A Discrete Contextual Stochastic Model for the Offline Recognition of Handwritten Chinese Characters

Yan Xiong, Qiang Huo, *Member*, *IEEE*, and Chorkin Chan

Abstract—We study a discrete contextual stochastic (CS) model for complex and variant patterns like handwritten Chinese characters. Three fundamental problems of using CS models for character recognition are discussed and several practical techniques for solving these problems are investigated. A formulation for discriminative training of CS model parameters is also introduced and its practical usage investigated. To illustrate the characteristics of the various algorithms, comparative experiments are performed on a recognition task with a vocabulary consisting of 50 pairs of highly similar handwritten Chinese characters. The experimental results confirm the effectiveness of the discriminative training for improving recognition performance.

Index Terms—Offline recognition of handwritten Chinese characters, contextual stochastic model, discriminative training, Markov random field.

1 Introduction

Offline recognition of handwritten Chinese characters is known as a challenging pattern recognition problem, mainly because of the large vocabulary size, complex character shapes, many confusable subsets of characters with slightly different shapes, and great variations of character samples written by the same or different writers. In the past several decades, a great deal of efforts have been made towards solving this problem (e.g., [1], [2], [3], [4], [5], and the references therein). A statistical pattern recognition approach (including artificial neural network) remains one of the most popular approaches being adopted to construct character recognizers in practice. Many recognition systems adopt a maximum discriminant function-based approach (e.g., [6]) with some simple discriminant functions derived from the metrics such as the Euclidean distance, the Mahalanobis distance, the posterior probability of a hypothesized class given the pattern to be recognized, etc. Inspired by the success of hidden Markov model methodology (HMM) in the automatic speech recognition (ASR) field (e.g., [7]), in the past decade, there have been also many research efforts that use HMM in both online handwriting recognition and the offline character recognition. Although quite encouraging results have been achieved by using these approaches in various character recognition tasks, offline recognition of handwritten Chinese characters remains largely an unsolved problem that requests more researches. This fact encourages us to explore new modeling techniques for attacking this difficult problem. Among many possibilities, one direction to pursue is to appropriately model the contextual information of a character.

- *Y. Xiong is with Hewlett-Packard Laboratories, 1501 page Mill Road, MS 1L-15, Palo Alto, CA 94304-1126. E-mail: yan_xiong@hp.com.*
- *Q. Huo is with the Department of Computer Science and Information Systems, The University of Hong Kong, Pokfulam Road, Hong Kong. E-mail: qhuo@csis.hku.hk.*
- *C. Chan was with the Department of Computer Science and Information Systems, The University of Hong Kong, Pokfulam Road, Hong Kong. He is now retired.*

Manuscript received 14 Jan. 1999; revised 19 Dec. 2000; accepted 13 Jan. 2001.
Recommended for acceptance by R. Plamonodon.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 108971.

Contextual or, more generally, structural information manifests itself under a wide variety of guises. One possibility is to use a *syntactic pattern recognition* approach which identifies the strokes of each character template and recognizes an unknown image by matching it against every template by means of stroke matching. However, it is well-known that strokes are difficult to identify robustly. Another possibility is to adopt a *statistical pattern recognition* approach as people currently do in ASR area. A number of statistical approaches aimed at exploiting the contextual information had appeared in the pattern recognition literature (e.g., [8], [9], [10]). In the past two decades, a significant amount of research activities in image modeling with a special emphasis on Hidden Markov Random Field (HMRF) which can be viewed as a 2D counterpart of HMM in modeling 2D signals like images (e.g., [11], [12], [13], [14], [15], [16], [17], and the references therein) has been witnessed. The HMRF interprets the observed image as a partly observed version of a complete data and can be used to model and exploit the contextual information in an image. The MRF technique has been applied to many applications such as texture analysis, image restoration, classification, and segmentation. Some researchers in the OCR (optical character recognition) field adopted causal HMRFs for character modeling (e.g., [18], [19]) where an asymmetric *local dependence* structure of the hidden states (regions) is assumed as in causal MRFs (e.g., [17]). Motivated by the developments of MRF techniques in the image processing field, we also became interested in MRF theory and tried to use HMRF in some pattern recognition applications. From the very first beginning, we intend to model the *local dependence* structure of the hidden states in a symmetric manner. At the same time, we want to develop computationally efficient algorithms so that they can be applied to practical pattern recognition problems. This prevents us from directly using the relevant techniques in the MRF literature and we need to develop our own solution. Consequently, several simplified models which take into account *heuristically* the local contextual dependence information were developed and first applied to speech recognition application [20], [21]. Interestingly, not much performance improvement was achieved in comparison with conventional HMM techniques. One possible explanation could be that for a temporal signal-like speech, HMM with an assumption of a casual local dependence structure for hidden states, might be good enough to capture the contextual dependence information. Considering the spatial nature of character images, it is natural to believe that omnidirectional information of spatially contextual dependence might be useful for character recognition if a model similar to HMRF is used to model a character. One of the techniques, called contextual vector quantization (CVQ), developed in [20], [21] was then applied to handwritten Chinese character recognition [22]. The same technique was later employed in implementing an offline recognizer supporting a vocabulary of 4,616 Chinese characters as well as alphanumeric and punctuation symbols, where a similar CVQ technique is also used for language modeling [23]. In order to avoid the possible confusion caused by using the term VQ which usually has slightly different meanings in different research areas, from now on, a new term, namely, *contextual stochastic* (CS) model is adopted to refer to this technique.

Although strongly motivated by the MRF theory, strictly speaking, our model cannot be called an HMRF model because we are not following the theoretical rigor implied in the MRF theory due to the practical difficulties arisen from the assumption of the noncasual *local dependence* structure of the hidden states. Instead, we are using the principle of *maximum discriminant decision rule* [6] to guide us to design our character model and the *discriminant function*. In this paper, we shall discuss in detail how the idea of contextual stochastic modeling for Chinese character recognition is formulated and developed under

this framework. Several new training and recognition algorithms will also be introduced and investigated.

2 CS Models for the Offline Recognition of Handwritten Chinese Characters

The discrete CS modeling framework for character recognition can be outlined as follows: Given an original binary image of a character, each character can be abstracted into a matrix of feature vectors $\mathbf{O} = [\mathbf{o}_{i,j}]$ with $\mathbf{o}_{i,j}$ being (i,j)th feature vector. The meaning of $o_{i,j}$ depends on what feature extraction method and sampling scheme are used. In [22], [23], and this study as well, five features at each pixel are extracted to form a feature vector. So, $o_{i,j}$ represents a 5D feature vector indexed by (i,j) (also called site (i,j)) which happens to have the same index as the pixel in the original character image. More specifically, the features employed are the so-called cellular features [1], namely, the total number of contiguous black pixels along the upwards and downwards directions from the pixel under consideration and the number of strokes encountered along the upwards, downwards, left, and right directions from the pixel to the image boundary. One can imagine that each character can be partitioned into a set of regions. The concept of regions here is quite general which can correspond to any intrinsic structure of a character under certain criterion. Unlike strokes, there are many ways of partitioning a character into such-defined regions. $\mathbf{Z} = [z_{i,j}]$ represents a region map, where $z_{i,j}$ can take one of K qualitative values $\{G_1, G_2, \cdots, G_K\}$ each of which corresponds to a unique region label. The structural information of a character image in terms of the contextual relationship between its regions can be *characterized* statistically by using the overall prior region distributions and the directional region to region conditional probability distributions as detailed in the following: One can imagine that for each unique region, there is a unique distribution for observed feature vectors belonging to the region. So, one can view $\mathbf{o}_{i,j}$ as a realization of a random vector observable in a region with label $z_{i,j}$. Because one cannot observe $\{z_{i,j}\}$ directly, regions are referred to as being *hidden*. For the readers who are familiar with the HMM framework, it becomes clear that the *hidden region* concept here corresponds to the *hidden state* concept in HMM. The terms *region* and *state* will be used interchangeably hereinafter. Elements of a CS model can now be formally defined.

