
r_s = rank of the string by the SBM,
g_h = density of the HMM classifier,
r_h = rank of the string by the HMM,
g_f = density of the FHMM classifier,
r_f = rank of the string by the FHMM.

If a string is in the top n choices of the segmentation-based system, then

$$g_s = \max(\varepsilon,\ \alpha * C_s),$$
$$g_h = \beta * \sqrt{(1 - C_s) * (1 - |r_h - r_s|)},$$
$$g_f = \gamma * \sqrt{(1 - C_s) * (1 - |r_f - r_s|)}.$$

Otherwise

$$g_s = \max(\varepsilon,\ \alpha * C_s),$$
$$g_h = \beta * \sqrt{(1 - C_s) * (1 - |r_h - r_f|)},$$
$$g_f = \gamma * \sqrt{(1 - C_s) * (1 - |r_h - r_f|)}.$$

Here α, β and γ are parameters that can be optimized and $\varepsilon > 0$ is very small. The expression $\max(\varepsilon,\ \alpha * C_s)$ is used to keep the densities for the SBM nonnegative in the case that an appropriate segmentation cannot be found. This same method can be used to define weights for the weighted Borda count.

As an example, we show how the data-dependent Choquet combination method would assign confidence values for the strings "island" and "grant" in Table 1. The values of the parameters used for the computations shown in Table 2 are $\alpha = 0.900$, $\beta = 0.100$ and $\gamma = 0.400$.

5. Experimental results for the fuzzy integral fusion

The experiments were performed on handwritten words from the SUNY CDROM database. The BD words were used (Hull, 1994) The FHMM and HMM were trained using the standard training set in that database. The SBM method was trained using a different set of data. We used 126 words from the training set to "train" densities and weights for

Table 2
Example of data-dependent Choquet integral fusion

String	C_s	g_s	g_h	g_f	Choquet integral
"island"	0.75	0.68	0.02	0.04	0.92
"grant"	0.46	0.42	0.27	0.08	0.86

Table 3
Recognition results for individual classifiers

Classifier	Training result		Testing result	
	Top 1	Top 3	Top 1	Top 3
HMM	74.6%	86.5%	71.6%	86.8%
FHMM	74.6%	84.1%	73.2%	87.1%
SBM	82.5%	92.1%	83.9%	92.4%

some of our experiments. These words were not hand chosen; a random directory was chosen from the training directories. The directory just happened to have 126 words. All of the 317 BD city names from the test set were used for testing. We used the sets of lexicons that have average length 100 for both training and testing. The results of the individual classifiers are shown in Table 3.

The "training" method that we used for the weighted Borda count and the fixed density fuzzy integral approaches was exhaustive search. We let the weights/densities vary from 0.05 to 0.95 by increments of 0.05. The training method that was used for the data-dependent densities was exhaustive search on α, β and γ. In each case, optimal values for the parameters were found on the training set and then used on the testing set. The top 1 and top 3 results are shown in Table 4.

6. Comparison to neural networks

We attempted to train several neural networks to learn the classifier fusion from the same data that was used by the fuzzy integral. Each feature vector contained ten inputs: the segmentation confidence (C_s), the word ranks for the three classifiers (r_s, r_h, r_f), the data dependent densities for the three classifiers (g_s, g_h, g_f), and the fuzzy measures of the three 2-element subsets of classifiers, say g_{sh}, g_{sf} and g_{hf}. Hence, the neural networks had as input the segmentation confidence, the classifier outputs, and the fuzzy measure as defined earlier. The target was set to 0.9 if the string represented the correct choice for the current word image, and 0.1 if it was incorrect. Many architectures were investigated. Table 5 shows the best results obtained.

The neural networks were unable to come close to the results obtained by the fuzzy integral. Upon close examination, however, this is not surprising. In handwritten word recognition, we are not learning a nonlinear function in the same sense that standard pattern recognizers do. Since we must rank strings, there is a very large number of possible classes and we cannot use the standard class coding approach. This makes the task for a neural network extremely difficult, as seen in Table 5. In fact, it turns out that for this application, we may not even be learning a true function. It is possible that essentially identical input patterns (for different word images) can be

Table 5
Training and testing results of neural networks with crisp outputs

Architecture	Number of iterations	Training results	Testing results
10–5–1	1000	84.1%	80.4%
10–5–1	3000	84.9%	82.3%
10–5–1	6000	84.9%	81.4%
10–5–1	21000	86.5%	79.5%
10–10–1	2000	83.3%	81.4%
10–10–1	4000	83.3%	81.4%
10–10–1	10000	86.5%	81.7%
10–10–1	15000	88.1%	80.8%
10–10–5–1	5000	84.9%	82.0%
10–10–5–1	9000	86.5%	81.4%

Table 4
Training and testing results for combinations of classifiers

Combination approach	Optimal training result		Testing result	
	Top 1	Top 3	Top 1	Top 3
Data-dependent Choquet	89.7%	93.7%	88.0%	95.9%
Data-dependent Sugeno	89.7%	94.4%	86.4%	95.0%
Data-dependent weighted Borda	88.9%	94.4%	85.5%	94.0%
Fixed Choquet	88.1%	92.9%	82.0%	95.0%
Fixed Sugeno	88.1%	92.1%	85.2%	87.4%
Fixed weighted Borda	88.1%	92.9%	86.4%	94.3%
Borda	84.1%	92.9%	83.3%	93.7%

