2009 10th International Conference on Document Analysis and Recognition

Statistical Modeling and Learning for Recognition-based Handwritten Numeral String Segmentation

Yanjie Wang Xiabi Liu* Yunde Jia
Beijing Laboratory of Intelligent Information Technology
School of Computer Science and Technology
Beijing Institute of Technology, Beijing 100081, China
{wangyanjie, liuxiabi, jiayunde}@bit.edu.cn

Abstract

This paper proposes a recognition based approach to handwritten numeral string segmentation. We consider two classes: numeral strings segmented correctly or not. The feature vectors containing recognition information for numeral strings segmented correctly are assumed to be of the distribution of Gaussian mixture model (GMM). Based on this modeling, the recognition based segmentation is solved under the Max-Min posterior Pseudo-probabilities (MMP) framework of learning Bayesian classifiers. In the training phase, we use the MMP method to learn a posterior pseudo-probability measure function from positive samples and negative samples of numeral strings segmented correctly. In the process of recognition based segmentation, we generate all possible candidate segmentations of an input string through contour and profile analysis, and then compute the posterior pseudo-probabilities of being the numeral string segmented correctly for all the candidate segmentations. The candidate segmentation with the maximum posterior pseudo-probability is taken as the final result. The effectiveness of our approach is demonstrated by the experiments of numeral string segmentation and recognition on the NIST SD19 database.

1. Introduction

Character segmentation is a main problem in state-of-the-art off-line handwriting recognition. The unsatisfactory results of character segmentation often lead to failures of character string recognition.

Early methods of character segmentation are free of recognition, which does not concern the following recognition results [4, 18, 11, 3, 6, 14]. The main drawback of recognition-free methods is in their experiential solution to amphibolous breaking of touching characters. Therefore, recognition based methods are introduced and developed to deal with the segmentation of touching characters in recent years [5, 15, 7, 16, 13, 1, 8, 9, 17]. In recognition based character segmentation, candidate segmentation hypotheses are generated firstly using recognition-free methods, then from which the optimal one is selected based on corresponding recognition information. How to determining the optimal result in hypotheses is a key problem of recognition based character segmentation. Several methods have been presented in previous work to solve this problem, such as beam search [9], dynamic programming [16], Bayesian decision [13, 8], HMM based methods [15, 1], etc.

In this paper, we propose a novel recognition based approach to segmentation of off-line handwritten numeral strings. The main contribution is the determination method of the optimal result in candidate segmentations, which is developed under the Max-Min posterior Pseudo-probabilities (MMP) framework of learning Bayesian classifiers [10]. We extract the feature vectors from segmented and recognized numeral strings. The feature vectors for correct segmentations are assumed to be of the distribution of Gaussian mixture model (GMM). A corresponding posterior pseudo-probability measure function is obtained to discriminate correct segmentations from wrong segmentations. Unknown parameters in the posterior pseudo-probability measure function are learned from the training data using the MMP method. After MMP learning of numeral strings segmented correctly, an input numeral string is segmented and recognized in three steps. Firstly, we employ contour and profile analysis to generate candidate segmentations. Then a feature vector with recognition information is extracted for each candidate segmentation. Finally, we compute the posterior pseudo-probability of numeral strings segmented correctly for each candidate segmentation. The candidate segmentation with the maximum pos-

*The corresponding author: Tel.: +86-10-68913447, Fax: +86-10-86343158

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.25

terior pseudo-probability is selected as the optimal one. We conducted the experiments of segmentation and recognition of off-line handwritten numeral strings on NIST SD19 database. In the length-free test where the number of digits in the string is not been preset, 97.45% and 94.6% correct rates of segmentation were achieved for 2-digit and 3-digit strings, respectively. The corresponding results were 97.95% and 95.4% in the length-fixed test where we preset the numbers of digits in the strings as the true ones. These results are comparable to those reported previously.

The rest of this paper is organized as follows. Section 2 describes the generation method of candidate segmentations. Section 3 presents the GMM modeling of numeral strings segmented correctly and the MMP method of determining the optimal segmentation. Section 4 discusses the experimental results. We conclude the paper in Section 5.