A CS model is characterized by the following:

1. K is the number of regions in the model. The collection of all regions are denoted as $G = \{G_1, G_2, \cdots, G_K\}$ and the region at site (i,j) is denoted as $z_{i,j}$.
2. T is the number of distinct observation symbols, i.e., the discrete alphabet size. Individual symbols are denoted as $V = \{v_1, \ldots, v_T\}$ so that any $\mathbf{o}_{i,j}$ is one such observation symbol.
3. $\pi = \{\pi_k = Pr(z_{i,j} = G_k)\}$ is a collection of prior region distributions, where π_k measures the relative size in terms of the number of feature vectors belonging to a region labeled as G_k.
4. $\mathbf{A} = \{a_{kl}^{m,n} = Pr^{m,n}(z_{i+m,j+n} = G_l | z_{i,j} = G_k)\}$ is the set of directional region to region conditional probability distributions which supply the contextual information between a feature vector and its immediate 4-neighbors. Here, $(m,n) \in \{(-1,0),(1,0),\ (0,-1),(0,1)\}$, each of which corresponds to one of the four directions: up, down, left, and right, respectively. For each direction (m,n), $a_{kl}^{m,n}$ denotes the conditional probability from region G_k to region G_l along that direction.
5. $\mathbf{B} = \{b_{k,t} = b_k(v_t) = Pr(o_{i,j}{=}v_t | z_{i,j} = G_k)\}$ is a collection of region output probability distributions. For each region, there is a discrete output probability distribution. $b_{k,t}$ refers to the probability of observing discrete symbol v_t at a site in region G_k. This is why the current model is named *discrete CS model*.

For convenience, the compact notation $\lambda = (\pi, \mathbf{A}, \mathbf{B})$ is used to indicate the complete parameter set of the model. In order to use this model for character recognition, three basic problems similar to what were discussed in HMM (e.g., [7]), must be solved. They are:

Problem 1. *Given an observation matrix $\mathbf{O} = [\mathbf{o}_{i,j}]$ and a model λ, how can one choose a corresponding region map $\mathbf{Z} = [z_{i,j}]$ which is optimal in some meaningful sense? It will be seen later that the solution of this problem is important in answering the following two questions as well.*

Problem 2. *Given an observation matrix $\mathbf{O} = [\mathbf{o}_{i,j}]$ and a model λ, how can one appropriately define and efficiently compute a similarity measure $g(\mathbf{O}, \lambda)$ between the observation $\mathbf{O}$ and the given model λ so that it can be used to serve as a discriminant function for recognition purposes with a maximum discriminant decision rule?*

Problem 3. *Given a set of training observations $\{\mathbf{O}\}$, how can one estimate the model parameters λ under certain meaningful criteria?*

In order to develop a methodology of region labeling and parameter estimation for a CS model, the following two assumptions are made:

Assumption 1. *Let $\eta_{i,j}$ be the immediate 4-neighbor neighborhood of the site (i,j) and $\eta_{i,j}^{+}$ be the union of $\eta_{i,j}$ and (i,j). Given $z_{\eta_{i,j}^{+}}$, the random feature vectors $\mathbf{o}_{i',j'}$s, with $(i',j') \in \eta_{i,j}^{+}$, are conditionally independent. That is,*

$$Pr(\mathbf{o}_{i,j}, \mathbf{o}_{\eta_{i,j}} | z_{i,j}, z_{\eta_{i,j}}) = \prod_{(i',j') \in \eta_{i,j}^{+}} Pr(\mathbf{o}_{i',j'} | z_{i',j'}). \tag{1}$$

Assumption 2. *Given $z_{i,j}$ and the $\eta_{i,j}$ defined as above, the $z_{i',j'}$s, with $(i',j') \in \eta_{i,j}$, are conditionally independent. That is,*

$$Pr(z_{\eta_{i,j}} | z_{i,j}) = \prod_{(i',j') \in \eta_{i,j}} Pr^{m,n}(z_{i',j'} | z_{i,j}), \tag{2}$$

where $m = i' - i$ and $n = j' - j$.

This assumption was made to simplify the relevant derivations to be discussed in the next section. Readers are referred to [8] for discussions on its heuristic justification.

In the next section, some solutions to the above three fundamental problems of CS modeling will be presented. One will see that the three problems are linked together tightly under a probabilistic framework.

3 Solutions to the Three Basic Problems of CS Models

3.1 Solution to Problem 1

Ignoring any context, observation $\mathbf{o}_{i,j}$ can be classified to the region G_k by choosing

$$G_k = \underset{z_{i,j}}{\operatorname{argmax}}\, Pr(z_{i,j} | \mathbf{o}_{i,j}). \tag{3}$$

If one wants to take contextual information into account in labeling an observed feature vector, depending on the criterion used, one has many ways to perform contextual labeling.

3.1.1 Individual Contextual Labeling

As discussed in [20], one way to obtain an optimal region map for an observed $\mathbf{O}$ is to individually consider the labeling of the observed feature vector on the basis of its posterior probabilities of group membership given all the observation vectors in the character. The feature vector $\mathbf{o}_{i,j}$ is then labeled as

$$G_k = \underset{z_{i,j}}{\operatorname{argmax}} Pr(z_{i,j}|\mathbf{O}). \tag{4}$$

However, such a scheme would be too difficult to implement. In order to reduce the complexity of the problem, we further *assume* that $\mathbf{o}_{i,j}$ might be labeled to maximize $Pr(z_{i,j}|\mathbf{o}_{i,j}, \mathbf{o}_{\eta_{i,j}})$, i.e.,

$$\begin{aligned} G_k &= \underset{z_{i,j}}{\operatorname{argmax}} Pr(z_{i,j}|\mathbf{o}_{i,j}, \mathbf{o}_{\eta_{i,j}}) \\ &= \underset{z_{i,j}}{\operatorname{argmax}} Pr(z_{i,j}, \mathbf{o}_{i,j}, \mathbf{o}_{\eta_{i,j}}). \end{aligned} \tag{5}$$

Note that

$$Pr(z_{i,j}, \mathbf{o}_{i,j}, \mathbf{o}_{\eta_{i,j}}) = \sum_{z_{\eta_{i,j}}} Pr(z_{i,j}, z_{\eta_{i,j}}, \mathbf{o}_{i,j}, \mathbf{o}_{\eta_{i,j}})$$

and

$$Pr(z_{i,j}, z_{\eta_{i,j}}, \mathbf{o}_{i,j}, \mathbf{o}_{\eta_{i,j}}) = Pr(\mathbf{o}_{i,j}, \mathbf{o}_{\eta_{i,j}}|z_{i,j}, z_{\eta_{i,j}}) \cdot Pr(z_{\eta_{i,j}}|z_{i,j}) \cdot Pr(z_{i,j}). \tag{6}$$

Substituting (1) and (2) into (6), one gets

$$\begin{aligned} &Pr(z_{i,j}, z_{\eta_{i,j}}, \mathbf{o}_{i,j}, \mathbf{o}_{\eta_{i,j}}) = \\ &\quad Pr(z_{i,j}) \cdot Pr(\mathbf{o}_{i,j}|z_{i,j}) \cdot \prod_{(i',j')\in\eta_{i,j}} Pr^{m,n}(z_{i',j'}|z_{i,j}) \cdot Pr(\mathbf{o}_{i',j'}|z_{i',j'}). \end{aligned} \tag{7}$$

So, the feature vector $\mathbf{o}_{i,j}$ can be labeled as

$$\begin{aligned} G_k = \underset{z_{i,j}}{\operatorname{argmax}}\, &Pr(z_{i,j}) \cdot Pr(\mathbf{o}_{i,j}|z_{i,j}) \cdot \\ &\prod_{(i',j')\in\eta_{i,j}} \sum_{z_{i',j'}} Pr^{m,n}(z_{i',j'}|z_{i,j}) \cdot Pr(\mathbf{o}_{i',j'}|z_{i',j'}). \end{aligned} \tag{8}$$

The above labeling scheme is a noniterative one and all $z_{i,j}$s can be updated synchronously. This method was originally developed in [20] and called CVQ labeling, and later was adopted in [22], [23] for character recognition.

