PERGAMON

Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Pattern Recognition 38 (2005) 11–28

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Classifier combination based on confidence transformation

Cheng-Lin Liu*

Central Research Laboratory, Hitachi, Ltd., 1-280 Higashi-koigakubo, Kokubunji-shi, Tokyo 185-8601, Japan

Received 17 November 2003; accepted 24 May 2004

Abstract

This paper investigates the effects of confidence transformation in combining multiple classifiers using various combination rules. The combination methods were tested in handwritten digit recognition by combining varying classifier sets. The classifier outputs are transformed to confidence measures by combining three scaling functions (global normalization, Gaussian density modeling, and logistic regression) and three confidence types (linear, sigmoid, and evidence). The combination rules include fixed rules (sum-rule, product-rule, median-rule, etc.) and trained rules (linear discriminants and weighted combination with various parameter estimation techniques). The experimental results justify that confidence transformation benefits the combination performance of either fixed rules or trained rules. Trained rules mostly outperform fixed rules, especially when the classifier set contains weak classifiers. Among the trained rules, the support vector machine with linear kernel (linear SVM) performs best while the weighted combination with optimized weights performs comparably well. I have also attempted the joint optimization of confidence parameters and combination weights but its performance was inferior to that of cascaded confidence transformation-combination. This justifies that the cascaded strategy is a right way of multiple classifier combination.

Keywords: Classifier combination; Confidence transformation; Evidence combination; Fixed combination rule; Linear discriminant; Linear SVM; Weighted combination

1. Introduction

The combination of multiple classifiers has been intensively studied with the aim of overcoming the limitations of individual classifiers [1–3]. Classifiers differing in feature representation, architecture, learning algorithm, or training data exhibit complementary classification behavior and the fusion of their decisions can yield higher performance than the best individual classifier. The performance of a multiple classifier system relies on both the complementariness of the participating classifiers and the combination method. Hence, the research efforts in this field have focused on either the generation of complementary classifiers or the combination of a given set of classifiers.

Based on a given classifier set, the combination methods can be categorized according to the level of classifier outputs: abstract level (class label), rank level (rank order), and measurement level (class scores) [1,4]. In principle, the class scores provide richer information than the class label and the rank order and should give higher combination performance. However, measurement-level combination relies on a sophisticated combination method, especially when the constituent classifiers have different discriminant functions, which give measurements with diverse scales and physical meanings. The conversion of classifier outputs to crisp class label or rank order simplifies combination but loses useful information and will deteriorate the combination performance. In essence, the classifier outputs should be transformed to uniform measures that have similar scales. Preferably, the transformed measures represent the degree of confidence of

* Tel.: +81-42-323-1111; fax: +81-42-327-7746.
E-mail address: liucl@crl.hitachi.co.jp (C.-L. Liu).

0031-3203/$30.00
doi:10.1016/j.patcog.2004.05.013

decision, like the class posterior probability or likelihood [5–8].

In the context of measurement-level classifier combination, numerous methods have been proposed. Depending on whether tunable parameters are used or not, the combination methods can be divided into fixed rules and trained rules [9]. Fixed rules include the Dempster–Shafer (D–S) theory of evidence [10,11], the sum-rule (averaging), product-rule, max-rule, min-rule, median-rule [3], generalized order statistics [12], etc. Trained rules include various trainable classifiers (called meta-classifiers or meta-learners, including statistical classifiers, neural networks [13], support vector machines [14], etc.), optimized evidence combination [15], and typically, the weighted combination (weighted average). The weighted combination has a very small number of parameters and can yield competitive performance to sophisticated meta-classifiers if the classifier weights are optimized. Both meta-classifiers and weighted combination can cope well with the imbalanced individual performance and the dependence between the constituent classifiers to yield high combination performance [16].

The weighted combination is also called linear opinion pool or logarithm opinion pool depending on whether the class measures to combine represent either the class probability or log-likelihood [17]. The classifier weights can be estimated using various techniques, which I roughly categorize into three classes: behavior- (e.g., [17–19]), density- [20,21], and regression-based methods [22–24]. Behavior-based methods exploit the individual classifier behavior (such as confusion matrix) and the correlation between classifiers (such as joint confusion matrix) to estimate the classifier reliability, class-dependent reliability and data-dependent reliability, and integrate them into classifier weights. By density-based methods, the classifier weight is related to the prior probability that the input pattern is likely to be generated by the underlying density model of the participating classifier (veridical probability). While by regression, the classifier weights are tuned by optimizing an objective function, like the cross entropy (CE) [25], mean square error (MSE) [22] or minimum classification error (MCE) criterion [24].

In the context of classifier combination, the transformation of classifier outputs into confidence measures has not been addressed adequately except the works of Refs. [5,26,27]. Rather, many previous works assumed that the classifier outputs inherently represent the class probabilities or likelihood, as for neural networks [28,29] or parametric statistical classifiers [30]. A number of works have contributed to the confidence transformation of classifier outputs for general purpose [31–35]. My research group has compared a variety of confidence transformation methods which can transform the classifier outputs (similarity measures or dissimilarity measures in any scale) into four types of confidence measures: log-likelihood (linear), likelihood (exponential), sigmoid, and evidence [8], where the combination performance was evaluated on various classifier sets using the fixed sum-rule for combination. The results justified the promise of confidence transformation.

In this paper, I turn to evaluate the performance of various combination rules with selected confidence transformation methods. A confidence transformation method is the combination of a scaling function and a confidence type. I select three scaling functions: global normalization, one-dimensional Gaussian density modeling, and logistic regression (LR) with one input variable. The parameters of scaling functions (confidence parameters) are estimated on a validation data set of classifier outputs. Among the four confidence types, the exponential measure was abandoned since it has very wild range of value and makes the parameter estimation of trained combination difficult. The combination of three scaling functions and three confidence types gives nine confidence measures. The evidence measure of a class is obtained by combining the sigmoid measures of the pre-defined classes using the D–S theory of evidence and is expected to better represent the multi-class probability.

The transformed confidence measures are combined to make the final decision of classification. As to the fixed combination rules, I will present the results of the sum-rule, product-rule, max-rule, and median-rule. Other rules based on order statistics (like the min-rule) were shown to not outperform the max-rule in my experiments. As to the trained rules, I use the linear discriminant function (LDF) as the representative of meta-classifiers. Sophisticated classifiers with more parameters do not necessarily outperform the LDF because the classifier outputs are distributed in a restrictive subspace and the classes are well separated. The parameters of LDF are estimated by linear discriminant analysis (LDA) assuming multivariate Gaussian density for each class with equal covariance [30]. This is also called supra Bayesian procedure in Ref. [20]. On the other hand, the nearest-mean rule [9] as a special case of the decision template method [36] reduces to a linear discriminant. The parameters of linear discriminant can also be estimated by regression via optimizing the CE, MSE, or MCE criterion. I also test the SVM with linear kernel (linear SVM), which results in a linear discriminant as well. For weighted combination, I optimize the classifier weights by regression under the CE, MSE, and MCE criteria. I also attempt the weighted combination of order statistics, as has been used in other applications [37].

The above methods undergo a two-stage cascaded procedure: confidence transformation followed by confidence fusion. Alternatively, we can optimize the confidence parameters and classifier weights simultaneously on a validation data set by optimizing an integrated objective function related to the combination performance. I experimented this strategy but the promise was not justified. The combination performance of joint confidence and combination parameter optimization is inferior to that of cascaded confidence transformation-combination. I will present the results of joint parameter optimization, however.

The rest of this paper is organized as follows: Section 2 gives an formulation of classifier combination problem;

Section 3 reviews the confidence transformation methods; Section 4 describes the combination rules; Section 5 presents the experimental results and Section 6 provides my concluding remarks.

2. Problem formulation

To classify a pattern X into M classes $\{\omega_1, \ldots, \omega_M\}$, assume that we have K classifiers (classification experts) $\{E_1, \ldots, E_K\}$, each using a feature vector $\mathbf{x}_k$, $k=1, \ldots, K$. On an input pattern, each classifier E_k outputs discriminant measures (scores) to all classes: $d_{kj}(\mathbf{x}_k)$, $j=1, \ldots, M$. The measurements represent the class membership/similarity or dissimilarity/distance. The input pattern is classified to the class of maximum similarity or minimum dissimilarity.

By measurement-level combination, the decisions of the constituent classifiers are deferred and the final decision is made after the outputs of multiple classifiers are fused to give combined class membership measures. Formally, the combined measures are computed by a meta-classifier

$$D_j(X)=F\begin{pmatrix} d_{11}(\mathbf{x}_1), & \cdots, & d_{1M}(\mathbf{x}_1), \\ \vdots & \ddots & \vdots \\ d_{K1}(\mathbf{x}_K), & \cdots, & d_{KM}(\mathbf{x}_K) \end{pmatrix}, \quad j=1, \ldots, M,$$

where $D_j(X)$ denotes the combined measure for class ω_j. For many combination rules (such as fixed rules and weighted combination), the combined measure of one class considers solely the corresponding outputs of the same class

$$D_j(X) = F(d_{1j}(\mathbf{x}_1), \ldots, d_{Kj}(\mathbf{x}_K)), \qquad j=1, \ldots, M.$$

Transforming the raw outputs to confidence measures can improve the classification performance of combination. The outputs of classifier E_k are transformed to confidence measures z_{kj}, $j=1, \ldots, M$, by

$$z_{kj}(\mathbf{x}_k) = z_{kj}(\mathbf{d}_k) = T(d_{k1}, \ldots, d_{kM}). \tag{1}$$

Based on this, the combination rules become

$$D_j(X)=F\begin{pmatrix} z_{11}(\mathbf{x}_1), & \cdots, & z_{1M}(\mathbf{x}_1), \\ \vdots & \ddots & \vdots \\ z_{K1}(\mathbf{x}_K), & \cdots, & z_{KM}(\mathbf{x}_K) \end{pmatrix}, \quad j=1, \ldots, M \tag{2}$$

for meta-classifier and

$$D_j(X) = F(z_{1j}(\mathbf{x}_1), \ldots, z_{Kj}(\mathbf{x}_K)), \quad j=1, \ldots, M \tag{3}$$

for fixed/weighted combination.