2. Generating Candidate Segmentations

In this stage, the numeral string is over-segmented to generate candidate segmentations in order that the correct segmentation is guaranteed to be included in these candidate segmentations. Our method is a modified version of Lei et al.'s [8].

Firstly, two kinds of points in the contour of a numeral string are extracted as candidate segmentation points. We use contour analysis to extract the points with high curvature in the contour, and employ profile analysis to detect the points around which the Euclidean distance between top profile and bottom profile change greatly.

Then we travel the set of candidate segmentation points to get candidate segmentation lines. For each candidate segmentation point, a corresponding segmentation line is obtained as follows. Let A denote a candidate segmentation point under consideration. Suppose A is in the top contour. We compute Euclidean distances between A and all candidate segmentation points in the bottom contour. If the least distance is smaller than some threshold, then the line corresponding with the least distance will be taken as a candidate segmentation line. Or else, a vertical line passing through A towards the bottom contour will be taken as a candidate segmentation line. The process above is similar if A is in the bottom contour. After all candidate segmentation points are processed in this way, we check the obtained set of candidate segmentation lines. If several candidate segmentation lines are adjacent to each other, only one of them is preserved and others are deleted.

Finally, candidate segmentations are generated based on candidate segmentation lines. Each candidate segmentation line can be used to cut the string into two parts. So let m be the number of candidate segmentation lines, then the possible number of segmented digits is in $\{1, 2, \cdots, m+1\}$. If the number of digits in the string is known beforehand, let it be n, then C_m^{n-1} candidate segmentations will be generated. Oppositely, if the number of digits in the string is unknown, then we must consider all possible numbers of segmented digits, thus the number of candidate segmentations generated will be $\sum_{n=1}^{m+1} C_m^{n-1}$.

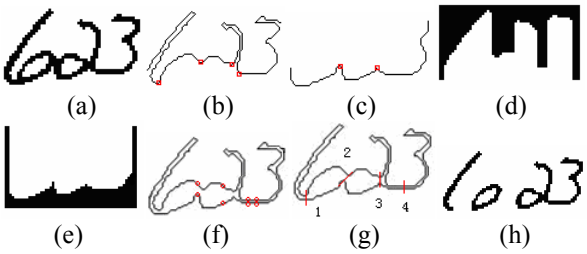

Figure 1. The illustration of candidate segmentation generation: (a) a numeral string image; (b) the top contour and points with high curvature in it; (c) the bottom contour and points with high curvature in it; (d) the top profile; (e) the bottom profile; (f) the points around which Euclidean distances between top and bottom profile change greatly; (g) candidate segmentation lines; (h) the candidate segmentation corresponding with (1, 2) numbered in Fig. 1g.

In Fig. 1, we take an image of the numeral string "623" as an example to illustrate the process of generating candidate segmentations.

3. Determining the Optimal Segmentation

The optimal segmentation is selected from the set of candidate segmentations generated in Section 2. To solve this problem, we apply the digit classifier of Chen et al. [2] to recognize segmented blocks in each candidate segmentation and design a 5-D feature vector with corresponding recognition information to represent each candidate segmentation. The feature vectors for correct segmentations are then assumed to be of the distribution of GMM. A resultant posterior pseudo-probability measure function is obtained to discriminate the correct segmentations from the wrong ones.

3.1 Gaussian Mixture Modeling of Numeral Strings

After segmentation and recognition, a numeral string is represented as a 5-D feature vector $\mathbf{x} = \{x_1, x_2, x_3, x_4, x_5\}$, elements in which are explained as follows.

1) x_1: the average similarity between segmented digits and corresponding recognition results in the string.

2) x_2: the smallest similarity between segmented digits and corresponding recognition results in the string.

3) x_3: the ratio of the largest height to the smallest height for segmented blocks in the string.

4) x_4: the ratio of the largest width to the smallest width for segmented blocks in the string.

5) x_5:the ratio of the average width of segmented blocks to the height of the whole string.

Generally speaking, digits segmented correctly are more similar to corresponding recognition results than digits segmented improperly. Therefore it is reasonable to evaluate candidate segmentations using similarities between digits segmented and corresponding recognition results. Here we consider the average similarity and the smallest similarity for a numeral string. Furthermore, digits in the string are usually written in regular size. It is singular to write a digit in much larger or smaller size than other digits in a string. Therefore, we use x_3, x_4, and x_5 to evaluate the regularity of each candidate segmentation.

