IJDAR (2013) 16:17–31
DOI 10.1007/s10032-011-0178-0

ORIGINAL PAPER

A Bayesian-based method of unconstrained handwritten offline Chinese text line recognition

Nan-Xi Li · Lian-Wen Jin

Received: 12 December 2010 / Revised: 6 October 2011 / Accepted: 10 October 2011 / Published online: 8 November 2011

Abstract This paper presents a new Bayesian-based method of unconstrained handwritten offline Chinese text line recognition. In this method, a sample of a real character or non-character in realistic handwritten text lines is jointly recognized by a traditional isolated character recognizer and a character verifier, which requires just a moderate number of handwritten text lines for training. To improve its ability to distinguish between real characters and non-characters, the isolated character recognizer is negatively trained using a linear discriminant analysis (LDA)-based strategy, which employs the outputs of a traditional MQDF classifier and the LDA transform to re-compute the posterior probability of isolated character recognition. In tests with 383 text lines in HIT-MW database, the proposed method achieved the character-level recognition rates of 71.37% without any language model, and 80.15% with a bi-gram language model, respectively. These promising results have shown the effectiveness of the proposed method for unconstrained handwritten offline Chinese text line recognition.

Keywords Handwritten character recognition · Text line recognition · Verification · Negative training · Linear discriminant analysis

N.-X. Li
College of Educational Information Technology,
South China Normal University,
Guangzhou, 510631, People's Republic of China
e-mail: pumpkinLNX@gmail.com

L.-W. Jin (✉)
School of Electronic and Information Engineering,
South China University of Technology,
Guangzhou, 510641, People's Republic of China
e-mail: lianwen.jin@gmail.com

1 Introduction

Unconstrained handwritten offline Chinese text line recognition is currently one of the most challenging problems in handwritten character recognition [4,6,18,20,21,23–25]. Solutions to problems in this area have potential applications, such as automatic manuscript reading and document retrieval. The development of a method that allows for free handwriting styles that are not specific to individual writers of text lines is attractive, but resolving this problem involves recognizing a text line with high accuracy, comparable to that of human reading. One of the main challenges in achieving this goal is the proper evaluation of segmentation hypotheses containing non-characters. After pre-segmentation (as shown in Fig. 1b) of an original text line (Fig. 1a), all possible segmentation hypotheses of a text line can be represented in a segmentation candidate lattice (as shown in Fig. 1d), where each node corresponds to a segmentation position in pre-segmentation, and each edge corresponds to a sample of a real character or non-character. Any path from the first node to the last node in the segmentation candidate lattice is a possible segmentation hypothesis of the text line, which may contain lots of non-characters. Since non-characters are extremely difficult to differentiate from characters, a segmentation hypothesis containing non-characters might be evaluated as the optimal choice among all possible segmentation hypotheses (as shown in Fig. 1c), a problem that decreases the accuracy of text line recognition.

Probabilistic model based on the maximum a posteriori (MAP) criterion [2] is one of the frequently used methods for segmentation hypothesis evaluation [3,4,6,18,20,21,23–25]. Several probabilistic models utilized recognition scores and segmentation scores to reduce errors in text line evaluation [4,6,18]. And others employed both character recognizers and verifiers to discriminate between real characters and

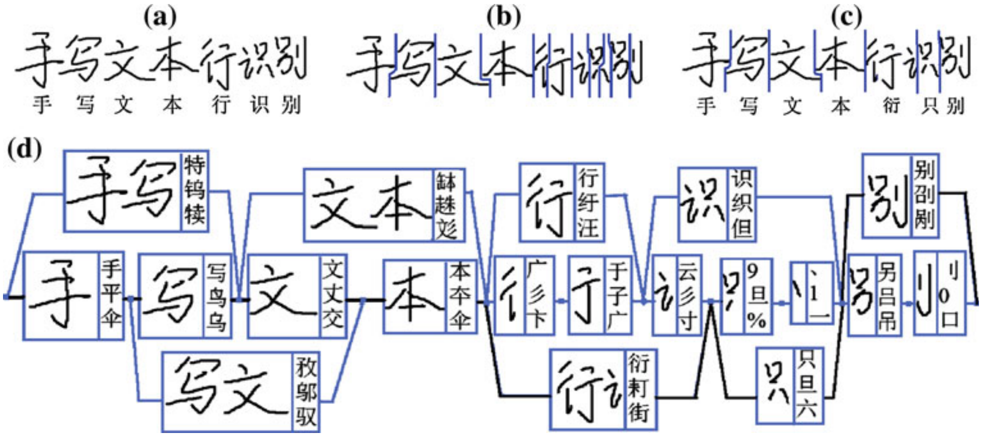

Fig. 1 Examples of mis-recognition of an input text line. **a** An original text line of “手写文本行识别”, **b** text line pre-segmentation, **c** the optimal segmentation hypothesis recognized as “手写文本衍只别, and **d** the segmentation candidate lattice corresponding to (b), where the optimal segmentation hypothesis is linked by the *dark line*

non-characters [3,23–25]. However, the trade-off between evaluation robustness and computational complexity is an important concern for these probabilistic models. Although first type of probabilistic models is relatively easy in computation, the assumptions for calculating the segmentation scores are too empirical to robustly evaluate realistic handwritten text lines. On the other hand, in the second type of probabilistic models, character verifiers can provide satisfactory robustness in discriminating characters from non-characters. However, lots of verifiers might be required in a large character set (such as Chinese), which is computationally prohibitive.

Another problem with most existing probabilistic models is that they use traditional isolated character recognizers for recognizing both characters and non-characters, and the latter are usually mis-recognized as the former. For instance, [20,21,23–25] employed class conditional probability densities and [4,6] utilized posterior probabilities, respectively, of traditional character recognition, which have not introduced non-character resistance into their models of segmentation hypothesis evaluation. This problem might be solved by the negative training of an isolated character recognizer. For distance classifiers used in Chinese character recognition, the following negative training strategies have been frequently used: the *k*-means clustering strategy [8] and the thresholding strategy [13]. However, in the *k*-means clustering strategy, the cluster number *k* of the class of non-characters is often empirically set, which is a trial-and-error process. In comparison, the thresholding strategy simply discards non-characters using a pre-defined threshold whereas failed to output a posterior probability for the class of non-characters as *k*-means clustering strategy does. Both strategies own obvious drawbacks for the negative training of distance classifiers.

In this paper, a novel probabilistic model is proposed to evaluate possible segmentation hypotheses of a text line. The probabilistic model can be implemented using just two classifiers, an isolated character recognizer and a character verifier, in a simple way that follows Bayesian rules. In addition, a linear discriminant analysis (LDA)-based negative training strategy is applied to an isolated character recognizer. Without any requirement of empirical parameters, this strategy is able to provide the posterior probability of non-characters in a distance classifier, which further improves the performance of the proposed probabilistic model. Experiments with the HIT-MW database [19] showed that the proposed method works well in unconstrained handwritten offline Chinese text line recognition and compares favorably with previous methods tested on the same data.

The rest of this paper is organized as follows: Sect. 2 presents an overview of previous works on unconstrained handwritten offline Chinese text line recognition. Section 3 introduces a Bayesian-based probabilistic model for segmentation hypothesis evaluation. Section 4 discusses the LDA-based negative training strategy for distance classifiers. Section 5 describes the experimental results on unconstrained handwritten offline Chinese text line recognition, and Sect. 6 draws conclusions from the experiments.

2 Previous works

In previous studies, methods used for unconstrained handwritten offline Chinese text line recognition fall into two categories: segmentation-based recognition [4,6,18,21,23–25] and segmentation-free recognition [20]. In segmentation-based recognition, a text line is first pre-segmented into characters or radicals. All possible segmentation hypotheses are then evaluated, among which the optimal hypotheses are regarded as the recognition candidates of the input text line. One of the main challenges in segmentation-based

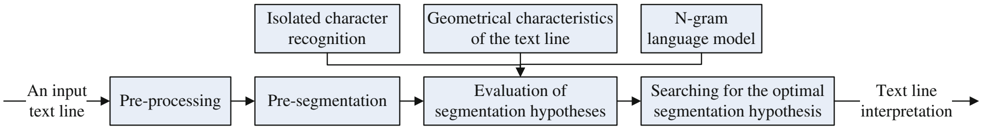
Isolated character recognition
Geometrical characteristics of the text line
N-gram language model
An input text line
Pre-processing
Pre-segmentation
Evaluation of segmentation hypotheses
Searching for the optimal segmentation hypothesis
Text line interpretation

Fig. 2 Framework of the segmentation-based methods of unconstrained handwritten offline Chinese *text line* recognition

recognition is that a large number of non-characters are generated by pre-segmentation, which severely interferes with the evaluation of possible segmentation hypotheses. Incorrect evaluation may result in serious errors in the final text line recognition, as shown in Fig. 1c.

In contrast, segmentation-free recognition requires no explicit segmentation of a text line. Instead, a whole text line is modeled using mathematical tools such as hidden Markov models (HMMs), and the recognition candidates of the text line are given by finding the optimal matches of the model. Because precise modeling requires a large number of training samples of realistic handwritten text lines, segmentation-free recognition tends to suffer from the problem of data sparseness [20], especially for a large character set such as Chinese characters, which may lead to a sharp deterioration in its recognition performance.

In previous studies of unconstrained handwritten offline Chinese text line recognition, segmentation-based recognition approaches have been found to perform much better than segmentation-free recognition methods [20,21]. In the two most recent studies using data from the HIT-MW database, character-level recognition rates were 44.22% for segmentation-free recognition [20] and 57.60% for segmentation-based recognition [21], without using a language model. A study using the segmentation-based recognition method reported that higher recognition rates of 77.18 and 78.44% were achieved using a bi-gram language model and a tri-gram language model, respectively [21]. The proposed method also adopts the general framework of segmentation-based recognition methods, as illustrated in Fig. 2. In the next sections, we will focus on the segmentation hypothesis evaluation module and introduce a Bayesian-based probabilistic model.

