Objective Function Design for MCE-based Combination of On-line and Off-line Character Recognizers for On-line Handwritten Japanese Text Recognition

Bilan Zhu, JinFeng Gao and Masaki Nakagawa
Department of Computer and Information Sciences,
Tokyo University of Agriculture and Technology,
Tokyo 184-8588, Japan
{zhubilan, nakagawa}@cc.tuat.ac.jp

***Abstract*—This paper describes effective object function design for combining on-line and off-line character recognizers for on-line handwritten Japanese text recognition. We combine on-line and off-line recognizers using a linear or nonlinear function with weighting parameters optimized by the MCE criterion. We apply a *k*-means method to cluster the parameters of all character categories into groups so that the categories belonging to the same group have the same weight parameters. Moreover, we apply a genetic algorithm to estimate super parameters such as the number of clusters, initial learning rate and maximum learning times as well as the sigmoid function parameter for MCE optimization. Experimental results on horizontal text lines extracted from the TUAT Kondate database demonstrate the superiority of our method.**

Keywords-Classifier combination; On-line recognition; string recognition; Character rcognition

I. INTRODUCTION

Handwritten character pattern recognition methods are generally divided into two types of approaches. One is on-line recognition and another is off-line recognition [1]. The on-line method regards each character pattern as a temporal feature sequence of pen movements. On the other hand, the off-line method regards it as a two dimensional image. The on-line method is very sensitive to stroke order variations while it is robust against character shape variations. On the other hand, the off-line method is robust against the stroke order variations, but it is very weak to character shape variations. Since off-line features are easily extracted from an on-line handwritten pattern by discarding temporal and structural information, we can apply the off-line method and thus complement the weakness of the on-line method. By combining the on-line method with the off-line method, the recognition accuracy is improved since they compensate their disadvantages reciprocally.

How to combine different classifiers is an important problem in multiple classifier approaches. In Japanese character recognition, Oda et al. improved recognition performance by combining on-line and off-line recognizers using probabilistic tables to normalize the combination scores [2]. The combination method by probabilistic tables is a generative method, and applying a discriminative method such as the minimum classification error (MCE) criterion and neural network to estimate and to optimize the combination may bring higher performance.

Liu investigated the effects of confidence transformation in combining multiple classifiers using various combination rules [3]. Kermorvant et al. constructed a neural network to combine the top rank candidates of three word recognizers [4]. The two works used the discriminative methods to estimate the combination parameters. However, when optimizing the parameters the previous works always only considered the character/word recognition performance, and did not consider the string recognition performance. In fact, real applications usually employ the string recognition rather than the character recognition. The character recognition is a part of the string recognition. Therefore, when we create a character recognizer, we have to consider the string recognition performance as done in [5][6]. The methods that only guarantee the character recognition accuracy do not necessarily bring well string recognition performance. They cannot even be applied for string recognition. In this paper, the word "string" denotes on-line handwritten text composed of a sequence of characters. It is also called digital ink.

On the other hand, we have to point out that introducing more parameters for a discriminative method dose not bring higher performance, since we have only a limited amount of samples for training. However, previous works tended to introduce too many parameters for a discriminative method. To introduce an effective set of parameters, we apply a *k*-means method to cluster the parameters of all character categories into groups, and for categories belonging to the same group we introduce the same weight parameters. We consider three types of functions with a different number of parameters, investigate how to construct the function and how to introduce effective parameters for discriminative methods under the condition of a limited amount of samples for classifier training.

In this paper, we apply a discriminative method MCE to optimize the parameters for combination of on-line and off-line recognizers with a linear or nonlinear function. We design the objective functions of parameter optimization so as to optimize the string performance. Moreover, we employ a genetic algorithm (GA) to estimate super parameters such as the number of clusters, initial learning rate and maximum learning times as well as the sigmoid function parameter for MCE optimization. Experimental results on horizontal text lines extracted from the TUAT Kondate database demonstrate the superiority of our method.