Since the confidence measures are transformed from the classifier outputs and are dependent on the input pattern, it has been shown that the confidence measures function as data-dependent weights to some extent [8]. For example, assume that the confidence measure approximates the Bayesian likelihood

$$z_{kj} = p(\mathbf{x}_k|\omega_j)P(\omega_j),$$

the posterior probabilities are calculated by

$$p(\omega_j|\mathbf{x}_k) = \frac{z_{kj}}{\sum_{i=1}^{M} z_{ki}}.$$

By averaging the likelihood measures, the combined class measures are

$$D_j(X) = \frac{1}{K}\sum_{k=1}^{K} z_{kj} = \frac{1}{K}\sum_{k=1}^{K}\left(\sum_{i=1}^{M} z_{ki}\right) p(\omega_j|\mathbf{x}_k), \tag{4}$$

wherein the sum of likelihood $\sum_{i=1}^{M} z_{ki}$ can be viewed as the weight of classifier E_k, akin to density-based weighted combination. By averaging the normalized posterior probabilities, this weight will be lost. This may partially explain why normalizing the class confidences to unity of sum deteriorates the combination performance, as has been shown by our previous results [8].

3. Confidence transformation

In our formulation, a confidence transformation method is the combination of a scaling function and an activation function. The scaling function shifts and re-scales the classifier output to a moderate range such that the outputs of different classifiers are comparable. The re-scaled output is transformed to confidence measure using an activation function corresponding to one of four confidence types: log-likelihood (linear), likelihood (exponential), sigmoid, and evidence. The scaling functions include global normalization, one-dimensional Gaussian density modeling, multivariate Gaussian density modeling, and LR with varying input variables. The Gaussian models and the LR yield higher combination performance than the global normalization. The LR with one input variable performs as well as or better than that with multiple inputs. For evaluating various combination rules, I select the global normalization (as a baseline), the one-dimensional Gaussian model, and the LR with one input variable. The confidence types and scaling functions are briefly reviewed in the following, and more details can be found in Ref. [8].

3.1. Confidence types

For probabilistic fusion of classifier outputs, the transformed confidence measures are desired to approximate the Bayesian likelihood $p(\mathbf{x}|\omega_j)P(\omega_j)$ or class posterior probability $p(\omega_j|\mathbf{x})$ (in this section, I concern the confidence transformation of one classifier, so the subscript k of classifier index is dropped).

The likelihood and the class posterior probability are approximated with exponential measure and sigmoid measure, respectively. For transforming the classifier outputs $\mathbf{d} = [d_1 \dots d_M]^T$, a scaling function is used to shift and rescale the outputs to give a basic measure $f_j(\mathbf{d})$ for each class (the scaling functions will be described later). Then the exponential measure and sigmoid measure are computed by

$$z_j^e = \exp[f_j(\mathbf{d})] \tag{5}$$

and

$$z_j^s = \frac{1}{1+\exp[-f_j(\mathbf{d})]}, \tag{6}$$

respectively.

The normalization of exponentials to unity of sum results in the soft-max of posterior probabilities. The logarithm of Bayesian likelihood (log-likelihood) can be used as another type of confidence. In the case of exponential form, the log-likelihood is simply the scaling function

$$z_j^l = f_j(\mathbf{d}). \tag{7}$$

The log-likelihood measure is also called linear measure since the scaling function is basically linear with respect to the classifier outputs.

The sigmoid function is prevalently used for the neuronal outputs of neural networks to approximate the class posterior probabilities [28,29]. On the other hand, assuming that the output measure of one class separates the class from the others and the two meta-classes have one-dimensional Gaussian densities, the class posterior probability is a sigmoid function [31]. The plausibility of approximating Bayesian likelihood using exponential measure is rooted in parametric classification with Gaussian density assumptions, where the discriminant function is often the logarithm of Bayesian likelihood [30].

Viewing the sigmoid measure as two-class (one class and the others) probability, the multi-class probabilities can be obtained by combining the sigmoid measures according to the D–S theory of evidence [38,39]. In the framework of discernment, we have $2M$ focal elements (singletons and negations) $\{\omega_1, \overline{\omega}_1, \dots, \omega_M, \overline{\omega}_M\}$ with basic probability assignments (BPAs) $m_j(\omega_j) = z_j^s$, $m_j(\overline{\omega}_j) = 1 - z_j^s$, then the combined evidence of ω_j is

$$\begin{aligned} z_j^c = m(\omega_j) &= A \cdot m_j(\omega_j) \prod_{i=1, i\neq j}^{M} m_i(\overline{\omega}_i) \\ &= A \cdot z_j^s \prod_{i=1, i\neq j}^{M} (1 - z_i^s), \end{aligned} \tag{8}$$

where

$$A^{-1} = \sum_{j=1}^{M} z_j^s \prod_{i=1, i\neq j}^{M} (1 - z_i^s) + \prod_{i=1}^{M} (1 - z_i^s).$$

The exponential, sigmoid, and evidence measures can be normalized to sum up to one. The normalization of sum, however, may deteriorate the combination performance since the sum of class probabilities or likelihood is related to the prior probability that the input pattern is generated by the underlying density model, as shown in Eq. (4). Still, I will give the combination results using normalized confidence measures.

Functions (5)–(8) are called activation functions [8]. The exponential measure is not used in our experiments of various combination rules because it has very wild range of value and makes the parameter estimation of regression-based combination difficult. As to the normalized exponential measure, it has been shown that by normalizing the sum, the exponential measure is equivalent to the evidence measure [8].

3.2. Scaling functions

To manage the range of classifier outputs, one simple strategy is to re-scale the output measures to zero mean and standard deviation one

$$f_j(\mathbf{d}) = \frac{d_j - \mu_0}{\sigma_0}, \tag{9}$$

where μ_0 and σ_0^2 are the mean value and the variance of the classifier outputs pooled in one Gaussian distribution. This scaling function is referred to as *Global normalization*. For classifiers that output dissimilarity measures, the sign of the scaling function should be reversed. The global normalization does not generate proper class probability or likelihood estimate because the class information is not considered in scaling parameter estimation.

Schürmann gave a sigmoid measure as class posterior probability by assuming that the class output d_j functions as a two-class discriminant that separates class ω_j from the other classes. Assuming one-dimensional Gaussian densities for the two meta-classes (ω_j and the others) with equal variance,

$$\begin{aligned} p(d_j|\omega_j) &= \frac{1}{\sqrt{2\pi}\sigma_j^+} \exp\left[-\frac{(d_j-\mu_j^+)^2}{2(\sigma_j)^2}\right], \\ p(d_j|\overline{\omega}_j) &= \frac{1}{\sqrt{2\pi}\sigma_j^-} \exp\left[-\frac{(d_j-\mu_j^-)^2}{2(\sigma_j)^2}\right], \end{aligned}$$

the class posterior probability is

$$P(\omega_j|d_j) = \frac{1}{1+\exp\left\{-\alpha\left[d_j - (\beta + \gamma/\alpha)\right]\right\}}, \tag{10}$$

where

$$\alpha = \frac{\mu_j^+ - \mu_j^-}{\sigma_j^2},$$

$$\beta = \frac{\mu_j^+ + \mu_j^-}{2},$$

$$\gamma = \log \frac{P(\overline{\omega}_j)}{P(\omega_j)}.$$

From the sigmoid form (10) the scaling function is extracted as

$$f_j(\mathbf{d}) = \alpha\left[d_j - \left(\beta + \frac{\gamma}{\alpha}\right)\right], \tag{11}$$

which is referred to as *Gaussian* scaling function. The parameters $\{\mu_j^+, \mu_j^-, \sigma_j^2\}$ are estimated on a validation data set by maximum likelihood (ML). The ratio of prior probability is set to $\gamma = \log(P(\overline{\omega}_j)/P(\omega_j)) = \log M$ by assuming equal prior probabilities and that out-of-class patterns are present as well as the M defined classes.

LR has been applied to the confidence evaluation of classifiers [32–34]. In LR, a weighted linear combination of input variables is used to approximate the log-odds (logit) of the probability of response. For multi-class problem, assume that m variables $x_1, \ldots, x_m$ are used to estimate the probabilities of each class

$$\pi_j(\mathbf{x}) = \frac{1}{1 + \exp[-(\sum_{i=1}^m \beta_{ji} x_i + \beta_{j0})]}, \quad j = 1, \ldots, M,$$

where $\beta_{j1}, \ldots, \beta_{jm}$ are weights and β_{j0} is the bias. On a validation sample set $(\mathbf{x}^n, c^n)$, $n = 1, \ldots, N$, the target probability of true class c^n is $t_{c^n} = 1$ and that of the other classes is $t_j = 0$, $j \neq c^n$. The objective is to maximize the likelihood of validation samples

$$\max\ L = \prod_{n=1}^{N} \prod_{j=1}^{M} \pi_j^{t_j} (1 - \pi_j)^{1 - t_j},$$

and equivalently, to minimize the negative log-likelihood

$$\min\ J = -\sum_{n=1}^{N} \sum_{j=1}^{M} [t_j \log \pi_j + (1 - t_j) \log(1 - \pi_j)]. \tag{12}$$

Criterion (12) is often referred to as cross entropy (CE) [25].

In our implementation, a term of weight decay is added to the criterion to alleviate overfitting

$$\min\ J = -\sum_{n=1}^{N} \sum_{j=1}^{M} [t_j \log \pi_j + (1 - t_j) \log(1 - \pi_j)] + \lambda \sum_{j=1}^{M} \sum_{i=1}^{m} \beta_{ji}^2, \tag{13}$$

where λ is a pre-specified coefficient for weight decay. The weights and biases are computed in minimizing the criterion by stochastic gradient descent [40].

For confidence transformation, the variables are selected from the classifier outputs, and the scaling function is extracted from the sigmoid form of probability

$$f_j(\mathbf{d}) = \sum_{i=1}^{m} \beta_{ji} x_i + \beta_{j0}.$$

The scaling function is combined with all the four activation functions to give four confidence measures.

Our previous results of confidence-based classifier combination show that the LR with one input variable (*LR-1*) performs as well as or better than those with multiple inputs [8]. In *LR-1*, the scaling function of each class uses its own output as the only variable

$$f_j(\mathbf{d}) = \beta_{j1} d_j + \beta_{j0}. \tag{14}$$

This function has the identical form to that of *Gaussian* scaling function (11) yet the parameters are estimated in different ways.