3.1.2 Global Labeling of the Feature Map

Alternatively, one can define an optimal region map $\mathbf{Z}_{opt}$ as follows:

$$\mathbf{Z}_{opt} = \underset{\mathbf{Z}}{\operatorname{argmax}} Pr(\mathbf{Z}|\mathbf{O}). \tag{9}$$

According to Besag's ICM (iterated conditional modes) method [12], a suboptimal region map can be obtained by applying the following single feature vector labeling procedure iteratively which relabels $\mathbf{o}_{i,j}$ to G_k by:

$$G_k = \underset{z_{i,j}}{\operatorname{argmax}} Pr(z_{i,j}|\mathbf{O}, z_{\gamma_{i,j}}), \tag{10}$$

where $\gamma_{i,j}$ is the set of all sites of the observation image except (i,j). By making an analogy to the original ICM method, a global region map can be derived by *iteratively* applying the following single feature vector labeling procedure [24]:

$$G_k = \underset{z_{i,j}}{\operatorname{argmax}} Pr(z_{i,j}|\mathbf{o}_{i,j}, \mathbf{o}_{\eta_{i,j}}, z_{\eta_{i,j}}) \tag{11}$$

$$= \underset{z_{i,j}}{\operatorname{argmax}} Pr(z_{i,j}, z_{\eta_{i,j}}, \mathbf{o}_{i,j}, \mathbf{o}_{\eta_{i,j}}). \tag{12}$$

Substituting (7) into the above equation, we have

$$\begin{aligned} G_k = \underset{z_{i,j}}{\operatorname{argmax}}\, &Pr(z_{i,j}) \cdot Pr(\mathbf{o}_{i,j}|z_{i,j}) \cdot \\ &\prod_{(i',j')\in\eta_{i,j}} Pr^{m,n}(z_{i',j'}|z_{i,j}) \cdot Pr(\mathbf{o}_{i',j'}|z_{i',j'}), \end{aligned} \tag{13}$$

where each term on the second line of (13) represents a source of contextual information. Consequently, for a given CS model, one gets the following "modified" ICM feature vector labeling algorithm:

Step a. Get an initial region map for the image. This can be achieved by ignoring contextual considerations and merely choosing $z_{i,j}$ to maximize $Pr(\mathbf{o}_{i,j}|z_{i,j})$ at each (i,j) separately.

Step b. Let (i,j) move from the top left corner of the image to the bottom right corner along a row-wise raster scan. Reidentify feature vector $\mathbf{o}_{i,j}$ to G_k according to (13) and replace the old $z_{i,j}$ value with G_k. Then, this process of feature vector reidentification is repeated following a path of a row-wise raster scan but in the opposite direction, from bottom to top. Repeat again along columnwise raster scans in two opposite directions.

Step c. Repeat **Step b** until the labeling of all feature vectors becomes stable.

The above iterative algorithm can also be viewed as a deterministic relaxation version of the above noniterative CVQ labeling scheme.

3.2 Solution to Problem 2

Let there be a collection of M CS models, $\Lambda = \{\lambda_1, \lambda_2, \cdots, \lambda_M\}$, one for each character class in the vocabulary. Furthermore, let the prior uniform probability distribution for all M character classes be assumed. There are many possible ways to define a discriminant function which characterizes the similarity between the observation $\mathbf{O}$ and the given model λ of a character class. In the following, we highlight three discriminant functions which we have studied.

One possibility is to define the discriminant function as follows [22]:

$$\begin{aligned} g_1(\mathbf{O}, \lambda) = g_1(\mathbf{O}, \mathbf{Z}, \lambda) = &\prod_{(i,j)} Pr(z_{i,j}) \cdot Pr(\mathbf{o}_{i,j}|z_{i,j}) \cdot \\ &\prod_{(i',j')\in\eta_{i,j}} \sum_{z_{i',j'}} Pr^{m,n}(z_{i',j'}|z_{i,j}) \cdot Pr(\mathbf{o}_{i',j'}|z_{i',j'}), \end{aligned} \tag{14}$$

where $\mathbf{Z} = [z_{i,j}]$ is the region map obtained by labeling $\mathbf{O} = [\mathbf{o}_{i,j}]$ with respect to λ using (8).

Another way to define the discriminant function is as follows [24]:

$$\begin{aligned} g_2(\mathbf{O}, \lambda) = g_2(\mathbf{O}, \mathbf{Z}, \lambda) = &\prod_{(i,j)} Pr(z_{i,j}) \cdot Pr(\mathbf{o}_{i,j}|z_{i,j}) \cdot \\ &\prod_{(i',j')\in\eta_{i,j}} Pr^{m,n}(z_{i',j'}|z_{i,j}) \cdot Pr(\mathbf{o}_{i',j'}|z_{i',j'}), \end{aligned} \tag{15}$$

where $\mathbf{Z}$ is the region map obtained by using the above modified iterative ICM labeling algorithm.

A third way is to define the discriminant function as follows [24]:

$$\begin{aligned} g_3(\mathbf{O}, \lambda) = &\prod_{(i,j)} \sum_{z_{i,j}} Pr(z_{i,j}) \cdot Pr(\mathbf{o}_{i,j}|z_{i,j}) \cdot \\ &\prod_{(i',j')\in\eta_{i,j}} \sum_{z_{i',j'}} Pr^{m,n}(z_{i',j'}|z_{i,j}) \cdot Pr(\mathbf{o}_{i',j'}|z_{i',j'}). \end{aligned} \tag{16}$$

This equation differs from (14) only in the presence of a summation over all possible values of $z_{i,j}$.

By using any of the above definitions of the discriminant function, an unknown image $\mathbf{O}$ will be classified to class d if

$$g(\mathbf{O}, \lambda_d) > g(\mathbf{O}, \lambda_c) \quad \forall c \neq d. \tag{17}$$

This is known as the *maximum discriminant decision rule* for pattern recognition.

3.3 Solution to Problem 3

The third, and by far the most difficult, problem of CS modeling is to determine a method to adjust the model parameters $(\pi, \mathbf{A}, \mathbf{B})$ under certain reasonable criterion. Several solutions will be provided below.

3.3.1 Decision-Directed Method

In section 3.1, it is assumed that, if a particular model λ is given, a particular feature vector labeling procedure can be used to generate a "meaningful" region map $\mathbf{Z}$. In this subsection, conversely, assuming a particular region map $\mathbf{Z}$ is given, the way to estimate the parameters of λ will be discussed.

Let $[\mathbf{O}^1, \mathbf{O}^2, \ldots, \mathbf{O}^S]$ denote a set of S observation samples with $\mathbf{O}^s = [\mathbf{o}^s_{i,j}]$ being the sth training sample, and $[\mathbf{Z}^1, \mathbf{Z}^2, \ldots, \mathbf{Z}^S]$ denote the corresponding region maps with $\mathbf{Z}^s = [z^s_{i,j}]$ being the given region map associated with the sth image. Assuming observation samples are drawn independently, an objective function can be defined as follows:

$$F(\lambda) = \sum_{s=1}^{S} \ln g(\mathbf{O}^s, \mathbf{Z}^s, \lambda), \tag{18}$$

where $g(\mathbf{O}^s, \mathbf{Z}^s, \lambda)$ can take the form of $g_1(\cdot)$ in (14), or $g_2(\cdot)$ in (15). The parameter estimation of a CS model is then based on the maximization of the above-defined objective function. This type of objective functions will be called hereinafter "pseudolikelihood" functions.

If $g_2(\cdot)$ is adopted, the "pseudolikelihood" function becomes

$$F_2(\lambda) = \sum_{s=1}^{S} \ln g_2(\mathbf{O}^s, \mathbf{Z}^s, \lambda) \tag{19}$$

and the reestimation formulas for π_k, $a^{m,n}_{kl}$, and $b_{k,t}$ can be derived as follows:

$$\hat{\pi}_k = \frac{\sum_{s=1}^{S} N^s_k}{\sum_{s=1}^{S} N^s} \tag{20}$$

$$\hat{a}^{m,n}_{kl} = \frac{\sum_{s=1}^{S} \left| \{(i,j) \mid z^s_{i,j} = G_k, z^s_{i+m,j+n} = G_l\} \right|}{\sum_{s=1}^{S} N^s_k} \tag{21}$$

$$\hat{b}_{k,t} = \frac{\sum_{s=1}^{S} \left| \{(i,j) | \mathbf{o}^s_{i,j} = v_t, z^s_{i,j} = G_k\} \right|}{\sum_{s=1}^{S} N^s_k}, \tag{22}$$

where N^s_k denotes the number of feature vectors in the sth image assigned to region G_k, N^s denotes the total number of feature vectors in that image, and $|\{\ldots\}|$ denotes the number of events in the set defined within the two bars.