3 The proposed probabilistic model

In segmentation-based text line recognition methods, the optimal recognition candidate C^* of a handwritten text line can be evaluated as follows according to MAP [2] criterion:

$$C^* = \arg\max_C \{\log P(C|E)\}, \tag{1}$$

where E is a sequence of characters or radicals produced by text line pre-segmentation, C is a possible recognition candidate of the text line, and $P(C|E)$ is the posterior probability of C given E. For the probability $P(C|E)$ on the right hand side of (1), we introduce two hidden variables S and V, where S is the segmentation hypothesis that can be interpreted as C, and V is a binary sequence denoting the validity of each segment in S. As illustrated in Fig. 3, these sequences E, S, C, and V can be unfolded as follows:

$$\begin{cases} E = \{e_1, e_2, \ldots, e_M\} \\ S = \{s_1, s_2, \ldots, s_K\} \\ C = \{c_1, c_2, \ldots, c_K\} \\ V = \{v_1, v_2, \ldots, v_K\} \end{cases}, \tag{2}$$

where $e_i (i = 1, 2, \ldots, M)$ is a character or radical, $s_i (i = 1, 2, \ldots, K)$ is a text line segment combined by several neighboring $e_j (j = 1, 2, \ldots, M)$, $c_i (i = 1, 2, \ldots, K)$ is the character recognition candidate of s_i, and $v_i (i = 1, 2, \ldots, K)$ is a binary value denoting the validity of s_i ($v_i = 1$ means that s_i is a character or otherwise a non-character).

In interpreting a segmentation hypothesis of a text line, we may notice the presence of invalid segments (non-characters, e.g., s_4 and s_{26} in Fig. 3) which should not be recognized as any real character. However, since a traditional isolated character recognizer always recognize a non-character as a real character (e.g., c_4 and c_{26} in Fig. 3), it is improper to directly apply traditional isolated character recognition to recognize each segment $s_i (i = 1, 2, \ldots, K)$ in a segmentation hypothesis. To solve this problem, we employ a validity constraint described by a binary value $v_i \in \{0, 1\}$ for recognizing each segment s_i. Thus, for any possible segmentation hypothesis S' given E, and for any possible binary sequence V' given S', the probability $P(C|E)$ on the right hand side of (1) can be rewritten as follows:

$$\begin{aligned} P(C|E) &= \sum_{V'} \sum_{S'} P(C, S', V'|E) \approx P(C, S, V_0|E) \\ &= P(C|S, V_0, E) * P(V_0|S, E) * P(S|E), \end{aligned} \tag{3}$$

where S is the segmentation hypothesis coupled with the interpretation sequence C, and $V_0 = \{v_1 = 1, v_2 = 1, \ldots, v_K = 1\}$ is an all-one sequence denoting that each segment $s_i (i = 1, 2, \ldots, K)$ in the segmentation hypothesis S is a real character, so that the joint probability $P(C, S, V_0|E)$ is dominant over any other probability $P(C, S', V'|E) (\forall S' \neq S, V' \neq V_0)$.

The first probability $P(C|S, V_0, E)$ on the right hand side of (3) can be unfolded as

$$P(C|S, V_0, E) = \prod_{i=1}^{K} P\left(c_i | c_1, c_2, \ldots, c_{i-1}, s_1, s_2, \ldots, s_K,\right.$$

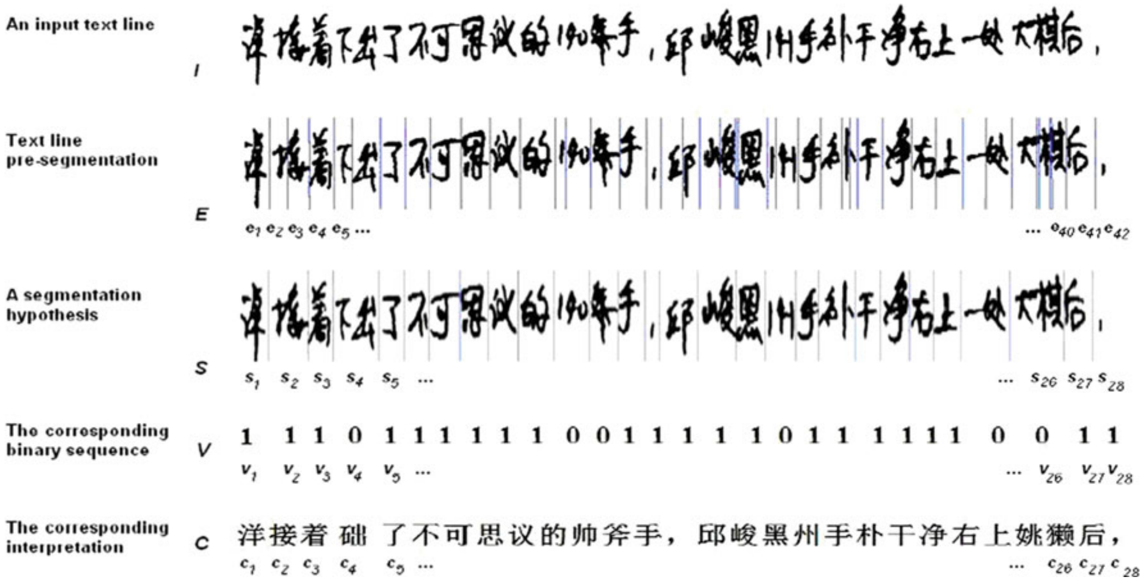
An input text line
I
Text line pre-segmentation
E
e_1 e_2 e_3 e_4 e_5 ...
... e_{40} e_{41} e_{42}
A segmentation hypothesis
S
s_1 s_2 s_3 s_4 s_5 ...
... s_{26} s_{27} s_{28}
The corresponding binary sequence
V
1 1 1 0 1 1 1 1 1 1 0 0 1 1 1 1 1 0 1 1 1 1 1 1 0 0 1 1
v_1 v_2 v_3 v_4 v_5 ...
... v_{26} v_{27} v_{28}
The corresponding interpretation
C
洋接着 础 了不可思议的帅斧手，邱峻黑州手朴干净右上姚獭后，
c_1 c_2 c_3 c_4 c_5 ...
... c_{26} c_{27} c_{28}

Fig. 3 Meaning of the variables I, E, S, V, and C in segmentation-based *text line* recognition

$$
\begin{aligned}
& v_1 = 1, v_2 = 1, \ldots, v_K = 1, e_1, e_2, \ldots, e_M) \\
&= \prod_{i=1}^{K} P(c_i|c_1, c_2, \ldots, c_{i-1}, s_i, v_i = 1) \\
&= \prod_{i=1}^{K} \{P(c_i|c_1, c_2, \ldots, c_{i-1}) * P(c_i|s_i, v_i = 1)/P(c_i)\} \\
&\approx \prod_{i=1}^{K} \{P(c_i|c_{i-n+1}, c_{i-n+2}, \ldots, c_{i-1}) * \\
&\quad (P(c_i|s\_\mathrm{norm}_i, v_i = 1))^{k_i}/P(c_i)^{k_i}\}, \qquad (4)
\end{aligned}
$$

where $k_i (i = 1, 2, \ldots, K)$ is the number of characters or radicals contained in the segment s_i, and s_norm_i is the normalized segment s_i used for isolated character recognition. The third equal mark of (4) comes from the assumption that the linguistic contexts $c_1, c_2, \ldots, c_{i-1}$ are conditionally independent of the procedure of isolated character recognition, which meets

$$
\begin{cases}
P(s_i|c_1, c_2, \ldots, c_i, v_i = 1) = P(s_i|c_i, v_i = 1) \\
P(s_i|c_1, c_2, \ldots, c_{i-1}, v_i = 1) = P(s_i|v_i = 1)
\end{cases} \qquad (5)
$$

And the fourth equal mark of (4) is based on the following approximation:

$$
\begin{cases}
P(c_i|c_1, c_2, \ldots, c_{i-1}) \approx P(c_i|c_{i-n+1}, .c_{i-n+2}, .., c_{i-1}) \\
P(c_i|s_i, v_i = 1) = P(c_i|e_{i_1}, e_{i_2}, \ldots, e_{i_{k_i}}, v_i = 1) \\
\quad \approx (P(c_i|s\_\mathrm{norm}_i, v_i = 1))^{k_i}/P(c_i)^{k_i - 1}
\end{cases}, \qquad (6)
$$

where $e_j (j = i_1, i_2, \ldots, i_{k_i})$ is any character or radical composing the segment s_i. The first line in (6) approximates the linguistic contexts of character c_i using its previous n characters. And the second line in (6) approximates the procedure of recognizing k_i neighboring characters or radicals $e_{i_i}, e_{i_2}, \ldots, e_{i_{k_i}}$ by isolated character recognition, where the power index k_i empirically compensates for the k_i parts of patterns in $e_{i_i}, e_{i_2}, \ldots, e_{i_{k_i}}$ [21,23].

The second probability $P(V_0|S, E)$ on the right hand side of (3) can be unfolded as

$$
\begin{aligned}
P(V_0|S, E) &= \prod_{i=1}^{K} P\,(v_i = 1|v_1 = 1, v_2 = 1, \ldots, v_{i-1} = 1, \\
&\quad s_1, s_2, \ldots, s_K, e_1, e_2, \ldots, e_M) \\
&\approx \prod_{i=1}^{K} P(v_i = 1|s_{i-1}, s_i, s_{i+1}). \qquad (7)
\end{aligned}
$$

The second line in (7) is based on the assumption that the validity of a segment $s_i (i = 1, 2, \ldots, K)$ largely depends on s_i itself and its two nearest neighbors s_{i-1}, s_{i+1}.

And the third probability $P(S|E)$ on the right hand side of (3) can be treated as a constant. As will be discussed in the subsequential subsections, the probability $P(c_i|s\_\mathrm{norm}_i, v_i = 1)$ on the last line of (4) and the probability $P(v_i = 1|s_{i-1}, s_i, s_{i+1})$ on the last line of (7) can be calculated by isolated character recognition and character verification, respectively.

