Improving HMM-Based Chinese Handwriting Recognition Using Delta Features and Synthesized String Samples

Tong-Hua Su, Cheng-Lin Liu
National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences
95 Zhongguancun East Road, Beijing 100190, P.R. China
{thsu, liucl}@nlpr.ia.ac.cn

Abstract

The HMM-based segmentation-free strategy for Chinese handwriting recognition has the advantage of training without annotation of character boundaries. However, the recognition performance has been limited by the small number of string samples. In this paper, we explore two techniques to improve the performance. First, Delta features are added to the static ones for alleviating the conditional independence assumption of HMMs. We then investigate into techniques for synthesizing string samples from isolated character images. We show that synthesizing linguistically natural string samples utilizes isolated samples insufficiently. Instead, we draw character samples without replacement and concatenate them into string images through between-character gaps. Our experimental results demonstrate that both Delta features and synthesized string samples significantly improve the recognition performance. Combining these with a bigram language model, the recognition accuracy has been increased by 36~38% compared to our previous system.

1. Introduction

The powerful tool, hidden Markov model (HMM), has been widely applied to sequence analysis tasks such as speech recognition and handwriting recognition. Equipped with efficient learning and inference algorithms like the Baum-Welch algorithm and the Viterbi algorithm, the HMM scales well to large data set. HMM-based handwriting recognition has the merit that the model parameters can be trained with string samples without annotation of character boundaries. We have employed HMM into Chinese handwriting recognition to build a segmentation-free recognizer [1][2][3], where each character was modeled by a continuous-density HMM. Our previous practice suggests that besides modeling techniques, carefully deriving strong features [2] and reasonably alleviating data sparseness [3] are critical to the recognition performance. However, due to the unavailability of large data set of string samples, the recognition accuracy of our previous system has been limited to a low level.

In this paper, we improve the performance of HMM-based Chinese handwriting recognition by exploring the Delta features and synthesized string samples. The Delta feature was first proposed in speech recognition [4] and is widely adopted, but it has been rarely used in handwriting recognition. We utilize such features to express the local feature slope across neighboring windows for alleviating the conditional independence assumption of HMMs.

Considering the large number of Chinese character categories and the severe sparseness of handwritten string image data, we synthesize string images from existing isolated character samples for HMM training. Synthesizing string samples has been practised in English word recognition. In [5], training samples were expanded by distorting realistic text lines. In [6], individual characters are concatenated using a ligature joining algorithm for a natural-shape handwriting. We investigate into the techniques for synthesizing Chinese string samples from isolated character samples and our emphasis is how to effectively utilize these samples rather than generate linguistically natural handwriting. In light of above principle, we approximate the distribution of between-character gaps and draw character samples without replacement. By combining Delta features and training with synthetic string samples, we have achieved a large improvement of recognition accuracy in Chinese handwriting recognition.

The rest of this paper is organized as follows. Sec-
3. Delta Features

Delta features are equipped in most speech recognition systems for their ability to grasp the temporal dependence. The Delta features are the first order linear regression coefficients derived from static features:

$$
\Delta \mathbf{o}_t = \frac{\sum_{i=-n}^{n} i \cdot \mathbf{o}_{t+i}}{\sum_{i=-n}^{n} i^2} = \frac{\sum_{i=1}^{n} i \cdot (\mathbf{o}_{t+i} - \mathbf{o}_{t-i})}{2 \sum_{i=1}^{n} i^2},
$$

(1)

where \( \mathbf{o}_t \) is the vector of static features. The Delta feature involves \( 2n+1 \) windows for regression. In handwriting images, it reflects the slope in each feature dimension. The derivative features used in \( \delta \) and \( \delta \) can be viewed as special cases of Eq. (1). Delta features are dynamically derived and can express the dependence among consecutive observations. This compensates HMMs for the loss resulting from the conditional independence assumption.

4. Synthesizing String Samples

There have been large databases of isolated Chinese character samples for training classifiers but the number of text line images is very small. Isolated character samples can be directly used to train character HMMs but the collocation between characters cannot be modeled by training with isolated characters. To alleviate the sparseness of text line data, we synthesize string image samples from isolated character images.