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.125

The rest of this paper is organized as follows: Section 2 presents an overview of our on-line handwritten text recognition system. Section 3 describes objective function design and introducing the combination parameters. Section 4 describes parameter optimization. Section 5 presents the experimental results, and Section 6 draws conclusion.

II. Recognition System Overview

We process each on-line handwritten string pattern as follows:

(1) Candidate lattice construction.

Strokes in a string are grouped into blocks (primitive segments) according to the features such as off-stroke (pen lift between two adjacent strokes) distance and overlap of bounding boxes of adjacent strokes. Each primitive segment is assumed to be a character or a part of a character. An off-stroke between adjacent blocks is called a candidate segmentation point, which can be a true segmentation point (SP) or a non-segmentation point (NSP). One or more consecutive primitive segments form a candidate character pattern. The combination of all candidate patterns is represented by a candidate lattice.

(2) Character pattern recognition.

For an input pattern, for accelerating on-line and off-line recognitions, we first select 40 top rank candidate classes according to the Euclidean distance to class means using a two layers Euclidean distance coarse classifier. The accumulated accuracy of top 40 candidate classes is mostly over 99.9%. After coarse classification, we apply an on-line recognizer and an off-line recognizer to recognize the input pattern, and obtain two sets of character candidate classes from on-line and off-line recognizers. Each candidate class of each set has a corresponding on-line or off-line recognition score. We combine the two sets of candidate classes considering their recognition scores to output a set of candidate classes to save them into the candidate lattice.

For the on-line recognizer, we extract feature points along the pen-tip trace from pen-down to pen-up. We employ the coordinates of feature points as unary features and the differences in coordinates between the neighboring feature points as binary features. Then we use a MRF model to match the feature points with the states of each character class of candidates and obtain a similarity for each character class. We then select the top character classes with the largest similarities as the output candidates of the fine classifier [7].

For the off-line recognizer, from an on-line character patterns (a sequence of stroke coordinates) we extract directional features: histograms of normalized stroke direction [8]. For coordinate normalization we apply pseudo 2D bi-moment normalization (P2DBMN) [9]. The local stroke direction is decomposed into 8 directions and from the feature map of each direction, 8x8 values are extracted by Gaussian blurring so that the dimensionality of feature vectors is 512. To improve the Gaussianity of feature distribution, each value of the 512 features is transformed by the Box-Cox transformation (also called variable transformation). The input feature vector is reduced from 512D to nD by Fisher linear discriminant analysis (FLDA) [10]. Then we use the nD feature vectors to create a modified quadratic discriminant function (MQDF) recognizer [11] as follows:

$$g_2(\mathbf{x},\omega_i) = \sum_{j=1}^{k} \frac{1}{\lambda_{ij}} [\varphi_{ij}^T(\mathbf{x}-\boldsymbol{\mu}_i)]^2 + \frac{1}{\delta}\|\mathbf{x}-\boldsymbol{\mu}_i\|^2 - \sum_{j=1}^{k} [\varphi_{ij}^T(\mathbf{x}-\boldsymbol{\mu}_i)]^2 + \sum_{j=1}^{k} \log \lambda_{ij} + (n-k)\log\delta \quad (1)$$

where μ_i is the mean vector of class ω_i, λ_{ij} (j = 1, …, k) are the largest eigenvalues of the covariance matrix and φ_{ij} are the corresponding eigenvectors, k denotes the number of principal axes and δ is a modified eigenvector which is set as a constant. The value of δ can be optimized on the training data set, however for a convenience we simply set it as $\gamma\lambda_{average}$ where $\lambda_{average}$ is the average of λ_{ij} (i,j = 1, …, n) for all features of all classes and γ is a constant that is larger than 0 and smaller than 1.

According to the previous works [8, 12], the best off-line recognition performance is obtained when n is about 160 and k is about 50 for the off-line MQDF recognizer. When combining on-line and off-line recognizers and then combining them with linguistic context and geometric features for the string recognition, however, we have found the best combination performance is obtained when n is about 90 and k is about 10 for the off-line MQDF recognizer. Therefore, we take n as 90 and k as 10, respectively.

