Pattern Recognition 47 (2014) 1904–1916

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Minimum-risk training for semi-Markov conditional random fields with application to handwritten Chinese/Japanese text recognition

Xiang-Dong Zhou [a,*,1], Yan-Ming Zhang [b], Feng Tian [c], Hong-An Wang [c], Cheng-Lin Liu [b]

[a] Intelligent Media Technique Research Center, Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 400714, PR China
[b] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, 95 Zhongguan East Road, Beijing 100190, PR China
[c] Beijing Key Lab of Human–Computer Interaction, Institute of Software, Chinese Academy of Sciences, Beijing 100190, PR China

ARTICLE INFO

Article history:
Received 6 July 2013
Received in revised form
24 October 2013
Accepted 2 December 2013
Available online 12 December 2013

Keywords:
Semi-Markov conditional random fields
Minimum-risk training
Character string recognition

ABSTRACT

Semi-Markov conditional random fields (semi-CRFs) are usually trained with maximum a posteriori (MAP) criterion which adopts the 0/1 cost for measuring the loss of misclassification. In this paper, based on our previous work on handwritten Chinese/Japanese text recognition (HCTR) using semi-CRFs, we propose an alternative parameter learning method by minimizing the risk on the training set, which has unequal misclassification costs depending on the hypothesis and the ground-truth. Based on this framework, three non-uniform cost functions are compared with the conventional 0/1 cost, and training data selection is incorporated to reduce the computational complexity. In experiments of online handwriting recognition on databases CASIA-OLHWDB and TUAT Kondate, we compared the performances of the proposed method with several widely used learning criteria, including conditional log-likelihood (CLL), softmax-margin (SMM), minimum classification error (MCE), large-margin MCE (LM-MCE) and max-margin (MM). On the test set (online handwritten texts) of ICDAR 2011 Chinese handwriting recognition competition, the proposed method outperforms the best system in competition.

1. Introduction

Due to the large character set and the ambiguity of character segmentation, handwritten Chinese/Japanese text recognition (HCTR) is generally accomplished by an integrated segmentation and recognition approach based on character over-segmentation [1]. The input string (text line image for offline data or pen-tip trajectory for online data) is over-segmented into a sequence of components according to the overlapping between strokes (Fig. 1(a)), with each component (consisting a block of strokes) being a character or part of a character. Subject to constraints of character width, consecutive components are combined to generate candidate characters, which constitute the segmentation candidate lattice (Fig. 1(b) and (c)). On assigning each candidate character a number of candidate classes using a character classifier, we construct the segmentation-recognition candidate lattice (referred to as lattice for brevity). Each path in the lattice corresponds to a segmentation-recognition hypothesis, which is evaluated by a parameterized function combining the character recognition score, geometric and linguistic contexts, and the string recognition result is obtained by searching for the optimal path with maximum score.

The performance of integrated segmentation-recognition of character strings (handwritten texts) largely relies on the parameterized path evaluation function. Although many function forms have been proposed, which are usually the heuristic approximation of posterior probability for a segmentation-recognition path (see [1], for a review), only several papers address the problem of parameter learning [2–4]. In our previous work [5], a semi-Markov conditional random field (semi-CRF) [6] based approach has been proposed for HCTR. A semi-CRF outputs a segmentation S of the observation sequence X, together with the label sequence Y assigned to the segments (sub-sequences) of X. In other words, unlike the linear-chain CRF [7] which models $P(Y|X)$, the semi-CRF explicitly estimates $P(S,Y|X)$. For HCTR, if X is the component sequence after over-segmentation, the segments will be the candidate characters (cf. Fig. 1). Semi-CRFs have the advantages that they allow the use of segment features and between-segment dependencies. This attribute is important for HCTR, since the state-of-the-art Chinese character classifiers, such as the modified quadratic discriminant function (MQDF) [8], usually take the holistic character features as input. The semi-CRF model for HCTR is defined on the lattice to directly estimate the a posteriori probability of a segmentation-recognition hypothesis, in which the information of

* Corresponding author. Tel./fax: +86 2365935537.
E-mail addresses: zhouxiangdong@cigit.ac.cn, xiangdongzhou@foxmail.com (X.-D. Zhou), ymzhang@nlpr.ia.ac.cn (Y.-M. Zhang), tianfeng@iscas.ac.cn (F. Tian), hongan@iscas.ac.cn (H.-A. Wang), liucl@nlpr.ia.ac.cn (C.-L. Liu).

[1] This work was partially done when the first author was with the Beijing Key Lab of Human–Computer Interaction, Institute of Software, Chinese Academy of Sciences.

0031-3203/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2013.12.002

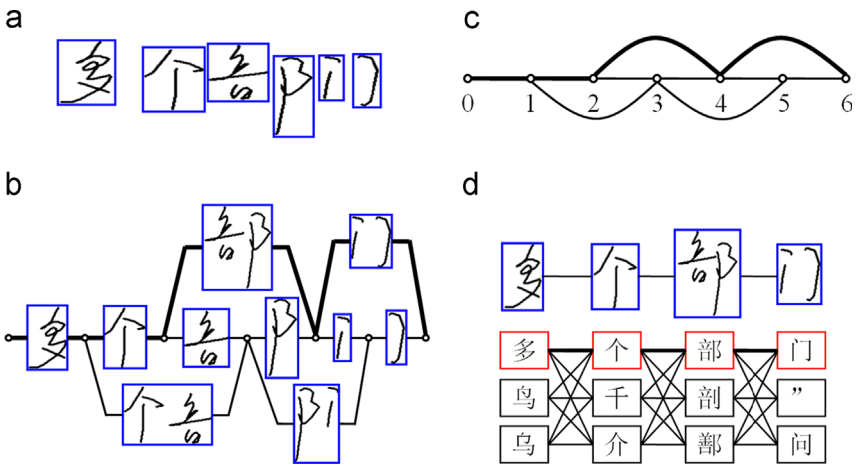

Fig. 1. Generation of segmentation-recognition candidate lattice. (a) Component sequence, (b) candidate characters, (c) segmentation candidate lattice, where each node denotes a candidate segmentation point, each edge corresponds to a candidate character, and the bold lines indicate the desired segmentation and (d) candidate classes of the desired segmentation path.

character recognition, geometric and linguistic contexts are defined as feature functions [5]. This model provides principled tools for both parameter learning and decoding under the maximum a posteriori (MAP) criterion and enables the fusing of high-order features (long range context, such as the trigram language model).

According to the Bayesian decision theory [9], the optimal decision should be made by minimizing the overall risk associated with a cost function measuring the loss of misclassification. When the 0/1 cost is adopted in the Bayes decision rule, we get the popular MAP criterion. The 0/1 cost simply assigns no loss to a correct prediction and a unit loss to an error. Consequently, for HCTR, it aims to minimize the string error rate rather than the character error rate. For example, a hypothesized path (cf. Fig. 1) containing one or more character-level errors, or a totally different path, as compared to the correct, will incur the same amount of loss. However, the performance of HCTR is usually measured in terms of character errors (insertions, deletions and substitutions) [2,5,10], instead of the errors of whole sentence or text line. So, for HCTR, the use of 0/1 cost will lead to a mismatch between classifier training and performance evaluation.

For structured prediction on sequence labeling problems, various discriminative learning techniques have been proposed for training hidden Markov models (HMMs) and CRFs (see Section 2 for a review). Generalization ability is one of the key issues in discriminative training, since the learned models will be finally tested on the unseen data. According to statistical learning theory [11], the test-set error rate is bounded by the sum of the empirical error rate on the training set and a generalization term associated with the margin. Traditional discriminative learning methods, such as maximum mutual information (MMI, an instance of MAP criterion) [12], minimum classification error (MCE) [13] and minimum phone/word error (MPE/MWE) [14], focus more on reducing the empirical error rate rather than decreasing the generalization term [15]. Many attempts have been made to incorporate the principle of large margin into the training of HMMs or CRFs to further improve the generalization abilities [16–18]. However, the performances of these training criteria have not been comprehensively evaluated on HCTR tasks.

