A Study of Feature Design for Online Handwritten Chinese Character Recognition based on Continuous-Density Hidden Markov Models

Lei Ma, Qiang Huo, Yu Shi
Microsoft Research Asia, Beijing, China
(E-mails: {lema,qianghuo,yushi}@microsoft.com)

Abstract

We present a new feature extraction approach to online Chinese handwriting recognition based on continuous-density hidden Markov models (CDHMM). Given an online handwriting sample, a sequence of time-ordered dominant points are extracted first, which include stroke-endings, points corresponding to local extrema of curvature, and points with a large distance to the chords formed by pairs of previously identified neighboring dominant points. Then, at each dominant point, a 6-dimensional feature vector is extracted, which consists of two coordinate features, two delta features, and two double-delta features. Its effectiveness has been confirmed by experiments for a recognition task with a vocabulary of 9119 Chinese characters and CDHMMs trained from about 10 million samples using both maximum likelihood and discriminative training criteria.

1. Introduction

In order to improve the performance of an HMM-based handwriting recognition system for East Asian (EA) languages such as Chinese, Japanese, and Korean (CJK), the quest for more effective features has never stopped. For a given online handwriting sample represented as a temporal sequence of points, $(P_0, P_1, \cdots, P_t, \cdots, P_T)$, where $P_t = (x_t, y_t)$ is the coordinates of the tth point, following traditional feature extraction approaches have been tried before for CJK handwriting recognition based on continuous-density hidden Markov models (CDHMMs):

- $O_t = (\Delta x_t, \Delta y_t)^{Tr}$, where $\Delta x_t = x_t - x_{t-1}$, $\Delta y_t = y_t - y_{t-1}$ (e.g., [12, 8]);
- $O_t = (r_t, \theta_t)^{Tr}$ where $r_t = \sqrt{(\Delta x_t)^2 + (\Delta y_t)^2}$, $\theta_t = \arctan \frac{\Delta y_t}{\Delta x_t}$ (e.g., [14]); or $O_t = \theta_t$ for equal-arc-length re-sampled sample (i.e., r_t is a constant);
- $O_t = (x_t, y_t, r_t, \theta_t)^{Tr}$ (e.g., [14]); or $O_t = (x_t, y_t, \theta_t)^{Tr}$ for equal-arc-length re-sampled sample.

A lesser-known feature extraction approach was reported in [3, 4]. For this approach, the raw "ink" is segmented first into a sequence of line segments. The segmentation is based on the local maximum of directional angle changes in a stroke. For each segment, the coordinate difference between its ending and starting points is used to form a 2-dimensional feature vector. Consequently, the number of feature vectors extracted from each handwriting sample is much less than that of the aforementioned more traditional approaches, which leads to a much more efficient recognizer. Interestingly, we observed in a preliminary study that the recognition accuracy based on this "line segment" feature is also higher than that based on the traditional feature vector $O_t = (\Delta x_t, \Delta y_t)^{Tr}$ extracted from the fine-trajectory of a handwriting sample.

Encouraged by the promising results reported in [3, 4] and inspired by the traditional idea of using both pen-direction and pen-coordinate features (e.g., [14]), in this paper, we propose a new approach to extracting features at each dominant point (to be explained later) of an online handwriting sample, which consists of two coordinate features, two delta features, and two double-delta features. In terms of using additional coordinate features and double-delta features, our approach can be viewed as an extension of [3, 4]. Because we extract both delta and coordinate features at each dominant point, this makes our approach different from all the previous approaches which extract the relevant features from the fine-trajectory of a handwriting sample, including a recent work reported in [10], where pen-coordinate features are extracted and used only at the ending points of a line segment yet the pen-direction features are extracted from other sampling points of the fine-trajectory of a handwriting sample. The hybrid approach in [10] necessitates the modification of HMM modeling techniques, while our approach extracts much less feature vectors which can be dealt with by standard CDHMM modeling, training, and recognition techniques.

The rest of this paper is organized as follows. In section 2, we present details of our feature extraction approach as well as the corresponding techniques for CDHMM model-

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.24

ing, training, and model compression. In section 3 we report experimental results to demonstrate the effectiveness of our approach. Finally we conclude the paper in section 4.

2. Our Approach

2.1. Preprocessing

Given the captured raw "ink" of an online handwritten character, it is first normalized to a 256×256 sample using an aspect-ratio preserving linear mapping. Then, for each stroke, any point (except for ending points) which has a distance less than 3 to the previous point is treated as redundant and is removed accordingly. If the number of points in a stroke is less than 3 and the length of the stroke is less than 15, this stroke is treated as an artifact therefore is also removed. Given the processed "ink", a procedure adapted from [9] is used to detect a sequence of time-ordered *dominant points* which include

- stroke-endings,
- any point where the trajectory direction changes more than 60 degrees, and
- any point which has the large enough maximum distance to the chord formed by the pair of previously identified neighboring dominant points.