(3) Search and recognition.

We apply the beam search strategy to search the candidate lattice. When searching the paths are evaluated according to the path evaluation criterion proposed in [13] that combines the scores of character recognition, linguistic context and geometric features (character pattern sizes, inner gaps, single-character positions, pair-character positions, candidate segmentation points) with the weighting parameters estimated by the genetic algorithm. This method selects an optimal path as recognition result.

Denote $\boldsymbol{X} = x_1 \ldots x_m$ as successive candidate character patterns of one path, and every candidate character pattern x_i is assigned candidate class C_i. Then $f(\boldsymbol{X},\boldsymbol{C})$ is the score of the path $(\boldsymbol{X},\boldsymbol{C})$ where $\boldsymbol{C} = C_1 \ldots C_m$. The path evaluation criterion is expressed as:

$$f(\mathbf{X},\mathbf{C}) = \sum_{i=1}^{m} \left\{ \begin{array}{l} \sum_{h=1}^{6} [\lambda_{h1} + \lambda_{h2}(k_i - 1)] \log P_h \\ \lambda_{71} \log P(g_{j_i} \mid SP) + \lambda_{72} \sum_{j=j_i+1}^{j_i+k_i-1} \log P(g_j \mid NSP) \end{array} \right\} + m\lambda. \quad (2)$$

where P_h, h=1,…,6, stand for the probabilities of $P(C_i|C_{i-2}C_{i-1})$, $P(b_i|C_i)$, $P(q_i|C_i)$, $P(p^u{}_i|C_i)$, $P(x_i|C_i)$ and $P(p^b{}_i|C_{i-1}C_i)$, respectively. b_i, q_i, $p^u{}_i$ and $p^b{}_i$ are the feature vectors for character pattern sizes, inner gaps, single-character positions and pair-character positions, respectively. g_i is the between-segment gap feature vector. $P(C_i|C_{i-2},C_{i-1})$ is the tri-gram probability. k_i is the number of primitive segments contained in the candidate character pattern x_i. λ_{h1}, λ_{h2} (h=1~7) and λ are the weighting parameters. $P(x_i|C_i)$ is estimated by the combination score of the on-line and off-line recognizers. We can also divide it into two parts $P(x^{on}{}_i|C_i)$, $P(x^{off}{}_i|C_i)$ where $x^{on}{}_i$ denotes the on-line features of x_i, $x^{off}{}_i$ denotes the off-line features of x_i, $P(x^{on}{}_i|C_i)$ is estimated by the score of the on-line recognizer and $P(x^{off}{}_i|C_i)$ is estimated by the score

of the off-line recognizer. The path evaluation criterion is changed as:

$$f^{1}(\mathbf{X},\mathbf{C})=\sum_{i=1}^{m}\left\{\begin{array}{l}\sum_{h=1}^{7}[\lambda_{h1}+\lambda_{h2}(k_i-1)]\log P_h \\ \lambda_{81}\log P(g_{j_i}\mid SP)+\lambda_{82}\sum_{j=j_i+1}^{j_i+k_i-1}\log P(g_j\mid NSP)\end{array}\right\}+m\lambda. \quad (3)$$

where P_h, h=1,…,7, stand for the probabilities of $P(C_i|C_{i-2}C_{i-1})$, $P(b_i|C_i)$, $P(q_i|C_i)$, $P(p^u{}_i|C_i)$, $P(x^{on}{}_i|C_i)$, $P(x^{off}{}_i|C_i)$ and $P(p^b{}_i|C_{i-1}C_i)$, respectively. λ_{h1}, λ_{h2} (h=1~8) and λ are the weighting parameters. By the path evaluation criterion we re-estimate the combination of the on-line and off-line recognizers.