In this paper, based on our previous work on HCTR [5], we propose a lattice-based minimum-risk (MR) estimation framework for parameter learning of semi-CRFs. By incorporating the non-uniform (non-0/1) misclassification cost, this criterion is more directly related to the character error rate in contrast to the MAP rule which aims at minimizing the string error rate. With this method, the cost functions initially used for training HMMs in speech recognition can be conveniently applied to HCTR. We also investigate edge selection in MR training, attempting to improve the generalization ability and reduce the computational complexity. Further, we compare the proposed method with several prevalently used learning criteria, including conditional log-likelihood (CLL) [19], MCE [13], max-margin (MM) [17,18], and margin-based extensions of CLL and MCE. We believe that it is the first work that evaluates these training techniques on HCTR tasks. In experiments on three online handwriting databases, the proposed MR training method has yielded superior string recognition performances compared to the state-of-the-art methods.

This work is an extension of a conference paper [20]. The extension includes the comparison with other training criteria, the details of derivations, the effects of edge selection, extensive experimental results and discussions. The remainder of this paper is organized as follows: Section 2 reviews the related work. Section 3 gives a brief introduction to the semi-CRF model defined on the candidate lattice. Section 4 details the minimum-risk training framework for semi-CRFs. Section 5 describes the learning criteria for comparison. Section 6 presents our experimental results and Section 7 draws concluding remarks.

2. Related work

In speech recognition, it has been shown that discriminative learning of HMMs is able to produce consistent improvements in performance compared to the conventional maximum likelihood training criterion, which aims at modeling the data distribution instead of directly separating class categories [21]. In contrast, discriminative learning typically bypass the stage of building the joint-probability model while directly managing to minimize the classification errors. A central issue in the development of discriminative learning methods is the construction of objective function (learning criterion). Popular discriminative learning techniques for HMMs are MMI [12], MCE [13] and MPE/MWE [14]. MMI estimation tries to maximize the a posteriori probability of the training utterances, whereas in MCE training, an approximation to the sentence error rate on the training data is minimized. In contrast to MMI and MCE, which are typically designed to optimize the string-level errors, MPE/MWE aims at performance optimization at the substring pattern level, such as phones and words. Traditional discriminative training aims to find classification boundaries that minimize empirical error rates on training sets, which may not be well generalized to test sets [16]. Many attempts have been made to incorporate the principle of large-

margin into the training of HMMs to improve the generalization abilities (see [16] for a review). Do and Artières [22] also applied large-margin learning of HMMs to handwriting recognition. In [23], the modified MMI/MPE criteria are prosed, which allow large-margin training in speech recognition using the same optimization algorithms as the conventional MMI and MPE criteria. Similar extensions for MCE can be found in [24].

MAP training for CRFs is usually accomplished by minimizing the conditional log-likelihood (CLL) loss [19]. Starting from CLL, several discriminative training criteria have been proposed for parameter learning of CRFs, and superior recognition performances have been observed in contrast to MAP. Like modified MMI [23], softmax-margin CRF [25] is an extension of traditional CRF by incorporating a task-specific cost function into the CLL loss, and has been demonstrated to perform significantly better than CLL and max-margin on named-entity recognition problem. A similar method, which is referred to as large margin cost-sensitive learning of CRFs, is proposed in [26]. Analogous extension to semi-CRF for HCTR is proposed in our former work [5]. Inspired by MCE [13], Suzuki et al. [27] proposed a framework for training CRFs by optimizing approximated non-linear measures, and the maximum labelwise accuracy criterion proposed in [28] aims to maximize the per-label predictive accuracy on the training set. Stoyanov and Eisner [29] tried to minimize the empirical risk, i.e., the average task-specific loss on training data. Considering that in many cases the test performance only depends on the quality of marginal distribution, rather than a joint conditional distribution, Kakade et al. [30] proposed the average single-time prediction cost. Altun et al. [31] investigated different loss functions and optimization methods for discriminative learning of sequence labeling problems. Large-margin framework has been used in parameter learning of structured linear classifiers to improve the generalization ability [17,18,32]. Specially, this technique is applied to parameter estimation of semi-Markov models for phonetic recognition [33] as well as human action segmentation and recognition [34].

Risk minimization has been used for parameter learning of HMMs in the community of speech recognition. The overall risk criterion [35], whose calculation is based on the Levenshtein distance between the correct transcription and the N-best recognized transcriptions, can consistently decrease the recognition errors when compared to the standard maximum likelihood training. Minimum phone/word error (MPE/MWE) training [14] can be interpreted as an instance of minimum-risk training where the set of all possible phone/word sequences forms the hypothesis space. Use of this criterion has been shown to outperform the MMI criterion (the MAP rule) on several speech recognition tasks [14]. Heigold et al. [36] pointed out that MMI is vulnerable to label noise (i.e., incorrectly labeled training examples), while MPE/MWE tends to be less sensitive to outliers than MMI. In [37], several objective functions based on the MPE/MWE criterion are compared on the task of broadcast news recognition, and the results show that the most promising technique is the minimum phone frame error rule, which is a frame-level version of MPE/MWE. Gibson and Hain [38] evaluated different error approximation strategies based on the frame error metric and demonstrated that significant improvements can be observed on a large vocabulary speech recognition task when the symmetrically normalized frame error (SNFE) is adopted. In natural language processing, Smith and Eisner [39] trained log-linear models by risk minimization where the distribution over output variables is defined on N-best lists, and Li and Eisner [40] applied the expectation semiring to minimizing risk using dynamic programming. Xiong et al. [41] proposed a minimum tag error criterion for discriminative training of linear-chain CRFs, which is an average of the raw tag accuracy over all possible label sequences weighted by their likelihood. However, to the best of our knowledge, such risk minimization technique has not been applied to semi-CRFs, whose inference algorithms and parameter learning techniques need more computation than linear-chain CRFs. For HCTR, the only work exploiting this idea is the maximum character accuracy method [2], which takes the N-best list as the hypothesis space when calculating the risk. In contrast, the use of a lattice to represent the hypothesis space is favored because it is a more compact representation of usually many segmentation-recognition paths. Moreover, lattice-based training can directly take advantage of the inference algorithms of semi-CRFs.

Among the above mentioned training techniques, only several are used to learn HCTR models [2,3,5]. In our experiments, MR training is compared with several typical learning criteria on HCTR tasks, and their performances are evaluated on three public data sets.

3. Semi-CRFs for string recognition

In our previous work [5], a semi-CRF based approach for HCTR is proposed, in which the semi-CRF model is defined on the lattice (cf. Fig. 1) to directly estimate the a posteriori probability $P(S, Y|X;\Lambda)$ of a hypothesized segmentation-recognition path (S, Y) given the string X:

$$P(S,Y|X;\Lambda)=\frac{1}{Z(X;\Lambda)}\prod_{c\in S}\Psi_c(X,Y_c;\Lambda)=\frac{1}{Z(X;\Lambda)}\exp\{-E(X,S,Y;\Lambda)\},\quad(1)$$

where S denotes a segmentation (character sequence) of X and Y denotes a label sequence of S. $\Psi_c(X,Y_c;\Lambda)$ is the potential function on maximal clique c (consecutive characters with fixed length in the lattice):

$$\Psi_c(X,Y_c;\Lambda)=\exp\left\{\sum_{k=1}^{K}\lambda_k f_k(X_c,Y_c)\right\}.\quad(2)$$

$f_k(X_c,Y_c)$ is the k-th feature function defined on clique c, which models character recognition, geometric or linguistic context. We also refer to Y_c as a labeling of clique c. $\Lambda=\{\lambda_k|k=1,\ldots,K\}$ are the weighting parameters to be learned. $E(X,S,Y;\Lambda)$ is the energy function:

$$E(X,S,Y;\Lambda)=-\sum_{c\in S}\sum_{k=1}^{K}\lambda_k f_k(X_c,Y_c).\quad(3)$$

$Z(X;\Lambda)$ is the partition function defined as the summation over all the paths in the lattice:

$$Z(X;\Lambda)=\sum_{(S',Y')}\prod_{c\in S'}\Psi_c(X,Y'_c;\Lambda).\quad(4)$$

Given N training samples: $\{(X^i,S^i,Y^i)|i=1,\ldots,N\}$ (strings with segmentation points and character classes labeled), following the standard MAP estimation, the conventional training criterion is to minimize the conditional log-likelihood (CLL) loss with L_2-norm regularization:

$$L_{CLL}(\Lambda)=-\frac{1}{N}\sum_{i=1}^{N}\log P(S^i,Y^i|X^i;\Lambda)+\frac{C}{2}\|\Lambda\|^2\quad(5)$$

where C is a positive constant balancing the loss term against the regularization term.