3.1 Isolated character recognition

In Chinese character recognition, the classifier with the form of modified quadratic discriminant functions (MQDF) [11,15] has shown state-of-the-art recognition accuracy. Since a traditional MQDF classifier is trained and tested both on real characters, the normalized segment $s\_\mathrm{norm}_i (i = 1, 2, \ldots, K)$ fed into the classifier is implicitly treated as a real character. The posterior probability $P(c_i | s\_\mathrm{norm}_i, v_i = 1)$ of a traditional MQDF classifier can be obtained as follows:

$$\begin{aligned} P(c_i | s\_\mathrm{norm}_i, v_i = 1) &= \frac{p(s\_\mathrm{norm}_i | c_i, v_i = 1)^* P(c_i)}{\sum_{j=1}^{N} p(s\_\mathrm{norm}_i | c_j, v_j = 1)^* P(c_j)} \\ &= \frac{p(s\_\mathrm{norm}_i | c_i, v_i = 1)}{\sum_{j=1}^{N} p(s\_\mathrm{norm}_i | c_j, v_j = 1)}, \end{aligned} \tag{8}$$

where N is the number of character classes in the MQDF classifier, $P(c_j)(j = 1, 2, \ldots, N)$ is the prior probability assumed to follow a uniform distribution, and $p(s\_\mathrm{norm}_i | c_i, v_i = 1)$ is the conditional probability density of the normalized segment s_norm_i given class c_i. When $p(s\_\mathrm{norm}_i | c_i, v_i = 1)$ is under Gaussian distribution, it can be computed as

$$p(s\_\mathrm{norm}_i | c_i, v_i = 1) \propto \exp(-d(s\_\mathrm{norm}_i; c_i, v_i = 1)/\alpha), \tag{9}$$

where $d(s\_\mathrm{norm}_i; c_i, v_i = 1)$ is the output of class c_i given the normalized segment s_norm_i in the MQDF classifier, and α is a positive constant to tune $d(s\_\mathrm{norm}_i; c_i, v_i = 1)$ to a reasonable scale to avoid the zero value after taking negative exponential function. Empirically, we choose the maximal integer l that satisfies $\alpha = 2^l$ and the following constraints:

$$\begin{cases} \exp(-d(s\_\mathrm{norm}_i; c_i^{\mathrm{cand\_10}}, v_i = 1)/\alpha) \leq 10^{-10} \\ P(c_i^{\mathrm{cand\_1}} | s\_\mathrm{norm}_i, v_i = 1) \geq 0.5 \end{cases}, \tag{10}$$

where $c_i^{\mathrm{cand\_}j} (j = 1, 10)$ is the top jth recognition candidate of the normalized segment s_norm_i, $d(s\_\mathrm{norm}_i; c_i^{\mathrm{cand\_10}}, v_i = 1)$ is the output of the top 10th recognition candidate of s_norm_i in the classifier, and $P(c_i^{\mathrm{cand\_1}} | s\_\mathrm{norm}_i, v_i = 1)$ is the posterior probability of the topmost recognition candidate of s_norm_i calculated by (8). By observations on a training set of real characters, the integer l is set as 9 and the constant α is set as 512.

3.2 Character verification

The process of verifying each segment $s_i (i = 1, 2, \ldots, K)$ can be performed using another MQDF classifier, which includes the following five classes:

$$\begin{cases} \omega_0 : \text{Chinese character} \\ \omega_1 : \text{digit} \\ \omega_2 : \text{punctuation} \\ \omega_3 : \text{over-segmented character} \\ \omega_4 : \text{under-segmented character} \end{cases} \tag{11}$$

And the feature vector fed into this MQDF classifier is defined as

$$\mathbf{f}_i = \begin{cases} \Big[\frac{w_i}{\overline{w}}, \frac{h_i}{w_i}, d(s\_\mathrm{norm}_i; c_i^{\mathrm{cand\_1}}, v_i = 1), \\ \quad P(c_i^{\mathrm{cand\_1}} | s\_\mathrm{norm}_i,\ v_i = 1), \\ \min(1, \frac{d_{i,i+1}}{\overline{w}}), \max\left(1, \frac{d_{i,i+1}}{\overline{w}}\right)\Big]^T, \quad \text{if} \quad i = 1 \\ \Big[\frac{w_i}{\overline{w}}, \frac{h_i}{w_i}, d(s\_\mathrm{norm}_i; c_i^{\mathrm{cand\_1}}, v_i = 1), \\ \quad P(c_i^{\mathrm{cand\_1}} | s\_\mathrm{norm}_i,\ v_i = 1), \\ \min\left(\frac{d_{i-1,i}}{\overline{w}}, \frac{d_{i,i+1}}{\overline{w}}\right), \max\left(\frac{d_{i-1,i}}{\overline{w}}, \frac{d_{i,i+1}}{\overline{w}}\right)\Big]^T, \\ \quad \text{if} \quad 1 < i < K \\ \Big[\frac{w_i}{\overline{w}}, \frac{h_i}{w_i}, d(s\_\mathrm{norm}_i; c_i^{\mathrm{cand\_1}}, v_i = 1), \\ \quad P(c_i^{\mathrm{cand\_1}} | s\_\mathrm{norm}_i,\ v_i = 1), \\ \min\left(\frac{d_{i-1,i}}{\overline{w}}, 1\right), \max\left(\frac{d_{i-1,i}}{\overline{w}}, 1\right)\Big]^T, \quad \text{if} \quad i = K \end{cases}, \tag{12}$$

where $\overline{w}$ is the average width of all characters or radicals generated from text line pre-segmentation, w_i is the width of the ith segment s_i, h_i is the height of s_i, $d_{i-1,i}$ is the horizontal gravity distance between segments s_{i-1} and s_i, $d_{i,i+1}$ is the horizontal gravity distance between s_i and s_{i+1}, $d(s\_\mathrm{norm}_i; c_i^{\mathrm{cand\_1}}, v_i = 1)$ and $P(c_i^{\mathrm{cand\_1}} | s\_\mathrm{norm}_i, v_i = 1)$ are defined in (10). By observations on a training set of handwritten text lines, we notice that the probability density of each dimension of $\mathbf{f}_i$ almost renders a single peak in each class defined in (11), which is suitable for an MQDF classifier, as illustrated in Fig. 4.

The probability $P(v_i = 1 | s_i, s_{i-1}, s_{i+1})$ of character verification can then be transformed to the posterior probability of a five-class MQDF classifier according to the following:

$$\begin{aligned} & P(v_i = 1 | s_{i-1}, s_i, s_{i+1}) \\ & \quad = \sum_{j=0}^{4} P(v_i = 1, s_i \in \omega_j | s_{i-1}, s_i, s_{i+1}) \\ & \quad \approx P(v_i = 1, s_i \in \omega_{s_i} | s_{i-1}, s_i, s_{i+1}) \\ & \quad = \begin{cases} P(s_i \in \omega_{s_i} | s_{i-1}, s_i, s_{i+1}), & \text{if } \omega_{s_i} \in \{\omega_0, \omega_1, \omega_2\}, \\ 0, & \text{if } \omega_{s_i} \in \{\omega_3, \omega_4\} \end{cases} \\ & \quad \approx \begin{cases} (P(\omega_{s_i} | \mathbf{f}_i))^{k_i}, & \text{if } \omega_{s_i} \in \{\omega_0, \omega_1, \omega_2\} \\ 0, & \text{if } \omega_{s_i} \in \{\omega_3, \omega_4\} \end{cases} \end{aligned} \tag{13}$$

where ω_{s_i} is the class defined in (11) that segment s_i most probably belongs to, so that the joint probability $P(v_i = 1, s_i \in \omega_{s_i} | s_{i-1}, s_i, s_{i+1})$ is dominant over any other probability $P(v_i = 1, s_i \in \omega_j | s_{i-1}, s_i, s_{i+1}) (\forall \omega_j \neq \omega_{s_i})$, $\mathbf{f}_i$ is the feature vector defined in (12), k_i is the number of characters or radicals composing the segment s_i, and the posterior probability $P(\omega_{s_i} | \mathbf{f}_i)$ can be calculated similar to (8). The third equal mark in (13) is based on the assumption that a valid segment just comes from Chinese characters, digits, and punctuations. The fourth equal mark in (13) approximates the probability $P(s_i \in \omega_{s_i} | s_{i-1}, s_i, s_{i+1})$ using the posterior probability of a five-class MQDF classifier, where

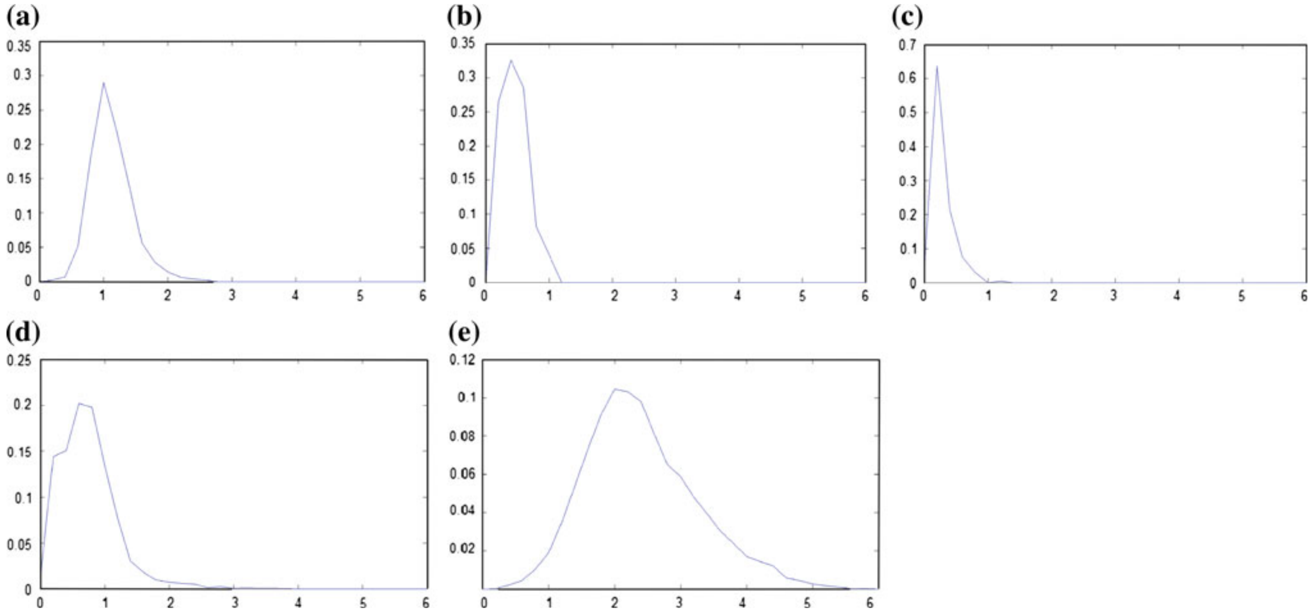

Fig. 4 Examples of the probability densities of the first dimension of the vector defined in (12) among **a** Chinese characters, **b** digits, **c** punctuations, **d** over-segmented characters, and **e** under-segmented characters

the power index k_i empirically compensates for the k_i parts of patterns in segment s_i (similar to the approximation in (4)).