III. Objective Function Design and Combination Parameters

A. Preliminary Investigation

We made a preliminary investigation before designing the objective functions. We can combine the on-line and off-line scores by a linear function as follows:

$$Score^{kj}{}_{comb}=w_1 Score^{kj}{}_{on}+w_2 Score^{kj}{}_{off} \quad (4)$$

where $Score^{kj}{}_{comb}$, $Score^{kj}{}_{on}$ and $Score^{kj}{}_{off}$ stand for the combination score, the on-line recognition score and the off-line recognition score between the character pattern x_k (k=1~P, P is the number of patterns) and the character class C_j (j=1~Q, Q is the number of the character classes) , respectively. These scores indicate the similarities between x_k and C_j. $\exp(\alpha Score^{kj}{}_{comb})$ is proportional to the joint probability $P(x_k,C_j)$ that is explained in detail in the next Section so that $\log P(x_k|C_j)$ is estimated by $\alpha Score^{kj}{}_{comb}+\log P(C_j)-\log\beta$ where β is a constant. w_1 and w_2 are the weighting parameters for combination, and they can be class-independent or class-dependent. When they are class-independent, all character classes share the same pair of w_1 and w_2. When they are class-dependent, each character class has a pair of w_1 and w_2, and the number of the pairs of parameters is equal to the number of the character classes.

We can apply the MCE criterion [14] optimized by stochastic gradient descent [15] to find the optimal parameter vector $\boldsymbol{w}=\{w_1, w_2\}$ by minimizing the following difference between the score of the most confusing character class and that of the correct one:

$$\begin{array}{l} L_{MCE}(\mathbf{w},x_k,C_j)=\sigma(\max(Score^{k\tilde{j}}{}_{comb})-Score^{kj}{}_{comb}) \\ \sigma(x)=(1+e^{-\alpha x})^{-1} \\ Score^{k\tilde{j}}{}_{comb}=\text{score of } x_k \text{ with incorrect character class } C_{\tilde{j}} \\ Score^{kj}{}_{comb}=\text{scores of } x_k \text{ with the correct character class } C_j \end{array} \quad (5)$$

where α is a super parameter for the sigmoid function.

We trained the on-line and off-line character recognizers, the weighting parameters for combination and geometric scoring functions using a Japanese online handwriting database Nakayosi [16]. For scoring linguistic context, we prepared a tri-gram table from the year 1993 volume of the ASAHI newspaper and the year 2002 volume of the NIKKEI newspaper. For training the weight parameters and evaluating the performance of character string recognition, we extracted horizontally written text lines from the database Kondate which were collected from 100 people. We used 75 persons' text lines for training the SVM classifier for the candidate segmentation point probability and the weighting parameters of path evaluation. The performance test for the character recognizers used an on-line Japanese handwriting database called Kuchibue [16] and that for the string recognizer used the text lines of the remaining 25 persons of Kondate. For the string recognizer, the candidate lattice retains 20 candidate classes for each character pattern. Table 1 and Table 2 show the details of the databases. The experiments were implemented on an Intel(R) Xeon(R) CPU W5590 @ 3.36 GHz 3.36 GHz (2 processers) with 12 GB memory.

TABLE I. Statistics of character pattern databases.

		Nakayosi_t	Kuchibue_d
#writers		163	120
#characters /each writer	Total	11,962	10,403
	Kanji/Kana/ Symbol/alpha numerals	5,643/5,068/ 1,085/166	5,799/3,723/ 816/65
#character categories /each writer	Total	3,356	4,438
	Kanji/Kana/ Symbol/alpha numerals	2976/169/ 146/62	4058/169 149/62
#average category characters	Total	3.6	2.3
	Kanji/Kana/ Symbol/alpha numerals	1.9/30.0/ 7.4/2.7	1.4/22.0 5.5/1.0

TABLE II. Statistics of training/test text lines of Kondate.