Given a test string X, we first over-segment it into a component sequence and construct the segmentation-recognition candidate lattice (cf. Fig. 1). The Viterbi-like decoding tries to find the optimal path with maximum a posteriori probability:

$$(S^*,Y^*)=\arg\max_{(S,Y)}P(S,Y|X;\Lambda)=\arg\min_{(S,Y)}E(X,S,Y;\Lambda).\quad(6)$$

Table 1
Definitions and explanations of frequently used terms.

Component: A component, consisting a block of strokes, is hoped to be a character or part of a character after over-segmentation of the text line (cf. Fig. 1(a))
Candidate character: The candidate characters are generated by combining consecutive components, subject to constraints of character width (cf. Fig. 1(b))
Candidate class: Each candidate character in the lattice is assigned several candidate labels (classes) of high probability by a character classifier
Edge: Each character-label pair (a candidate character coupled with one of its candidate label) in the lattice is referred to as an edge
Sub-path: A sequence of consecutive edges in the lattice is referred to as a sub-path
Sub-segmentation path: A sequence of consecutive characters in the lattice is referred to as sub-segmentation-path
Maximal clique: A maximal clique is usually a sequence of m consecutive characters (sub-segmentation path) in the lattice, where m is called the maximal clique size and is predefined

In practice, to accelerate the recognition process, approximate decoding, such as beam search is usually adopted [2–5].

4. Minimum-risk training

Before introducing the minimum-risk training framework for HCTR, we first summarize the definitions of some frequently used terms in Table 1. For more details of these definitions, refer to [5].

In contrast to the CLL loss (cf. Eq. (5)) which aims to maximize the posterior probability on the training set, the minimum-risk (MR) criterion is to minimize the expected cost

$$L_{MR}(\Lambda)=\frac{1}{N}\sum_{i=1}^{N}\sum_{(S,Y)\in\mathcal{H}}P(S,Y|X^i;\Lambda)\ell((S,Y),(S^i,Y^i)), \tag{7}$$

where the summation space $\mathcal{H}$ is the entire lattice. $\ell((S,Y),(S^i,Y^i))$ signifies the cost when recognizing string X^i as (S,Y) instead of the ground-truth (S^i,Y^i), which is nonnegative and has the following property:

$$\begin{cases}\ell((S,Y),(S^i,Y^i))>0 & \text{if } (S,Y)\neq(S^i,Y^i)\\ \ell((S,Y),(S^i,Y^i))=0 & \text{if } (S,Y)=(S^i,Y^i)\end{cases} \tag{8}$$

As the commonly used evaluation measures of HCTR systems are derived from the Levenshtein distance (cf. Section 6.2), one would ideally like to take this metric as cost function in MR training. However, to calculate the Levenshtein distance between each path in the lattice (which usually encodes many paths) and the reference label sequence is computationally expensive. Heigold et al. [42] also demonstrated that it is not evident that more accurate error approximation during model estimation can lead to improved model generalization. In Eq. (7), to avoid explicitly enumerating numerous paths in the lattice, we consider the cost functions that can be decomposed onto each character along the hypothesized path

$$\ell((S,Y),(S^i,Y^i))=\sum_{q\in S}\tilde{\ell}((q,Y_q),(S^i,Y^i)), \tag{9}$$

where $\tilde{\ell}((q,Y_q),(S^i,Y^i))$ is the cost for an edge (character-label pair) (q,Y_q) on path (S,Y).

With the above defined cost function, the minimum-risk loss can be rewritten as

$$L_{MR}(\Lambda)=\frac{1}{N}\sum_{i=1}^{N}\sum_{(q,Y_q)\in\mathcal{H}}P(q,Y_q|X^i;\Lambda)\tilde{\ell}((q,Y_q),(S^i,Y^i)), \tag{10}$$

where $P(q,Y_q|X^i;\Lambda)$ denotes the marginal probability on (q,Y_q):

$$P(q,Y_q|X^i;\Lambda)=\sum_{(S,Y)\in\mathcal{H}:(q,Y_q)\in(S,Y)}P(S,Y|X^i;\Lambda). \tag{11}$$

A detailed derivation of Eq. (10) is provided in Appendix A. It can be seen from Eq. (10) that, to optimize the objective function with gradient descent, we should first calculate the derivatives of the marginal probabilities. Note that Eqs. (7) and (10) are actually regularized as in Eq. (5). We drop the regularization terms for succinct representation.

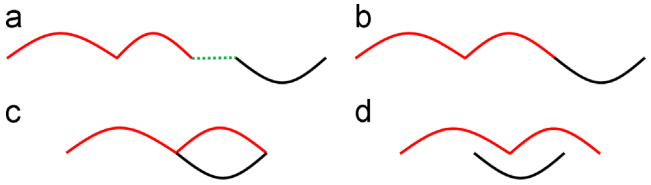

Fig. 2. Illustration of the relationships between a maximal clique (red) and a candidate character (black). The clique size is 2. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

4.1. Derivatives of marginal probabilities

The partial derivatives of $P(q,Y_q|X^i;\Lambda)$ with respect to the weighting parameters can be computed by

$$\frac{\partial P(q,Y_q|X^i;\Lambda)}{\partial\lambda_k}=\sum_{(c,Y_c)\in\mathcal{H}}f_k(X_c^i,Y_c)\times(P(c,Y_c,q,Y_q|X^i;\Lambda)-P(c,Y_c|X^i;\Lambda)P(q,Y_q|X^i;\Lambda)), \tag{12}$$

where c denotes the maximal clique (cf. Section 3). The detailed derivation of Eq. (12) can be found in Appendix B. The inference algorithms for calculating the marginal probabilities can be found in [5]. Here (q,Y_q) is relaxed to be an arbitrary sub-segmentation path (character sequence) and its labeling, i.e., Eq. (12) is actually applicable to any sub-paths in the lattice. In Eq. (12), to avoid summing over all clique-labeling pairs in the lattice, we adopt the approximation that if c and q are disjoint (Fig. 2(a)), i.e., there is at least one character between them, (c,Y_c) and (q,Y_q) are conditionally independent:

$$P(c,Y_c,q,Y_q|X^i;\Lambda)\approx P(c,Y_c|X^i;\Lambda)P(q,Y_q|X^i;\Lambda). \tag{13}$$

The assumption is reasonable because the constrains between disconnected handwritten characters are weak. With the above approximation, the summation space in Eq. (12) is reduced to those cliques that adjoin (Fig. 2(b)) or overlap (Fig. 2(c) and (d)) q. Note that $P(c,Y_c,q,Y_q|X^i;\Lambda)$ is the marginal probability that both (c,Y_c) and (q,Y_q) are on a hypothesized path, so if c and q overlaps but there is at least one common component whose label is different in Y_c and Y_q, $P(c,Y_c,q,Y_q|X^i;\Lambda)$ will be zero. Another case that makes $P(c,Y_c,q,Y_q|X^i;\Lambda)$ zero is that the overlapping part of c and q does not form common character(s) of them (Fig. 2(d)), considering that a segmentation path should be composed of concatenated characters (cf. Fig. 1(c)).