3.3 The implementation of the probabilistic model

By taking logarithmic function on both sides of (3), and substituting (4), (7), and (13) into (3) as well as ignoring the constant term $P(S|E)$, the proposed probabilistic model for handwritten text line recognition can be rewritten as follows:

$$\begin{aligned}\log P(C|E) \approx & \sum_{i=1}^{K} \log P(c_i|c_{i-n+1}, c_{i-n+2}, \ldots, c_{i-1}) \\ & - \sum_{i=1}^{K} k_i * \log P(c_i), \\ & + \sum_{i=1}^{K} k_i * \log P(c_i|s\_\text{norm}_i, v_i = 1) \\ & + \sum_{i=1}^{K} k_i * \log P(\omega_{c_i}|\mathbf{f}_i) \qquad (14)\end{aligned}$$

where ω_{c_i} is the class defined in (11) that the character recognition candidate c_i belongs to, $\mathbf{f}_i$ is the feature vector defined in (12), k_i is the number of characters or radicals (generated by pre-segmentation) composing the segment s_i, the probability $P(c_i|s\_\text{norm}_i, v_i = 1)$ is the posterior probability of isolated character recognition calculated by (8), the probability $P(\omega_{c_i}|\mathbf{f}_i)$ is the posterior probability of a five-class MQDF classifier calculated similar to (8), the probabilities $P(c_i|c_{i-n+1}, c_{i-n+2}, \ldots, c_{i-1})$ and $P(c_i)$ can be given by an n-gram language model (note that in our experiments the prior probability $P(c_i)$ was empirically regarded to follow a uniform distribution, thus the second term in (14) became a constant $M * \log P(c_i)$ which can be omitted in path searching process, where M is the total number of segments produced from text line pre-segmentation).

The implementation of the above probabilistic model is illustrated in Fig. 5. Given a possible segmentation hypothesis $S = \{s_1, s_2, \ldots s_i, \ldots, s_K\}$, firstly, each segment $s_i (i = 1, 2, \ldots, K)$ is recognized by an MQDF classifier for isolated character recognition. Another MQDF classifier is then employed to verify the segment s_i. Both of the two classifiers jointly decide on the posterior probability of recognizing segment s_i no matter it is a real character or a non-character. When considering the n-gram language model, the segmentation hypothesis can be evaluated by simple summation or subtraction over the log-likelihood of each probability, as indicated in (14).

4 LDA-based negative training for isolated character recognition

In the proposed probabilistic model described in (14), the fourth dimension of feature vector $\mathbf{f}_i$ defined in (12) for character verification comes from the posterior probability of isolated character recognition. However, a traditional isolated character recognizer cannot correctly recognize non-characters, so that the above posterior probability may not be accurate at presence of non-characters in isolated character recognition. To solve this problem, we propose an LDA-based negative training strategy for isolated character

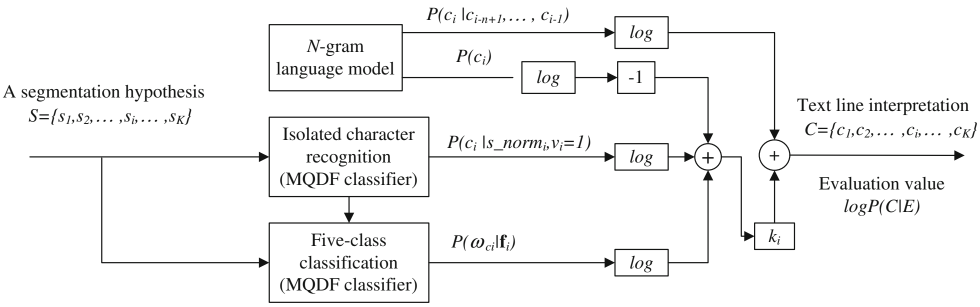
P(c_i |c_{i-n+1},… , c_{i-1})
N-gram language model
P(c_i)
log
log
-1
A segmentation hypothesis S={s_1,s_2,… ,s_i,… ,s_K}
Isolated character recognition (MQDF classifier)
P(c_i |s_norm_i,v_i=1)
log
+
+
Text line interpretation C={c_1,c_2,… ,c_i,… ,c_K}
Evaluation value logP(C|E)
Five-class classification (MQDF classifier)
P(ω_{ci}|f_i)
log
k_i

Fig. 5 Flowchart of the proposed probabilistic model to evaluate a possible segmentation hypothesis

recognition, which is discussed in the context of an MQDF classifier but is also applicable to other kinds of distance classifiers.

In the negative training of distance classifiers (such as an MQDF classifier), a class of non-characters can be added to the original N classes of real characters, modifying the posterior probabilities of the total $N+1$ classes. According to the following property of a joint probability:

$$P(AB) = P(B|A) * P(A) = P(A),\ \text{if}\ P(B|A) = 1, \quad (15)$$

the posterior probability of the MQDF classifier, which includes a class of non-characters, can be calculated as

$$P(\omega_i|s) = \begin{cases} P(\omega_i, \Omega_{\text{pos}}|s) = P(\omega_i|\Omega_{\text{pos}}, s) * P(\Omega_{\text{pos}}|s), \\ \quad \text{if} \quad i = 1, 2, \ldots N \\ P(\Omega_{\text{neg}}|s), \\ \quad \text{if} \quad i = N+1 \end{cases} \quad (16)$$

where s is the segment fed into the MQDF classifier, $\omega_i (i = 1, 2, \ldots, N+1)$ is the ith class in the MQDF classifier, $\Omega_{\text{pos}} = \bigcup_{i=1}^{N} \omega_i$ represents the class of real characters, $\Omega_{\text{neg}} = \omega_{N+1}$ represents the class of non-characters, $P(\omega_i|\Omega_{\text{pos}}, s)(i = 1, 2, \ldots, N)$ is the posterior probability in a traditional MQDF classifier calculated by (8), $P(\Omega_{\text{pos}}|s)$ is the posterior probability that segment s is a real character, and $P(\Omega_{\text{neg}}|s)$ is the posterior probability that segment s is a non-character. The last two probabilities, $P(\Omega_{\text{pos}}|s)$ and $P(\Omega_{\text{neg}}|s)$, can be calculated as follows:

$$\begin{cases} P(\Omega_{\text{pos}}|s) = \frac{p_s(s|\Omega_{\text{pos}})P(\Omega_{\text{pos}})}{p_s(s|\Omega_{\text{pos}})P(\Omega_{\text{pos}})+p_s(s|\Omega_{\text{neg}})P(\Omega_{\text{neg}})} \\ \quad = \frac{p_s(s|\Omega_{\text{pos}})}{p_s(s|\Omega_{\text{pos}})+p_s(s|\Omega_{\text{neg}})} \\ P(\Omega_{\text{neg}}|s) = \frac{p_s(s|\Omega_{\text{neg}})P(\Omega_{\text{neg}})}{p_s(s|\Omega_{\text{pos}})P(\Omega_{\text{pos}})+p_s(s|\Omega_{\text{neg}})P(\Omega_{\text{neg}})} \\ \quad = \frac{p_s(s|\Omega_{\text{neg}})}{p_s(s|\Omega_{\text{pos}})+p_s(s|\Omega_{\text{neg}})} \end{cases}, \quad (17)$$

where $P(\Omega_{\text{pos}})$ and $P(\Omega_{\text{neg}})$ are the prior probabilities assumed to follow a uniform distribution, $p_s(s|\Omega_{\text{pos}})$ and $p_s(s|\Omega_{\text{neg}})$ are the conditional probability densities of segment s in class Ω_{pos} and class Ω_{neg}, respectively. To calculate the above two conditional probability densities, we assume that the output of each class in a traditional MQDF classifier contains relevant information on revealing the validity of the input segment s, as indicated in previous methods of character recognition and verification [5,7,14]. Then, we have

$$\begin{cases} p_s(s|\Omega_{\text{pos}}) \approx p_{\mathbf{f}_N}(\mathbf{f}_N|\Omega_{\text{pos}}) = p_t(t|\Omega_{\text{pos}}) \\ \quad = \frac{1}{\sqrt{2\pi}\sigma_{\text{pos}}} \exp\left\{-\frac{(t-\mu_{\text{pos}})^2}{2\sigma_{\text{pos}}^2}\right\} \\ p_s(s|\Omega_{\text{neg}}) \approx p_{\mathbf{f}_N}(\mathbf{f}_N|\Omega_{\text{neg}}) \\ \quad = p_t(t|\Omega_{\text{neg}}) = \frac{1}{\sqrt{2\pi}\sigma_{\text{neg}}} \exp\left\{-\frac{(t-\mu_{\text{neg}})^2}{2\sigma_{\text{neg}}^2}\right\} \end{cases}, \quad (18)$$