Now, an algorithm can be formulated for CS model parameter estimation by taking an initial set of model parameters and iteratively improving it as follows:

Step 1. An initial estimate of parameters of a CS model can be derived with a bootstrapping region segmentation algorithm to be explained later.

Step 2. Based on the current estimate of model parameters, a region map for every training image is generated with feature vectors identified according to the modified ICM labeling method described previously.

Step 3. Based on the current region maps, the model parameters are updated by using (20), (21), and (22). $\hat{b}_{k,t}$ for any v_t not observed in region G_k should be assigned a small constant ϵ followed by normalization instead of leaving it at zero because such a lack of observation may simply be due to the finite size of the training image set.

Step 4. Repeat **Step 2** and **Step 3** until convergence (i.e., the change in (19) drops below a predefined threshold).

The above algorithm can be viewed as a standard technique for finding a fixed-point via the method of successive approximation [25]. In practice, convergence to what must generally be a local maximum of the above-defined "pseudolikelihood" function seems very rapid with little change of the function value after six or seven iterations on the average. Readers can examine the rate of convergence in Section 5.

If the modified ICM labeling in Step 2 of the above training algorithm is replaced with the noniterative CVQ labeling procedure defined in (8), the resulting training algorithm can be viewed as maximizing the "pseudolikelihood" function defined in (18) with $g(\cdot)$ taking the form of $g_1(\cdot)$ as follows:

$$F_1(\lambda) = \sum_{s=1}^{S} \ln g_1(\mathbf{O}^s, \mathbf{Z}^s, \lambda). \tag{23}$$

In practice, the convergence of this simplified algorithm is still acceptable. Readers can refer to its learning curve in Section 5. In fact, this simplified training and the associated recognition algorithm were first presented in [22] as an extension of the CVQ model used in speech recognition [20], [21] and, later, used to build the character recognizer reported in [23].

In order to start the training process of λ for a character, the initial model parameters can be specified according to the initial region identifications of an arbitrarily chosen training sample of the character. Such a bootstrapping region segmentation procedure is described as follows [22], [23]:

- Each row of the selected original binary character image as a bit map is decomposed into alternate white and black segments. Each segment in the first row is assigned a unique region identity.
- For each segment in the next and subsequent rows,
  - if there is a segment in the previous row having the same color and approximately the same starting and ending columns, say, differing by no more than one pixel position, the same region identity will be inherited from the segment of the previous row; otherwise,
  - a new region identity will be created for the segment of the new row.

In this way, pixels of the same stroke may therefore belong to multiple regions and blank spaces between strokes will be divided into regions also. Thus, a region map is created for the image. Based on the initial region map, one computes the initial parameter estimates of the CS model by using (20), (21), and (22) (note that $S = 1$ in this case). As a remark, the above initialization method is not necessarily the best one. As a future work, other possibilities deserve being explored.

3.3.2 Mixture-Region Algorithm

Identifying each feature vector to a region will inevitably end up with some quantization error. Each feature vector can be considered belonging to all regions stochastically instead of a particular one in the labeling procedure. That is, one considers $Pr(z_{i,j} = G_k, \mathbf{o}_{i,j}, \mathbf{o}_{\eta_{i,j}})$ over all possible $k = 1, 2, \ldots, K$, and not just

$$G_k = \underset{z_{i,j}}{\operatorname{argmax}}\, Pr(z_{i,j}, \mathbf{o}_{i,j}, \mathbf{o}_{\eta_{i,j}}),$$

as in (8). The associated training algorithm uses the following "pseudolikelihood" function [24]:

$$F_3(\lambda) = \sum_{s=1}^{S} \ln g_3(\mathbf{O}^s, \lambda), \tag{24}$$

where $g_3(\mathbf{O}^s, \lambda)$ is defined in (16). One can observe that in (20), (21), and (22), N_k^s is the number of feature vectors from the training image assigned to region G_k. It's useful to think of $\sum_{i,j} Pr(z_{i,j}^s = G_k, \mathbf{o}_{i,j}^s, \mathbf{o}_{\eta_{i,j}}^s)$ as a fractional number of feature vectors from the training image assigned to region G_k. Then, one can modify the parameter reestimation formulas (20), (21), and (22) into the ones shown as follows:

$$\hat{\pi}_k = \frac{\sum_{s=1}^{S}\sum_{i,j} Pr(z_{i,j}^s = G_k, \mathbf{o}_{i,j}^s, \mathbf{o}_{\eta_{i,j}}^s)}{\sum_{k'}\sum_{s=1}^{S}\sum_{i,j} Pr(z_{i,j}^s = G_{k'}, \mathbf{o}_{i,j}^s, \mathbf{o}_{\eta_{i,j}}^s)} \tag{25}$$

$$\hat{a}_{kl}^{m,n} = \frac{\sum_{s=1}^{S}\sum_{i,j} Pr(z_{i,j}^s = G_k, \mathbf{o}_{i,j}^s, \mathbf{o}_{\eta_{i,j}}^s) \cdot Pr(z_{i',j'}^s = G_l, \mathbf{o}_{i',j'}^s, \mathbf{o}_{\eta_{i',j'}}^s)}{\sum_{s=1}^{S}\sum_{i,j}\sum_{l'} Pr(z_{i,j}^s = G_k, \mathbf{o}_{i,j}^s, \mathbf{o}_{\eta_{i,j}}^s) \cdot Pr(z_{i',j'}^s = G_{l'}, \mathbf{o}_{i',j'}^s, \mathbf{o}_{\eta_{i',j'}}^s)} \tag{26}$$

$$\hat{b}_{k,t} = \frac{\sum_{s=1}^{S} \sum_{\{(i,j)|\mathbf{o}_{i,j}^s = v_t\}} Pr(z_{i,j}^s = G_k, \mathbf{o}_{i,j}^s, \mathbf{o}_{\eta_{i,j}}^s)}{\sum_{t=1}^{T}\sum_{s=1}^{S} \sum_{\{(i,j)|\mathbf{o}_{i,j}^s = v_t\}} Pr(z_{i,j}^s = G_k, \mathbf{o}_{i,j}^s, \mathbf{o}_{\eta_{i,j}}^s)}, \tag{27}$$

where

$$Pr(z_{i,j}^s = G_k, \mathbf{o}_{i,j}^s, \mathbf{o}_{\eta_{i,j}}^s) = Pr(G_k) \cdot Pr(\mathbf{o}_{i,j}^s | G_k) \cdot \prod_{(i',j') \in \eta_{i,j}} \sum_{z_{i',j'}^s} Pr^{m,n}(z_{i',j'}^s | G_k) \cdot Pr(\mathbf{o}_{i',j'}^s | z_{i',j'}^s). \tag{28}$$

The mixture-region algorithm has a merit of being more robust by considering all possible region labels and not just the most likely one, which can be regarded as a smoothing technique. Little is known of the convergence properties of the above training algorithm, but limited experience thus far seems encouraging. Readers can refer to the relevant learning curve in Section 5.

3.3.3 Training a CS Model with the Gradient Projection Method

With the well-defined objective function as in (24), CS model parameters $(\pi, \mathbf{A}, \mathbf{B})$ can be considered as variables of this function subject to the following linear constraints:

$$\sum_{k=1}^{K} \pi_k = 1 \text{ and } \pi_k \geq 0, \quad k = 1, 2, \cdots, K \tag{29}$$

$$\sum_{l=1}^{K} a_{k,l}^{m,n} = 1 \text{ and } a_{k,l}^{m,n} \geq 0, \; k, l = 1, \cdots, K; \; (m,n) \in \Delta \tag{30}$$

$$\sum_{t=1}^{T} b_{k,t} = 1 \text{ and } b_{k,t} \geq \epsilon, \quad k = 1, \cdots, K; \; t = 1, \cdots, T, \tag{31}$$

where ϵ is a small positive value and $\Delta = \{(-1,0), (1,0), (0,-1), (0,1)\}$. If one looks at the training problem of a CS model as a problem of classical constrained optimization, then the standard optimization techniques can be used to solve for the "optimal" model parameters. There are many general purposed procedures for linear constrained optimization (e.g., [25]) that can be used to solve the training problem. One of them is called, the *gradient projection method* (GPM), which was proposed and extensively analyzed by Rosen in [26].