Our previous results also show that sharing class parameters in LR does not sacrifice the combination performance. In the experiments of this paper, I share the class parameters for both the Gaussian scaling method and LR. In my implementation of LR, the classifier outputs are first re-scaled by global normalization and in stochastic gradient search, the initial values of parameters are set to $\beta_{j1} = 1$ and $\beta_{ji} = 0$, $i \neq 1$.

4. Combination rules

The combination rules to be evaluated include fixed rules, trained rules, and integrated confidence transformation and weighted combination with joint parameter optimization.

4.1. Fixed rules

Given K classifiers for M-class classification, the classifier outputs $d_{kj}(\mathbf{x}_k)$, are transformed to confidence measures z_{kj}, $k = 1, \ldots, K$, $j = 1, \ldots, M$. In combination, the sum-rule computes the combined class scores by

$$y_j(\mathbf{z}) = \sum_{k=1}^{K} z_{kj}, \qquad j = 1, \ldots, M. \tag{15}$$

This is equivalent to the averaging of confidence measures over the classifiers.

The product-rule gives the combined class scores by

$$y_j(\mathbf{z}) = \prod_{k=1}^{K} z_{kj}, \qquad j = 1, \ldots, M. \tag{16}$$

If the classifiers are mutually independent and the confidence measures represent the class posterior probabilities or likelihood, the product-rule gives Bayesian classification. However, since the posterior probability or likelihood can not be precisely estimated and the classifiers are not independent, some approximate fusion rules such as the sum-rule or median-rule often perform better [3,41].

The combination rules based on order statistics include the max-rule, min-rule, median-rule, trim and spread rules [12],

as well as the weighted combination of order statistics. Basically, the confidence measures of a class, z_{kj}, $k=1,\ldots,K$, are sorted in decreasing order

$$z_j^{(1)} \geqslant z_j^{(2)} \geqslant \cdots \geqslant z_j^{(K)}.$$

The max-rule, min-rule, and median-rule take from the ordered list the first item, the last item, and the middle one (or the average of two items in the middle), respectively, as the combined class measure. The spread rule averages the first item and the last one, while the trim rule averages the items of a segment in the ordered list. In my experiments of order statistics combination, the max-rule mostly outperforms other rules. I will show the results of max-rule and median-rule in this paper.

4.2. LDF

For combination using a meta-classifier, the $K \times M = p$ confidence measures (or classifier outputs) are viewed as the input features of the meta-classifier and are concatenated into a vector. On an input pattern X, denote the vector of confidence measure as $\mathbf{z} = [z_1, z_2, \ldots, z_p]^{\mathrm{T}}$, the combined class score is the class output of the meta-classifier. In the case of LDF, the combined class score is computed by

$$y_j(\mathbf{z}) = \sum_{i=1}^{p} w_{ji} z_i + w_{j0} = \mathbf{w_j}^{\mathrm{T}}\mathbf{z} + w_{j0}, \quad j = 1, \ldots, M. \tag{17}$$

I estimate the parameters (weights and biases) of LDF in four ways: nearest mean (N-Mean), LDA, regression, and linear SVM.

4.2.1. N-Mean and LDA

By N-Mean, each class has a template that is the mean vector of confidence measures over the class-specific samples in the validation data set. Denote the mean vectors by μ_j, $j=1,\ldots,M$, the input pattern is classified according to the minimum Euclidean distance to the class means. The square Euclidean distance is computed by

$$\delta(\mathbf{z}, \mu_j) = (\mathbf{z}-\mu_j)^{\mathrm{T}}(\mathbf{z}-\mu_j) = \mathbf{z}^{\mathrm{T}}\mathbf{z} - 2\mu_j^{\mathrm{T}}\mathbf{z} + \mu_j^{\mathrm{T}}\mu_j, \tag{18}$$

from which excluding the common term $\mathbf{z}^{\mathrm{T}}\mathbf{z}$, the discriminant function is linear.

LDA assumes that the feature space of each class undergoes a Gaussian density and the classes share a common covariance matrix. Further assuming equal prior probability, the discriminant function (the logarithm of Bayesian likelihood, excluding common terms) is a linear form [30]

$$y_j(\mathbf{z}) = \mu_j^{\mathrm{T}}\Sigma^{-1}\mathbf{z} - \frac{1}{2}\mu_j^{\mathrm{T}}\Sigma^{-1}\mu_j, \tag{19}$$

where Σ is the common covariance matrix.

4.2.2. Regression

In parameter estimation by regression, the weights and biases are adjusted by optimizing an objective function, typically, the CE, MSE, or MCE criterion. For the criteria CE and MSE, since each class has a target output, the output of meta-classifier need to be transformed by sigmoid function to approximate the class probability

$$y_j(\mathbf{z}) = \frac{1}{1+\exp[-(\mathbf{w_j}^{\mathrm{T}}\mathbf{z} + w_{j0})]}, \quad j = 1, \ldots, M. \tag{20}$$

On a validation data set $(\mathbf{z}^n, c^n)$, $n=1,\ldots,N$, the target output of true class c^n is $t_{c^n}=1$ and that of the other classes is $t_j = 0$, $j \neq c^n$. The criterion functions of CE and MSE are given by

$$\min\ J = -\sum_{n=1}^{N}\sum_{j=1}^{M}[t_j \log y_j + (1-t_j)\log(1-y_j)] + \lambda\sum_{j=1}^{M}\sum_{i=1}^{p} w_{ji}^2, \tag{21}$$

and

$$\min\ E = \sum_{n=1}^{N}\sum_{j=1}^{M}(t_j - y_j)^2 + \lambda\sum_{j=1}^{M}\sum_{i=1}^{p} w_{ji}^2, \tag{22}$$

respectively. The parameter values are constrained by weight decay to overcome the overfitting to validation data.

The MCE criterion was proposed by Juang et al. [42,43] and is supposed to be more relevant to the classification error than CE and MSE. Following Ref. [43], the misclassification measure of a pattern $\mathbf{z}$ from class c is defined by

$$h_c(\mathbf{z}) = -y_c(\mathbf{z}) + \log\left[\frac{1}{M-1}\sum_{i\neq c} \mathrm{e}^{\eta y_i(\mathbf{z})}\right]^{1/\eta}, \tag{23}$$

where η is a positive number. When $\eta \to \infty$, the misclassification measure becomes

$$h_c(\mathbf{z}) = -y_c(\mathbf{z}) + y_r(\mathbf{z}), \tag{24}$$

where $y_r(\mathbf{z})$ is the discriminant measure of the most competing class

$$y_r(\mathbf{z}) = \max_{i\neq c} y_i(\mathbf{z}).$$

The simplification of misclassification measure by setting $\eta \to \infty$ is helpful to speed up the learning process by stochastic gradient descent [40],[1] where only the parameters involved in the loss function are updated on a training

[1] Stochastic gradient descent is also referred to as generalized probabilistic descent (GPD) [44].

pattern. The loss of misclassification is computed by

$$l_c(\mathbf{z}) = l_c(h_c) = \frac{1}{1+\mathrm{e}^{-\xi h_c}}. \tag{25}$$

The parameter ξ was set to one in my experiments.

On the validation data set, the empirical loss is

$$L_0 = \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{M} l_i(\mathbf{z}^n) I(\mathbf{z}^n \in \omega_i), \tag{26}$$

where $I(\cdot)$ is an indicator function which takes value 1 when the condition in the parentheses is satisfied, otherwise takes value 0. Adding a weight decay to constrain the parameters, the MCE criterion is

$$\min L_1 = \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{M} l_i(\mathbf{z}^n) I(\mathbf{z}^n \in \omega_i) + \lambda \sum_{j=1}^{M}\sum_{i=1}^{p} w_{ji}^2. \tag{27}$$

In parameter estimation by optimizing the CE, MSE, or MCE criterion, the parameters are adjusted by stochastic gradient descent. The discriminant function for CE and MSE is the sigmoid form (20), whereas that for MCE can be either form (17) or (20) because the criterion does not depend on the target output. I use the linear form (17) for the simplicity of computation.

4.2.3. Linear SVM

I use a meta-classifier composed of M binary SVMs for combining the confidence measures of K classifiers. Each binary SVM aims to separate one class from the others. The principle of SVM is outlined as follows. On an input pattern $\mathbf{x}$, the discriminant function of binary SVM is computed by

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} y_i \alpha_i \cdot k(\mathbf{x}, \mathbf{x}_i) + b, \tag{28}$$

where ℓ is the number of learning patterns, y_i is the target value of learning pattern $\mathbf{x}_i$ ($+1$ for the first class and -1 for the second class), b is a bias, and $k(\mathbf{x}, \mathbf{x}_i)$ is a kernel function which implicitly defines an expanded feature space

$$k(\mathbf{x}, \mathbf{x}_i) = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}_i).$$

In the case of linear kernel,

$$k(\mathbf{x}, \mathbf{x}_i) = \mathbf{x} \cdot \mathbf{x}_i = \mathbf{x}_i^{\mathrm{T}}\mathbf{x},$$

the discriminant function is a linear combination of the input features

$$\begin{aligned} f(\mathbf{x}) &= \sum_{i=1}^{\ell} y_i \alpha_i \cdot \mathbf{x}_i^{\mathrm{T}}\mathbf{x} + b \\ &= \left(\sum_{i=1}^{\ell} y_i \alpha_i \cdot \mathbf{x}_i\right)^{\mathrm{T}} \mathbf{x} + b \\ &= \mathbf{w}^{\mathrm{T}}\mathbf{x} + b. \end{aligned}$$

The weight vector is estimated by solving a quadratic programming problem

$$\begin{aligned} &\text{minimize} \quad \tau(\mathbf{w}) = \tfrac{1}{2}\|\mathbf{w}\|^2 \\ &\text{subject to} \quad y_i \cdot f(\mathbf{x}_i) \geqslant 1 - \xi_i, \quad \xi_i \geqslant 0, \quad i = 1, \ldots, \ell, \end{aligned}$$

which is converted to the following dual problem [14]:

$$\begin{aligned} &\text{maximize} \quad W(\alpha) \\ &\qquad = \sum_{i=1}^{\ell} \alpha_i - \tfrac{1}{2}\sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \\ &\text{subject to} \quad 0 \leqslant \alpha_i \leqslant C, \quad i = 1, \ldots, \ell, \quad \text{and} \\ &\qquad \sum_{i=1}^{\ell} \alpha_i y_i = 0, \end{aligned} \tag{29}$$

where C is a parameter to control the tolerance of classification errors in learning. It was set to 0.1 in my experiments.