	#Text lines	#Character patterns	#Character classes	#Characters per line
Training	10, 174	104, 093	1, 106	10.23
Testing	3, 511	35, 686	790	16.89

Table 3 shows the results where *Rc_training* is the character recognition rate of training data after applying the on-line and off-line combined recognition, *Rc_testing* is the character recognition rate of testing data after applying the combined recognition, *Rs_e1* is the character recognition rate of testing data after applying the string recognition that uses the path evaluation criterion as shown in (2), and *Rs_e2* is the character recognition rate of testing data after applying the string recognition that uses the path evaluation criterion as shown in (3). The character recognition rate of testing data for the on-line recognizer is 88.66%, and that for the off-line recognizer is 89.21%.

TABLE III. Comparison of class-independent and class-dependent weighting parameters

Method / Performance	Class-independent	Class-dependent
Rc_training(%)	93.85	96.48
Rc_testing(%)	92.00	94.45
Rs_e1(%)	92.10	90.09
Rs_e2(%)	92.93	92.92

From the results, we can see that for the class-dependent weighting parameters the character recognition rates after applying the combination recognition are remarkably higher, but those after applying the string recognition are lower compared with the class-independent weighting parameters. The optimization method for the combination weighting parameters by MCE as shown in Eq. (5) is to optimize the performance of the combined recognizer. Therefore, taking personal weighting parameters for each character class causes the performance of the character classes with a larger number of the training character patterns to have priority, and it lets the combined recognizer to have high recognition rate. However, it causes low performance of the character classes with small number of the training character patterns

and even adding the considerations for the scores of linguistic context and geometric features at the string recognition step cannot save the failures for the character classes with the result of low recognition rate for the string recognition. We can also see that the path evaluation criterion in (3) that re-estimates the combination of on-line and off-line recognitions has brought better recognition accuracies.

B. Objective Function

According to the multinomial logistic regression model, the posterior probability of a character class C_j is given by:

$$P(C_j \mid x_k) = \frac{\exp(\alpha Score_{comb}^{kj})}{\sum_{\tilde{j}} \exp(\alpha Score_{comb}^{k\tilde{j}}) + \exp(\alpha Score_{comb}^{kj})}. \tag{6}$$

For the scores $Score_{comb}^{k\tilde{j}}$ other than the largest score $\max(Score_{comb}^{k\tilde{j}})$, $\exp(\alpha Score_{comb}^{k\tilde{j}})$ is very small compared to $\exp\left(\alpha \max(Score_{comb}^{k\tilde{j}})\right)$ and $\exp(\alpha Score_{comb}^{kj})$. Therefore, the posterior probability can be approximated as:

$$P(C_j \mid x_k) \approx \frac{\exp(\alpha Score_{comb}^{kj})}{\exp\left(\alpha \max(Score_{comb}^{k\tilde{j}})\right) + \exp(\alpha Score_{comb}^{kj})}. \tag{7}$$

Then, we can obtain the relation:

$$\begin{aligned} P(C_j \mid x_k) &\approx \frac{\exp\left(-\alpha \max(Score_{comb}^{k\tilde{j}}) + \alpha Score_{comb}^{kj}\right)}{1 + \exp\left(-\alpha \max(Score_{comb}^{k\tilde{j}}) + \alpha Score_{comb}^{kj}\right)} \\ &= 1 - \frac{1}{1 + \exp\left(-\alpha\left(\max(Score_{comb}^{k\tilde{j}}) - Score_{comb}^{kj}\right)\right)} \\ &= 1 - L_{MCE}(\mathbf{w}, x_k, C_j) \end{aligned} \tag{8}$$

Therefore, to minimize the MCE criterion is equal to maximize the posterior probability of the character class. The estimated $\exp(\alpha Score_{comb}^{kj})$ is proportional to the joint probability $P(x_k,C_j)$: $\exp(\alpha Score_{comb}^{kj}) = \beta P(x_k,C_j)$. However, to maximize the posterior probability of the character class can only realize high performance of the character recognition, it cannot obtain high performance of the string recognition. That is the reason why the class-dependent weighting parameters has brought higher the character recognition rates of the combined recognition, but has brought lower the character recognition rates of the string recognition. To realize high performance of the string recognition it is necessary to maximize the posterior probability of the string class $P(\boldsymbol{C}|\boldsymbol{X})$. The path evaluation criterion in (2) is to maximize $P(\boldsymbol{C}|\boldsymbol{X})$ where we need to select a set of candidate classes with $P(x_k|C_j)$ as larger as possible to retain them in the candidate lattice for each character pattern. The MCE criterion as shown in (5) is for maximizing the posterior probability $P(C_j|x_k)$. Therefore, we modify it as follows:

Using the Basian law:

$$P(x_k \mid C_j) = \frac{P(C_j \mid x_k) P(x_k)}{P(C_j)} \tag{9}$$

We assume $P(x_k)$ is a constant for all character patterns, then:

$$P(x_k \mid C_j) = \rho \frac{P(C_j \mid x_k)}{P(C_j)} \tag{10}$$

where ρ is a constant. From (10), maximizing $P(x_k|C_j)$ is equal to maximizing $P(C_j|x_k)/P(C_j)$. From (8) we obtain:

$$\frac{P(C_j \mid x_k)}{P(C_j)} \approx \frac{1}{P(C_j)} - \frac{L_{MCE}(\mathbf{w}, x_k, C_j)}{P(C_j)}. \tag{11}$$

Since $P(C_j)$ is not changed when optimizing the weight parameters $\boldsymbol{w}$, minimizing $L_{MCE}(\boldsymbol{w},x_k,C_j)/P(C_j)$ is equal to maximizing $P(C_j|x_k)/P(C_j)$. Therefore, we set a new MCE criterion as follows:

$$L_{MCE}^{1}(\mathbf{w}, x_k, C_j) = \frac{\sigma(\max(Score_{comb}^{k\tilde{j}}) - Score_{comb}^{kj})}{P(C_j)} \tag{12}$$

The new MCE criterion is for maximizing the conditional probability $P(x_k|C_j)$ so as to maximize $P(\boldsymbol{C}|\boldsymbol{X})$ and to optimize the string performance.

For the path evaluation criterion as shown in (3), it re-estimates the combination of on-line and off-line recognitions and does not use the combination scores $Score^{kj}_{comb}$ estimated by MCE, we only use $Score^{kj}_{comb}$ to select a set of candidate classes to retain them in the candidate lattice for each character pattern. Therefore, we consider another MCE criterion to select a set of candidate classes with candidate correct rate as higher as possible that will bring high performance of string recognition as follows:

$$\begin{aligned} &L_{MCE}^{2}(\mathbf{w}, x_k, C_j) = \\ &\begin{cases} \sigma(\min(Score_{comb}^{k\tilde{j}} - Score_{comb}^{kj}), \\ \qquad \text{if } C_{\tilde{j}} \text{ isnot included in the 20 top candidate classes} \\ 0, \text{otherwise} \end{cases} \\ &Score_{comb}^{k\tilde{j}} = \text{score of } x_k \text{ with incorrect character class } C_{\tilde{j}} \\ &\qquad\qquad \text{included in the 20 top candidate classes} \\ &Score_{comb}^{kj} = \text{scores of } x_k \text{ with the correct character class } C_j \end{aligned} \tag{13}$$

C. Combination parameters

From the result as shown in Table 3, we can see that introducing more parameters will brings lower performance, because we have only a limited amount of samples for training. We consider the estimated each pair of class-dependent parameters for each character class reflects the attribute of the character class. Therefore, we apply a *k*-means method to cluster the pairs of class-dependent parameters of all character classes into *G* groups, and for the classes belonging to the same group we employ the same parameters. We expect the problem of the limited samples for training can be solved by it.

We also investigate a nonlinear function to combine the on-line and off-line scores as follows:

$$Score_{comb}^{kj} = \log\left(\sum_{\mu=1}^{n_{mu}} c_\mu \, \sigma(w_{\mu 1} Score_{on}^{kj} + w_{\mu 2} Score_{off}^{kj} + b_\mu)\right) \tag{14}$$

$$\sigma(x) = (1 + e^{-\omega x})^{-1}$$

where n_{mu} is the number of middle layer units, $w_{\mu 1}$, $w_{\mu 2}$, b_μ, and c_μ are the weighting parameters for combination, and ω is a super parameter for the sigmoid function.