The main idea of the GPM is to search along the projection of the gradient of the objective function on the constraint space for a local maximum. So, the GPM is essentially a steepest ascent method in the subspace defined by "the active constraints" of model parameters. The method had been adopted and tailored for HMM training in [27], [28]. As in the HMM case, when this GPM method is applied to train CS models, a very simple formulation can be derived due to the special structure of the constraints on CS model parameters. One can see that the constraints in (29), (30), and (31) can be divided into disjoint groups, i.e., no two constraint groups have any variable in common. Each constraint group takes the form $\sum_{i=1}^{N} x_i = 1$ and $x_i \geq \epsilon, i = 1, 2, \cdots, N$. So, all the CS model parameters and their associated constraints can be divided into disjoint subsets, with the corresponding search directions computed and the working set for each subset determined independently. The overall search direction is just the concatenation of search directions of the disjoint subsets of CS model parameters. In this way, the same formulation in [27], [28] can also be used for CS model parameter estimation because of the same linear constraint properties. What needs to change is the evaluation of the objective function $F_3(\lambda)$ and its partial derivatives. These partial derivatives can be calculated as follows:

$$\frac{\partial F_3}{\partial \pi_k} = \sum_{s=1}^{S}\sum_{i,j} \frac{b_k(\mathbf{o}_{i,j}^s) Y_{i,j,k}^s}{X_{i,j}^s} \tag{32}$$

$$\frac{\partial F_3}{\partial a_{k,l}^{m,n}} = \sum_{s=1}^{S}\sum_{i,j} \frac{\pi_k b_k(\mathbf{o}_{i,j}^s) b_l(\mathbf{o}_{i+m,j+n}^s) Y_{i,j,k}^{m,n}(s)}{X_{i,j}^s} \tag{33}$$

$$\frac{\partial F_3}{\partial b_{k,t}} = \sum_{s=1}^{S}\sum_{i,j} \frac{1}{X_{i,j}^s} \sum_{k'=1}^{K} \Big[1(k'=k) 1(\mathbf{o}_{i,j}^s = v_t) \pi_k Y_{i,j,k}^s + \pi_{k'} b_{k'}(\mathbf{o}_{i,j}^s) \sum_{(m,n)\in\Delta} 1(\mathbf{o}_{i+m,j+n}^s = v_t) a_{k',k}^{m,n} Y_{i,j,k'}^{m,n}(s) \Big], \tag{34}$$

where

$$Y_{i,j,k}^s = \prod_{(m,n)\in\Delta} \sum_{l=1}^{K} a_{k,l}^{m,n} b_l(\mathbf{o}_{i+m,j+n}^s)$$

$$Y_{i,j,k}^{m,n}(s) = \prod_{(m',n')\in\Delta-\{(m,n)\}} \sum_{l=1}^{K} a_{k,l}^{m',n'} b_l(\mathbf{o}_{i+m',j+n'}^s)$$

$$X_{i,j} = \sum_{k=1}^{K} \pi_k b_k(\mathbf{o}_{i,j}^s) Y_{i,j,k}$$

and $1(\cdot)$ is an indicator function:

$$1(\hbar) = \begin{cases} 1 & \text{if } \hbar \text{ is true} \\ 0 & \text{otherwise.} \end{cases} \tag{35}$$

Readers are referred to [27] for details of the GPM algorithm. Note that the GPM algorithm does not require the objective function to assume any special form. This fact may prove to be an advantage since the decision-directed algorithms and the mixture-region algorithm discussed previously are not applicable to a general objective function which is usually demanded in discriminative training. In the following section, we discuss how to discriminatively train the CS models by using GPM.

4 Discriminative Training with GPM

For any CS model-based character recognizer, generally speaking, the purpose of CS model training is to yield a recognizer of the lowest possible recognition error rate. This objective is achieved by maximizing a suitable objective function $R(\lambda)$. Thus, there are two important and difficult problems to consider. The first is to

determine a meaningful objective function such that, if $R(\bar{\lambda}) > R(\lambda)$, then $\bar{\lambda}$ produces a better recognizer than that by λ. Once a function $R(\lambda)$ has been chosen, the second problem (the estimation problem) is to find the parameter set $\bar{\lambda}$ which maximizes it. The maximum "pseudolikelihood" training of CS models described in the previous section will not necessarily lead to maximum recognition accuracy. Discriminative training methods such as the *minimum classification error (MCE)* training (e.g., [29], [30], [28]) had been proposed for a speech recognition problem to address this issue. In this section, we demonstrate how to use the MCE/GPM method in [28] for discriminative training of CS models [31].

Let's consider a collection of M CS models, $\Lambda = (\lambda_1, \lambda_2, \cdots, \lambda_M)$, where λ_m denotes the set of parameters of the mth CS model. Let $\mathbf{O}^{(m,n)}$ denote the nth training sample associated with λ_m, and each model has W_m such training samples. The objective function for discriminative training adopted in this section is the minimum recognition error formulation which is a three-step procedure proposed in [29]. The three-step definition emulates the classification/recognition operation as well as the performance evaluation, particularly in terms of classification errors. Readers are referred to [29], [30] for details of the rationale of the MCE framework. In the following, we briefly outline how to form the objective function for MCE training.

The first step of the formulation is to prescribe an appropriate discriminant function $f_i(\mathbf{O};\Lambda)$ which is used by the classifier to make its decision for each input $\mathbf{O}$ by choosing the largest of the discriminants evaluated on $\mathbf{O}$, i.e.,

$$\mathbf{O} \text{ is classified as class } i, \text{if } f_i(\mathbf{O};\Lambda) = \max_j f_j(\mathbf{O};\Lambda). \tag{36}$$

The ith discriminant function $f_i(\mathbf{O};\Lambda)$ is chosen to be $\ln g_3(\mathbf{O},\lambda_i)$.

A misclassification measure is then introduced in the second step to embed the decision process in a function form. While there are many alternatives, one misclassification measure for each class i can be defined as:

$$d_i(\mathbf{O};\Lambda) = -f_i(\mathbf{O};\Lambda) + \ln\left[\frac{1}{M-1}\sum_{j\neq i} e^{f_j(\mathbf{O};\Lambda)\zeta}\right]^{\frac{1}{\zeta}}, \tag{37}$$

where ζ is a positive number. This misclassification measure is a quantity that indicates whether an input token $\mathbf{O}$ of the ith class will be misclassified according to the decision rule of (36), implemented by the classifier parameter set Λ. A larger $d_i(\mathbf{O};\Lambda)$ definitely implies that more of the input will be misclassified. By varying the value of ζ, one can, to a degree, take all the competing classes into consideration in the process of optimizing the classifier parameter set Λ.

The third step is to define the smoothed loss function $l_i(\mathbf{O};\Lambda)$ of the misclassification measure for each class i. One possibility is to choose

$$l_i(\mathbf{O};\Lambda) = l_i(d_i(\mathbf{O};\Lambda)) = \frac{1}{1+e^{-\xi d_i(\mathbf{O};\Lambda)}}, \tag{38}$$

where ξ is a positive number. Thus, for any unknown $\mathbf{O}$, the classifier performance is measured by

$$l(\mathbf{O};\Lambda) = \sum_{i=1}^{M} l_i(\mathbf{O};\Lambda)1(\mathbf{O}\in C_i), \tag{39}$$

where $1(\cdot)$ is the same indicator function as in (35) and C_i is used to denote both the class and the data set of it.

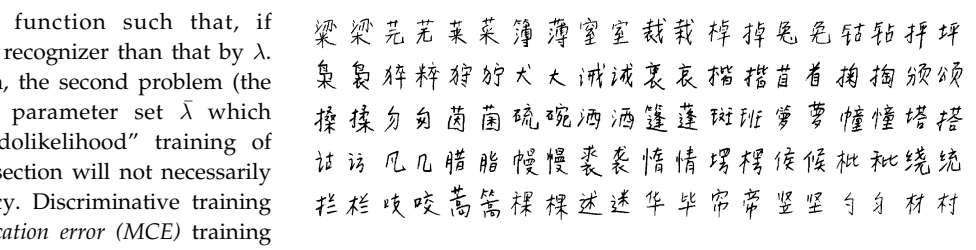

Fig. 1. A vocabulary of 50 pairs of highly similar Chinese characters.