where $\mathbf{f}_N = [d(c^{\text{cand\_1}}; s, \Omega_{\text{pos}}), d(c^{\text{cand\_2}}; s, \Omega_{\text{pos}}), \ldots, d(c^{\text{cand\_}N}; s, \Omega_{\text{pos}})]^T$ is a vector consisting of the output $d(c^{\text{cand\_}i}; s, \Omega_{\text{pos}})(i = 1, 2, \ldots, N)$ of the top ith character recognition candidate $c^{\text{cand\_}i}$ in a traditional MQDF classifier given segment s, and $t = W_{lda}^T \mathbf{f}_N$ is the scalar generated by LDA transform of vector $\mathbf{f}_N$, where W_{lda} is the LDA transform matrix defined as [2]

$$\begin{cases} W_{lda} = S_w^{-1}(\mu_1 - \mu_2) \\ S_w = \sum_{i=1}^{2}\sum_{j=1}^{M_i}(\mathbf{x}_i^j - \mu_i)(\mathbf{x}_i^j - \mu_i)^T \end{cases}, \quad (19)$$

where $\mu_i (i = 1, 2)$ is the mean feature vector of the ith class ($i = 0$ is class Ω_{pos} and $i = 1$ is class Ω_{neg} as defined in (16)), S_w is the within-class matrix, $M_i (i = 1, 2)$ is the number of samples in the ith class, $\mathbf{x}_i^j (i = 1, 2, j = 1, \ldots, M_i)$ is the feature vector of the jth sample in the ith class.

By observations on a set of training text lines, the probability density of scalar t in either class Ω_{pos} or Ω_{neg}, $p(t|\Omega_{\text{pos}})$ or $p(t|\Omega_{\text{neg}})$ defined in (18), approximately follows a Gaussian distribution, as illustrated in Fig. 6. Thus, the four parameters on the right hand side of (18), μ_{pos}, σ_{pos}, μ_{neg}, and σ_{neg}, can be determined using maximum likelihood estimation [2] after manually labeling whether the segment s is a real character or non-character in training phase.

In summary, the process of the LDA-based negative training for an MQDF classifier can be illustrated in Fig. 7. Firstly, a traditional MQDF classifier is employed to recognize the segment s, whose outputs form an N-dimensional vector $\mathbf{f}_N$ define in (18). LDA is then applied to transforming the vector $\mathbf{f}_N$ to a scalar t defined in (18). After estimating the means and variances of the scalar t in two classes, Ω_{pos} and Ω_{neg}, respectively, the probabilities $P(\Omega_{\text{pos}}|s)$ and

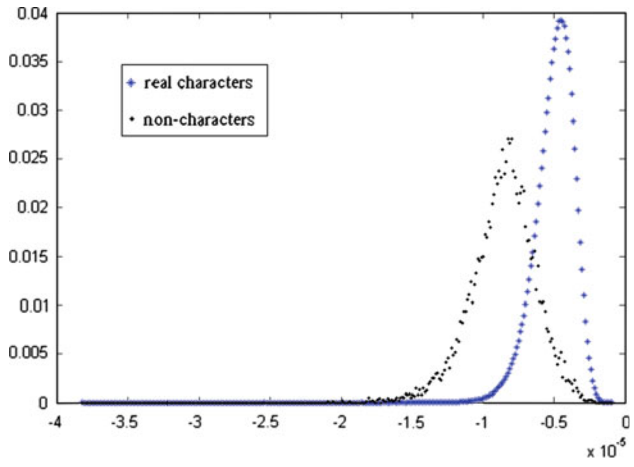

Fig. 6 The probability densities of scalar t defined in (19) in class of real characters or non-characters

$P(\Omega_{\text{neg}}|s)$ can be determined by (17) and (18). Finally, the posterior probability $P(\omega_i|s)(i = 1, 2, \ldots, N + 1)$ in the MQDF classifier, which includes a class of non-characters, can be output directly for negative samples or indirectly for positive samples using simple multiplication, as indicated in (16). It is worthwhile to mention that we processed this posterior probability as follows before applying it to the third term of (14): we considered only the classes of real characters in top $L(L = 1, 2, \ldots)$ recognition candidates by excluding any class of non-characters and putting forward its subsequent classes of real characters. This did not conflict with our understanding of isolated character recognition, because the much lower posterior probabilities of the classes subsequent to a non-character class also indicated the more reliable interpretation of a sample as a non-character. Then, the probability $P(\omega_i|s)(i = 1, 2, \ldots, N + 1)$ calculated by (16) became $P(\omega_i|s, \Omega_{\text{pos}})(i = 1, 2, \ldots, N)$ since the candidate classes were restricted to real characters as if the segment s was of a real character.

5 Experimental results

5.1 Experimental setup

The experiments consisted of two parts: one was for the LDA-based negative training strategy and the other was for the proposed probabilistic model of handwritten text line recognition. For the experiments on isolated character recognition, the following datasets were employed as training data: 105 sets of Chinese characters from the HCL2000 database [22], 97 sets of Chinese characters and 43 sets of digits from the SCUT-IRAC database [9], 295 sets of characters including Chinese characters, digits, and punctuation marks from the CASIA-HWDB1.1 database [16] (Chinese characters included the 3,755 classes in the first-level set of the GB2312-80, digits included the 10 classes from integer 0–9, and punctuation marks included 14 frequently used classes such as ! % ( ) : ; ? , ` ○ “” ≪≫). The miscellaneous training samples from different databases may improve the generality of the MQDF classifier. The testing data for isolated character recognition consisted of the characters labeled in the 383 text lines in the test set of HIT-MW database [19], including 7,401 samples of Chinese characters (in total 1,319 classes), 226 samples of digits (in total 10 classes), and 806 samples of punctuation marks (in total 24 classes). For handwritten text line recognition, we adopted the randomly selected 100 text lines in the train set of HIT-MW database as training data, and the whole 383 text lines in the test set of HIT-MW database as testing data, respectively. And the experimental platform was a PC with a dual-core 2.66 GHz CPU and 4 G of memory. All the experiments were implemented using VC++2008 except for the module of pre-segmentation which was firstly written using Matlab2008 and then was invoked in a VC++2008 project.

For pre-segmentation of a text line, we adopted the algorithm proposed in [12] to generate curved segmentation paths, which sequentially deals with naturally separated characters, overlapped characters, and touched characters in unconstrained handwritten offline Chinese text lines.

For an n-gram language model, we adopted the same bi-gram language model used in [21], which is trained on a Chinese corpus from the Chinese Linguistic Data Consortium (CLDC) and modified with Katz smoothing and entropy-based pruning. The top ten recognition candidates in isolated character recognition were preserved when using the bi-gram language model.

For searching for the optimal recognition candidate of a text line, we employed a beam search algorithm [13,25], which preserved the top ten partial paths at each step of path searching. As the proposed probabilistic model in (14) needed to consider both the previous neighbor s_{i-1} and the next neighbors s_{i+1} of the current segment s_i, we assumed that the search direction from segment index $i = 1$ to $i = K$ was forward direction, and that the last segment on the previously selected partial path adjacent to s_i was its previous neighbor s_{i-1}, and that the character or radical generated by pre-segmentation next to s_i was its next neighbor s_{i+1}. As the feature vector for character verification defined in (12) simply measured the horizontal gravity distance between s_i and its neighboring segment, the above assumption worked satisfactorily in path searching given the proposed probabilistic model. Furthermore, to accelerate the searching process, we adopted a two-stage searching strategy: firstly, without any language model, the above beam search algorithm was performed to seek for the top ten paths of a text line given (14) preserving only the last two terms, where for isolated character recognition, only the topmost candidate character was preserved to limit the searching space. Then at the second

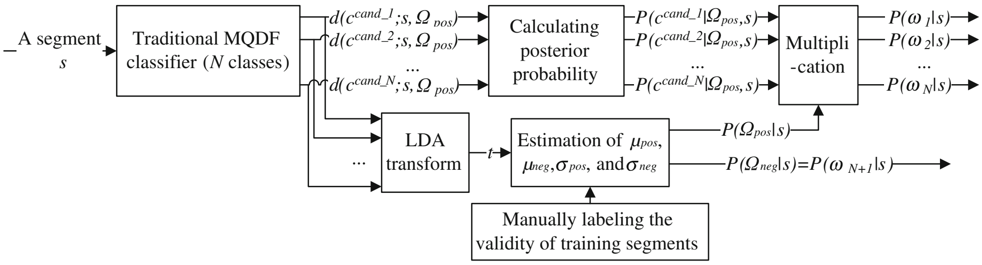

Fig. 7 Flowchart of the LDA-based negative training for an MQDF classifier

stage of searching, each of these ten paths was treated as an unrecognized segmentation configuration of the text line which needed to be re-evaluated given (14). With a bi-gram language model, the best recognition candidate of a segmentation configuration was found using Viterbi algorithm [17], where for isolated character recognition, the top ten candidate characters were preserved. Finally, the searching results of these ten paths were re-ranked, in order to get the optimal segmentation hypothesis and the corresponding interpretation of the input text line.

For training an isolated character recognizer, maximum likelihood estimation [2] was employed to determine the parameters of the classifier. Each input segment was firstly resized to $64*64$ using linear normalization and then divided to $8*8$ meshes by local elastic meshing [10]. At the center of each mesh, 8-directional gradient features [1] were extracted. An LDA transform was then applied to reducing the original 512-dimensional feature vector to a 160-dimensional feature vector, and the latter was fed into the recognizer.