Several heuristic rules [7] are further used to obtain a refined set of dominant points, from which a sequence of feature vectors are extracted as described in the following subsection.

2.2. Feature Extraction

For notational simplicity, we reuse the notation, $(P_1, P_2, \cdots, P_t, \cdots, P_T)$, to denote the sequence of time-ordered dominant points extracted from an online handwriting sample, where $P_t = (x_t, y_t)$ is the coordinates of the tth dominant point. At each dominant point, the following four types of feature vector are extracted and compared in experiments:

- F_D: $O_t = (\Delta x_t, \Delta y_t)^{Tr}$, where $\Delta x_t = x_t - x_{t-1}$ and $\Delta y_t = y_t - y_{t-1}$ are called "delta" features;
- F_{DA}: $O_t = (\Delta x_t, \Delta y_t, \Delta^2 x_t, \Delta^2 y_t)^{Tr}$, where $\Delta^2 x_t = \Delta x_t - \Delta x_{t-1}$ and $\Delta^2 y_t = \Delta y_t - \Delta y_{t-1}$ are called "double-delta" or "acceleration" features;
- F_{CD}: $O_t = (x_t, y_t, \Delta x_t, \Delta y_t)^{Tr}$, where x_t and y_t are called "coordinate" features;
- F_{CDA}: $O_t = (x_t, y_t, \Delta x_t, \Delta y_t, \Delta^2 x_t, \Delta^2 y_t)^{Tr}$.

In calculating "delta" and "double-delta" features, some modifications are needed to deal with "boundary" problem as follows:

- For F_D features, we only use $T-1$ feature vectors $O_2, O_3, \cdots, O_T$;
- For F_{DA} features, we also only use $T-1$ feature vectors $O_2, O_3, \cdots, O_T$, where $O_2 = (\Delta x_2, \Delta y_2, 0, 0)^{Tr}$;
- For F_{CD} features, we use T feature vectors $O_1, O_2, \cdots, O_T$, where $O_1 = (x_1, y_1, 0, 0)^{Tr}$;
- For F_{CDA} features, we also use T feature vectors $O_1, O_2, \cdots, O_T$, where $O_1 = (x_1, y_1, 0, 0, 0, 0)^{Tr}$, and $O_2 = (x_2, y_2, \Delta x_2, \Delta y_2, 0, 0)^{Tr}$.

It is noted that the above coordinate, delta, and double-delta features have comparable dynamic ranges, which makes jobs of modeling, training, model compression (e.g., [5]), and fixed-point implementation of recognizers, relatively easy and straightforward. If directional feature, $\theta_t = \arctan \frac{\Delta y_t}{\Delta x_t}$, was used, the story would be more complicated.

2.3. CDHMM-based Character Modeling

Although other options exist (e.g., [12, 3, 14]), we use CDHMM to model the whole character directly for simplicity. Assume that there are M character classes, C_i, $i = 1, 2, ..., M$, each is modeled by a left-to-right CDHMM allowing state transitions of skipping one state and having mixture of Gaussians with diagonal covariance matrices as probability density function (PDF) for each state. We did not use multiple CDHMMs to deal with explicitly the variability of writing orders of strokes, because it can be addressed more effectively by using a classifier (e.g., MQDF-based classifier [11]) with features (e.g., [1]) insensitive to the aforementioned variability. The latest development for such type of classifiers is reported in [15]. A state-of-the-art online CJK handwriting recognition system typically consists of at least these two types of classifiers.

Let λ_i denote the set of CDHMM parameters for class C_i. The number of HMM states for λ_i is set as the median value of the numbers of feature vectors per character sample calculated over the set of training samples for class C_i. For simplicity, we use the same number, K, of Gaussians for each HMM state.