For classifier combination, the confidence measures are input to M linear SVMs, one for each class, and the input pattern is classified to the class of maximum SVM output.

4.3. Weighted combination

In weighted combination, each constituent classifier has exactly one weight for each class or for sharing for all classes. I use a shared weight for all classes. Accordingly, the combined class score is computed by

$$y_j(\mathbf{z}) = \sum_{k=1}^{K} w_k z_{kj}, \qquad j = 1, \ldots, M. \tag{30}$$

In weighted order statistics (WOS), a weight is given for each rank of confidence measure, unlike that the conventional weighted combination has a weight for each constituent classifier. The combined class score of WOS is computed by

$$y_j(\mathbf{z}) = \sum_{k=1}^{K} w_k z_j^{(k)}, \qquad j = 1, \ldots, M. \tag{31}$$

The classifier weights are estimated by regression on a validation data set to optimize the CE, MSE, or MCE criterion. For regression with CE or MSE criterion, it is necessary to transform the combined class score to sigmoid measure

$$y_j(\mathbf{z}) = \frac{1}{1+\exp[-\sum_{k=1}^{K} w_k z_{kj} - b]}, \quad j = 1, \ldots, M \tag{32}$$

for weighted combination, and

$$y_j(\mathbf{z}) = \frac{1}{1+\exp[-\sum_{k=1}^{K} w_k z_j^{(k)} - b]}, \quad j = 1, \ldots, M \tag{33}$$

for WOS. Note that an intercept b (common to all classes) is added to the sigmoid form such that the weighted sum is shifted to be around 0 without forcing the weights to

Table 1
Recognition rates (%) of individual classifiers and plurality vote

Expert	Feature	Classifier	Valid	Test-1	Test-2
E0	e-ncf-p	SLP	96.19	99.01	77.55
E1	ncf-p	RBF	97.13	99.53	86.02
E2	e-grd-g	PC	98.92	**99.77**	**89.93**
E3	e-grd	PC	98.38	98.25	82.90
E4	e-grd	LVQ	98.39	99.64	87.58
E5	des	DLQDF	98.61	98.65	82.43
E6	blr	N-Mean	88.83	92.81	60.44
	Plurality{E0–3} (CS1)		98.86	99.74	88.45
	Plurality{E1–5} (CS2)		99.21	**99.83**	**91.42**
	Plurality{E0–6} (CS3)		99.10	99.78	89.67

negative values. The intercept and the weights are adjusted in regression on a validation data set to optimize the CE (13) or MSE (22) criterion.

For regression with the MCE criterion (27), the transformation of combined class score to sigmoid measure and the intercept are not needed.

Whether to constrain the weights (non-negativity and sum to unity) or not has been widely discussed [20,22] and some experiments have shown that this does not influence the combination performance. I did not impose constraints in my experiments and actually, the resulting weights are rarely negative. The classifier weights also converge to small values because of the weight decay in the criterion.

4.4. Joint optimization of parameters

Both regression-based confidence evaluation and regression-based weighed combination provide good combination performance because either the confidence measure or the classifier weights are optimized with regard to a criterion function. Since the confidence parameters and classifier weights are optimized separately in two stages, they do not necessarily globally optimize the combination performance. The joint optimization of confidence parameters and classifier weights is supposed to achieve this goal. I experiment this strategy with the sigmoid confidence measure.

For classifiers $E_k, k=1,\ldots,K$, the output measurements are transformed to sigmoid confidence measures by

$$z_{kj}(\mathbf{d}_k)=s(a_k d_{kj}+b_k) = \frac{1}{1+\exp[-(a_k d_{kj}+b_k)]}, \quad j=1,\ldots,M, \quad (34)$$

where $s(\cdot)$ denote the sigmoid function. The parameters $\{a_k, b_k\}$ are shared for all classes, as in the case of confidence transformation by *LR-1*. For regularizing the values of parameters, the measurements d_{kj} are re-scaled by global normalization (9) prior to regression. The combined class measures are computed by

$$y_j(X)=\sum_{k=1}^{K} w_k s(a_k d_{kj}+b_k), \quad j=1,\ldots,M. \quad (35)$$

By joint optimization, the parameters $\{a_k, b_k, w_k\}$, $k=1,\ldots,K$, are estimated simultaneously to optimize the CE, MSE, or MCE criterion. For CE and MSE, the combined class measures need to be transformed to sigmoid.

5. Experimental results

To evaluate the performance of the classifier combination methods, I conducted experiments in handwritten digit recognition with variable classifier settings. Handwritten digit recognition is an important problem in document analysis applications and has long been intensively studied by the pattern recognition community. Many applications require very high accuracy of digit recognition, and combining multiple classifiers can break through the bottleneck of accuracy of individual classifiers.

5.1. Classifiers and data sets

I experimented with seven digit classifiers, including four neural classifiers and three dissimilarity-based classifiers. The specifications of individual classifiers are given in Table 1. The features used by the classifiers are blurred chaincode feature (blr), deslant chaincode feature (des), normalization-cooperated chaincode feature (ncf, feature extracted from original image with normalization skipped [45]), and gradient direction feature (grd). The prefix "e-" denotes 8-direction features, while other features are of 4-orientation; the suffix "-p" means that profile structure feature is added to enhance the feature representation, and "-g" denotes feature extraction from gray-scale normalized images. Various aspect ratio functions are used in coordinate mapping of image normalization [46,47].

The classifier structures are single-layer perceptron (SLP, trained by MSE minimization), radial basis function (RBF)

classifier [48], polynomial classifier [49], learning vector quantization (LVQ) classifier [50], discriminative learning quadratic discriminant function (DLQDF) [51], and the N-Mean classifier. The outputs of DLQDF classifier represent negative log-likelihood, and the outputs of LVQ and N-Mean classifiers are square Euclidean distances. For the neural classifiers, the output values before sigmoid squashing are used as the inputs of scaling functions for confidence transformation. Transforming the linear outputs using the sigmoid activation function gives the original sigmoid outputs of neural networks.

The seven classifiers were trained with two data sets, one collected by Hitachi, Ltd., and one extracted from the NIST special database 19 (NIST-SD19) [52]. The Hitachi data set contains 164,158 digit samples. The NIST data set contains 66,214 digit samples written by 600 writers [53]. Two test data sets were used to evaluate the classification performances. "Test-1" contains 9725 samples, collected in Japan (ATM transfer forms). "Test-2" contains 36,473 difficult samples that were mis-classified or rejected by an old recognizer of Hitachi. The images in Test-2 are either highly distorted or degraded. The recognition accuracies of the test sets by the individual classifiers are shown in Table 1. We can see that the highest accuracy on Test-2 is lower than 90%.

To estimate the parameters of confidence transformation and trained combination, I made a validation data set containing 30,000 samples collected in Japan ("Valid-1") and 10,000 samples from NIST-SD19 ("Valid-2"). In the validation set, each class has the same number of samples. The samples of Valid-1 were collected in similar environment to Test-1. The samples of Valid-2 were the leading samples of each class in a previously generated validation set of 200 writers [53].

To investigate the combination performance of variable classifier settings, I tested three classifier subsets, $CS1 = \{E0,E1,E2,E3\}$, $CS2 = \{E1,E2,E3,E4,E5\}$, and $CS3 = \{E0,E1,E2,E3,E4,E5,E6\}$. CS1 contains four neural classifiers, whose (sigmoid) outputs approximate the class posterior probabilities by MSE training. I hoped to see whether the re-estimation of confidences benefit the combination performance or not. CS2 contains the classifiers that give high accuracies (strong classifiers), while CS3 contains both high accuracy classifiers and low accuracy ones (weak classifiers, E0 and E6). As the baseline of combination performance, the accuracies of abstract-level combination by plurality vote are shown in Table 1 as well. We can see that when combining five strong classifiers (CS2), the accuracies of plurality vote are higher than those of the best individual classifier.

5.2. Results of fixed rules

On transforming the classifier outputs to three types of confidence measures (linear, sigmoid, and evidence) using three scaling functions (global normalization, Gaussian, LR-1), the transformed confidence measures of multiple classifiers are fused using either fixed rules or trained rules. In the classifier subset CS1, since the neural network outputs (weighted sum before sigmoid squashing) inherently range around 0 and the sigmoid outputs approximate the class posterior probabilities, I also transformed the raw outputs to confidence measures directly using the activation functions.

Tables 2, 3, and 4 show the classification accuracies on two test sets by combining four neural classifiers (CS1), five strong classifiers (CS2), and seven mixed classifiers (CS3), respectively, using fixed combination rules. For saving space, for order statistics rules, I only show the results of max-rule and median-rule. Other fixed order statistics rules (min-rule, trim and spread rules) did not outperform the max-rule in my experiments.

The product-rule is not applicable to log-likelihood (linear) measures because as the output of scaling function, the value can be either positive or negative. For the sigmoid measure and the evidence measure, whether to normalize the class measures (to unity of sum) or not does not change the classification result of product-rule. So, for the product-rule, we show the results of un-normalized measures only.

Some observations can be drawn from the combination accuracies of Tables 2–4. Comparing the combination rules, it is evident that the accuracy of product-rule is mostly lower than that of sum-rule, for any scaling function or confidence type. The inferior performance of product-rule is probably due to the dependence between the constituent classifiers. For the combination rules except the max-rule, the combination accuracy on normalized confidence measures is mostly lower than that on un-normalized measures. The max-rule performs on normalized measures as well as on un-normalized measures. The accuracy of max-rule is consistently higher than that of media-rule, but is mostly lower than that of sum-rule, especially when combining five strong (CS2) or seven mixed (CS3) classifiers.