We assume that $\mathbf{x} = \{x_1, x_2, x_3, x_4, x_5\}$ for numeral strings segmented correctly is of the distribution of GMM. Let C denote the class of numeral strings segmented correctly, K be the number of components in the GMM, $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, and w_k be the mean, the covariance matrix, and the weight of the k-th Gaussian component respectively, $\sum_{k=1}^{K} w_k = 1$, then we have

$$p(\mathbf{x}|C) = \sum_{k=1}^{K} w_k N(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{1}$$

where

$$\begin{aligned} &N(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\ &= (2\pi)^{-\frac{5}{2}} |\boldsymbol{\Sigma}_k|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)' \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}-\boldsymbol{\mu}_k)\right), \end{aligned} \tag{2}$$

For practical computation, $\boldsymbol{\Sigma}_k$ is assumed as a diagonal matrix, i.e. $\boldsymbol{\Sigma}_k = [\sigma_{kj}]_{j=1}^{5}$.

3.2. MMP Based Segmentation Evaluation

Based on the GMM modeling of numeral string segmented correctly, we identify the correct segmentation from candidate segmentations under the MMP framework of learning Bayesian classifiers. In this section, the resultant MMP algorithm of identifying correct segmentation is described. The reader is referred to the paper of Liu et al. [10] for more details of the MMP framework.

Let $\{\mathbf{x}_i\}_{i=1}^{m}$ be the set of feature vectors extracted from candidate segmentations, then the corresponding posterior pseudo-probability measure function is

$$f(p(\mathbf{x}_i|C)) = 1 - \exp(-\lambda p(\mathbf{x}_i|C)), \tag{3}$$

where λ is a positive number. In fact, the value of $f(p(\mathbf{x}_i|C))$ imitate the posterior probability of being correct segmentation for each candidate segmentation. Therefore, we select the candidate segmentation with the maximum posterior pseudo-probability as the optimal result i^*, i.e.

$$i^* = \arg \max_{i \in \{1,2,\cdots,m\}} f(p(\mathbf{x}_i|C)). \tag{4}$$

The set of unknown parameters in Eq.3 is

$$\boldsymbol{\Lambda} = \{\lambda, w_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}, k = 1, \cdots, K, \tag{5}$$

Accordingly, the posterior pseudo-probability measure function can be expressed as $f(\mathbf{x}_i; \boldsymbol{\Lambda})$.

We estimate $\boldsymbol{\Lambda}$ from the training data using the MMP learning method. In the MMP learning, the posterior pseudo-probabilities of the class for its positive samples are maximized towards 1, while those for its negative samples are minimized towards 0. In this paper, the class of numeral strings segmented correctly is under consideration. Correct segmentations for arbitrary numeral strings are its positive samples, and wrong segmentations are its negative samples.

Let m and n be the numbers of positive and negative samples of numeral strings segmented correctly in the training data, $\hat{\mathbf{x}}_i$ and $\bar{\mathbf{x}}_j$ be the feature vector of arbitrary positive and negative sample, respectively. According to the MMP learning, the objective function is designed as

$$\begin{aligned} &F(\boldsymbol{\Lambda}) \\ &= \frac{n}{m+n} \sum_{i=1}^{m} [f(\hat{\mathbf{x}}_i; \boldsymbol{\Lambda}) - 1] + \frac{m}{m+n} \sum_{j=1}^{n} [f(\bar{\mathbf{x}}_j; \boldsymbol{\Lambda})], \end{aligned} \tag{6}$$

$F(\boldsymbol{\Lambda}) = 0$ means the hundred-percent separability between correct and wrong segmentation. Consequently, we can obtain the optimal parameter set $\boldsymbol{\Lambda}^*$ by minimizing $F(\boldsymbol{\Lambda})$:

$$\boldsymbol{\Lambda}^* = \arg \min_{\boldsymbol{\Lambda}} F(\boldsymbol{\Lambda}). \tag{7}$$

In this paper, the gradient descent algorithm is employed to optimize the parameter set $\boldsymbol{\Lambda}^*$ according to Eq. 7.