Table 6
Example of two very similar training patterns that are to be mapped to different outputs

r_h	r_f	r_s	g_h	g_f	g_s	C_s	g_{hf}	g_{hs}	g_{fs}	Desired output
1.000	0.800	1.000	0.264	0.059	0.563	0.626	0.332	0.911	0.641	0.900
1.000	0.800	1.000	0.262	0.059	0.572	0.635	0.328	0.913	0.648	0.100

Table 7
Training and testing results for neural networks with FI outputs

Architecture	Number of iterations	Training results	Testing results
10–10–1	1500	87.3%	83.9%
10–10–1	2500	87.3%	83.6%
10–5–1	1000	87.3%	84.5%
10–5–1	5000	88.1%	83.6%
10–2–1	500	88.1%	83.3%
10–2–1	1500	88.9%	83.3%
10–2–1	6000	88.9%	83.6%
10–2–1	20000	88.9%	83.3%

obtained, one for a correct match (desired output of 0.9) while the other is incorrect (desired output of 0.1). A real example of this from our training data set is shown in Table 6. The values of the features are extremely close but the desired outputs are different. It is very difficult for the neural network to learn without overtraining. One could argue that this argument shows that the features are not good. However, the fuzzy integral is able to use these features to achieve a significantly higher score than the neural network.

Since the results of the neural network fusion were lower than those generated by the fuzzy integral, we attempted to determine if a neural network could learn the fuzzy integral outputs. By using the fuzzy integral outputs as the desired outputs, we eased the problem of forcing the neural net to try to learn a nonfunction. These neural net outputs were then used to test the classifier fusion. Table 7 highlights the best results obtained. As can be seen, the nets did considerably better than those in the previous experiment. However, none approached the results generated by the fuzzy integral. We believe that this can be attributed to two factors. First, as stated above, we are not learning a straightforward classification function, which makes the task for the neural network considerably more difficult. Secondly, the very nature of the fuzzy integral ignores some of the input data – a mixture of confidence values and measures of worth. We believe that the neural networks have a hard time emulating that behavior. The fuzzy integral therefore provides a robust methodology for the fusion of information in decision making.

7. Conclusions

A major contribution of this paper is that it describes novel methods for using fuzzy integrals to combine word recognition results from multiple classifiers. In our experiments, Choquet fuzzy integrals with data dependent densities outperformed several popular methodologies.

References

Cho, S.-B. (1995). Combining multiple neural networks by fuzzy integral for robust classification. IEEE Trans. Syst. Man Cybernet. 25 (2), 380–384.

Gader, P.D., D. Hepp, B. Forester, T. Peurach and B.T. Mitchell (1990). Pipelined systems for recognition of handwritten digits in USPS zip codes. Paper presented at the *United States Postal Service Advanced Technology Conf.*, Washington, DC.

Gader, P.D., M. Mohamed and J. Chiang (1995a). Comparison of crisp and fuzzy character neural networks in handwritten word recognition. *IEEE Trans. Fuzzy Systems* 3 (3), 357–364.

Gader, P.D., M. Mohamed and J.H. Chiang (1995b). Handwritten word recognition with character and inter-character neural networks. *IEEE Trans. Syst. Man Cybernet.*, accepted.

Gader, P.D., M.P. Whalen, M.J. Ganzberger and D. Hepp (1995). Handprinted word recognition on a NIST data set. *Machine Vision and Its Applications* 8, 31–40.

Grabisch, M. (1994). Classification by fuzzy integral: performance and tests. *Fuzzy Sets and Systems* 65 (2), 255–273.

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.

Huang, T.S. and C.Y. Suen (1995). Combination of multiple experts for the recognition of unconstrained handwritten numerals. *IEEE Trans. Pattern Anal. Mach. Intell.* 17 (1), 90–94.

Hull, J., T. Ho, J. Favata, V. Govindaraju and S. Srihari (1992). Combination of segmentation-based and holistic handwritten word recognition algorithms. Paper presented at the *Second Internat. Workshop on Frontiers in Handwriting Recognition*, Chateau de Bonas, France.

Hull, I.J. (1994). A database for handwritten text recognition research. *IEEE Trans. Pattern Anal. Mach. Intell.* 16 (5), 550–554.

Keller, J.M., P.D. Gader, H. Tahani, J.-H. Chiang and M. Mohamed (1994). Advances in fuzzy integration for pattern recognition 65, 273–283.

Lee, D.-S. and S.N. Srihari (1993). Handprinted digit recognition: a comparison of algorithms. Paper presented at the *Third Internat. Workshop Frontiers Handwriting Recognition*, Buffalo, NY.

Mohamed, M. and P.D. Gader (1995). Handwritten word recognition using segmentation-free hidden Markov modeling and segmentation-based dynamic programming techniques. *IEEE Trans. Pattern Anal. Mach. Intell.*, accepted for publication.

Mohamed, M.A. (1995). Handwritten Word Recognition using Generalized Hidden Markov Models. Unpublished Ph.D., University of Missouri – Columbia, Columbia, MO.

Murofushi, T. and M. Sugeno (1991). A theory of fuzzy measures: representations, the Choquet integral, and null sets. *J. Math. Anal. Appl.* 159, 532–549.

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

Sugeno, M. (1977). Fuzzy measures and fuzzy integrals – a survey. In: M.M. Gupta, G.N. Saridis and B.R. Gaines, Eds., *Fuzzy Automata and Decision Processes*. North-Holland, New York, 89–102.

Yamaoka, F., Y. Lu, A. Shaout and M. Shridhar (1994). Fuzzy integration of classification results in a handwritten digit recognition system. Paper presented at the *Fourth Internat. Workshop Frontiers Handwriting Recognition*, Taipei, Taiwan.