Now I would focus on the accuracies of the sum-rule to compare the confidence transformation methods. First, in the combination of four neural classifiers (CS1), it is evident that the confidence measures with the Gaussian and LR-1 scaling functions outperform those with the raw outputs, especially on Test-2. This indicates that the re-estimation of confidences is beneficial for the combination of neural classifiers, whose (sigmoid) outputs inherently approximate the posterior probabilities though. The results on all the three classifier subsets show that the Gaussian scaling function and LR-1 outperform the global normalization, while the accuracies of Gaussian scaling and those of LR-1 are comparable. The global normalization, as a baseline, also performs fairly well.

Comparing the confidence types, we can see that with the same scaling function and combination rule, the accuracies of sigmoid measure is mostly higher than that of linear measure, while the accuracy of evidence measure is comparable to or higher than that of sigmoid measure, especially on Test-2.

Table 2
Combination results (%) of four neural classifiers (CS1) using fixed rules. The highest rate for each scaling function is highlighted

Test	Scaling	Confid.	Un-normalized				Normalized		
data	funct.	type	Sum	Product	Max	Median	Sum	Max	Median
		Linear	99.76		99.78	99.75			
	Raw	Sigm	99.77	99.73	99.78	99.75	99.75	99.73	99.75
		Evid	99.79	99.76	**99.80**	99.75	99.78	99.75	99.76
		Linear	99.77		99.78	99.72			
	Global	Sigm	99.75	99.73	99.78	99.72	99.74	99.65	99.74
		Evid	99.78	99.77	**99.81**	99.76	99.78	**99.81**	99.76
Test-1		Linear	99.78		99.79	99.72			
	Gauss	Sigm	99.80	99.74	99.79	99.76	99.80	99.80	99.76
		Evid	99.80	99.78	99.79	99.76	**99.81**	99.80	99.76
		Linear	99.78		99.79	99.75			
	LR-1	Sigm	99.78	99.73	99.79	99.74	99.77	99.79	99.76
		Evid	99.79	99.78	99.80	99.76	99.78	**99.81**	99.76
	Average		99.779	99.752	99.790	99.745	99.776	99.767	99.756
		Linear	90.10		90.68	89.82			
	Raw	Sigm	91.27	88.86	90.68	90.29	89.98	89.44	89.72
		Evid	**91.29**	90.10	90.95	90.21	90.23	90.60	89.92
		Linear	90.55		90.89	90.01			
	Global	Sigm	90.13	89.89	90.89	89.94	90.12	88.99	89.70
		Evid	91.24	90.55	91.07	89.96	**91.27**	91.08	89.94
Test-2		Linear	91.42		91.56	90.89			
	Gauss	Sigm	91.81	90.48	91.56	90.63	90.58	91.45	90.28
		Evid	91.82	91.42	91.67	90.52	90.48	**91.97**	90.26
		Linear	90.66		91.37	90.07			
	LR-1	Sigm	91.34	89.50	91.37	90.30	90.27	90.72	89.82
		Evid	91.50	90.66	**91.63**	90.29	90.37	91.60	90.06
	Average		91.094	90.182	91.193	90.244	90.412	90.731	89.962

Table 3
Combination results (%) of five strong classifiers (CS2) using fixed rules

Test	Scaling	Confid.	Un-normalized				Normalized		
data	funct.	type	Sum	Product	Max	Median	Sum	Max	Median
		Linear	99.78		99.80	99.80			
	Global	Sigm	99.76	99.74	99.80	99.80	99.76	99.71	99.73
		Evid	99.80	99.78	**99.84**	**99.84**	99.79	**99.84**	99.83
		Linear	99.80		99.80	99.78			
Test-1	Gauss	Sigm	**99.84**	99.75	99.80	99.78	99.83	99.79	99.80
		Evid	99.81	99.80	99.83	99.79	99.81	99.80	99.80
		Linear	99.76		99.81	99.74			
	LR-1	Sigm	99.83	99.73	99.81	99.74	99.80	99.77	99.83
		Evid	99.83	99.76	**99.84**	99.80	99.83	99.79	99.83
	Average		99.801	99.760	99.814	99.786	99.803	99.783	99.803
		Linear	92.04		92.27	91.59			
	Global	Sigm	91.65	91.27	92.27	91.59	91.55	89.90	91.02
		Evid	**92.52**	92.04	92.20	91.71	92.50	92.12	91.7
		Linear	92.04		92.09	91.54			
Test-2	Gauss	Sigm	**92.89**	91.22	92.09	91.54	92.04	91.45	91.68
		Evid	92.87	92.04	92.20	91.83	92.14	92.00	91.73
		Linear	92.05		92.35	91.64			
	LR-1	Sigm	92.87	90.76	92.35	91.64	92.14	91.24	91.57
		Evid	**92.97**	92.05	92.45	92.01	92.42	92.31	91.61
	Average		92.433	91.563	92.252	91.677	92.132	91.503	91.552

Table 4
Combination results (%) of seven mixed classifiers (CS3) using fixed rules

Test data	Scaling funct.	Confid. type	Un-normalized				Normalized		
			Sum	Product	Max	Median	Sum	Max	Median
		Linear	99.77		99.75	99.78			
	Global	Sigm	99.72	99.70	99.75	99.78	99.72	99.27	99.76
		Evid	99.78	99.77	99.72	**99.80**	99.78	99.72	**99.80**
		Linear	99.79		99.79	99.79			
Test-1	Gauss	Sigm	99.81	99.74	99.79	99.79	99.80	99.79	99.79
		Evid	**99.83**	99.79	99.79	99.78	**99.83**	99.80	99.80
		Linear	99.79		99.78	99.73			
	LR-1	Sigm	**99.81**	99.75	99.78	99.73	**99.81**	99.79	99.80
		Evid	**99.81**	99.79	99.80	99.78	**99.81**	99.78	99.78
	Average		99.790	99.757	99.772	99.773	99.792	99.692	99.788
		Linear	90.52		89.43	90.23			
	Global	Sigm	90.06	89.50	89.43	90.23	89.82	87.05	89.66
		Evid	**91.27**	90.52	90.81	89.90	91.25	90.76	89.84
		Linear	91.48		91.54	90.41			
Test-2	Gauss	Sigm	92.14	90.63	91.54	90.41	91.13	91.45	90.02
		Evid	**92.22**	91.48	91.73	90.35	91.18	91.98	89.92
		Linear	91.09		91.57	90.02			
	LR-1	Sigm	92.06	89.82	91.57	90.02	90.98	90.60	89.88
		Evid	**92.13**	91.09	91.84	90.26	91.06	91.70	89.87
	Average		91.441	90.507	91.051	90.203	90.903	90.590	89.865

Last, comparing the accuracies of confidence-based combination using the sum-rule with those of individual classifiers and plurality vote (in Table 1), we can see when combining five strong classifiers (CS2), the accuracies of sum-rule are significantly higher than that of abstract-level combination and the best individual classifier, especially on Test-2 (92.97% of sum-rule vs. 91.42% of plurality vs. 89.93% of best individual). The combination of CS3 (seven classifiers), however, gives lower accuracies than the combination of CS2, though CS2 is a subset of CS3. This is because CS3 contains weak classifiers and the combination is not weighted.

5.3. Results of linear discriminants

The combination accuracies using LDFs are shown in Tables 5, 6, and 7, for combining four neural classifiers (CS1), five strong classifiers (CS2), and seven classifiers (CS3), respectively. The parameters of LDFs were estimated by N-Mean, LDA, regression and linear SVM. For saving space, for regression-based LDFs, I only present the results of MSE criterion. In my experiments, the combination performance of MSE regression is superior to that of the CE criterion and comparable to that of the MCE criterion.

I again compare both the combination rules and the confidence transformation methods. First, it is evident that when normalizing the class confidence measures to unity of sum, the combination accuracies mostly deteriorate, especially on the Test-2 data set. Hence, I focus on the results of un-normalized measures from now on.

Comparing the combination rules, I focus on the highest accuracies given by each rule over the confidence types. In combining four neural classifiers (CS1), the highest accuracies on both Test-1 and Test-2 were given by the linear SVM. In combining five strong classifiers (CS2), the nearest-mean rule and the linear SVM give comparably high accuracies to both Test-1 and Test-2; The MSE regression performs fairly well on both test sets whereas the LDA performs inferiorly. In combining seven classifiers (CS3), the linear SVM gives the highest accuracies to both test sets; The nearest-mean rule and the MSE regression perform fairly well on both test sets, whereas the LDA performs well on Test-2 only.[2] In summary, in combination using LDF with various parameter estimation techniques, the linear SVM performs best.

Comparing the confidence transformation methods, I focus on the accuracies of linear SVM. First, in combining the neural classifiers (Table 5), we can see that the highest accuracies on both Test-1 and Test-2 are given by combining the evidence measures computed from the raw

[2] In Tables 2–10, the average accuracy is not a reliable indicator of performance because some combination rules (e.g., N-Mean and SVM) are susceptible to the linear confidence (log-likelihood) measure, while the others are not. So, in comparing the combination rules, we mainly consider the accuracies on sigmoid and evidence measures with Gaussian and LR-1 scaling.