IV. Parameter Optimization

We apply a MCE criterion as shown in (5) or (12) or (13) to learn the weighting parameters. We use the stochastic gradient descent to find the optimal parameter vector. Before learning the on-line and off-line scores are normalized by their means and variances. The weighting parameters are initialized with random values, and then they are changed to the direction that will reduce the MCE criterion on all training patterns. For the learning rate η, we initialize it as a large value ε, and update it at each iteration t as follows:

$$\text{if}(\,J(t)-J(t-1)>=0\;\&\&\;\text{It occurs 3 times continuously}\,)\quad(14)$$
$$\eta=\eta-0.5\,\eta$$

If $t > T$ (the maximum learning times) the learning is stopped.

For the super parameters of the number of clusters G, the sigmoid function parameters α and ω, the initial learning rate ε and the maximum learning times T, we use a genetic algorithm to optimize them. We treat each one of $\{G, \alpha, \omega, \varepsilon, T\}$ as an element of a chromosome. Each chromosome has 5 elements. At the step of the fitness evaluation of GA, firstly we use the k-means method to cluster the class-dependent parameters into G groups, and for the classes belonging to the same group we introduce the same parameters, secondly, we apply α, ω, ε, T to learn the weighting parameters by MCE to obtain a smallest MCE criterion $L^{min}{}_{MCE}$, and set 1-$L^{min}{}_{MCE}$ as the fitness of each chromosome $\{G, \alpha, \omega, \varepsilon, T\}$.

V. EXPERIMENTS

We compared the performance of the three objective functions: L_{MCE} as shown in (5), $L^1{}_{MCE}$ as shown in (12), $L^2{}_{MCE}$ as shown in (13), and the three combination functions: linear function as shown in (4), nonlinear function with n_{mu}=2 and that with n_{mu}=3 as shown in (14). The training and testing data as well as the environment of the experiments (CPU and memory size) are the same as the experiments for the preliminary investigation described in Section III. Table 4 shows the results where the character recognition rates highlighted by underlines represent the candidate correct rate in the top 20 rank candidate classes, and the numbers enclosed in square brackets are the numbers of clusters G.

TABLE IV. COMPARISON OF OBJECTIVE FUNCTIONS AND COMBINATION FUNCTIONS.

Performance \ Method		L_{MCE}	$L^1{}_{MCE}$	$L^2{}_{MCE}$
linear	[*G*] *Rc_training*(%)	[1364] 96.60	[3] 93.76	[772] 98.91
	Rc_testing(%)	94.43	91.52	99.27
	Rs_e1(%)	89.60	**92.29**	90.86
	Rs_e2(%)	92.93	92.93	**92.99**
Nonlinear n_{mu}=2	[*G*] *Rc_training*(%)	[2816] 96.00	[5] 93.70	[1665] 98.91
	Rc_testing(%)	93.88	91.40	99.29
	Rs_e1(%)	90.13	92.00	91.78
	Rs_e2(%)	92.93	92.93	92.96
Nonlinear n_{mu}=3	[*G*] *Rc_training*(%)	[1535] 95.93	[7] 93.45	[1500] 98.91
	Rc_testing(%)	93.68	91.13	99.27
	Rs_e1(%)	90.73	92.19	91.55
	Rs_e2(%)	92.93	92.93	92.97

From the results, we can see that $L^1{}_{MCE}$ achieves the best recognition rate for the path evaluation criterion as shown in (2), the path evaluation criterion in (3) that re-estimates the combination of the on-line and off-line recognitions brings about better recognition accuracies. $L^2{}_{MCE}$ realizes the best recognition rate for the path evaluation criterion as shown in (3) because it maximizes the candidate correct rate in the top 20 rank candidate classes. The linear functions bring about better performances.