4. Experimental Results

We conducted experiments of numeral string segmentation and recognition on NIST SD19 database [12], which is one of the most popular databases in the field of digit recognition. NIST SD19 database contains 3699 forms filled by different writers, from which we manually collected 2300 two-digit strings and 700 three-digit strings. The digits are touched with each other in these strings.

300 two-digit strings and 200 three-digit strings were used for training, and other strings for test. That is to say,

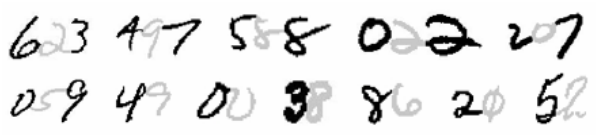

Figure 2. Examples of numeral strings which are correctly segmented and recognized.

(a) (b) (c) (d)

Figure 3. Examples of wrong segmentation or recognition.

we have 500 positive samples among which 300 and 200 for two-digit strings and three-digit strings, respectively. As for negative samples, we performed the over-segmentation procedure described in Section 2 on training strings, all candidate segmentations corresponding with wrong segmentations were taken as negative samples. In this way, 4025 negative samples, including 1047 for two-digit strings and 2978 for three-digit strings, were obtained. After the training, we conducted two kinds of experiments on the test set: length-fixed and length-free. In length-fixed test, the numbers of digits in the strings are preset as the true ones. In length-free test, the numbers of digits in the strings are determined automatically in the segmentation and recognition process.

In length-free test, we achieved 97.45% and 94.6% correct rates of segmentation for 2-digit and 3-digit strings, respectively. The better results were obtained in length-fixed test, where 97.95% and 95.4% correct rates of segmentation were achieved for 2-digit and 3-digit strings, respectively. These experimental results are listed in Table 1 where we also collected and listed the results of similar experiments reported in previous works.

Fig. 2 shows some of numeral strings which were segmented and recognized correctly in experiments. Fig. 3 shows some examples of wrong segmentation or recognition. The reasons behind the errors are analyzed in the following: (1) There are distinct differences between widths of digits in the string, as shown in Fig. 3a; (2) There are essential segmentation ambiguities which cannot be solved by even human, as shown in Fig. 3b; (3) The strings are segmented correctly, but recognized incorrectly, as shown in Fig. 3d. Based on the analysis above, the performance of our method could be improved through exploring more effective features of numeral strings segmented correctly and using more sophisticated digit classifiers.

5. Conclusions

In this paper, a recognition based approach to numeral string segmentation has been proposed under the Max-Min posterior Pseudo-probabilities (MMP) framework of learning Bayesian classifiers. We extract the feature vectors with recognition information from numeral strings segmented and recognized. The feature vectors for numeral strings segmented correctly are assumed to be of the distribution of Gaussian mixture model. A corresponding posterior pseudo-probability measure function is obtained to discriminate correct segmentations from wrong segmentations. Unknown parameters in the posterior pseudo-probability measure function are learned from the training data by the MMP method. For an input image of numeral string, candidate segmentations are generated firstly according to contour and profile analysis, then the optimal one with the maximum posterior pseudo-probability is selected as the final result.

We conducted the experiments of numeral string segmentation and recognition on NIST SD19 database. 97.45% and 94.6% correct rates of segmentation were achieved respectively for 2-digit and 3-digit strings in length-free test, and 97.95% and 95.4% respectively for 2-digit and 3-digit strings in length-fixed test. These results are comparable to those reported in previous works. The analysis on experimental results reveals that the performance of our method could be improved through exploring more effective features of numeral strings segmented correctly and using more sophisticated digit classifiers in the future.

6. Acknowledgements

This research was partially supported by 973 Program of China (No. 2006CB303103), Excellent Young Scholars Research Fund of Beijing Institute of Technology (No. 2008YS1203), Open Fund of the National Laboratory of Pattern Recognition, and MCE Theme Project Fund of Microsoft Research Asia (No. FY08-RES-THEME-158).

References

[1] A. S. Britto, R. Sabourin, F. Bortolozzi, and C. Y. Suen. The recognition of handwritten numeral strings using a two-stage hmm-based method. *International Journal of Document Analysis and Recognition*, 5:102–117, 2003.