String samples are generated through concatenating existing isolated character samples of the same image resolution. If we have already picked \( M \) character samples up, then the task is to concatenate them into a string image whose purpose is to effectively utilize isolated character samples. To synthesize a text line, thus the following problems should be solved: how to draw a character class, how to select a sample for a certain class, how to join a sequence of character samples together. Since the samples in each class often are assumed of equal importance, we draw samples uniformly providing that the character class is known.
Character samples are concatenated together through between-character gaps. Gap between two samples are randomly generated from a gap histogram. To approximate the gaps, we first mark each left and right boundaries of characters in the training text lines. Then the difference between the left position of current character and the right position of previous character is collected to derive a gap histogram. If character overlapping occurs, the gap may be negative. The accumulative gap histogram is illustrated in Fig. 2. The binary search is used to locate the gap corresponding to a random number.

The general algorithm to produce a text line is shown in Alg. 1. Step 2∼Step 6 recursively draw samples. All selected class names and samples in current text line are temporarily recorded. Step 4 and 5 express a stratified sampling paradigm. We can first draw a class from a lexicon, then we dispatch a sample instance to the class. Each class may associate a weight, indicating its probability to be sampled in Step 4. We may consider uniform (zero-gram), unigram and bigram statistics in Sect. 5 respectively. Step 7 and 8 normalize current text line to a standard height. Step 9 concatenates a sample sequence together with random gaps. The \( x_{kj} \)s are aligned in vertical middle line. The last two steps of the algorithm produce feature observations and corresponding ground truth. Alg. 1 can be called as many times as needed or till there leaves no available isolated character samples.

Also, sampling can be done with replacement or without replacement. In our system, isolated character samples are drawn without replacement. This kind of sampling deliberately avoids choosing any sample more than once. If all samples of a character class are exhausted, the class will never be picked up again in the following steps. It stops when samples of all classes are exhausted. Unlike, random sampling can be done with replacement, however, even in a Bootstrap manner, only 63.21% of samples can be picked up.

## 5. Experiments

Proposed methods are evaluated on a test set from HIT-MW database [10]. Six Gaussian components are used in our systems for the state emission probability. The experimental setup is the same with [2] and [3], unless declared explicitly.

The database and the criteria for performance evaluation are presented in subsection 5.1. The next subsection evaluates the renewed baseline system. Finally, experiments on Delta features and the algorithm for synthesizing string samples are conducted in subsection 5.3 and 5.4 respectively.

### 5.1. Database and evaluation criteria

The HIT-MW database can be seen as a representative subset of real Chinese handwriting [10]. It includes 853 legible Chinese handwriting samples from more than 780 participants with a systematic way. Portions of text lines are partitioned into training set (953 text lines), validation set (189 text lines), and test set (383 text lines) according to random sampling theory to reproduce a realistic scenario where the handwritten text lines for recognition in test set have not been seen before. These text lines are realistic samples. Since Chinese character recognition involves a large categories,
many classes associate few samples. Thus we also synthesize text lines to alleviate the severe data sparseness (Cf. [3]). This paper utilizes 100 sets of isolated character samples in CASIA database [1].

The output of certain recognizer is compared with the reference transcription and two metrics, the correct rate (CR) and accurate rate (AR), are calculated to evaluate the results. Supposing the number of substitution errors \( S_e \), deletion errors \( D_e \), and insertion errors \( I_e \) are known, CR and AR are defined respectively as:

\[
\begin{align*}
CR &= \frac{N_t - D_e - S_e}{N_t} \\
AR &= \frac{N_t - D_e - S_e - I_e}{N_t}
\end{align*}
\]

where \( N_t \) is the total characters in the reference transcription.

5.2. Evaluation of renewed baseline

Techniques used to renew the baseline are evaluated and results are listed in Table 4. Seen from the table, VT increases the CR/AR by 8~10 percentages, BCT receives 1.1~1.7% improvements and HN with about 1% increases. Renewed baseline yields at least 11% improvements than previous results totally. Note that the results of [2] presented here have slight differences with that of the original paper, since the training iterations are fixed in this paper instead of tuning up on validation set.

5.3. Evaluation of Delta features

We merely append (first order) Delta features for the demonstration purpose. The static features are enFPF features. The parameter \( n \) in Eq. [1] is set as 8 through validation. Compared with the baseline, a steady improvement is observed with about 3% (see Table 2). Further investigation shows that Delta features facilitate classes of large samples, without loss in those of small samples.