At this point, the objective function of discriminative training is defined as the following *empirical average cost* for the entire training data set:

$$L(\Lambda) = \frac{1}{W}\sum_{m=1}^{M}\sum_{n=1}^{W_m} l_m(\mathbf{O}^{(m,n)};\Lambda), \tag{40}$$

where $W = \sum_{m=1}^{M} W_m$ is the total number of training samples. By controlling parameters ζ and ξ, and minimizing this *empirical average cost*, one can have an accurate approximation to the minimization of the classification error probability. Due to the fact that the GPM formulation in [27] is for maximization, the actual objective function adopted is

$$F_4(\Lambda) = -L(\Lambda). \tag{41}$$

To compute the gradient $\nabla F_4(\Lambda)$, let θ_k denote a particular parameter of model k, then one has

$$\frac{\partial F_4(\Lambda)}{\partial \theta_k} = -\frac{1}{W}\sum_{m=1}^{M}\sum_{n=1}^{W_m}\frac{\partial l_m(\mathbf{O}^{(m,n)};\Lambda)}{\partial \theta_k}. \tag{42}$$

After some algebraic manipulation, one gets

$$\begin{aligned}\frac{\partial F_4(\Lambda)}{\partial \theta_k} = &\frac{\xi}{W}\sum_{n=1}^{W_k}\left\{l_k(\mathbf{O}^{(k,n)};\Lambda)\left[1-l_k(\mathbf{O}^{(k,n)};\Lambda)\right]\cdot\frac{\partial f_k(\mathbf{O}^{(k,n)};\Lambda)}{\partial\theta_k}\right\}\\ &-\frac{\xi}{W}\sum_{m\neq k}^{M}\sum_{n=1}^{W_m}\left\{l_m(\mathbf{O}^{(m,n)};\Lambda)\left[1-l_m(\mathbf{O}^{(m,n)};\Lambda)\right]\cdot\right.\\ &\left.\frac{e^{f_k(\mathbf{O}^{(m,n)};\Lambda)\zeta}}{\sum_{j\neq m}^{M} e^{f_j(\mathbf{O}^{(m,n)};\Lambda)\zeta}}\cdot\frac{\partial f_k(\mathbf{O}^{(m,n)};\Lambda)}{\partial\theta_k}\right\}.\end{aligned} \tag{43}$$

By substituting the related derivatives $\frac{\partial f_k(\mathbf{O};\Lambda)}{\partial\theta_k}$ (which are special cases in (32), (33), and (34)) into the above equation, the final derivatives used in the gradient projection method can be obtained.

Given the above objective function and its partial derivatives, one can now apply the GPM to discriminatively adjust the CS model parameters Λ which equivalently minimizes the cost function.

5 Experiments and Results

5.1 Experimental Setup

In this study, 50 pairs of highly similar Chinese characters, as shown in Fig. 1, are used as the recognition vocabulary to study the characteristics and the effectiveness of the various training algorithms and the discriminant functions discussed in this paper. This vocabulary is formed by choosing the most confusable pairs from the confusion matrix of the testing result of a previous recognizer reported in [23]. Each character is written by 200 writers with 150 of them used for training and the remaining 50 samples for testing. The same preprocessing techniques as in [22], [23] are adopted. The characters are scanned by a 300dpi scanner into

TABLE 1
Character Recognizers Constructed by Using Different Methods

Recognizer Number	Discriminant Function Used in Recognizer	Training Algorithm For CS Models	
		Objective Function	Estimation Method
1	$g_1(\mathbf{O}, \lambda)$ in Eq.(14)	$F_1(\lambda)$ in Eq.(23)	CVQ training in [22], [23]
2	$g_2(\mathbf{O}, \lambda)$ in Eq.(15)	$F_2(\lambda)$ in Eq.(19)	DD method in section 3.3.1
3	$g_3(\mathbf{O}, \lambda)$ in Eq.(16)	$F_3(\lambda)$ in Eq.(24)	Mixture-region method in section 3.3.2
4	$g_3(\mathbf{O}, \lambda)$ in Eq.(16)	$F_4(\Lambda)$ in Eq.(41)	MCE/GPM in section 4

TIFF files, which are then noise removed, thinned, segmented, and size normalized (with a resolution of 30×30 pixels) into individual character files. At each pixel, a feature vector is constructed by computing five features, as described in Section 2. Four character recognizers are constructed by using four combinations of the model training algorithms and the discriminant functions described in Sections 3 and 4. These recognizers are described briefly in Table 1. A series of experiments are then carried out to compare the performance of these recognizers.

5.2 Experimental Results

The first experiment is to test Recognizer 1. After 20 iterations of training, the recognizer achieves a recognition rate of 98.1 percent on the training set (close-test) and 93.7 percent on the testing set (open-test). The objective function values, and the close- and open-test recognition rates at various iterations are plotted in Fig. 2 to illustrate the performance improvement as a function of the training process. The relative change of the corresponding pseudolikelihood function is less than 10^{-3} after nine iterations.

The second experiment is to test Recognizer 2. After 20 iterations of training, the close- and open-test rates are 98.5 percent and 93.8 percent, respectively. The related learning curves are plotted in Fig. 3. The algorithm converges rapidly with a relative change in the pseudolikelihood function value less than 10^{-3} after seven iterations.

The third experiment is to test the mixture-region algorithm. After 20 iterations, the close- and open-test rates are 98.1 percent

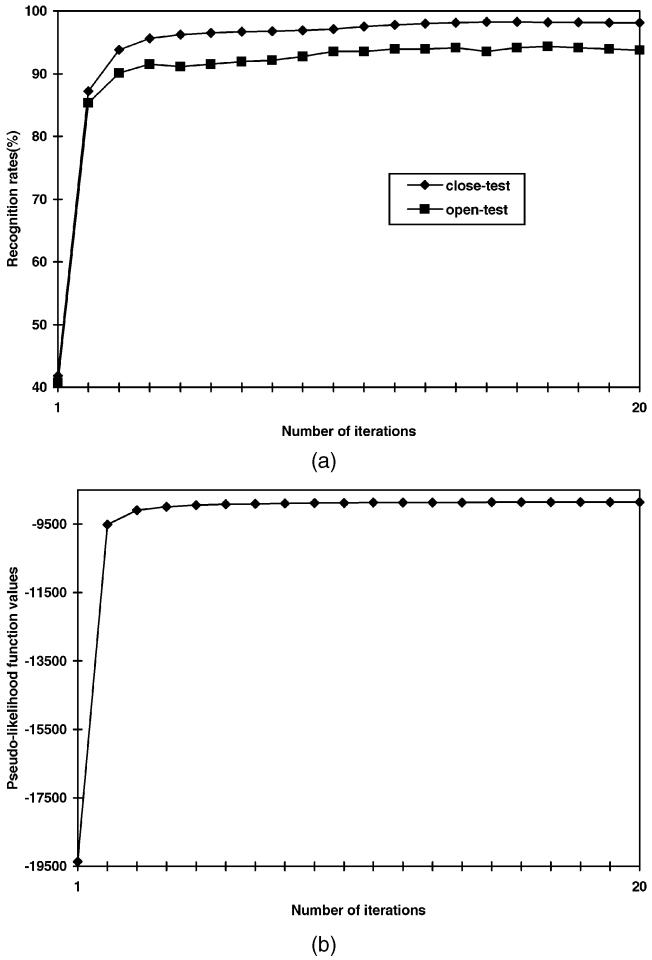

Fig. 2. Learning curves for training Recognizer 1: (a) Close-test and open-test recognition rates (percent correct). (b) The corresponding pseudolikelihood function values.