Table 5
Combination results (%) of four neural classifiers (CS1) using linear discriminants

Test	Scaling	Confid.	Un-normalized				Normalized			
data	funct.	type	N-Mean	LDA	MSE	SVM	N-Mean	LDA	MSE	SVM
		Linear	99.62	99.72	99.76	99.74				
	Raw	Sigm	99.79	99.78	99.79	99.84	99.77	99.77	99.77	99.84
		Evid	99.80	99.77	99.81	**99.86**	99.79	99.78	99.77	99.85
		Linear	99.60	99.72	99.77	99.77				
	Global	Sigm	99.58	99.63	99.76	99.70	99.59	99.59	99.72	99.69
		Evid	99.70	99.78	**99.83**	99.81	99.72	99.77	**99.83**	99.81
Test-1		Linear	99.59	99.72	99.79	99.78				
	Gauss	Sigm	99.79	99.75	99.78	99.83	99.80	99.76	99.71	99.77
		Evid	99.80	99.75	99.77	**99.84**	99.81	99.76	99.69	99.77
		Linear	99.65	99.72	99.78	99.78				
	LR-1	Sigm	99.79	99.78	99.78	99.85	99.79	99.74	99.76	99.83
		Evid	99.80	99.79	99.77	**99.86**	99.80	99.78	99.77	99.81
	Average		99.709	99.743	99.782	99.805	99.759	99.744	99.752	99.796
		Linear	85.92	89.37	90.78	89.80				
	Raw	Sigm	91.48	92.46	91.93	92.50	90.20	91.92	91.41	91.90
		Evid	91.55	92.22	91.99	**92.68**	90.43	92.02	91.29	92.09
		Linear	88.57	89.37	91.2	90.51				
	Global	Sigm	86.50	88.04	90.12	89.09	86.93	87.69	89.63	88.76
		Evid	90.56	90.65	**91.77**	91.71	90.74	90.56	91.75	91.71
Test-2		Linear	90.01	89.37	91.29	91.12				
	Gauss	Sigm	91.92	91.37	91.36	**92.45**	90.82	91.36	90.69	91.37
		Evid	91.91	91.20	91.35	92.40	90.70	91.34	90.51	91.17
		Linear	88.71	89.36	91.11	90.62				
	LR-1	Sigm	91.53	92.30	91.83	92.52	90.51	91.62	91.21	91.71
		Evid	91.69	92.19	91.84	**92.67**	90.53	91.75	91.02	91.74
	Average		90.029	90.658	91.381	91.506	90.108	91.032	90.939	91.306

Table 6
Combination results (%) of five strong classifiers (CS2) using linear discriminants

Test	Scaling	Confid.	Un-normalized				Normalized			
data	funct.	type	N-Mean	LDA	MSE	SVM	N-Mean	LDA	MSE	SVM
		Linear	99.71	99.71	99.81	99.77				
	Global	Sigm	99.65	99.68	99.75	99.73	99.57	99.64	99.75	99.73
		Evid	99.63	99.76	**99.84**	99.79	99.65	99.75	**99.84**	99.78
		Linear	99.58	99.71	99.81	99.78				
Test-1	Gauss	Sigm	99.81	99.76	99.77	99.83	**99.84**	99.75	99.81	99.77
		Evid	99.83	99.75	99.78	**99.84**	99.83	99.75	99.83	99.78
		Linear	99.71	99.71	99.80	99.76				
	LR-1	Sigm	99.83	99.79	99.80	**99.84**	99.80	99.72	99.80	99.79
		Evid	**99.84**	99.78	99.83	**99.84**	**99.84**	99.76	99.80	99.80
	Average		99.732	99.739	99.799	99.798	99.755	99.728	99.805	99.775
		Linear	88.87	89.68	91.25	90.90				
	Global	Sigm	86.74	88.09	90.51	89.66	86.92	87.49	90.04	89.38
		Evid	91.02	89.77	92.01	**92.17**	91.19	89.65	91.98	92.15
		Linear	90.20	89.68	91.23	91.27				
Test-2	Gauss	Sigm	92.82	91.40	92.15	92.89	91.95	91.54	91.73	92.05
		Evid	92.88	91.17	92.11	**92.95**	92.21	92.11	91.91	92.24
		Linear	87.99	89.68	91.14	90.77				
	LR-1	Sigm	92.85	92.30	92.49	92.84	92.29	91.76	91.77	91.94
		Evid	**93.02**	92.09	92.36	92.94	92.50	92.21	91.59	92.16
	Average		90.71	90.429	91.694	91.821	91.177	90.793	91.503	91.653

Table 7
Combination results (%) of seven mixed classifiers (CS3) using linear discriminants

Test data	Scaling funct.	Confid. type	Un-normalized				Normalized			
			N-Mean	LDA	MSE	SVM	N-Mean	LDA	MSE	SVM
		Linear	99.63	99.70	99.81	99.79				
	Global	Sigm	99.52	99.62	99.77	99.73	99.52	99.62	99.79	99.72
		Evid	99.70	99.81	**99.84**	99.79	99.72	99.80	**99.84**	99.79
		Linear	99.62	99.70	99.83	99.78				
Test-1	Gauss	Sigm	99.80	99.76	99.80	**99.84**	99.83	99.76	99.81	99.76
		Evid	99.81	99.75	99.81	**99.84**	99.83	99.76	99.80	99.78
		Linear	99.69	99.70	99.80	99.75				
	LR-1	Sigm	99.84	99.79	99.85	**99.87**	99.83	99.74	99.80	99.76
		Evid	99.84	99.79	99.81	99.83	99.81	99.76	99.81	99.79
	Average		99.717	99.736	99.813	99.802	99.757	99.740	99.808	99.767
		Linear	86.33	88.95	91.23	90.15				
	Global	Sigm	83.21	87.05	90.06	88.34	83.61	86.63	89.66	88.06
		Evid	90.64	90.15	**91.98**	91.84	90.71	90.20	91.92	91.81
		Linear	90.04	88.95	91.17	91.01				
Test-2	Gauss	Sigm	92.36	91.56	91.96	92.73	91.42	91.67	91.57	91.94
		Evid	92.49	91.28	91.96	**92.76**	91.43	92.22	91.78	92.16
		Linear	86.88	88.95	90.98	90.20				
	LR-1	Sigm	92.18	92.37	92.20	92.70	91.33	91.90	91.79	91.80
		Evid	92.36	92.13	92.30	**92.82**	91.34	92.28	91.75	92.07
	Average		89.610	90.154	91.538	91.394	89.973	90.817	91.412	91.307

outputs. Nevertheless, the accuracies of confidence measures from re-scaled outputs by LR-1 are comparably high, and those of Gaussian scaling function are marginally lower. This indicates that for combining neural classifiers by meta-classification, confidence transformation does not deteriorate the combination performance.

Comparing the scaling functions and confidence types (focusing on the accuracies of linear SVM), it is evident in Tables 5–7 that, the Gaussian scaling function and LR-1 perform comparably well and both outperform the global normalization. Regarding the confidence type, the highest accuracies are mostly given by the evidence measure, especially on Test-2.

It is interesting to compare the accuracies of fixed combination and trained combination. Comparing the accuracies of linear SVM with those of sum-rule (focusing on the highest accuracies over the confidence transformation methods), we can see that in combining CS1 and CS3 that contain weak classifiers, the linear SVM gives significantly higher accuracies than the fixed sum-rule. While in combining five strong classifiers (CS2), the accuracies of linear SVM are comparable to those of sum-rule. This indicates that trained combination is beneficial especially for combining the classifiers with imbalanced performances. Similar observations have been reported by Fumera and Roli [16]. We can also conjecture that trained classification benefits the combination of dependent classifiers but this needs deeper investigation in the future.

5.4. Results of weighted combination

The test accuracies of weighted combination are shown in Tables 8, 9, and 10, for combining four neural classifiers (CS1), five strong classifiers (CS2), and seven classifiers (CS3), respectively. The classifier weights were estimated by regression with the CE, MSE, and MCE criterion. The rank weights for WOS were estimated by regression with the MCE criterion.

Since it remains true that the normalized confidence measures give lower combination accuracies than the un-normalized measures (especially on Test-2), I focus on the accuracies of un-normalized measures to compare the methods. Comparing the regression criteria for weight estimation, I pay attention to the highest accuracies of each criterion. In combining four neural classifiers (Table 8), the accuracies of MSE and MCE are comparably high. The accuracy of CE criterion on Test-2 is lower than those of MSE and MCE. The accuracies of WOS are apparently lower than those of weighted combination by regression with CE, MSE and MCE criteria. In combining five strong classifiers (Table 9), the accuracies of weighted combination by regression with CE, MSE, and MCE criteria are comparable to those of WOS. The results of combining seven classifiers (Table 10) show the same tendency as combining four neural classifiers, i.e., MSE and MCE are superior to CE, and WOS is inferior. In summary, the weighted combination with MSE or MCE regression perform well in

Table 8
Combination results (%) of four neural classifiers (CS1) using weighted rules

Test	Scaling	Confid.	Un-normalized				Normalized			
data	funct.	type	CE	MSE	MCE	WOS	CE	MSE	MCE	WOS
		Linear	99.79	99.78	99.79	99.79				
	Raw	Sigm	99.81	99.83	99.83	99.78	99.79	99.79	99.80	99.76
		Evid	99.84	**99.85**	99.84	99.81	99.81	99.81	99.79	99.79
		Linear	99.78	99.78	99.79	99.79				
	Global	Sigm	99.73	99.73	99.75	99.76	99.76	99.76	99.75	99.74
		Evid	99.80	**99.81**	**99.81**	99.78	99.80	**99.81**	**99.81**	99.78
Test-1		Linear	99.79	99.79	99.79	99.81				
	Gauss	Sigm	**99.83**	99.81	99.80	99.80	99.80	99.78	99.80	99.80
		Evid	99.81	99.81	99.80	99.81	99.80	99.80	99.81	99.80
		Linear	99.79	99.79	99.77	99.79				
	LR-1	Sigm	99.81	99.81	99.81	99.79	99.77	99.77	99.77	99.77
		Evid	99.84	**99.84**	99.83	99.80	99.80	99.81	99.82	99.79
	Average		99.802	99.802	99.801	99.792	99.791	99.791	99.794	99.779
		Linear	91.93	91.99	92.10	91.15				
	Raw	Sigm	92.44	92.53	92.51	91.43	92.05	92.15	91.97	90.22
		Evid	92.40	**92.63**	92.51	91.50	92.07	92.13	92.13	90.49
		Linear	91.99	92.03	91.90	91.42				
	Global	Sigm	91.30	91.40	91.50	90.86	90.57	90.65	90.98	90.17
		Evid	91.80	91.94	**92.20**	91.45	91.80	91.95	92.19	91.45
Test-2		Linear	91.78	91.78	92.21	92.02				
	Gauss	Sigm	91.76	92.31	**92.37**	91.90	91.36	91.53	91.64	90.69
		Evid	91.62	92.21	92.16	91.84	91.03	91.46	91.54	90.54
		Linear	91.91	91.97	91.65	91.55				
	LR-1	Sigm	92.39	92.50	92.48	91.59	91.83	91.96	91.80	90.70
		Evid	92.33	**92.59**	92.53	91.73	91.84	91.93	91.98	90.68
	Average		91.971	92.157	92.177	91.537	91.569	91.720	91.779	90.618