Let $\mathcal{O} = \{(\mathbf{O}_r, i_r) | r = 1, \cdots, R\}$ denote the set of training samples, where $\mathbf{O}_r$ is the r-th training sample and i_r denotes the index of its true class label. Given $\mathcal{O}$, the set of CDHMM parameters, $\Lambda = \{\lambda_i | i = 1, \cdots, M\}$, is first estimated by using ML training as implemented in HTK toolkit [16]. Starting from well-trained ML models, Λ can be further refined by discriminative training to maximize the following objective function:

$$F(\Lambda) = \frac{1}{R}\sum_{r=1}^{R} \log \frac{p(\mathbf{O}_r|\lambda_{i_r})^{\kappa}}{\sum_{j=1}^{M} p(\mathbf{O}_r|\lambda_j)^{\kappa}} , \tag{1}$$

which is known in speech recognition community as maximum mutual information (MMI) or conditional maximum likelihood (CML) training using uniform unigram as language model (e.g., [13]). However, because of the introduction of the exponential scaling factor κ, the term MMI or CML is not mathematically accurate any more, therefore we prefer to use simply the term discriminative training (DT) in this paper. The version of a so-called extended Baum-Welch (EBW) algorithm as described in [13] is implemented[1] to optimize the above objective function.

In recognition phase, an unknown character sample $\mathbf{O}$ will be classified as class C_i, if

$$i = \arg\max_{j}\{\max_{S} \log p(\mathbf{O}, S|\lambda_j)\} \tag{2}$$

where $p(\mathbf{O}, S|\lambda_j)$ is the joint likelihood of the observation $\mathbf{O}$ and the associated hidden state sequence S given the CDHMM λ_j, and $\max_{S} \log p(\mathbf{O}, S|\lambda_j)$ can be calculated efficiently by using the Viterbi algorithm (e.g., [16]).

2.4. Compression of CDHMM Parameters

In order to reduce memory requirement, CDHMM parameters can be easily compressed by using some well-established techniques without incurring much degradation of recognition accuracy. Transition probabilities can be compressed aggressively using scalar quantization [6]. Mixture coefficients can be compressed using scalar quantization or just be thrown away [5]. Mean vectors and diagonal covariance matrices can be compressed by using a technique known as subspace distribution clustering HMM (SDCHMM) originally proposed in [2]. In our experiments, instead of using *Bhattacharyya distance* to measure the dissimilarity between two Gaussians as described in [2], we use Kullback-Leibler (KL) divergence as suggested in [5] because it is computationally more efficient yet leads to recognizers with similar recognition accuracies.

3. Experiments and Results

3.1. Experimental Setup

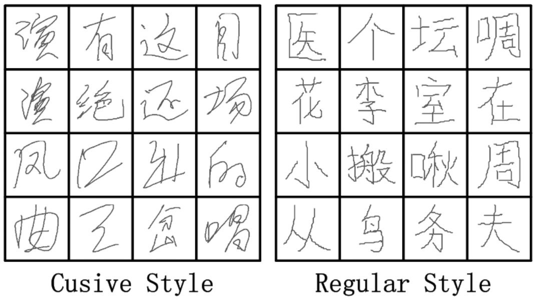

Figure 1. Examples of testing samples with different writing styles.

In order to evaluate and compare the effectiveness of the proposed feature extraction approach, we conduct a series of experiments on the task of the recognition of isolated online handwritten characters with a vocabulary of 9,119 Chinese characters. An in-house developed online handwritten Chinese character corpus is used for our experiments. The training set contains 9,671,951 handwriting samples from the above 9,119 character classes. The number of training samples for each character class differs significantly, ranging from 18 to 5,679. The testing set contains 614,369 handwriting samples from 6763 character classes, which is further divided into two subsets according to the writing style (*cursive* or *regular*) of each testing sample as follows:

- **Cursive**: 431,546 cursive handwriting samples from 3863 character classes;
- **Regular**: 182,823 regular handwriting samples from 6763 character classes.

Fig. 1 gives some examples of testing samples selected randomly from the above two testing subsets with different writing styles.

In all the experiments, we use 4 (i.e., $K = 4$) Gaussians for each CDHMM state.

3.2. Effectiveness of Different Features

The first set of experiments is designed to compare recognition accuracies (in %) of four character classifiers with four sets of ML-trained CDHMMs using four types of features described in section 2, respectively. Two rows with a label "ML" in Table 1 summarize the corresponding results. It is observed that our proposed F_{CD} and F_{CDA} features perform much better, especially for cursive samples, than the F_D features proposed originally in [3]. F_{DA} features perform worse than the F_D features.

The second set of experiments is designed to give a similar comparison for DT-trained CDHMMs. Only mean vectors are updated during DT training. To save computations,

[1] Discriminative training experiments reported in this paper were conducted by using a training tool developed originally by our colleague, Dr. Peng Liu at MSRA, whose contribution is gratefully acknowledged.

Table 1. Comparison of recognition accuracies (in %) and footprints (in MB) of different sets of CDHMMs without using model compression. RERR stands for relative error rate reduction (in %) from ML-trained to DT-trained CDHMMs.