4.2. Cost functions

To facilitate lattice-based training (the summation space in Eq. (7) is the entire lattice), we select the cost functions which can decompose the errors along the hypothesized path (cf. Eq. (9)). In our experiments, three cost functions initially used for training HMMs in speech recognition are investigated, including the MPE cost [14], the Hamming distance (HD) cost (also called frame error

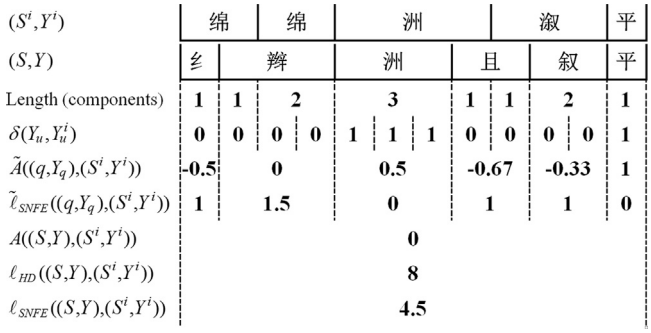

Fig. 3. Illustration of the calculation of cost functions.

in [43]) and the SNFE cost [38]. Unlike the 0/1 cost which characterizes whether a transcription is identical to the ground-truth or not, the non-uniform cost functions measure the character-level errors according to how many characters or components are incorrectly recognized. All the three cost functions are correlated with the Levenshtein distance [38], thus the minimization of the expected cost will lead to the reduction of character errors (substitutions, insertions and deletions) on the training set. In the following, we will detail the three cost functions and Fig. 3 illustrates the calculation process.

4.2.1. MPE cost

The MPE cost is derived from the accuracy function proposed in [14] for measuring the gains of classifying the genuine path (S^i, Y^i) as a hypothesized path (S, Y)

$$A((S,Y),(S^i,Y^i)) = \sum_{q \in S} \tilde{A}((q,Y_q),(S^i,Y^i)), \tag{14}$$

where (q, Y_q) is an edge on path (S, Y), and the edge accuracy $\tilde{A}((q,Y_q),(S^i,Y^i))$ is defined as

$$\tilde{A}((q,Y_q),(S^i,Y^i)) = \max_{q' \in S^i} \begin{cases} -1+2e(q,q') & \text{if } Y_q = Y^i_{q'} \\ -1+e(q,q') & \text{otherwise} \end{cases} \tag{15}$$

Here $(q', Y^i_{q'})$ is an edge on genuine path (S^i, Y^i), and $e(q, q')$ is defined as the common component number between q and q', divided by the component number of q'. With the above accuracy function, the MPE cost is defined as

$$\ell_{MPE}((S,Y),(S^i,Y^i)) = |Y^i| - A((S,Y),(S^i,Y^i)), \tag{16}$$

where $|Y^i|$ denotes the number of characters in the correct transcript. The per-edge cost (cf. Eq. (9)) can be written as

$$\tilde{\ell}_{MPE}((q,Y_q),(S^i,Y^i)) = \frac{|q| \cdot |Y^i|}{|S^i|} - \tilde{A}((q,Y_q),(S^i,Y^i)), \tag{17}$$

where $|q|$ and $|S^i|$ denote the length (in components) of q and S^i, respectively. Actually, considering that $|Y^i|$ is a constant, by substituting Eq. (16) into Eq. (7), we can derive that, when using the MPE cost, the learning criterion is equivalent to minimizing the following loss function:

$$L_{MR}(\Lambda) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{(q,Y_q)\in\mathcal{H}} P(q,Y_q|X^i;\Lambda)\tilde{A}((q,Y_q),(S^i,Y^i)). \tag{18}$$

In [14], to avoid the need for a dynamic programming alignment when calculating the Levenshtein distance, $A((S,Y),(S^i,Y^i))$ is used to approximate the true accuracy (the number of characters in the correct transcript minus the Levenshtein distance to the reference). Using the finite state transducer, Heigold et al. [42] proposed a training criterion aiming at minimizing the expected exact Levenshtein error, however, no significant differences in recognition performance are observed compared to the MPE/MWE criterion.

4.2.2. Hamming distance cost

The Hamming distance between two paths (S, Y) and (S^i, Y^i) is defined as the number of components at which the labels differ

$$\ell_{HD}((S,Y),(S^i,Y^i)) = \sum_u (1-\delta(Y_u, Y^i_u)), \tag{19}$$

where Y_u and $Y_u{}^i$ are the labels of component u on the hypothesized path (S, Y) and on the genuine path (S^i, Y^i), respectively. The label of a component on a path is identical to the label of the character containing it. Inspired by the work on minimum time frame error rate decoding [44], by substituting this cost function into Eq. (7), the risk loss can be rewritten as

$$L_{MR}(\Lambda) = \frac{1}{N}\sum_{i=1}^{N}\sum_u (1-P(Y^i_u|X^i;\Lambda)), \tag{20}$$

where $P(Y^i_u|X^i;\Lambda)$ is the marginal probability to observe Y^i_u at component u and can be calculated by summing over the probabilities of all edges overlapping component u and having identical label as Y^i_u:

$$P(Y^i_u|X^i;\Lambda) = \sum_{(q,Y_q)\in\mathcal{H}:u\in q \wedge Y^i_u = Y_q} P(q,Y_q|X^i;\Lambda). \tag{21}$$

in which (q, Y_q) denotes an edge in the lattice. The detailed derivation of Eqs. (20) and (21) are provided in Appendix C. Note that Eq. (19) can also be transformed into the form formulated in Eq. (9) by decomposing the errors onto each edge of (S, Y). However, instead of summing over all the lattice edges as in Eq. (10), the summation space in Eq. (20) is decomposed onto each component, and from Eq. (21) we can see that at each component, among the edges overlapping it, only those having identical label as the genuine component label are considered.

4.2.3. SNFE cost

In [38], limitations of the MPE cost are discussed, including the overestimation and the asymmetry (for two paths (S, Y) and (S', Y'), $A((S,Y),(S',Y')) \neq A((S',Y'),(S,Y))$) in error approximation, which causes an undesirable insertion to deletion bias. The SNFE cost [38] is also an approximation to the exact character errors. In contrast to the MPE cost, it is symmetric as the Levenshtein distance and yields more accurate approximations for the deletion and insertion errors. The SNFE cost between a hypothesized path (S, Y) and the reference (ground-truthed) path (S^i, Y^i) is defined as

$$\ell_{SNFE}((S,Y),(S^i,Y^i)) = \sum_{q\in S}\sum_{q'\in S^i} l((q,Y_q),(q',Y^i_{q'})), \tag{22}$$

where (q, Y_q) and $(q', Y^i_{q'})$ are edges on (S, Y) and (S^i, Y^i), respectively. $l((q,Y_q),(q',Y^i_{q'}))$ is defined as the number of overlapping components between q and q' at which Y_q and $Y^i_{q'}$ differ, divided by the smaller component number of q and q'. If no overlap exists between q and q', $l((q,Y_q),(q',Y^i_{q'}))$ is defined as zero. From Eq. (22), we can derive the per-edge cost (cf. Eq. (9)):

$$\tilde{\ell}_{SNFE}((q,Y_q),(S^i,Y^i)) = \sum_{q'\in S^i} l((q,Y_q),(q',Y^i_{q'})). \tag{23}$$

4.3. Edge selection

Reconsidering the derivatives of $P(q, Y_q|X^i;\Lambda)$ (cf. Eq. (12)), where (q, Y_q) denotes an edge in the lattice, Appendix D proves that when $P(q, Y_q|X^i;\Lambda)$ approaches the boundaries (0 or 1), its derivatives w.r.t. Λ will become close to zero. This means that when optimizing the MR criterion with gradient descent, Eq. (12) actually carries out an implicit edge selection, i.e., when updating

the model parameters, more contributions are donated by the confusing characters than those with almost definite decisions. To avoid calculating the derivatives on all lattice edges, we can also perform explicit edge selection, that is, only when $P(q, Y_q|X^i;\Lambda)$ satisfies the following condition, we calculate its derivatives:

$$\varepsilon < P(q, Y_q|X^i;\Lambda) < 1-\varepsilon, \tag{24}$$

where $0<\varepsilon<0.5$ is pre-defined. Otherwise, the derivatives are directly set as zero. For the three cost functions introduced in Section 4.2, the HD cost considers only the edges having identical labels as the overlapped components on the reference (ground-truthed) path (cf. Eq. (21)), while the MPE cost and the SNFE cost take all edges into account. Edge selection is equivalent to setting the per-edge cost $\tilde{\ell}((q, Y_q),(S^i, Y^i))$ (cf. Eq. (10)) to zero when $P(q, Y_q|X^i;\Lambda)$ violates the condition formulated in Eq. (24).