5.4. Evaluation of synthetic string samples

We evaluate the efficacy of synthetic string samples through recognition results. Before retraining the HMMs three passes, we append the synthetic string samples and reduce the variance truncation of Gaussian mixture density by half. The recognition results are given in Table 3 and Fig. 3. First, the effect of a bigram language model is considered. The bigram statistics are derived from People’s Daily corpus with 79,509,778 characters. Second, the sampling methods, with replacement and without replacement are evaluated separately. The sampling with replacement draws the same number of character samples as without replacement. Third, the process of drawing samples is guided by three kinds of linguistic contexts. More strong the linguistic contexts, more linguistically natural the synthetic text line looks. In addition, the results produced by character training are also provided. Recognition results of a text line are given in Fig. 4.

From Table 3 and Fig. 3 we simply highlight following three points. First, the sampling without replacement performs better. During sampling with replacement, if the underlying text is generated by bigram statistics, the synthetic image looks linguistically natural. However, the parameters of HMMs can hardly be robustly learned, since samples of some classes may be chosen multiple times, while portions of samples never be accessed (Cf. Fig. 3(a) and Fig. 3(b)).

Second, using synthetic text lines has clear advantages than character training to learn HMM parameters when recognition couples a bigram language model. The most fundamental distinction between text line training and character training is that the adjacent characters in a text line compete with each other for parameter estimation of their classes. Such competitions benefit postprocessing with a bigram language model.

Third, as for without replacement strategy, the uniform, unigram and bigram-guided sampling techniques perform comparably well. Though sampling can be guided by linguistic statistics, the underlying text of the synthetic images is biased much to uniform distribution, since the same sets of isolated characters are used. Further inspection (Cf. Fig. 3(c) and Fig. 3(d)) shows that uniform-guided technique performs slight better if bigram language model is equipped with recognition, while unigram and bigram-guided techniques are slight better when recognition without bigram language model. As for uniform-guided technique, a character class can follow any character classes, thus each character HMM has the equal chance to compete with each other. Unlikely, the character classes of high frequency are picked up with large chances using unigram or bigram-guided technique and their samples are exhausted in earlier stage. As a consequence, the competitions between the classes of high frequency and those of low frequency are much limited.

How about fixed between-character gap models instead of histogram one? We evaluate two kinds of fixed gaps: 1 pixel and one fifth of stdH (16 pixels). Character sampling is done uniformly without replacement. The average recognition rates are 70.77%/65.22%, 73.01%/69.51% (CR/AR) respectively. Histogram gap yields at least 0.88% improvement. From Fig. 3(c) and Fig. 3(d), synthesizing string samples using fixed gaps shares some properties with character training.
There exist a few related works in offline recognition of Chinese handwriting. Actually, they aren’t comparable, since they are not trained on the same data or they are developed from different motivations, and even they are tested on different test sets. As references to readers, we list them below. In [8], BBN recognition system is presented based on HMM, where the best AR metric is less than 38% on their in-house Chinese handwriting databases. In [12], segmentation-recognition-integrated strategy is presented. It recognizes 58% of the characters without language models, and the best recognition rate is about 78% after incorporating the scores from a restricted Gaussian classifier and language models.

### 6. Conclusion

Two techniques are presented to improve the Chinese handwriting recognition system. First of all, Delta features and static enFPF ones are incorporated together to compensate for the conditional independence assumption in HMM modeling. We further investigate techniques to synthesize string samples for alleviating data sparseness. Character samples are drawn without replacement and histogram of between-character gaps is utilized to join character samples. The recognition performance of Chinese handwriting is significantly improved after integrating these strategies.

### References


Figure 3. Chinese character CR over different sample sizes: (a) sample drawing with replacement & recognition without bigram; (b) sample drawing with replacement & recognition with bigram; (c) sample drawing without replacement & recognition without bigram, (d) sample drawing without replacement & recognition with bigram.

Table 3. Evaluation of synthesized methods (%).