Testing Set	Training Method	Types of Features			
		F_D	F_{DA}	F_{CD}	F_{CDA}
Cursive	ML	83.29	82.75	85.74	86.57
	DT	84.12	84.54	87.77	88.82
	RERR	4.97	10.38	14.24	16.75
Regular	ML	96.44	95.80	96.32	96.76
	DT	96.66	96.68	97.25	97.88
	RERR	6.18	20.95	25.27	34.57
Footprint (in MB)		19.68	34.69	33.66	48.90

Table 2. Comparison of recognition accuracies (in %) and footprints (in MB) of different sets of CDHMMs compressed with Method-1. RERR stands for relative error rate reduction (in %) from ML-trained to DT-trained CDHMMs.

Testing Set	Training Method	Types of Features			
		F_D	F_{DA}	F_{CD}	F_{CDA}
Cursive	ML	82.99	82.38	85.20	85.99
	DT	83.85	84.20	87.49	88.58
	RERR	5.06	10.33	15.47	18.49
Regular	ML	96.30	95.57	96.02	96.49
	DT	96.51	96.48	97.16	97.82
	RERR	5.68	20.54	28.64	37.89
Footprint (in MB)		7.74	9.45	9.83	11.62

we actually used top-10 candidates identified by using ML-trained seed models, instead of M classes, for each training sample to calculate the denominator term in Eq. (1). The setting of other control parameters is as follows: $\kappa = \frac{1}{15}$, and $E = 2$, where E is a constant which controls the setting of a "smoothing constant" D for updating the mean vector of each Gaussian component in the EBW algorithm (see [13] for more details). Two rows with a label "DT" in Table 1 summarize the corresponding experimental results. The same observations can be made except that F_{DA} features perform almost the same as that of the F_D features. By comparing DT results with that of ML results, DT brings a relative error rate reduction of about 5% to 35% for different sets of features on two testing sets. The power of DT is clearly demonstrated.

Table 3. Comparison of recognition accuracies (in %) and footprints (in MB) of different sets of CDHMMs compressed with Method-2. RERR stands for relative error rate reduction (in %) from ML-trained to DT-trained CDHMMs.

Testing Set	Training Method	Types of Features			
		F_D	F_{DA}	F_{CD}	F_{CDA}
Cursive	ML	81.87	81.75	84.64	85.67
	DT	83.15	83.83	87.27	88.48
	RERR	7.06	11.40	17.12	19.61
Regular	ML	95.51	95.05	95.57	96.20
	DT	96.02	96.19	97.04	97.71
	RERR	11.36	23.03	33.18	39.74
Footprint (in MB)		4.33	6.04	6.28	8.06

Table 4. Comparison of recognition accuracies (in %) and footprints (in MB) of different sets of CDHMMs compressed with Method-3. RERR stands for relative error rate reduction (in %) from ML-trained to DT-trained CDHMMs.

Testing Set	Training Method	Types of Features			
		F_D	F_{DA}	F_{CD}	F_{CDA}
Cursive	ML	81.87	79.40	82.45	82.45
	DT	83.15	81.52	85.36	84.55
	RERR	7.06	10.29	16.58	11.97
Regular	ML	95.51	94.10	94.81	95.04
	DT	96.02	95.36	96.36	96.35
	RERR	11.36	21.36	29.87	26.41
Footprint (in MB)		4.33	4.33	4.50	4.50

3.3. Effects of Model Compression

The next three sets of experiments are designed to study the effect of model compression using following three methods:

- **Method-1**: Keep uncompressed Gaussian mixture coefficients, and do subspace Gaussian clustering independently for each feature dimension;
- **Method-2**: Ignore Gaussian mixture coefficients, and do subspace Gaussian clustering independently for each feature dimension;
- **Method-3**: Ignore Gaussian mixture coefficients, and do subspace Gaussian clustering for two streams of subvectors detailed as follows:

- Δx_t and Δy_t for F_D feature based CDHMMs;
- $(\Delta x_t, \Delta^2 x_t)^{Tr}$ and $(\Delta y_t, \Delta^2 y_t)^{Tr}$ for F_{DA} feature based CDHMMs;
- $(x_t, \Delta x_t)^{Tr}$ and $(y_t, \Delta y_t)^{Tr}$ for F_{CD} feature based CDHMMs; and
- $(x_t, \Delta x_t, \Delta^2 x_t)^{Tr}$ and $(y_t, \Delta y_t, \Delta^2 y_t)^{Tr}$ for F_{CDA} feature based CDHMMs.