For discriminative training of acoustic models in speech recognition, Chen et al. [45] investigated utterance-level, phone-level and frame-level data selection in an attempt to reduce the time consumed in training. They pointed out that data selection is analogous to margin-based approaches, such as [15,46,47], which select the training samples (or fragments) close to the decision boundaries for better model discrimination and generalization. The counterparts for utterances, phones and frames in HCTR are strings, edges and components, respectively. From Eq. (21) we know that component-level selection can be accomplished by edge-level selection. In contrast to string-level selection, which may discard an entire text line, edge-level selection can utilize the training data more sufficiently.

5. Criteria for comparison

We compare the proposed method with several typical and widely used learning criteria, including CLL [19], MCE [13], max-margin (MM) [17,18], and the margin-based generalizations of CLL and MCE, which are referred to as softmax-margin (SMM) [25] and large-margin MCE (LM-MCE) [24], respectively. For margin-based methods (MM, SMM and LM-MCE in our experiments), the separation margin between the genuine path (S^i, Y^i) and a hypothesized path (S, Y) is chosen to scale with the error (cost) of choosing (S, Y) over the desired (S^i, Y^i). The cost $\ell((S, Y),(S^i, Y^i))$ should have the property formulated in Eq. (8). Here, like [17,25,26,33,47], we adopt Hamming distance (cf. Section 4.2.2) as the cost function. In contrast to CLL, SMM and MR, the optimization of MCE, LM-MCE and MM need to incorporate a decoding process to search for the most rival path.

5.1. CLL and softmax-margin criteria

Conditional log-likelihood (CLL, cf. Eq. (5)), which provides an upper bound on the empirical zero-one error rate [32], is the most commonly used objective for training CRFs under the MAP criterion. The convexity and differentiability ensure that gradient-based optimization procedures will not converge to suboptimal local minima of the objective function. However, considering the nature of 0/1 misclassification error, there is no guarantee that the parameters obtained by CLL training will lead to the best per-label predictive accuracy, even on the training set [28].

Softmax-margin CRFs [25] extend the CLL criterion by incorporating the margin concept into the loss function. The intuition is the same as that in max-margin learning (cf. Section 5.3): high-cost outputs should be penalized more heavily. This technique has also been applied to the training of HMMs in [47] and [23]. For HCTR, it is used for parameter learning of semi-CRFs in [5]. Gimpel and Smith [25] pointed out that SMM is a convex upper bound on CLL, risk and max-margin. Following [5], to introduce the margin concept into the CLL loss, we define the margin-posterior

$$P_m(S, Y|X^i;\Lambda) = \frac{\exp\{-E(X^i, S, Y;\Lambda)+\ell((S, Y),(S^i, Y^i))\}}{\sum_{(S', Y')}\exp\{-E(X^i, S', Y';\Lambda)+\ell((S', Y'),(S^i, Y^i))\}}. \tag{25}$$

Compared with the posterior formulated in Eq. (1), the margin-posterior includes a margin term $\exp(\ell((S, Y),(S^i, Y^i)))$, in which $\ell((S, Y),(S^i, Y^i))$ here adopts the Hamming distance between the genuine path (S^i, Y^i) and the rival path (S, Y). By replacing $P(S^i, Y^i|X^i)$ with $P_m(S^i, Y^i|X^i)$ in Eq. (5), we achieve the SMM criterion. Benefiting from the summation form of Hamming distance, the margin-based training criterion can take advantage of the same optimization algorithms as the conventional CLL-based training [5].

5.2. MCE and large-margin MCE criteria

The MCE criterion [13], which has been applied to HCTR in [3], minimizes an empirical loss corresponding to a smooth approximation of the string-level classification error rate. Following [3], we adopt 1-best MCE, which considers only the genuine path and the most competing path. For each training sample $(X^i, S^i, Y^i), i=1,\ldots,N$, the misclassification measure $d(X^i, S^i, Y^i;\Lambda)$ is defined as the energy difference between the genuine path and the most rival path:

$$d(X^i, S^i, Y^i;\Lambda) = E(X^i, S^i, Y^i;\Lambda) - \min_{(S,Y)\neq(S^i,Y^i)} E(X^i, S, Y;\Lambda). \tag{26}$$

Note that $d(X^i, S^i, Y^i;\Lambda)\geq 0$ indicates misclassification of string X^i under the 0/1 cost. Thus, the minimization of the string error rate can be rewritten as the minimization of averaged 0/1 losses on the training data:

$$L_{MCE}(\Lambda) = \frac{1}{N}\sum_{i=1}^{N}\mathbf{1}(d(X^i, S^i, Y^i;\Lambda)\geq 0). \tag{27}$$

However, the indicator function $\mathbf{1}(\cdot)$, which takes value 1 when the condition in the parentheses is satisfied, otherwise takes value 0, is not appropriate for optimization since it is discontinuous w.r.t. the parameters Λ. A typical choice is to approximate the indicator function with a sigmoidal function which is differentiable:

$$L_{MCE}(\Lambda) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{1+\exp(-\xi d(X^i, S^i, Y^i;\Lambda))}. \tag{28}$$

Via the smoothing parameter ξ, the MCE loss function can be made arbitrarily close to the binary classification error. Like the CLL criterion (cf. Eq. (5)), Eq. (28) is actually regularized.

Large-margin MCE (LM-MCE) extends the conventional MCE loss by incorporating a non-zero sigmoid bias, which can be interpreted as a soft margin [24]. Here, the bias term is chosen to be the error when predicting the genuine path as the most rival path:

$$d_m(X^i, S^i, Y^i;\Lambda) = E(X^i, S^i, Y^i;\Lambda) - \min_{(S,Y)\neq(S^i,Y^i)}\{E(X^i, S, Y;\Lambda)-\ell((S, Y),(S^i, Y^i))\}. \tag{29}$$

By replacing $d(X^i, S^i, Y^i;\Lambda)$ with $d_m(X^i, S^i, Y^i;\Lambda)$ in Eq. (28), we achieve the LM-MCE loss. Like MCE, the objective function of LM-MCE is not convex and the training is subject to the local minimum problem. As has been pointed out in [16], LM-MCE bears the same weaknesses as MCE, i.e., both methods optimize the sentence error rate on the training set. Stochastic gradient descent is generally used to optimize MCE and LM-MCE, and the most rival path in Eqs. (26) and (29) can be found by N-best based beam search [5].

5.3. Max-margin criterion

The max-margin learning framework for training structured prediction models inherits the merits of support vector machines (SVMs), which exhibit good generalization ability on unseen data. A popular implementation of this framework is the margin scaling method [17,18], which tries to ensure that the score of the correct labeling is separated from the score of the predicted by a margin in proportion to the error caused by the prediction. Kim et al. [33] apply this framework to discriminative training of semi-Markov models for phonetic recognition. Following [33], the goal of max-margin training for semi-CRFs is to find the parameter vector Λ such that the difference in energy of a predicted path (S, Y) from the correct path (S^i, Y^i) is at least $\ell((S,Y),(S^i,Y^i))$. Formally, this can be formulated as a convex programming problem

$$\begin{aligned}\min_{\Lambda,\boldsymbol{\xi}} \quad & \frac{C}{2}\|\Lambda\|^2+\frac{1}{N}\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \quad & \Delta E(X^i,S,Y;\Lambda)\geq \ell((S,Y),(S^i,Y^i))-\xi_i, \\ & \forall (S,Y)\neq (S^i,Y^i), \quad i=1,\ldots,N, \\ & \xi_i\geq 0, \quad i=1,\ldots,N,\end{aligned} \tag{30}$$

where $\Delta E(X^i,S,Y;\Lambda)=E(X^i,S,Y;\Lambda)-E(X^i,S^i,Y^i;\Lambda)$ and $C>0$ is a constant that controls the tradeoff between margin maximization and training error minimization. Considering the nonnegativity of $\ell((S,Y),(S^i,Y^i))$ (cf. Eq. (8)), the optimization problem formulated in Eq. (30) can be rewritten as an equivalent but unconstrained form [33], with the following loss function:

$$L_{MM}(\Lambda)=\frac{C}{2}\|\Lambda\|^2+\frac{1}{N}\sum_{i=1}^{N}\max_{(S,Y)\in\mathcal{H}}\left\{-\Delta E(X^i,S,Y;\Lambda)+\ell((S,Y),(S^i,Y^i))\right\}. \tag{31}$$

Due to the hard-max operation, Eq. (31) is not differentiable with respect to Λ. As [33], stochastic subgradient descent [48] is adopted to optimize this convex objective function. N-best based beam search [5] is used to find the most competing path in Eq. (31):

$$(S^*,Y^*)=\arg\min_{(S,Y)\in\mathcal{H}}\{E(X^i,S,Y;\Lambda)-\ell((S,Y),(S^i,Y^i))\}. \tag{32}$$

In contrast to LM-MCE, the most competing path in MM could be the genuine path itself. Note that the SMM criterion can be obtained by approximating the hard-max with soft-max in Eq. (31), considering that soft-max is a tight upper bound of hard-max [25,26,47].