Table 9
Combination results (%) of five strong classifiers (CS2) using weighted rules

Test	Scaling	Confid.	Un-normalized				Normalized			
data	funct.	type	CE	MSE	MCE	WOS	CE	MSE	MCE	WOS
		Linear	99.78	99.80	99.76	99.80				
	Global	Sigm	99.77	99.75	99.77	**99.81**	99.76	99.77	99.76	99.76
		Evid	99.80	**99.81**	99.80	**99.81**	99.80	**99.81**	99.80	99.80
		Linear	99.78	99.79	99.80	99.80				
Test-1	Gauss	Sigm	99.81	99.80	99.83	99.84	99.81	99.81	99.79	99.81
		Evid	99.81	**99.85**	99.84	99.84	99.80	99.83	99.81	99.81
		Linear	99.78	99.79	99.75	99.80				
	LR-1	Sigm	99.81	99.83	99.80	99.81	99.77	99.78	99.78	99.81
		Evid	99.83	99.83	99.81	**99.84**	99.80	99.80	99.81	**99.84**
	Average		99.797	99.806	99.796	99.817	99.79	99.80	99.792	99.805
		Linear	91.87	91.93	91.86	**92.82**				
	Global	Sigm	91.42	91.48	91.72	92.30	91.51	91.50	91.56	91.70
		Evid	92.47	92.44	92.46	92.65	92.41	92.42	92.44	92.64
		Linear	91.68	91.72	92.01	92.53				
Test-2	Gauss	Sigm	92.76	92.84	92.88	92.92	92.39	92.3	92.09	91.93
		Evid	92.90	**93.04**	92.87	92.95	92.50	92.56	92.34	92.11
		Linear	91.78	91.87	91.58	92.59				
	LR-1	Sigm	92.56	92.67	92.70	92.85	92.07	92.12	91.99	92.18
		Evid	92.83	**92.94**	92.77	92.93	92.42	92.45	92.47	92.57
	Average		92.252	92.326	92.317	92.727	92.217	92.225	92.148	92.188

Table 10
Combination results (%) of seven mixed classifiers (CS3) using weighted rules

Test data	Scaling funct.	Confid. type	Un-normalized				Normalized			
			CE	MSE	MCE	WOS	CE	MSE	MCE	WOS
		Linear	99.78	99.77	99.73	**99.84**				
	Global	Sigm	99.77	99.76	99.77	99.81	99.75	99.75	99.75	99.73
		Evid	99.81	99.81	99.80	99.78	99.77	99.81	99.80	99.78
		Linear	99.78	99.77	99.79	99.80				
Test-1	Gauss	Sigm	99.83	99.79	99.80	99.79	99.81	99.84	99.81	99.83
		Evid	99.83	99.83	99.83	99.80	99.84	99.84	**99.85**	99.83
		Linear	99.78	99.77	99.76	99.80				
	LR-1	Sigm	99.83	99.83	99.81	99.84	99.80	99.80	99.80	**99.85**
		Evid	99.84	99.83	99.84	99.81	99.81	99.83	99.83	99.84
	Average		99.806	99.796	99.792	99.808	99.707	99.812	99.807	99.810
		Linear	92.03	92.09	91.97	91.93				
	Global	Sigm	91.43	91.53	91.87	91.36	90.64	90.82	91.21	89.93
		Evid	92.19	92.30	**92.50**	91.65	92.22	92.30	92.48	91.56
		Linear	91.90	91.88	92.22	92.11				
Test-2	Gauss	Sigm	92.41	92.76	92.90	92.18	92.09	92.26	92.10	91.48
		Evid	92.35	92.88	**92.93**	92.27	92.30	92.39	92.32	91.47
		Linear	91.96	92.03	91.50	92.12				
	LR-1	Sigm	92.59	92.71	92.76	92.15	92.09	92.20	92.10	91.43
		Evid	92.64	92.87	**92.89**	92.32	92.37	92.51	92.43	91.64
	Average		92.167	92.339	92.393	92.01	91.952	92.08	92.107	91.252

variable classifier settings while WOS performs well only for combining strong classifiers.

In comparing the confidence transformation methods, I focus on the accuracies of weighted combination with MSE and MCE regression. More specifically, since for each scaling function, the evidence measure mostly give the highest accuracies to both test sets, I focus on the accuracies of evidence measure to compare the scaling methods. From the results of combining four neural classifiers (CS1), we can see that the accuracies of LR-1 are comparable to those of raw outputs (without re-scaling), while the accuracies of Gaussian scaling are evidently lower. In the combination of five classifiers (CS2) and seven classifiers (CS3), the Gaussian scaling and LR-1 perform comparably well and both outperform the global normalization.

Comparing the accuracies of weighted combination by MSE or MCE regression to those of meta-classification with linear SVM (in Tables 5–7), we can see that the accuracies of weighted combination are comparable to those of linear SVM. This indicates that because of the restrictive distribution of confidence measures transformed from the classifier outputs, weighted combination with one unique weight for each classifier is sufficient to achieve high combination performance, while the meta-classification with LDF is redundant.

5.5. Results of joint parameter optimization

In the experiments of joint optimization, the confidence parameters and classifier weights were simultaneously estimated by optimizing the MSE and MCE criterion on the validation data set. Since the classifier outputs are transformed to sigmoid measures, I compare the performance of joint optimization with that of two-stage procedure with (sigmoid) confidence transformation (LR-1 scaling) followed by weighted combination (MSE and MCE regression). The combination accuracies on the validation set and two test sets are shown in Table 11. In the table, Joint MSE and joint MCE denote joint parameter optimization with MSE and MCE criteria, respectively, while LR-MSE and LR-MCE denote confidence transformation by LR-1 followed by weighted combination with MSE and MCE regression, respectively.

In the combination of three classifier sets, I compare the combination accuracies of joint MSE with those of LR-MSE and the accuracies of joint MCE with those of LR-MCE. Comparing the accuracies on the validation set, we can see that because a very small number of parameters (shared by different classes) are tunable, the joint optimization does not improve the combination accuracy of validation data as compared to the two-stage strategy. Comparing the combination accuracies on the test sets, the accuracy of joint

Table 11
Combination results (%) of joint optimization and cascaded combination

Classifiers	Method	Valid	Test-1	Test-2
	Joint MSE	99.35	99.81	92.32
E0–3	Joint MCE	99.36	99.81	92.35
(CS1)	LR-MSE	99.34	99.81	92.50
	LR-MCE	99.34	99.81	92.48
	Joint MSE	99.37	99.80	92.36
E1–5	Joint MCE	99.41	99.80	92.36
(CS2)	LR-MSE	99.36	99.83	92.67
	LR-MCE	99.42	99.80	92.70
	Joint MSE	99.38	99.79	92.40
E0–6	Joint MCE	99.42	99.83	92.53
(CS3)	LR-MSE	99.38	99.83	92.71
	LR-MCE	99.42	99.81	92.76

optimization is comparable to that of two-stage strategy on Test-1 and is evidently lower than that of two-stage strategy on Test-2. This indicates that the two-stage strategy of cascaded confidence transformation and classifier weight estimation is a right way of classifier combination while the joint parameter estimation does not benefit the combination performance. The inferior performance of joint parameter optimization is not attributed to overfitting but because the output of the first layer (Eq. (34)) no longer represent class probability well.

6. Conclusion

This paper investigated the effects of confidence transformation in measurement-level classifier combination with various combination rules, including fixed rules and trained rules. The experimental results in handwritten digit recognition by combining variable sets of classifiers justified the promise of confidence transformation to yield high combination performance. The inferior combination performance of joint parameter optimization indicates that the two-stage strategy of confidence transformation followed by weighted combination is a right way. Each confidence transformation method is the combination of a scaling function and a confidence type. Comparing the scaling functions, the logistic regression (LR) with one input variable (LR-1) performs well in all the classifier settings, while the Gaussian scaling method occasionally suffers from the inappropriate density assumption. Regarding the confidence types, the evidence measure shows promise as a multi-class probability estimate.

Among the fixed combination rules that I tested, the sum-rule performs best. The fixed sum-rule, however, is susceptible to weak classifiers in the classifier set. The trained combination rules, including linear discriminant function (LDF) and weighted combination, outperform the sum-rule when combining classifiers with imbalanced performances and perform as well as the sum-rule when combining strong classifiers. The parameters of LDF can be estimated in various ways, among which the linear SVM performs best. For estimating the classifier weights of weighted average, the MSE regression and MCE regression perform comparably well. The weighted average with optimized classifier weights performs as well as the meta-classification with linear SVM, while it has much fewer free parameters. This is because the classifier outputs, especially the transformed confidence measures, distribute in a highly restricted subspace.

Acknowledgements

The author thanks Kazuki Nakashima and Ryuji Mine for providing the test data sets.

References

[1] L. Xu, A. Krzyzak, C.Y. Suen, Methods of combining multiple classifiers and their applications to handwriting recognition, IEEE Trans. Systems Man Cybernet. 22 (3) (1992) 418–435.
[2] T.K. Ho, J. Hull, S.N. Srihari, Decision combination in multiple classifier systems, IEEE Trans. Pattern Anal. Mach. Intell. 16 (1) (1994) 66–75.
[3] J. Kittler, M. Hatef, R.P.W. Duin, J. Matas, On combining classifiers, IEEE Trans. Pattern Anal. Mach. Intell. 20 (3) (1998) 226–239.
[4] C.Y. Suen, L. Lam, Multiple classifier combination methodologies for different output levels, in: J. Kittler, F. Roli (Eds.), Multiple Classifier Systems, Lecture Notes in Computer Science, vol. 1857, Springer, Berlin, 2000, pp. 52–66.
[5] J. Kittler, A framework for classifier fusion: is it still needed? in: F.J. Ferri, J.M. Inesta, A. Amin, P. Pudil (Eds.), Advances in Pattern Recognition, Lecture Notes in Computer Science, vol.1876, Springer, Berlin, 2000, pp. 45–56.
[6] R.P.W. Duin, The combining classifiers: to train or not to train, in: Proceedings of the 16th International Conference on Pattern Recognition, vol. 2, Que., Canada, 2002, pp. 765–770.
[7] H. Hao, C.-L. Liu, H. Sako, Confidence evaluation for combining diverse classifiers, in: Proceedings of the Seventh International Conference on Document Analysis and Recognition, Edinburgh, Scotland, 2003, pp. 760–764.
[8] C.-L. Liu, H. Hao, H. Sako, Confidence transformation for combining classifiers, Pattern Anal. Appl. 7 (1) (2004) 2–17.
[9] R.P.W. Duin, D.M.J. Tax, Experiments with classifier combining rules, in: J. Kittler, F. Roli (Eds.), Multiple Classifier Systems, Lecture Notes in Computer Science, vol. 1857, Springer, Berlin, 2000, pp. 16–29.
[10] E. Mandler, J. Schürman, Combining the classification results of independent classifiers based on the Dempster-Shafer theory of evidence, in: E.S. Gelsema, L.N. Kanal (Eds.), Pattern Recognition and Artificial Intelligence, Elsevier Science, Amsterdam, 1988, pp. 381–393.