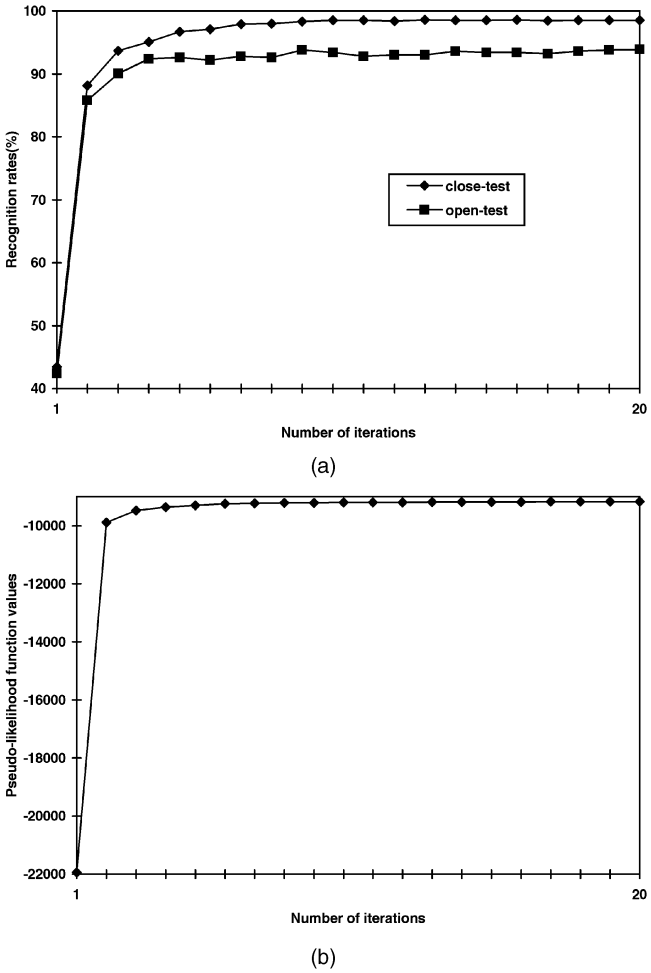

Fig. 3. Learning curves for training Recognizer 2: (a) Close-test and open-test recognition rates (percent correct). (b) The corresponding pseudolikelihood function values.

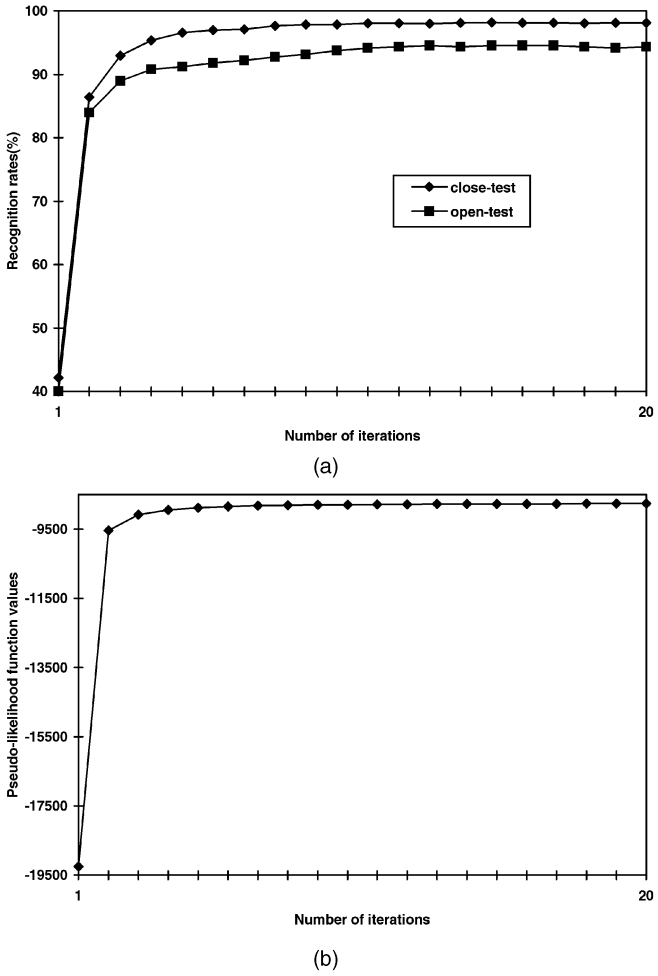

Fig. 4. Learning curves for training Recognizer 3 (mixture-region algorithm): (a) Close-test and open-test recognition rates (percent correct). (b) The corresponding pseudolikelihood function values.

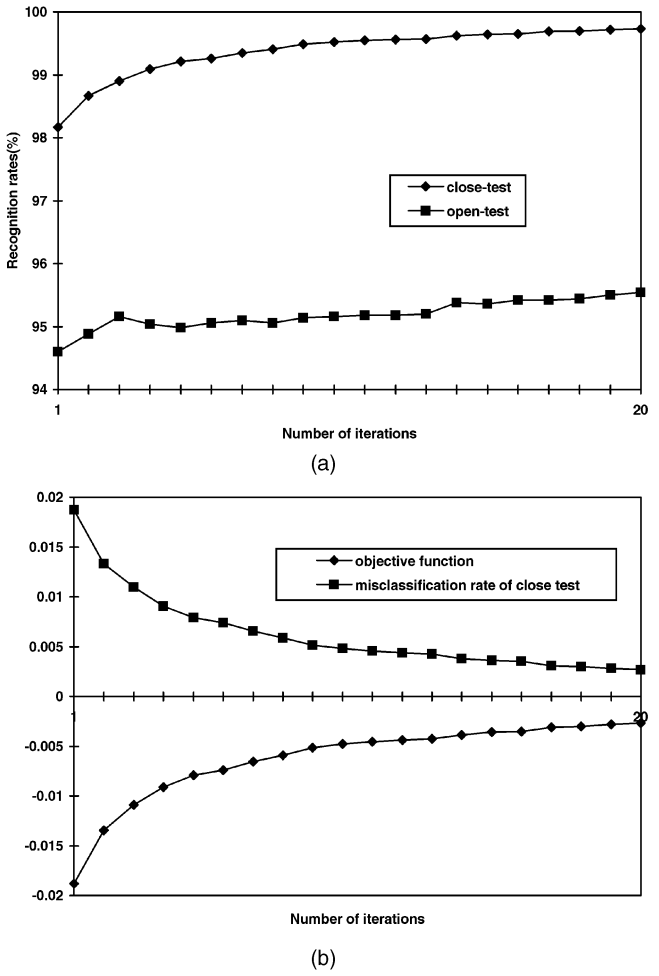

Fig. 5. Learning curves for discriminative training with initial models trained by mixture-region algorithm: (a) Close-test and open-test recognition rates (percent correct). (b) The objective function values and the misclassification rate of close-test.

and 94.4 percent, respectively. The related learning curves are plotted in Fig. 4. Although the close-test result is not as high as that of using the modified ICM algorithm, the open-test result is improved as a consequence of "smoothing" or "averaging" over the regions. The relative change of the corresponding pseudolikelihood function is less than 10^{-3} after 12 iterations.

The fourth experiment is to test the discriminative training based on the GPM. The training process starts with the well-trained initial models which themselves are trained with the mixture-region algorithm. The parameters ζ and ξ used in (37) and (38) are set to be ∞ and 0.1, respectively. When ζ approaches ∞, the misclassification measure becomes

$$d_i(\mathbf{O};\Lambda) = -f_i(\mathbf{O};\Lambda) + f_j(\mathbf{O};\Lambda), \tag{44}$$

where j is the index of the class with the largest discriminant value among those classes other than C_i. After 20 iterations, the close- and open-test recognition rates are 99.7 percent and 95.5 percent, respectively. Fig. 5 illustrates the rate of convergence of the discriminative training process in terms of the objective function and close- and open-test results. One can observe in Fig. 5b that the negative of the objective function has almost the same form of change as the misclassification rate, which demonstrates that the definition $L(\Lambda)$ emulates the classification errors very well.

5.3 Discussion

The close- and open-test recognition performance of the above four recognizers are summarized in Table 2 for comparison. Among the three maximum pseudolikelihood training methods, it can be observed that the training based on the modified ICM algorithm (Recognizer 2) can produce a better fitting to the training data than the other two algorithms (Recognizers 1 and 3). However, from the open-test results, it seems that the mixture-region training algorithm (Recognizer 3) tends to smooth the model thus, leading to a better generalization capability. In other words, if there is no shortage of training data, one can expect the modified ICM algorithm to be superior to the other two algorithms. The mixture-region algorithm seems to be more robust when there are insufficient training data which is the case in real life most of the time. By MCE/GPM training (Recognizer 4), about 86 percent error rate reduction is achieved for close-test and 20 percent for open-test. The very high close-test rate suggests the power of discriminative training in tuning the model parameters to the training data. This is not accomplished on the expense of model generalization to unseen samples, because effectively, the model of each character is now trained with not only its own samples but also those of the similar characters.