6. Experiments

We first evaluated the proposed method on unconstrained online handwritten text lines of a Chinese handwriting database CASIA-OLHWDB [49] and a Japanese handwriting database TUAT Kondate [4]. The test sets contain 10,510 text lines (269,674 characters of 2631 classes) and 3511 text lines (35,766 characters of 791 classes), respectively. Further, we compared with the best results in ICDAR 2011 Chinese handwriting recognition competition [50] on the same test data of online handwritten texts (3432 text lines, including 91,576 characters of 1375 classes), which is provided by the same group as the CASIA-OLHWDB database [49]. The text lines of all the three databases have been annotated with segmentation points and character labels.

6.1. Feature functions

The features functions employed in our experiments include character classification, geometric and linguistic contexts [5]. The character recognition scores are given by a character classifier (7356 classes for Chinese and 4438 classes for Japanese). Unless otherwise stated, the default character classifier is MQDF and the feature dimensionality is reduced from 512D to 160D by Fisher linear discriminant analysis (FDA) [51]. The scores for class-dependent and class-independent geometries are given by the quadratic discriminant functions (QDFs) and the linear SVMs, respectively. All the classifier outputs are transformed to confidences. Trigram language models are used in both training and decoding in our experiments. The details of classifier training and language models can be found in [5].

6.2. Performance metrics

Following [2,5,10], the string recognition performance is evaluated by character-level correct rate (CR) and accurate rate (AR) derived by aligning the result string with the transcript using dynamic programming. We also use the term character error rate (CER), which equals $1-AR$. And we denote the error rates of three error types (substitution, deletion and insertion) by SUB, DEL and INS, respectively. The string-level performance is measured by string error rate (SER), which is as the percentage of mis-recognized strings. The complexity of summation space $\mathcal{H}$ (cf. Eq. (7)) is measured by lattice edge density (LED), which is defined as the total number of edges divided by the total number of characters in the transcript.

6.3. Experimental results

Unless otherwise specified, the following settings are applicable to all the learning criteria investigated. We implemented the methods in MS Visual C++ 2008 and tested on a PC with Intel Quad Core 2.83 GHz CPU and 4 GB-RAM. In training, the string samples were processed iteratively for five cycles in stochastic gradient (or subgradient for max-margin) decent. Trigram language models were used in both training and decoding. Following our previous work [5], the candidate class number was set as 12 in training and 10 in testing, when constructing the lattice (cf. Fig. 1). To alleviate the computational burden, in training the lattice was first pruned with a pre-trained first-order semi-CRF with pruning threshold 12, and the default decoding method is ratio threshold based beam search with threshold 10. Enlarging these values will increase the time cost of training or decoding, while the improvement of performance is limited. The regularization constant C is tuned on the training data and set to 0.01 for all learning criteria (the performances are insensitive to C when it is not too large, e.g., $C<0.1$). In MCE and LM-MCE training, the smoothing parameter ξ (cf. Eq. (28)) increases progressively from 1 to 2, such that the loss approaches hard 0/1 decision. For CLL, SMM and MM, if not stated otherwise, training text lines with illegible characters were discarded to get higher recognition rates (cf. Section 6.3.5).

6.3.1. Effects of cost functions

Table 2 compares the string recognition results on test string sets together with the training time (averaged over the training string number times the iteration number in stochastic gradient descent as in [5]) for cost functions introduced in Section 4.2 (for MR training) and the conventional 0/1 cost (for MAP training using CLL loss). Since risk is non-convex [39], we took the learned parameters from CLL as initialization before optimization. The effects with and without pre-training were evaluated in Section 6.3.2. From Table 2 we can see that in contrast to the 0/1 cost, the use of non-uniform costs can generally improve the string recognition rates (AR and CR) and the correct rates of almost all character types. The HD cost and the SNFE cost outperform the MPE cost on both the two test string sets. The performances of

Table 2
Recognition results (including the correct rates for different character types) (%) and training time (s/str) for different cost functions.

Costs	AR	CR	Chinese	Symbol	Digit	Letter	Time
0/1	93.66	94.26	95.51	85.64	91.72	87.23	2.35
MPE	93.88	94.46	95.70	85.84	92.24	86.97	3.36
HD	93.95	94.53	95.76	85.98	92.47	86.97	2.82
SNFE	93.96	94.54	95.77	85.92	92.50	87.23	3.42
0/1	93.58	94.62	97.95	92.04	95.06	93.12	1.09
MPE	94.23	95.40	98.06	93.03	96.60	94.38	1.78
HD	94.34	95.43	98.15	93.10	96.42	94.20	1.35
SNFE	94.28	95.42	98.09	93.05	96.66	94.26	1.83

Upper – CASIA-OLHWDB; lower – TUAT Kondate. "Chinese" denotes Chinese characters or Japanese Kanji, "Symbol" includes both symbol characters and kana characters for Japanese.

Table 3
Error rates (%) on test strings.

Costs	SER	CER	SUB	DEL	INS
0/1	58.25	6.34	4.82	0.92	0.60
MPE	57.11	6.12	4.69	0.86	0.57
HD	56.80	6.05	4.65	0.82	0.58
SNFE	56.66	6.04	4.63	0.84	0.57
0/1	31.42	6.42	4.43	0.96	1.03
MPE	29.02	5.77	3.74	0.86	1.17
HD	28.62	5.66	3.68	0.89	1.08
SNFE	28.77	5.72	3.74	0.84	1.14

Upper – CASIA-OLHWDB; lower – TUAT Kondate.

SNFE and HD are comparable, however, the training time with HD is much lower than that with SNFE due to less computational complexity (cf. Section 4.2.2).

Table 3 lists the error rates on the test string sets for the four cost functions (0/1, MPE, HD and SNFE), from which we can see that MR training outperforms MAP training on both character error rates and string error rates. Although the 0/1 cost aims at minimizing the string-level errors, the non-uniform costs still achieve lower SERs by reducing the character-level errors. Compared to MAP training, MR training can reduce all the three types of errors (substitutions, deletions and insertions) on CASIA-OLHWDB, while slightly increasing the insertion errors on TUAT Kondate. With comparable CERs, the SERs on CASIA-OLHWDB are much higher than those on TUAT Kondate. This is because the text lines in the former database are usually much longer than those in the latter (25.66 characters per string vs. 10.19 characters per string, on average).

6.3.2. *Effects of pre-training*

In contrast to CLL, risk is non-convex [39], therefore the optimization procedure is prone to getting stuck in local optima. Inspired by the work of [29], in stochastic gradient descent, we took the learned parameters from CLL as initialization for risk minimization (Tables 2 and 3). This is because the convexity of the CLL loss can generally guarantee that the parameters are in the right region. Table 4 compares the recognition results with and without pre-training (the results are identical to those in Table 2 for the case with pre-training), from which we can see that pre-training significantly improves the performance on TUAT Kondate database which has less training samples, while the improvement is just limited on CASIA-OLHWDB having more training strings. Hereafter, if not stated otherwise, pre-training will be adopted for MR training.

Table 4
Recognition results for minimum-risk learning with (w/) and without (w/o) pre-training (%).

Criterion	Pre-training	AR	CR	AR	CR
MPE	w/	93.88	94.46	94.23	95.40
	w/o	93.85	94.47	93.07	93.85
HD	w/	93.95	94.53	94.34	95.43
	w/o	93.89	94.53	93.01	93.72
SNFE	w/	93.96	94.54	94.28	95.42
	w/o	93.92	94.53	92.51	93.26

Left – CASIA-OLHWDB; right – TUAT Kondate.