VI. CONCLUSION

This paper designed three objective functions and three functions for combining on-line and off-line character recognizers by the MCE criterion for on-line handwritten Japanese text recognition. We have shown the objective functions to optimize the string recognition performance bring better performance compared to that to optimize the character recognition performance. We have also shown introducing too many parameters for a discriminative method will bring lower performance because we have only a limited amount of samples for classifier training.

REFERENCES

[1] C.-L Liu, S. Jaeger, M. Nakagawa: On-line recognition of Chinese characters: The state of the art, IEEE Trans. Pattern Analysis and Machine Intelligence, vol.26, no.2, pp.198-213, 2004.

[2] H. Oda, B. Zhu, J. Tokuno, M. Onuma, A. Kitadai, M. Nakagawa: A compact on-line and off-line combined recognizer, Proc 10th International Workshop on Frontiers in Handwriting Recognition, pp. 133–138, La Baule, France, 2006.

[3] C.-L. Liu: Classifier combination based on confidence transformation, Pattern Recognition, vol. 38, no. 1, pp.11–28, 2005.

[4] C. Kermorvant, F. Menasri, A-L. Bianne, R. Al-Hajj, C. Mokbel, L. Likforman-Sulem: The A2iA-Telecom ParisTech-UOB system for the ICDAR 2009 handwriting recognition competition, Proc. 12th International Conference on Frontiers in Handwriting Recognition, pp.247-252, 2010.

[5] C.-L. Liu, H. Sako, H. Fujisawa: Effects of classifier structures and training regimes on integrated segmentation and recognition of handwritten numeral strings, IEEE Trans. Pattern Analysis and Machine Intelligence, vol.26, no. 11, pp. 1395-1407, 2004.

[6] Y. Tonouchi: Path evaluation and character classifier training on integrated segmentation and recognition of online handwritten Japanese character string, Proc. 12th International Conference on Frontiers in Handwriting Recognition, pp.513-517, 2010.

[7] B. Zhu, M. Nakagawa: A MRF model with parameters optimization by CRF for on-line recognition of handwritten Japanese characters, Proc. Document Recognition and Retrieval XVIII (DRR) that is part of IS&T/SPIE Electronic Imaging, San Jose, USA, 2011.

[8] C.-L. Liu, X.-D. Zhou: Online Japanese caracter recognition using trajectory-based normalization and direction feature extraction, Proc. 10th International Workshop on Frontiers in Handwriting Recognition, pp.217-222, 2006.

[9] C.-L. Liu, K. Marukawa: Pseudo two-dimensional shape normalization methods for handwritten Chinese character recognition, Pattern Recognition, 38(12), pp. 2242-2255, 2005.

[10] M. Cheriet, N. Kharma, C.-L. Liu, C. Y. Suen: Character Recognition Systems, A Guide for Students and Practioners, John Wiley & Sons, Inc., Hoboken, New Jersey, 2007

[11] F. Kimura: Modified quadratic discriminant function and the application to Chinese characters, IEEE Trans. Pattern Analysis and Machine Intelligence, 9 (1), pp.149-153, 1987.

[12] C.-L. Liu: High accuracy handwritten Chinese character recognition using quadratic classifiers with discriminative feature extraction, Proc. of the 18th International Conference on Pattern Recognition, vol. 2, Hong Kong, pp. 942–945, 2006.

[13] B. Zhu, X.-D. Zhou, C.-L. Liu, M. Nakagawa: A robust model for on-line handwritten Japanese text recognition, International Journal on Document Analysis and Recognition (IJDAR),Vol. 13, No. 2, pp.121-131, 2010.

[14] B.-H. Juang, S. Katagiri: Discriminative learning for minimum error classification, IEEE Trans. Signal Processing, 40(12), pp. 3043-3054, 1992.

[15] H. Robbins, S. Monro: A stochastic approximation method, Ann. Math. Stat. 22, pp. 400-407, 1951.

[16] M. Nakagawa, K. Matsumoto: Collection of on-line handwritten Japanese character pattern databases and their analysis, International Journal on Document Analysis and Recognition, 7(1), pp. 69-81, 2004.