Although a worst-case comparison of the computational complexity of the above recognizers can be easily made, it's not so meaningful here. In practical implementation of the above recognizers, many tricks and pruning techniques have been used. Consequently, different recognizers can be made equally efficient. A detailed description of those techniques is out of the scope of this paper.

TABLE 2
Performance (Character Recognition Rate in Percent) Comparison of Four Recognizers

Performance	Recognizer 1	Recognizer 2	Recognizer 3	Recognizer 4
close test	98.1	98.5	98.1	99.7
open test	93.7	93.8	94.4	95.5

In summary, the results show that all three pseudolikelihood-based algorithms are efficient for CS model training. The performance of such a recognizer can be substantially upgraded by parameter fine-tuning through discriminative training with the objective of minimum recognition error rate using the GPM. As a remark, like any local optimization procedure, the final result of the GPM-based training highly depends on the initial values of the CS model parameters. This also suggests that the algorithm based on the GPM is most attractive for final "tune-up" and will usually be bootstrapped from well-trained initial models trained with other methods such as pseudolikelihood-based algorithms.

6 Summary

In this paper, we present a study on using a discrete contextual stochastic model for handwritten Chinese character recognition. The capability of contextual stochastic models in modeling complex and variant patterns like handwritten Chinese characters has been demonstrated by the encouraging results obtained thus far. The limitations of the current discrete CS model are also apparent. First, it requires too much memory to store those discrete probability distribution parameters. Second, it is not clear if the adopted cellular features are the most efficient ones. Some perceptually-motivated features are currently under investigation, and the discrete density framework is being extended to a Gaussian-mixture continuous density framework. By extending a discrete CS model to a continuous density one, the number of model parameters will be reduced greatly. This will also facilitate the study of other more advanced topics such as recognizer adaptation, robustness issues, character verification, etc. As important future work, a carefully designed comparative study will be performed on a larger scale experiment to ascertain the performance difference among the contextual stochastic modeling approach, the HMM approach, and the more conventional multiple-prototype based approach. This will help us make an appropriate decision in choosing which approach to use when constructing a practical handwritten Chinese character recognizer.

Acknowledgments

The authors would like to thank Mr. Z.-Q. Ma for programming the original GPM algorithm. This work was supported by HK RGC Earmarked Grant under grant number HKU7020/98E and two internal HKU CRCG research grants.

References

[1] T.H. Hildebrandt and W.T. Liu, "Optical Recognition of Handwritten Chinese Characters: Advances since 1980," *Pattern Recognition,* vol. 26, pp. 205-225, 1993.
[2] Special Issue on Document Analysis and Recognition, *IEICE Trans. Information and Systems,* vol. E77-D, no. 7, July 1994.
[3] Special Issue on Character Recognition and Document Understanding, *IEICE Trans. Information and Systems,* vol. E79-D, no. 5, May 1996.
[4] Special Issue on Oriental Character Recognition, *Pattern Recognition,* vol. 30, no. 8, 1997.
[5] Special Issue on Advances in Oriental Document Analysis and Recognition Techniques, Part I, Part II, *Int'l J. Pattern Recognition and Artificial Intelligence,* vol. 12, nos. 1-2, 1998.
[6] R. Duda and P. Hart, *Pattern Classification and Scene Analysis.* 1973.
[7] L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE,* vol. 77, no. 2, pp. 257-286, 1989.
[8] K.S. Fu and T.S. Yu, *Statistical Pattern Classification Using Contextual Information.* Chichester: Research Studies Press, 1980.
[9] J. Kittler and D. Pairman, "Contextual Pattern Recognition Applied to Cloud Detection and Identification," *IEEE Trans. Geoscientific Remote Sensing,* vol. 23, pp. 855-863, 1985.
[10] J. Haslett, "Maximum Likelihood Discriminant Analysis on the Plane Using a Markovian Model of Spatial Context" *Pattern Recognition,* vol. 18, pp. 287-296, 1985.
[11] J. Besag, "Spatial Interactions and the Statistical Analysis of Lattice Systems," *J. Royal Statistical Soc. B,* vol. 36, pp. 192-236, 1974.
[12] J. Besag, "On the Statistical Analysis of Dirty Pictures," *J. Royal Statistical Soc. B,* vol. 48, pp. 259-302, 1986.
[13] R. Kindermann and J.L. Snell, *Markov Random Fields and Their Applications.* Providence, R.I.: Am. Math. Soc., 1980.
[14] R. Chellappa and A. Jain, *Markov Random Fields: Theory and Application.* Academic Press, 1993.
[15] S.Z. Li, *Markov Random Field Modeling in Computer Vision.* Springer-Verlag, 1995.
[16] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 6, no. 11, pp. 721-741, Nov. 1984.
[17] P.A. Devijver, "Hidden Markov Mesh Random Field Models in Image Analysis," *Advances in Applied Statistics (Statistics and Images: 1),* K.V. Mardia and G.K. Kanji, eds., pp. 187-227, Carfax, 1993.
[18] G. Saon and A. Belaid, "High Performance Unconstrained Word Recognition System Combining HMMs and Markov Random Fields," *Automatic Bankcheck Processing,* S. Impedovo, P.S.P. Wang, and H. Bunke, eds., pp. 309-326, World Scientific, 1997.
[19] H.-S. Park and S.-W. Lee, "A Truly 2D Hidden Markov Model for Off-Line Handwritten Character Recognition," *Pattern Recognition,* vol. 31, no. 12, pp. 1849-1864, 1998.
[20] Q. Huo and C. Chan, "Contextual Vector Quantization for Speech Recognition with Discrete Hidden Markov Model," *Pattern Recognition,* vol. 28, pp. 513-517, 1995.
[21] Q. Huo and C. Chan, "A Study on the Use of Bi-Directional Contextual Dependence in Markov Random Field-Based Acoustic Modeling for Speech Recognition," *Computer Speech and Language,* vol. 10, pp. 95-105, 1996.
[22] S.L. Leung, P.C. Chee, C. Chan, and Q. Huo, "Contextual Vector Quantization Modeling of Hand-Printed Chinese Character Recognition," *Proc. IEEE Int'l Conf. Image Processing,* pp. 432-435, Oct. 1995.
[23] P.K. Wong and C. Chan, "Off-Line Handwritten Chinese Character Recognition as a Compound Bayes Decision Problem" *IEEE Trans. Pattern Ananlysis and Machine Intelligence,* vol. 20, no. 9, pp. 1016-1023, 1998.
[24] Y. Xiong and C. Chan, "Contextual Modeling of Handwritten Chinese Character for Recognition (I)—A Comparative Study," *Proc. 13th Int'l Conf. Digital Signal Processing,* pp. 1099-1102, July 1997.
[25] D.G. Luenberger, *Linear and Nonlinear Programming.* Addison-Wesley, 1984.
[26] J.B. Rosen, "The Gradient Projection Method for Nonlinear Programming—Part I: Linear Constraints," *Proc. SIAM,* vol. 8, pp. 181-217, 1960.
[27] Q. Huo and C. Chan, "The Gradient Projection Method for the Training of Hidden Markov Models," *Speech Comm.,* vol. 13, pp. 307-313, 1993.
[28] Q. Huo and C. Chan, "Discriminative Training of HMM-Based Speech Recognizer with Gradient Projection Method," *Proc. Eurospeech-95,* pp. 101-104 Sept. 1995.
[29] B.-H. Juang and S. Katagiri, "Discriminative Learning for Minimum Error Classification," *IEEE Trans. Signal Processing,* vol. 40, pp. 3043-3054, 1992.
[30] B.-H. Juang, W. Chou, and C.-H. Lee, "Minimum Classification Error Rate Methods for Speech Recognition," *IEEE Trans. Speech and Audio Processing,* vol. 5, no. 3, pp. 257-265, 1997.
[31] Y. Xiong, Q. Huo, and C. Chan, "Contextual Modeling of Handwritten Chinese Character for Recognition (II)—Discriminative Training," *Proc. Int'l Conf. Digital Signal Processing '97,* pp. 1095-1098, 1997.