6.3.3. *Effects of lattice pruning*

For each training sample, the summation space $\mathcal{H}$ for calculating the risk is composed of all the paths in the lattice (cf. Eq. (7)). Although the lattice has greatly reduced the hypothesis space by assigning only a candidate class list to each candidate character rather than the entire state set which contains thousands of categories, to incorporate more competing paths, the initially constructed lattice is usually dense and comprises of many implausible edges. To alleviate the computational burden in training, following our previous work [5], we resorted to the forward–backward lattice pruning method, which reduces the lattice complexity while reserving the most rival paths. The risk was calculated on the reduced lattice.

Using the HD cost, we evaluated the effects of lattice pruning on string recognition errors. The results are shown in Fig. 4(a), in which γ denotes the pruning threshold. By enlarging γ, more edges will be reserved in the lattice. From Fig. 4(a) we can see that CER will saturate when γ grows large enough. The default lattice pruning threshold 12 (the corresponding LED is 3.41 for CASIA-OLHWDB and 4.75 for TUAT Kondate) performs sufficiently well with respect to the CERs. Increasing γ, though incorporates more rival paths, does not improve the performance.

6.3.4. *Effects of edge selection*

The purpose of edge selection is to reduce the computation cost in MR training. In Eq. (24), by enlarging ε, more edges will be filtered out in training and only the most confusing ones are selected for calculating the derivatives and updating the model parameters. Using the HD cost, on test text lines, Fig. 4(b) illustrates the character error rates over different ε, in which $\varepsilon=0$ means without edge selection. From Fig. 4(b) we can see that the character error rates with and without edge selection are comparable even with relatively larger ε (more edges are filtered out). For $\varepsilon=0.01$, Table 5 lists the string recognition rates on test string sets together with the proportion of edges discarded in training. As a comparison, the results without edge selection (cf. Table 2) are also provided. From Table 5 we can see that the recognition rates for the two cases are comparable, while a large part of edges are filtered out by edge selection, and consequently the amount of computations in optimization is reduced. The reason that the proportions for HD are relatively lower is because the HD cost has already left out those edges which do not have identical labels as the overlapped components on the reference path (cf. Eq. (21)).

Table 2 has shown that, without edge selection, MR takes a much longer training time than CLL. This is because only the derivatives on the reference path need to be calculated in CLL optimization [5], while for MR training, we need to calculate the derivatives on edges in the lattice (cf. Eq. (10)). In Table 6, the running time of MR training (using HD cost, with and without edge selection) is compared with that of CLL, MCE and MM. With comparable computational complexity as CLL and MCE

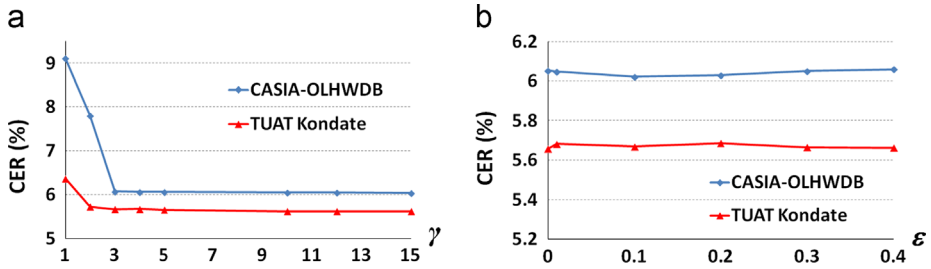

Fig. 4. (a) Effects of lattice pruning and (b) effects of edge selection.

Table 5
Recognition results for minimum-risk learning with (w/) and without (w/o) edge selection (%).

Criterion	Selection	AR	CR	Prop.	AR	CR	Prop.
MPE	w/	93.90	94.47	92.19	94.22	95.39	78.99
	w/o	93.88	94.46	0	94.23	95.40	0
HD	w/	93.95	94.51	28.33	94.32	95.41	62.56
	w/o	93.95	94.53	0	94.34	95.43	0
SNFE	w/	93.97	94.54	92.43	94.29	95.44	78.96
	w/o	93.96	94.54	0	94.28	95.42	0

Left – CASIA-OLHWDB; right – TUAT Kondate. "Prop." denotes the proportion of edges filtered out.

Table 6
Comparison of training time for different learning criteria (s/str).

Database	MR (w/)	MR (w/o)	CLL	MCE	MM
CASIA-OLHWDB	2.42	2.82	2.35	1.76	1.73
TUAT Kondate	1.15	1.35	1.09	0.68	0.64

"MR(w/)" and "MR(w/o)" denote minimum risk training with and without edge selection, respectively.

respectively, SMM and LM-MCE are not listed in Table 6, for the calculation of the margin term takes just a small amount of time. Table 6 shows that, by edge selection, the running time of MR training is almost comparable with that of CLL. However, it is still higher than MCE and MM. To calculate the derivatives in CLL and MR training, the exact forward and backward algorithms [5] are conducted to compute the marginal probabilities, while for MCE and MM, only the approximate forward algorithm (beam search) is run to search for the competing path (cf. Sections 5.2 and 5.3). This the reason why MCE and MM are more effective than CLL and MR in optimization.

6.3.5. Robustness of training criteria

For HCTR, due to variable writing styles of different authors, sloppy handwritings are commonly observed. In this part, we evaluated the robustness of training criteria to the quality of training samples. Recall the lattice construction procedure (cf. Fig. 1), in which each candidate character is given a candidate class list by a character classifier. In this step, the ground-truthed label of an illegibly written character or an outlier (incorrectly labeled character) may be excluded. We consider two strategies to deal with this case. The first is to simply discard the training strings containing such characters, and the second is to insert the ground-truthed labels into the candidate class lists. Training with the above two strategies are separately referred to as T_{rmv} and T_{ins}, and the corresponding recognition results are compared in Table 7. The results on CASIA-OLHWDB show that to remove the training samples (for T_{rmv}, 10,743 text lines were discarded from 41,710 training strings) containing illegible characters (or outliers) have

Table 7
Effects of training samples' quality (%).

Criterion	Strategy	AR	CR	AR	CR
CLL	T_{rmv}	93.66	94.26	93.58	94.62
	T_{ins}	93.46	93.99	93.57	94.66
SMM	T_{rmv}	93.73	94.33	94.05	95.18
	T_{ins}	93.65	94.20	94.00	95.16
MCE	T_{rmv}	93.67	94.29	94.07	95.24
	T_{ins}	93.71	94.32	94.06	95.23
LM-MCE	T_{rmv}	93.79	94.36	94.15	95.30
	T_{ins}	93.79	94.35	94.13	95.28
MM	T_{rmv}	93.74	94.36	94.03	95.21
	T_{ins}	93.67	94.26	94.05	95.22
MR	T_{rmv}	93.94	94.52	94.33	95.41
	T_{ins}	93.95	94.53	94.34	95.43

Left – CASIA-OLHWDB; right – TUAT Kondate.

just slight influence on the performances of MCE, LM-MCE and MR, however, for CLL, SMM and MM, T_{rmv} achieved higher recognition rates than T_{ins} even using less training samples. This means CLL, SMM and MM are more sensitive to illegible characters and outliers, that is, MCE, LM-MCE and MR are more robust to low quality training samples than CLL, SMM and MM. On TUAT Kondate, only 296 text lines were discarded from 9793 training samples for T_{rmv}, which is the reason that the results for the two cases are comparable. Unless otherwise stated, T_{rmv} strategy, which can be taken as a string-level training data selection (cf. Section 4.3), is adopted for CLL, SMM and MM to achieve better performances in our experiments.