In all the above cases, a separate codebook of 256 Gaussian PDFs is used for each stream of subvectors. For simplicity, transition probabilities are not compressed.

Tables 2 to 4 summarize comparisons of recognition accuracies (in %) and footprints (in MB) of different sets of CDHMMs compressed with the aforementioned three methods respectively. Following observations can be made:

- By comparing results in Table 2 with that in Table 1, Method 1 only incurs very small degradation of recognition accuracy, yet reduces the footprint significantly;
- By comparing results in Table 3 with that in Table 2, ignoring Gaussian mixture coefficients reduces the footprint further yet introduces small degradation of recognition accuracy;
- By comparing results in Table 4 with that in Table 3, significant degradation of recognition accuracy is incurred with further reduction of footprint, especially for F_{CDA} features. A codebook size of 256 may not be enough for 3-dimensional subvectors.

4. Conclusion

Given the above results, we draw the following conclusions:

- Among four types of features compared, F_{CDA} features give the highest recognition accuracy;
- The footprint of CDHMMs compressed with Method-1 or Method-2 is small enough for today's desktop PCs or notebook PCs;
- Using F_{CD} features and Method-3 for model compression achieves a remarkable memory-accuracy trade-off, therefore offers a good solution to designing compact handwriting recognition systems to be deployed on mobile devices for East Asian languages such as Chinese, Japanese, and Korean.

By sharing enough technical details and results with the research community, the above solutions could serve as a baseline for other researchers to compare with, and hopefully this would spur more researches in this exciting research field.

References

[1] Z.-L. Bai and Q. Huo, "A study on the use of 8-directional features for online handwritten Chinese character recognition," *Proc. ICDAR-2005*, pp.262-266.

[2] E. Bocchieri and B. Mak, "Subspace distribution clustering hidden Markov model," *IEEE Trans. on Speech and Audio Processing*, Vol. 9, No. 3, pp.264-275, 2001.

[3] Y. Ge, F.-J. Guo, and L.-X. Zhen, "A compact and flexible online Chinese handwriting recognizer based on hidden Markov models," *Proc. Int. Conf. on Image Science, Systems and Technology*, Las Vegas, Nevada, USA, 2004, pp.187-192.

[4] Y. Ge, F.-J. Guo, L.-X. Zhen, and Q.-S. Chen, "Online Chinese character recognition system with handwritten Pinyin input," *Proc. ICDAR-2005*, pp.1265-1269.

[5] Y. Ge and Q. Huo, "A study on the use of CDHMM for large vocabulary offline recognition of handwritten Chinese characters," *Proc. 8th IWFHR*, 2002, pp.334-338.

[6] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Springer, 1991.

[7] T. He, *A Study on Several Problems in Online Handwritten Chinese Character Recognition*, Ph.D. Thesis, The University of Hong Kong, 2008.

[8] Q. Huo and T. He, "A minimax classification approach to HMM-based online handwritten Chinese character recognition robust against affine distortions," *Proc. ICDAR-2007*, pp.43-47.

[9] K. Ishigaki and T. Morishita, "A top-down online handwritten character recognition method via the denotation of variation," *Proc. of Int. Conf. Computer Processing of Chinese and Oriental Languages*, 1988, pp.141-145.

[10] Y. Katayama, S. Uchida, and H. Sakoe, "A new HMM for on-line character recognition using pen-direction and pen-coordinate features," *Proc. ICPR-2008*.

[11] F. Kimura, K. Takashina, S. Tsuruoka, and Y. Miyake, "Modified quadratic discriminant functions and the application to Chinese character recognition," *IEEE Trans. on PAMI*, vol. 9, pp.149-153, 1987.

[12] M. Nakai, N. Akira, H. Shimodaira, and S. Sagayama, "Substroke approach to HMM-based on-line Kanji handwriting recognition," *Proc. ICDAR-2001*, pp.491-495.

[13] D. Povey, *Discriminative Training for Large Vocabulary Speech Recognition*, Ph.D. Thesis, University of Cambridge, 2004.

[14] J. Tokuno, Y. Yang, G. P. da Silva, A. Kitadai, M. Nakagawa, "Pen-coordinate information modeling by SCPR-based HMM for on-line Japanese handwriting recognition," *Proc. ICPR-2006*, pp.III-348-351.

[15] Y. Wang and Q. Huo, "Design compact recognizers of handwritten Chinese characters using precision constrained Gaussian models, minimum classification error training and parameter compression," *Proc. ICDAR-2009*.

[16] S. Young, *et al.*, *The HTK Book (for HTK Version 2.2)*, 1999.