For training the character verifier defined in (13), the samples of each class were collected from realistic handwritten text lines as follows: given a segmentation candidate lattice (see Fig. 8b) constructed form text line pre-segmentation, and the ground-truth segmentation positions of a text line (see Fig. 8a), we marked at each segmentation position a correct sign if a corresponding node existed in the segmentation candidate lattice, or a missing sign otherwise (see Fig. 8c). For each edge in the segmentation candidate lattice, it was treated as a real character if its two end nodes were both marked correct signs and meanwhile excluding any middle node marked with correct or missing sign. Otherwise, it is regarded as a non-character. The class label $\omega_j (j = 0, 1, 2,$ defined in (11)) of a real character was assigned according to the ground-truth content of the text line (known beforehand), whereas $\omega_j (j = 3, 4)$ of a non-character was assigned according to whether the edge contained any middle node marked with a correct or missing sign. In the 100 text lines for training as mentioned above, we collected 1,832 samples of Chinese characters, 49 samples of digits, 201 samples of punctuations, 3,229 samples of over-segmented characters and 7,328 samples of under-segmented characters. With the feature vector defined in (12), the parameters of the five-class MQDF classifier for character verification can be determined using maximum likelihood estimation [2].

5.2 Results of LDA-based negative training

We trained an MQDF classifier using the LDA-based negative training strategy. The real characters for training the classifier were detailed at the beginning of Sect. 5.1, and the non-characters for training the classifier consisting of the 10,557 non-character samples (3,229 samples of over-segmented characters and 7,328 samples of under-segmented characters) collected from the 100 training text lines as described at the end of Sect. 5.1. The MQDF classifier was then tested both on character level and text line level using the 383 text lines as detailed in Sect. 5.1, to show its performance on distinguishing between real characters and non-characters.

For comparison, we also trained and tested the MQDF classifier with two different negative training strategies: k-means clustering [8] and thresholding [13], on the same data. In k-means clustering strategy, the probability density of non-character samples was jointly approximated using k different Gaussian functions, where k was an empirical value set by observations of non-characters. In experiments, we empirically set k at 50. It should be noted that a new training procedure would be needed for the MQDF classifier whenever the cluster number k changed, even if the training samples of both real characters and non-characters remained unchanged. This added the computational burden to selecting a proper value of k. And in thresholding strategy, an input sample was treated as a non-character and rejected if its output of the topmost recognition candidate in a traditional MQDF classifier, $d(c^{\text{cand\_1}}; s, \Omega_{\text{pos}})$ defined in (18), exceeded a threshold. In experiments, we determined the threshold T as follows: by observations of the probability density of $d(c^{\text{cand\_1}}; s, \Omega_{\text{pos}})$ in class of either real characters or non-characters, we assumed that the intersection point of the two probability density curves was the threshold T, as illustrated in Fig. 9.

Two groups of experiments were conducted to compare these different negative training strategies: one was on the level of isolated character recognition and the other was

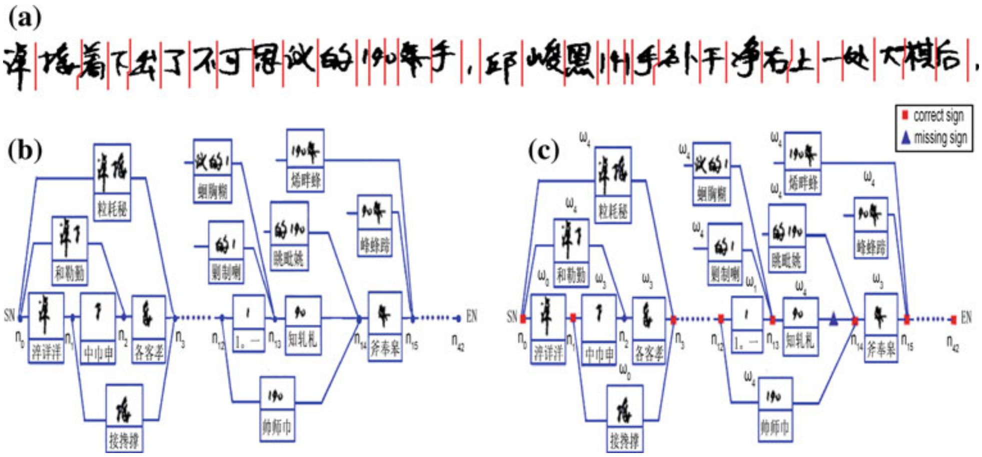

Fig. 8 Collecting training samples for a five-class MQDF classifier. **a** The ground-truth segmentation positions of a text line, **b** the segmentation candidate lattice of the *text line*, and **c** the collected training samples

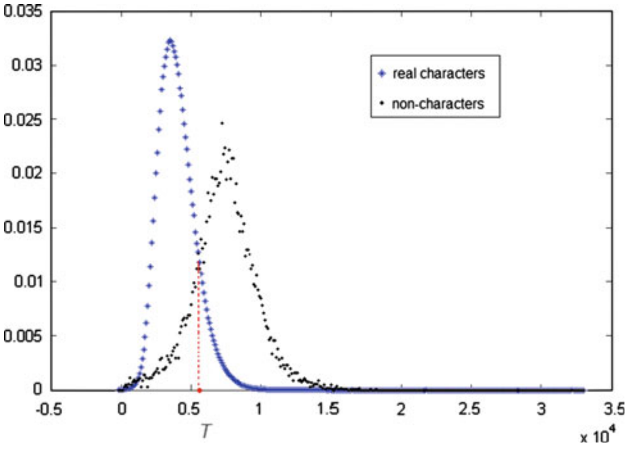

Fig. 9 Probability densities of $d(c^{\text{cand\_1}}; s, \Omega_{\text{pos}})$ defined in (18) in class of real characters or non-characters

on the level of text line recognition. In the experiments of isolated character recognition, the testing data were the labeled characters in the 383 text lines of HIT-MW dataset, as described at the beginning of Sect. 5.1. For the reasons stated at the end of Sect. 4, we just adopted the classes of real characters in the top $L (L = 1, 2, \ldots)$ recognition candidates of isolated character recognition, by excluding any class of non-characters and putting forward its subsequent classes of real characters. On the other hand, in the experiments of text line recognition, we used the 383 text lines in test set of HIT-MW dataset as testing data, as described at the beginning of Sect. 5.1. To evaluate possible segmentation hypotheses of a text line, we just preserved the third term in Eq. (14) $(P(C|E) = \sum_{i=1}^{K} k_i * \log P(c_i|s\_\text{norm}_i, v_i = 1))$, where the correctness of isolated character recognition decided on the accuracy of text line recognition. The measurements of the accuracy of text line recognition, correct recognition rate *CR* and segmentation measure *F*, were defined as follows:

$$\begin{cases} CR = NC_c/NT_c \\ F = 2/(R^{-1} + P^{-1}) \\ R = NC_s/NT_s \\ P = NC_s/NA_s \end{cases}, \tag{20}$$

where NC_c is the number of correctly recognized characters in the optimal segmentation hypotheses of text lines, NT_c is the number of characters in text lines, R is correct segmentation rate, P is valid segmentation rate, NC_s is the number of correct segmentation positions in the optimal segmentation hypotheses of text lines, NA_s is the number of all segmentation positions in the optimal segmentation hypotheses of text lines, and NT_s is the number of ground-truth segmentation positions in text lines. It is worth to note that the number of the correctly recognized characters in a text line was counted as follows: for each character in the optimal segmentation hypothesis of a text line, it might be aligned to a ground-truth character provided: (i) its image overlapped with the ground-truth position of this ground-truth character and (ii) this ground-truth character has not been aligned to any correctly recognized character in the optimal segmentation hypothesis. Check from left to right in the text line all possibly aligned ground-truth characters of the currently visited character. If the label of currently checked ground-truth character was the same as of the currently visited character, then the latter was treated to be correctly recognized and meanwhile be aligned to the former, so that we can stop this checking process. Otherwise, if none of these possibly aligned ground-truth characters had the same label as the currently visited character, the latter was regarded to be wrongly recognized. We repeated the above process from the beginning to the end of the text line until every character in the optimal segmentation hypothesis had been visited. The results of the above two groups of experiments were listed in Table 1 and Table 2, respectively.

Table 1 Correct recognition rates (%) of isolated character recognition using different recognition training strategies

	Negative training			Traditional training
	Thresholding [13]	k-means clustering ($k = 50$) [8]	**LDA-based method**	
Top 1	68.16	77.21	**77.36**	77.36
Top 5	74.94	86.12	**86.33**	86.33
Top 10	78.60	91.34	**91.89**	91.89

Bold values emphasize the best results achieved in comparison and their corresponding conditions in comparison

Table 2 Results of text line recognition preserving only the third term in (14) with different training strategies

	Negative training			Traditional training
	Thresholding [13]	k-means clustering ($k = 50$) [8]	**LDA-based method**	
CR (%)	50.55	61.86	**64.27**	52.51
F (%)	79.87	84.62	**86.92**	77.63
Time (s/line)	20.25	25.38	25.23	24.96

Bold values emphasize the best results achieved in comparison and their corresponding conditions in comparison

From Table 1, we can see that the thresholding strategy achieved the lowest accuracies in isolated character recognition, since the real character mis-recognized as a non-character was directly rejected, which may sharply reduce the number of correctly recognized characters. The character recognition rates of k-means clustering strategy slightly fluctuated around those of a traditional MQDF classifier, since the addition of k clusters of non-characters to a traditional MQDF classifier has changed the covariances of each class in the original classifier even if the training samples of real characters remained unchanged. The proposed LDA-based negative training strategy kept the same character recognition rates as in a traditional MQDF classifier. This is because the proposed strategy didn't need to change the parameters (such as mean, co-variance) of each real character class in a traditional MQDF classifier, and then kept unchanged the order and labels of the recognition candidates of real characters in the classifier. By excluding any class of non-characters from the top $L(L = 1, 2, \ldots)$ recognition candidates (as described at the end of Sect. 4), the remaining recognition candidates in a negatively trained classifier were the same as in a traditional classifier.