<table>
<thead>
<tr>
<th>Draw Methods</th>
<th>Digit CR</th>
<th>Digit AR</th>
<th>Punctuation CR</th>
<th>Punctuation AR</th>
<th>Chinese character CR</th>
<th>Chinese character AR</th>
<th>Average CR</th>
<th>Average AR</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Recognition without bigram language model</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Uniform &amp; with rep</td>
<td>59.13</td>
<td>50.00</td>
<td>41.92</td>
<td>39.27</td>
<td>61.51</td>
<td>58.43</td>
<td>59.55</td>
<td>56.36</td>
</tr>
<tr>
<td>Unigram &amp; with rep</td>
<td>60.00</td>
<td>47.39</td>
<td>41.79</td>
<td>40.03</td>
<td>63.97</td>
<td>61.06</td>
<td>61.38</td>
<td>58.07</td>
</tr>
<tr>
<td>Bigram &amp; with rep</td>
<td>58.26</td>
<td>50.43</td>
<td>37.50</td>
<td>35.86</td>
<td>61.29</td>
<td>58.58</td>
<td>58.81</td>
<td>56.18</td>
</tr>
<tr>
<td>Uniform &amp; without rep</td>
<td>56.09</td>
<td>47.39</td>
<td>41.29</td>
<td>39.39</td>
<td>64.10</td>
<td>61.45</td>
<td>61.73</td>
<td><strong>58.99</strong></td>
</tr>
<tr>
<td>Unigram &amp; without rep</td>
<td>57.39</td>
<td>48.70</td>
<td>41.29</td>
<td>39.39</td>
<td>64.10</td>
<td>61.45</td>
<td>61.73</td>
<td><strong>58.99</strong></td>
</tr>
<tr>
<td>Bigram &amp; without rep</td>
<td>55.22</td>
<td>46.96</td>
<td>39.02</td>
<td>37.37</td>
<td><strong>64.18</strong></td>
<td><strong>61.52</strong></td>
<td>61.52</td>
<td>58.81</td>
</tr>
<tr>
<td>Char training</td>
<td>62.17</td>
<td>47.39</td>
<td><strong>47.85</strong></td>
<td><strong>41.41</strong></td>
<td>63.54</td>
<td>60.38</td>
<td>61.97</td>
<td>58.20</td>
</tr>
<tr>
<td><strong>Recognition with bigram language model</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Uniform &amp; with rep</td>
<td>70.44</td>
<td>65.22</td>
<td>59.97</td>
<td>49.49</td>
<td>72.31</td>
<td>69.62</td>
<td>71.04</td>
<td>67.56</td>
</tr>
<tr>
<td>Unigram &amp; with rep</td>
<td><strong>72.61</strong></td>
<td><strong>69.13</strong></td>
<td>57.95</td>
<td>48.48</td>
<td>72.73</td>
<td>70.29</td>
<td>71.27</td>
<td>68.16</td>
</tr>
<tr>
<td>Bigram &amp; with rep</td>
<td>70.00</td>
<td>67.39</td>
<td>57.20</td>
<td>49.24</td>
<td>72.90</td>
<td>70.42</td>
<td>71.29</td>
<td>68.29</td>
</tr>
<tr>
<td>Uniform &amp; without rep</td>
<td>71.74</td>
<td>66.52</td>
<td>60.99</td>
<td>51.77</td>
<td><strong>75.41</strong></td>
<td><strong>73.05</strong></td>
<td><strong>73.89</strong></td>
<td><strong>70.81</strong></td>
</tr>
<tr>
<td>Unigram &amp; without rep</td>
<td>70.87</td>
<td>65.65</td>
<td>60.98</td>
<td><strong>52.02</strong></td>
<td>75.20</td>
<td>72.89</td>
<td>73.68</td>
<td>70.67</td>
</tr>
<tr>
<td>Bigram &amp; without rep</td>
<td>71.74</td>
<td>66.52</td>
<td>60.35</td>
<td>50.88</td>
<td>75.12</td>
<td>72.78</td>
<td>73.57</td>
<td>70.49</td>
</tr>
<tr>
<td>Char training</td>
<td>70.43</td>
<td>64.35</td>
<td><strong>62.37</strong></td>
<td>41.29</td>
<td>68.80</td>
<td>65.93</td>
<td>68.18</td>
<td>63.52</td>
</tr>
</tbody>
</table>

Figure 4. Recognition results coupling a bigram: (a) isolated character training; (b) synthetic text line training (uniform sampling, without replacement). The correct outputs are underlined.