[11] G. Rogova, Combining the results of several neural network classifiers, Neural Networks 7 (5) (1994) 777–781.
[12] K. Tumer, J. Ghosh, Robust combining of disparate classifiers through order statistics, Pattern Anal. Appl. 5 (2) (2002) 189–200.
[13] D.-S. Lee, S.N. Srihari, A theory of classifier combination: the neural network approach, in: Proceedings of the Third International Conference Document Analysis and Recognition, Montreal, Canada, 1995, pp. 42–45.
[14] C.J.C. Burges, A tutorial on support vector machines for pattern recognition, Knowledge Discovery Data Mining 2 (2) (1998) 1–43.
[15] A. Al-Ani, M. Deriche, A new technique for combining multiple classifiers using the Demspter–Shafer theory of evidence, J. Artif. Intell. Res. 17 (2002) 333–361.
[16] G. Fumera, F. Roli, Performance analysis and comparison of linear combiners for classifier fusion, in: T. Caelli, et al. (Eds.), Proceedings of the Joint International Workshop on Structural, Syntactic, and Statistical Pattern Recognition, Lecture Notes in Computer Science, vol. 2396, Springer, Berlin, 2002, pp. 424–432.
[17] J.A. Benediktsson, P.H. Swain, Consensus theoretic classification methods, IEEE Trans. Systems Man Cybernet 22 (1992) 688–704.
[18] C.C. Chibelushi, F. Deravi, J.S.D. Mason, Adaptive classifier integration for robust pattern recognition, IEEE Trans. Systems Man Cybernet. Part B 29 (6) (1999) 902–907.
[19] A. Verikas, A. Lipnickas, K. Malmqvist, M. Bacauskiene, A. Gelzinis, Soft combination of neural classifiers: a comparative study, Pattern Recognition Lett. 20 (4) (1999) 429–444.
[20] R.A. Jacobs, Methods for combining experts' probability assessments, Neural Comput. 7 (4) (1995) 867–888.
[21] V. Tresp, M. Taniguchi, Combining estimators using non-constant weighting functions, in: G. Tesauro, D.S. Touretzky, T.K. Leen (Eds.), Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 1995.
[22] S. Hashem, Optimal linear combinations of neural networks, Neural Networks 10 (4) (1997) 599–614.
[23] A.K. Jain, S. Prabhakar, S. Chen, Combining multiple matches for a high security fingerprint verification system, Pattern Recognition Lett. 20 (11–13) (1999) 1371–1379.
[24] N. Ueda, Optimal linear combination of neural networks for improving classification performance, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2) (2000) 207–215.
[25] J.S. Bridle, Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition, in: F. Fogelman-Soulie, J. Herault (Eds.), Neurocomputing: Algorithms, Architectures and Applications, Springer, Berlin, 1990, pp. 227–236.
[26] X. Lin, X. Ding, M. Chen, R. Zhang, Y. Wu, Adaptive confidence transform based classifier combination for Chinese character recognition, Pattern Recognition Lett. 19 (10) (1998) 975–988.
[27] A.S. Artukorale, P.N. Suganthan, Combining classifiers based on confidence values, in: Proceedings of the Fifth International Conference on Document and Recognition, Bangalore, India, 1999, pp. 37–40.
[28] D.W. Ruck, S.K. Rogers, M. Kabrisky, M.E. Oxley, B.W. Suter, The multilayer perceptron as an approximation to a Bayes optimal discriminant function, IEEE Trans. Neural Networks 1 (4) (1990) 296–298.
[29] M.D. Richard, R.P. Lippmann, Neural network classifiers estimate Bayesian a posteriori probabilities, Neural Comput. 4 (1991) 461–483.
[30] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, second ed., Wiley-Interscience, New York, 2001.
[31] J. Schürmann, Pattern Classification: A Unted View of Statistical and Neural Approaches, Wiley-Interscience, New York, 1996.
[32] A. Hoekstra, S.A. Tholen, R.P.W. Duin, Estimating the reliability of neural network classification, in: Proceedings of the International Conference on Artificial Neural Networks, Bochum, Germany, 1996, pp. 53–58.
[33] R.P.W. Duin, D.M.J. Tax, Classifier conditional posterior probabilities, in: A. Amin, D. Dori, P. Pudil, H. Freeman (Eds.), Advances in Pattern Recognition, Lecture Notes in Computer Science, vol. 1451, Springer, Berlin, 1998, pp. 611–619.
[34] L. Gillick, Y. Ito, J. Young, A probabilistic approach to confidence estimation and evaluation, in: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, Munich, Germany, vol. 2, pp. 879–882.
[35] J. Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, in: A.J. Smola, P. Bartlett, D. Scholköpf, D. Schuurmanns (Eds.), Advances in Large Margin Classifiers, MIT Press, Cambridge, MA, 1999.
[36] L.I. Kuncheva, J.C. Bezdek, R.P.W. Duin, Decision templates for multiple classifier fusion: an experimental comparison, Pattern Recognition 34 (2) (2001) 299–314.
[37] W.-T. Chen, P. Gader, H. Shi, Lexicon-driven handwritten word recognition using optimal linear combinations of order statistics, IEEE Trans. Pattern Anal. Mach. Intell. 21 (1) (1999) 77–82.
[38] G. Shafer, A Mathematical Theory of Evidence, Princeton University Press, Princeton, 1976.
[39] J.A. Barnett, Computational methods for a mathematical theory of evidence, in: Proceedings of the Seventh International Joint Conference on Artificial Intelligence,Vancouver, Canada, 1981, pp. 868–875.
[40] H. Robbins, S. Monro, A stochastic approximation method, Ann. Math. Stat. 22 (1951) 400–407.
[41] D.M.J. Tax, M. van Breukelen, R.P.W. Duin, J. Kittler, Combining multiple classifiers by averaging or by multiplying?, Pattern Recognition 33 (9) (2000) 1745–1785.
[42] B.-H. Juang, S. Katagiri, Discriminative learning for minimum error classification, IEEE Trans. Signal Process. 40 (12) (1992) 3043–3054.
[43] B.-H. Juang, W. Chou, C.-H. Lee, Minimum classification error rate methods for speech recognition, IEEE Trans. Speech Audio Process. 5 (3) (1997) 257–265.
[44] S. Katagiri, B.-H. Juang, C.-H. Lee, Pattern recognition using a family of design algorithms based upon the generalized probabilistic descent method, Proc. IEEE 86 (11) (1998) 2345–2375.
[45] M. Hamanaka, K. Yamada, J. Tsukumo, Normalization-cooperated feature extraction method for handprinted Kanji character recognition, in: Proceedings of the Third International Workshop on Frontiers of Handwriting Recognition, Buffalo, NY, 1993, pp. 343–348.
[46] C.-L. Liu, M. Koga, H. Sako, H. Fujisawa, Aspect ratio adaptive normalization for handwritten character recognition, in: T. Tan, Y. Shi, W. Gao (Eds.), Advances in Multimodal

Interfaces—ICMI2000, Lecture Notes in Computer Science, vol. 1948, Springer, Berlin, 2000, pp. 418–425.
[47] C.-L. Liu, K. Nakashima, H. Sako, H. Fujisawa, Handwritten digit recognition: investigation of normalization and feature extraction techniques, Pattern Recognition 37 (2) (2004) 265–279.
[48] C.M. Bishop, Neural Networks for Pattern Recognition, Clarendon Press, Oxford, 1995.
[49] U. Kreßel, J. Schürmann, Pattern classification techniques based on function approximation, in: H. Bunke, P.S.P. Wang (Eds.), Handbook of Character Recognition and Document Image Analysis, World Scientific, Singapore, 1997, pp. 49–78.
[50] C.-L. Liu, M. Nakagawa, Evaluation of prototype learning algorithms for nearest neighbor classifier in application to handwritten character recognition, Pattern Recognition 34 (3) (2001) 601–615.
[51] C.-L. Liu, H. Sako, H. Fujisawa, Discriminative learning quadratic discriminant function for handwriting recognition, IEEE Trans. Neural Networks 15 (2) (2004) 430–444.
[52] P.J. Grother, NIST special database 19: handprinted forms and characters database, Technical Report and CDROM, 1995.
[53] C.-L. Liu, H. Sako, H. Fujisawa, Performance evaluation of pattern classifiers for handwritten character recognition, Int. J. Doc. Anal. Recognition 4 (3) (2002) 191–204.

About the Author—CHENG-LIN LIU received the B.S. degree in electronic engineering from Wuhan University, Wuhan, China, the M.E. degree in electronic engineering from Beijing Polytechnic University, Beijing, China, the Ph.D. degree in pattern recognition and artificial intelligence from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 1989, 1992 and 1995, respectively. From March 1996 to March 1999, he was a postdoctoral fellow in Korea Advanced Institute of Science and Technology (KAIST), Taejon, Korea, and later in Tokyo University of Agriculture and Technology, Tokyo, Japan. Afterwards, he became a research staff member at the Central Research Laboratory, Hitachi, Ltd., Tokyo, Japan, where he was promoted to a senior researcher in 2002. His research interests include pattern recognition, artificial intelligence, image processing, neural networks, machine learning, and especially the applications to character recognition and document processing.