From Table 2, we can see that the negative training strategies were helpful to improve the accuracies of text line recognition, except for the thresholding strategy. This was because thresholding strategy simply rejected an input sample instead of adjusting its posterior probability, and the false-negative errors in character rejection may lead to lower recognition accuracy. The proposed LDA-based negative training strategy performed best among the three negative training strategies in text line recognition. It achieved a higher recognition rate (64.27%) than k-means clustering did (61.68%) by 2.41%, which suggested that the proposed strategy of posterior probability estimation was more suitable than the empirical approximation by using k different Gaussian functions, as in k-means clustering strategy.

5.3 Results of Bayesian-based probabilistic model

We applied the proposed probabilistic model in (14) to unconstrained handwritten offline Chinese text line recognition, with the 383 text lines for testing as detailed in Sect. 5.1. Two groups of experiments were conducted to test the performance of the proposed model on text line recognition: one was the proposed model under different conditions and the other was the comparison of the proposed model with other two models listed below [4,21]:

$$\log P(C|E) \approx \sum_{i=1}^{K} \log P(c_i|c_{i-1}) + \sum_{i=1}^{K} k_i * \log P(s\_\text{norm}_i|c_i), \tag{21}$$

and

$$\log P(C|E) \approx \sum_{i=1}^{K} \log P(c_i|c_{i-1}) - \sum_{i=1}^{K} k_i * \log P(c_i) + \sum_{i=1}^{K} k_i * \log P(c_i|s\_\text{norm}_i, v_i = 1) + \sum_{j=1}^{3} \log P(f_1^{(j)}, f_2^{(j)}, \ldots, f_K^{(j)}), \tag{22}$$

Table 3 Results of text line recognition using the proposed probabilistic model (14) under different conditions

	Cond.1		Cond.2		Cond.3		**Cond.4**	
	N/A	Bi-gram	N/A	Bi-gram	N/A	Bi-gram	N/A	**Bi-gram**
CR (%)	52.51	59.26	64.27	73.02	65.40	73.72	71.37	**80.15**
F (%)	77.63	78.97	86.92	88.28	89.35	90.00	92.03	**92.79**
Time (s/line)	24.96	28.11	25.23	29.27	26.38	31.77	27.99	32.07

N/A means that no language model is applied
Bold values emphasize the best results achieved in comparison and their corresponding conditions in comparison

where the probability $P(s\_\text{norm}_i|c_i)$ was the conditional probability of the normalized segment s_norm_i given class c_i in an isolated character recognizer, the value $f_i^{(j)}(i = 1, 2, \ldots, K, j = 1, 2, 3)$ was the jth dimension of the geometrical features of segment s_i, the joint probability $P(f_1^{(j)}, f_2^{(j)}, \ldots, f_K^{(j)})$ can be empirically approximated using a group of Gaussian functions [4], and other variables were the same as in (14). The Eq. (21) came from the model which employed the segment duration weighting and the class conditional probability of character recognition [21]. For more reasonable comparison with the proposed method, it discarded the two empirical weights that were used to balance the effects of isolated character recognition and the bi-gram language model. The Eq. (22) came form the model using empirical segmentation layout estimation for character verification [4], which was modified to be the same as (14) except for its last term for character verification. In our experiments, we implemented (21) and (22) using the same text line recognition system as for (14) by substituting the corresponding segmentation hypothesis evaluation criterion.

The results of the first group of the above experiments were listed in Table 3, with the following four different conditions:

Cond.1: Segmentation hypothesis evaluation using isolated character recognition (without negative training) and/or a bi-gram language model [discarding the fourth term in (14)].

Cond.2: Segmentation hypothesis evaluation using isolated character recognition (with the LDA-based negative training) and/or a bi-gram language model [discarding the fourth term in (14)].

Cond.3: Segmentation hypothesis evaluation using isolated character recognition (without negative training) and character verification and/or a bi-gram language model [preserving the third and fourth terms in (14)].

Cond.4: Segmentation hypothesis evaluation using isolated character recognition (with the LDA-based negative training) and character verification and/or a bi-gram language model [preserving the third and fourth terms in (14)].

From Table 3, we can see that compared to just using the posterior probability of a traditional MQDF classifier (Cond.1) for segmentation hypothesis evaluation, the assistance with either the LDA-based negative training (Cond.2) or the character verification (Cond.3) was able to greatly improve the accuracy of text line recognition. With the LDA-based negative training strategy, the correct recognition rates were increased by 11.76% (from 52.51 to 64.27%) and 13.76% (from 59.26 to 73.02%) without and with a bi-gram language model, respectively. Again it showed the effect of the proposed LDA-based negative training in resisting against non-characters. On the other hand, with the five-class MQDF classifier for character verification, the correct recognition rates were increased by 12.89% (from 52.51 to 65.40%) and 14.46% (from 59.26 to 73.72%) without and with a bi-gram language model, respectively. It implies that the features selected for character verification and the assumptions on their distributions in the five classes were reasonable. It's worth noticing that when using the LDA-based negative training and the character verification simultaneously (Cond.4), the correct recognition rates were further increased, reaching 71.37 and 80.15% without and with a bi-gram language model, respectively. This is mainly because the LDA-based negative training of an MQDF classifier has improved the accuracy of the posterior probability of the classifier, which rendered both the character recognition and the character verification [the fourth dimension in (12)] more proper for segmentation hypothesis evaluation.

The results of the second group of the experiments were listed in Table 4. In this group, the method 1 used (22) (modified from Ref. [4]) for segmentation hypothesis evaluation, the method 2 employed (21) (modified from Ref. [21]) for segmentation hypothesis evaluation, and the proposed method adopted (14) for segmentation hypothesis evaluation.

From Table 4, we can see that the proposed method achieved the best performance among the three methods. Compared to method 1, the proposed method improved the correct recognition rates by 8.96% (from 62.41 to 71.37%) and 11.08% (from 69.07 to 80.15%) without and with a bi-gram language model, respectively. Since the method 1 differs from the proposed method only in the ways of character verification [the last terms in (22) and (14)], we may

Table 4 Results of text line recognition using different probabilistic models

N/A means that no language model is applied

Bold values emphasize the best results achieved in comparison and their corresponding conditions in comparison

	Method 1 [4] (with negative training)		Method 2 [21]		**Proposed method (with negative training)**	
	N/A	Bi-gram	N/A	Bi-gram	N/A	**Bi-gram**
CR (%)	62.41	69.07	65.09	75.45	71.37	**80.15**
F (%)	90.98	91.00	87.09	88.71	92.03	**92.79**
Time (s/line)	22.38	23.81	22.25	24.35	27.99	32.07

say that the five-class MQDF classifier for character verification in the proposed method can perform better than the empirical segmentation layout evaluation for character verification in method 1. On the other hand, compared to method 2, the proposed method increased the correct recognition rates by 6.28% (from 65.09 to 71.37%) and 4.70% (from 75.45 to 80.15%) without and with a bi-gram language model, respectively. This suggests that the posterior probabilities of a negatively trained MQDF classifier along with the posterior probabilities of a five-class classifier can perform better than just using the class conditional probability densities of a traditional MQDF classifier for segmentation hypothesis evaluation. It is worth to note that compared to the posterior probabilities, the class conditional probability densities of an MQDF classifier were much less affected by non-character class, since the calculation of the former depended on all the classes in an MQDF classifier [see the summation under the fraction in (8)], whereas the calculation of the latter was independent for each class in the classifier. Specifically, when using the proposed LDA-based negative training strategy, the class conditional probability densities of an MQDF classifier can be calculated as $p(s|\omega_i) = \frac{P(\omega_i|s)p(s)}{P(\omega_i)} = \begin{cases} p(s|\omega_i, \Omega_{\text{pos}}), & \text{if } i = 1, 2, \ldots, N \\ p(s|\Omega_{\text{neg}}), & \text{if } i = N+1 \end{cases}$, where the probability $P(\omega_i|s)(i = 1, 2, \ldots, N+1)$ was defined in (16), the probability $p(s|\omega_i, \Omega_{\text{pos}})(i = 1, 2, \ldots, N)$ was the class conditional probability of a traditional MQDF classifier, and the probability $P(s|\Omega_{\text{neg}})$ can be calculated according to (18). The right hand side of the second equal mark implied that the proposed LDA-based negative training strategy remained the original value of the class conditional probabilities of each real character class in a traditional MQDF classifier, which accorded with the fact that the proposed negative training did not alter the mean and co-variance of each real character class in the classifier. As the proposed LDA-based negative training strategy was able to adjust the posterior probability of an MQDF classifier while keeping the class conditional probability density of each real character class unchanged, in our experiments, we still employed a traditional MQDF classifier to test the method 2 [21].

Some examples of text line recognition results using the proposed method were shown in Fig. 10, where the characters embraced in a rectangle were the ground-truth content of a text line, and the characters with underscores were the mis-recognized characters. Errors in text line recognition mainly arose from the following aspects: the incorrect segmentation of a text line (such as the under-segmented digits "4 0", and "4 1" in the first text line in Fig. 10, which were mis-recognized as characters"和" and "划", respectively), the inaccurate recognition of an individual character (such as the last character "以" in the second text line in Fig. 10, which was correctly segmented but wrongly recognized as "头") and the ineffectiveness in character verification (such as the two radicals of character "总" in the third text line in Fig. 10, which was verified as real characters and recognized as two characters "运" and "–"). The above recognition errors suggested that in order to achieve better performance on unconstrained handwritten offline Chinese text line recognition, we need to further improve the performance of the above important modules in text line recognition.