[2] X. Chen, X. Liu, and Y. Jia. Learning handwritten digit recognition by the max-min posterior pseudo-probabilities method. In *International Conference on Document Analysis and Recognition*, 2007.

Table 1. Performance comparison of our method and other methods.

Methods	Database	2-digit string number	Correct rate of seg. on 2-digit string	3-digit string number	Correct rate of seg. on 3-digit string
Length-free	NIST SD19	2000	97.45%	500	94.6%
Length-fixed	NIST SD19	2000	97.95%	500	95.4%
Lei, 2004	NIST SD19	3359	97.72%	525	93.33%
Chen, 2000	NIST+Unknown	4178+322	96%	-	-
Pal, 2003	French Check	2250	94.8%	-	-

[3] Y. K. Chen and J. F. Wang. Segmentation of single- or multiple-touching handwritten numeral string using background and foreground analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(11):1304–1317, 2000.

[4] H. Fujisawa, Y. Nakano, and K. Kurino. Segmentation methods for character recognition : from segmentation to document structure analysis. In *IEEE*, volume 80, pages 1079–1092, 1992.

[5] P. D. Gader, J. M. Keller, R. Krishnapuram, J. Chiang, and M. A. Mohamed. Neural and fuzzy methods in handwriting recognition. *Computer*, 30:79–86, 1997.

[6] K. K. Kim, J. H. Kim, and C. Y. Suen. Recognition of unconstrained handwritten numeral strings by composite segmentation method. In *Intel. Conf. on Pattern Recognition*, 2000.

[7] S. W. Lee and S. Y. Kim. Integrated segmentation and recognition of handwritten numerals with cascade neural network. *IEEE Trans. Systems, Man, and Cybernetics*, 29(2):285–290, 1999.

[8] Y. Lei, C. Liu, X. Ding, and Q. Fu. A recognition based system for segmentation of touching handwritten numeral strings. In *Intern. Workshop on Frontiers in Handwriting Recognition*, 2004.

[9] C. Liu, H. Sako, and H. Fujisawa. Effects of classifier structures and training regimes on integrated segmentation and recognition of handwritten numeral strings. *IEEE Tran. Pattern Analysis and Machine Intelligence*, 26(11):1395–1407, 2004.

[10] X. Liu, Y. Jia, X. Chen, Y. Deng, and H. Fu. Image classification using the max-min posterior pseudo-probabilities method. Technical Report BIT-CS-20080001, Beijing Institute of Technology http://www.mcislab.org.cn/member/˜xiabi/papers/2008_1.PDF, 2008.

[11] Z. Lu, Z. Chi, W. C. Siu, and P. Shi. A background-thinning based approach for separating and recognizing connected handwritten digit strings. *Pattern Recognition*, 32:921–933, 1999.

[12] D. Micheal, L. James, and T. Gerald. Nist form based handprint recognition system (release 2.0). Technical report, U.S. Department of Commerce Technology Administration: National Institute of Standards and Technology, 1997.

[13] L. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen. Automatic recognition of handwritten numerical strings: a recognition and verification strategy. *IEEE Tran. Pattern Analysis and Machine Intelligence*, 24(11):1438–1454, 2002.

[14] U. Pal, A. Belaid, and C. Choisy. Touching numeral segmentation using water reservoir concept. *Pattern Recognition Letters*, 24(1):261–272, 2003.

[15] S. Procter and A. J. Elms. The recognition of handwritten digit strings of unknown length using hidden markov models. In *Intern. Conference on Pattern Recognition*, 1998.

[16] Y. H. Tseng and H. J. Lee. Recognition-based handwritten chinese character segmentation using a probabilistic viterbi algorithm. *Pattern Recognition Letters*, 20(791-806), 1999.

[17] E. Vellasques, L. S. Oliveira, A. S. Britto, A. L. Koerich, and R. Sabourin. Modeling segmentation cuts using support vector machines. In *Intern. Workshop on Frontiers in Handwriting Recognition*, 2006.

[18] D. Yu and H. Yan. Separation of single-touching handwritten numeral strings based on structural features. *Pattern Recognition*, 31(12):1835–1847, 1998.