6 Discussions and conclusion

In this paper, a novel Bayesian-based probabilistic model was presented for unconstrained handwritten offline Chinese text line recognition. In this probabilistic model, traditional isolated character recognition was combined with character verification to jointly recognize each segment in a realistic handwritten text line. The character verifier can be trained with a moderate number of handwritten text lines (just 100 text lines in our experiments) and has shown effectiveness in improving the accuracies of text line recognition. To enhance the ability of an isolated character recognizer to distinguish between characters and non-characters, an LDA-based negative training strategy is presented. By employing the outputs of each class in a traditional MQDF classifier and the LDA transform, the posterior probability of each real character class or non-character class was re-computed. This strategy worked better than those either simply discarded a non-character by thresholding or empirically approximated the probability distribution of non-characters by *k*-means clustering. Experiments of the proposed method testing on 383 text lines in HIT-MW database showed that both the proposed character verification and the LDA-based negative training were

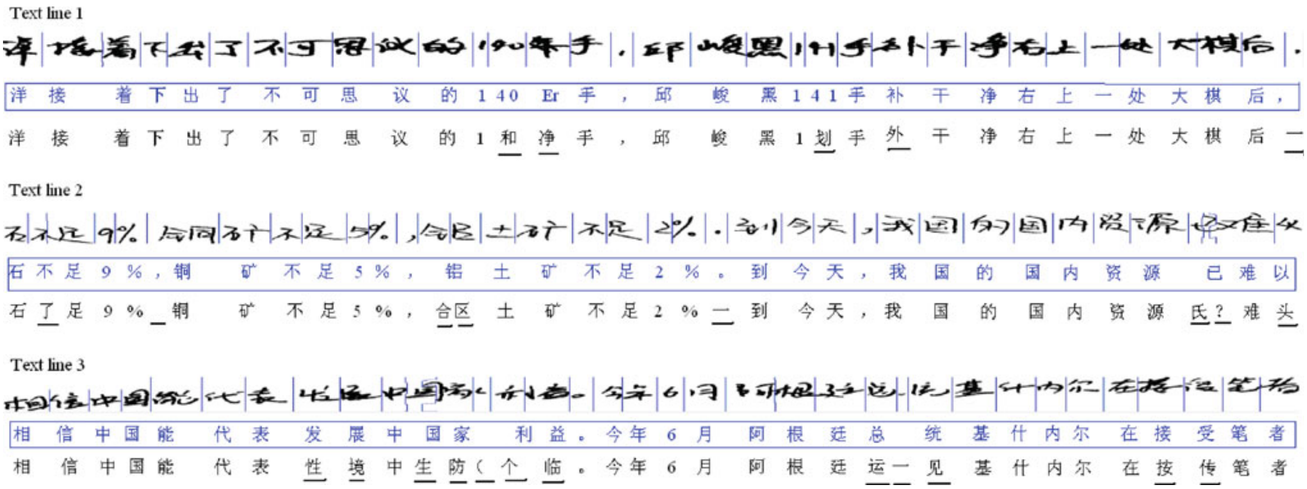
Text line 1
洋接着下出了不可思议的140Er手，邱峻黑141手补干净右上一处大棋后，
洋接着下出了不可思议的1和净手，邱峻黑1划手外干净右上一处大棋后一
Text line 2
石不足9%，铜矿不足5%，铝土矿不足2%。到今天，我国的国内资源已难以
石了足9%_铜矿不足5%，合区土矿不足2%一到今天，我国的国内资源氏?难头
Text line 3
相信中国能代表发展中国家利益。今年6月阿根廷总统基什内尔在接受笔者
相信中国能代表性境中生防(个临。今年6月阿根廷运一见基什内尔在按传笔者

Fig. 10 Examples of the *text line* recognition results using the proposed probabilistic model

effective to improve the accuracies of text line recognition. The character-level recognition rates of realistic handwritten text lines reached 71.37% without any language model and 80.15% with a bi-gram language model, respectively, outperforming the most recent methods tested on the same data.

In future, we can improve the proposed method for text line recognition in following three aspects: firstly, to improve the accuracies of isolated character recognition, methods of re-computing the scores of each class in a distance classifier could be considered. Secondly, to further increase the effectiveness of character verification, we may exploit more powerful features reflecting the geometrical characteristics of a handwritten text line. Lastly, the combination of the proposed method with some segmentation-free methods could be helpful to eliminate the mis-recognition arising from errors in text line segmentation.

Acknowledgments The authors would like to thank Cheng-Lin Liu, Fei Yin, Tong-Hua Su, Qiu-Feng Wang, et al at the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, for sharing the ground-truth labeling of the HIT-MW database and the *n*-gram language model. The authors also sincerely thank the anonymous reviewers for their valuable suggestions. This work was supported in part by the NSFC (Grant no. U0735004, 61075021) and GDSFC (Grant no. S2011020000541, GCZX-A1004).

References

1. Ding, K., Liu, Z.-B., Jin, L.-W., Zhu, X.-H.: A comparative study of Gabor feature and gradient feature for handwritten Chinese character recognition. In: Proceedings of the 2007 International Conference on Wavelet Analysis and Pattern Recognition, pp. 1182–1186. Beijing, China (2007)
2. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. China Machine Press, Beijing (2005)
3. Feng, Z.-D., Huo, Q.: Confidence guided progressive search and fast match techniques for high performance Chinese/English OCR. In: Proceedings of the16th International Conference on Pattern Recognition, pp. 89–92. Quebec City, Canada (2002)
4. Fu, Q., Ding, X.-Q., Liu, T., Jiang, Y., Ren, Z: A novel segmentation and recognition algorithm for Chinese handwritten address character strings. In: Proceedings of the 18th International Conference on Pattern Recognition, pp. 974–977. Hong Kong, China (2006)
5. Gao, T.-F., Liu, C.-L.: High accuracy handwritten Chinese character recognition using LDA-based compound distances. Pattern Recognit. **41**(11), 3442–3451 (2008)
6. Gu, J.-X., Ding, X.-Q.: Fusion recognition of courtesy and legal amounts on Chinese handwritten bank checks. In: Proceedings of the 8th International Conference on Signal Processing, pp. 1738–1741. Beijing, China (2006)
7. He, C.L., Lam, L., Suen, C.Y.: A novel rejection measurement in handwritten numeral recognition based on linear discriminant analysis. In: Proceedings of the 10th International Conference on Document Analysis and Recognition, pp. 451–455. Barcelona, Spain (2009)
8. Huo, Q., Feng, Z.-D.: Improving Chinese/English OCR performance by using MCE-based character-pair modeling and negative training. In: Proceedings of the 7th International Conference on Document Analysis and Recognition, pp. 364–368. Edinburgh, Scotland (2003)
9. Jin, L.-W.: SCUT-IRAC: the latest handwritten offline Chinese database (in Chinese). BYTE China (1), 101–102 (1998)
10. Jin, L.-W., Wei, G.: Handwritten Chinese character recognition with directional decomposition cellular features. J. Circ. Syst. Comput. **8**(4), 517–524 (1998)
11. Kimura, F., Takashina, K., Tsuruoka, S., Miyake, Y.: Modified quadratic discriminant functions and the application to Chinese Character Recognition. IEEE Trans. Pattern Anal. Mach. Intell. **9**(1), 149–153 (1987)
12. Li, N.-X., Gao, X., Jin, L.-W.: Curved segmentation path generation for unconstrained handwritten Chinese text lines. In: Proceedings of the 2008 IEEE Asia Pacific Conference on Circuits and Systems, pp. 501–505. Macao, China (2008)
13. Liu, C.-L., Sako, H., Fujisawa, H.: Effects of classifier structures and training regimes on integrated segmentation and recognition of handwritten numeral strings. IEEE Trans. Pattern Anal. Mach. Intell. **26**(11), 1395–1407 (2004)
14. Liu, C.-L., Nakagawa, M.: Precise candidate selection for large character set recognition by confidence evaluation. IEEE Trans. Pattern Anal. Mach. Intell. **22**(6), 636–642 (2000)

15. Long, T., Jin, L.-W.: Building compact MQDF classifier for large character set recognition by subspace distribution sharing. Pattern Recognit. **41**(9), 2916–2925 (2008)
16. National Laboratory of Pattern Recognition (NLPR): Institute of Automation, Chinese Academy of Sciences: CASIA Online and Offline Chinese Handwriting Databases. http://www.nlpr.ia.ac.cn/databases/handwriting/Home.html (2010)
17. Rabiner, L.R.: A tutorial on Hidden Markov models and selected applications in speech recognition. In: Proceedings of the IEEE, pp. 257–285 (1989)
18. Sadri, J., Suen, C.Y., Bui, T.D: A genetic framework using contextual knowledge for segmentation and recognition of handwritten numeral strings. Pattern Recognit. **40**(3), 898–919 (2007)
19. Su, T.-H., Zhang, T.-W., Guan, D.-J.: Corpus-based HIT-MW database for offline recognition of general-purpose Chinese handwritten text. Int. J. Doc. Anal. Recognit. **10**(1), 27–38 (2007)
20. Su, T.-H., Zhang, T.-W., Guan, D.-J., Huang, H.-J.: Off-line recognition of realistic Chinese handwriting using segmentation-free strategy. Pattern Recognit. **42**(1), 167–182 (2009)
21. Wang, Q.-F., Yin, F., Liu, C.-L.: Integrating language model in handwritten Chinese text recognition. In: Proceedings of the 10th International Conference on Document Analysis and Recognition, pp. 1036–1040. Barcelona, Spain (2009)
22. Zhang, H.-G., Guo, J., Chen, G., Li, C.-G.: HCL2000: a large-scale handwritten Chinese character database for handwritten character recognition. In: Proceedings of the 10th International Conference on Document Analysis and Recognition, pp. 286–290. Barcelona, Spain (2009)
23. Zhu, B.-L., Zhou, X.-D., Liu, C.-L., Nakagawa, M.: A robust model for on-line handwritten Japanese text recognition. Int. J. Doc. Anal. Recognit. **13**(2), 121–131 (2010)
24. Zhou, X.-D., Liu, C.-L., Nakagawa, M.: Online handwritten Japanese character string recognition using conditional random fields. In: Proceedings of the 10th International Conference on Document Analysis and Recognition, pp. 521–525. Barcelona, Spain (2009)
25. Zhou, X.-D., Yu, J.-L., Liu, C.-L., Nagasaki, T., Marukawa, K.: Online handwritten Japanese character string recognition incorporating geometric context. In: Proceedings of the 9th International Conference on Document Analysis and Recognition, pp. 48–52. Curitiba, Brazil (2007)