In stochastic gradient (or subgradient) descent, the semi-CRF parameters are updated using the derivatives of the loss functions. For CLL and SMM, the derivative of the logarithm $\log a$ diverges at $a = 0$. Thus when a training string (X^i, S^i, Y^i) contains illegible characters (or outliers), which usually makes $P(S^i, Y^i|X^i;\Lambda)$ a very small value approaching zero, the magnitude of the derivative will be very large. This is the reason why CLL and SMM are sensitive to low quality training samples. In contrast, for the sigmoid function $f(a) = 1/(1+e^{-a})$ in MCE and LM-MCE criteria, the derivative $f(a)(1-f(a))$ will become close to zero for extremely large misclassification measures (cf. Eqs. (26) and (29)). Yu et al. [24] also mentioned that MCE has the property of immunity to the outliers (i.e., incorrectly labeled training examples). The derivative of the MM criterion includes the difference of path features (summation of feature functions along the path, cf. Eq. (32)). The reason that MM is vulnerable to illegible characters is because the feature functions used are the logarithms of classification confidences [5], which are very small values for illegible characters. The derivative of the MR loss is determined by the derivatives of the marginal probabilities on lattice edges (cf. Eq. (10)). The marginal probability $P(q, Y_q|X^i;\Lambda)$ measures the possibility that the edge (q, Y_q) is on the genuine path, which will be close to zero if the similarity degree between q and Y_q is very low. In Appendix D, we have

Table 8
Effects of character classifiers (%).

Criterion	Classifier	AR	CR	AR	CR
CLL	FDA+MQDF	93.66	94.26	93.58	94.62
	DFE+MQDF	94.30	94.84	94.07	95.06
	DFE+DLQDF	94.49	95.15	92.71	93.60
SMM	FDA+MQDF	93.73	94.33	94.05	95.18
	DFE+MQDF	94.34	94.88	94.52	95.55
	DFE+DLQDF	94.54	95.20	93.00	93.91
MCE	FDA+MQDF	93.71	94.32	94.06	95.23
	DFE+MQDF	94.36	94.90	94.56	95.63
	DFE+DLQDF	94.54	95.21	92.99	93.90
LM-MCE	FDA+MQDF	93.79	94.35	94.13	95.28
	DFE+MQDF	94.38	94.90	94.76	95.73
	DFE+DLQDF	94.58	95.22	93.05	93.95
MM	FDA+MQDF	93.74	94.36	94.03	95.21
	DFE+MQDF	94.36	94.91	94.49	95.54
	DFE+DLQDF	94.54	95.22	92.97	93.89
MR	FDA+MQDF	93.95	94.53	94.34	95.43
	DFE+MQDF	94.51	95.03	94.87	95.86
	DFE+DLQDF	94.69	95.32	93.14	94.10

Left – CASIA-OLHWDB; right – TUAT Kondate.

proved that if $P(q, Y_q|X^i;\Lambda)$ approaches zero, its derivatives will also become close to zero, and thus the edge (q, Y_q) will not contribute much to the updating of model parameters. This the reason why MR is insensitive to low quality training samples. Heigold et al. [36] also pointed out that MMI is vulnerable to label noise (outliers), while MPE/MWE tends to be less sensitive to outliers than MMI on speech recognition tasks.

6.3.6. *Effects of character classifiers*

In the semi-CRF based HCTR approach, character classification is used in both candidate class selection and feature functions [5]. Using different feature dimensionality reduction and character classification methods, we compared the performances of MR training with other learning criteria introduced in Section 5. Before character classification, the feature dimensionality was first reduced to 160D. To achieve higher string recognition rates, we considered discriminative feature extraction (DFE) [52] and discriminative learning QDF (DLQDF) [53] in addition to the baseline FDA and MQDF. DFE optimizes the feature subspace (initialized by FDA) under a discriminative learning criterion, and DLQDF is a discriminatively updated version of MQDF [53]. The training set for DFE and DLQDF is the same as that for FDA and MQDF. On segmented string characters, DFE and DLQDF can generally achieve higher correct rates (top-1 accuracies) than FDA and MQDF due to discriminative learning [5]. Table 8 shows the recognition results on test string sets using different combinations of dimensionality reduction (FDA, DFE) and classification (MQDF, DLQDF) methods. For fair comparison with MR, pre-training (cf. Section 6.3.2) is also adopted for SMM, MCE, LM-MCE and MM.

For each training criterion, on CASIA-OLHWDB, superior string recognition rates are obtained by DFE+DLQDF, while on TUAT Kondate, DFE+MQDF performs better. The reason is because the string recognition performance owes much to the cumulative accuracies of character classifiers rather than the top-1 accuracy [5]. The cumulative accuracies (for top-10 candidates) for FDA+MQDF, DFE+MQDF and DFE+DLQDF on segmented test string characters of CASIA-OLHWDB are separately 97.75%, 98.04% and 98.71%, while on TUAT Kondate, the accuracies are 98.19%, 98.38% and 97.68%, respectively.

We put the emphasis on the comparison of different train criteria. For each character classifier, we can see that SMM outperforms CLL and LM-MCE outperforms MCE by incorporating the margin term. MCE achieves comparable or better recognition results than CLL, SMM and MM due to the robustness of sigmoid function (cf. Section 6.3.5), even when training data selection were conducted for CLL, SMM and MM. The performances of MM and SMM are just comparable. Among all the training criteria, MR achieves the best recognition performance due to the optimization of character-level errors and the robustness to low quality training samples. However, by comparing the results achieved by MR and those achieved by the baseline CLL, it can be seen that changing the learning criterion is not as effective as changing the character classification methods.

Table 9
Comparison with the best results of ICDAR 2011 Chinese handwriting competition (online handwritten texts).

System	Classifier	LM	AR (%)	CR (%)
Proposed	FDA+MQDF	char tri-gram	93.28	93.83
Proposed	DFE+MQDF	char tri-gram	93.99	94.45
Proposed	DFE+DLQDF	char tri-gram	94.22	94.76
Zhou et al. [5]	DFE+DLQDF	char tri-gram	94.06	94.62
VO	MLP	word tri-gram	93.56	94.33

6.3.7. *Experiments on competition set*

Finally, we conducted experiments on the test set of online handwritten texts in ICDAR 2011 Chinese handwriting recognition competition [50], in which the best results were achieved by Vision Objects Ltd. (VO), which adopts multilayer perceptron (MLP) as the character classifier. In Table 9, three semi-CRF based HCTR models using different character classification methods (cf. Section 6.3.6) are compared with the system of VO, in which the model parameters are learned by MR using HD cost and edge selection ($\varepsilon = 0.01$). As recommended by the competition, all the three models are trained with the samples of the entire CASIA-OLHWDB database. Table 9 shows that even with DFE+MQDF and a relatively weaker linguistic model (LM), the recognition rates (AR and CR) are already higher than those of VO. The best results are given by DFE+DLQDF, which are also higher than the best results achieved by our former work [5].

7. Conclusion

This paper presents a minimum-risk training method for handwritten Chinese/Japanese text recognition using semi-CRFs, which aims at minimizing the character error rate rather than the string error rate by taking advantage of the non-uniform (non-0/1) cost functions. An experimental evaluation on CASIA-OLHWDB database and TUAT Kondate database shows that minimum-risk training yields better string recognition rates than several widely used learning criteria, but changing the learning criterion is not as effective as changing the character classification methods. The HD cost [43] and the SNFE cost [38] outperform the MPE cost [14] on test sets of both the two databases. The performances of SNFE and HD are comparable, while the training time with HD is much lower than that with SNFE due to less computational complexity. Edge selection can help to reduce the computation cost of minimum-risk training. The proposed method also outperforms the best system on the test set (online handwritten texts) of ICDAR 2011 Chinese handwriting recognition competition.

Conflict of interest statement

None declared.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFC) under Grants nos. 61273269, 60933010, 61203296, 61232013 and the Chongqing Science & Technology Commission under Grants no. cstc2013yykfB0233. The authors would like to thank TUAT Nakagawa laboratory for providing the Japanese databases.

Appendix A. Derivation of Eq. (10)

With the cost function formulated in Eq. (9), the per-sample loss in Eq. (7) can be calculated by

$$\begin{aligned}
&\sum_{(S,Y)\in\mathcal{H}} P(S,Y|X^i;\Lambda)\ell((S,Y),(S^i,Y^i))\\
&\quad=\sum_{(S,Y)\in\mathcal{H}} P(S,Y|X^i;\Lambda)\sum_{q\in S}\tilde{\ell}((q,Y_q),(S^i,Y^i))\\
&\quad=\sum_{(S,Y)\in\mathcal{H}}\sum_{q\in S} P(S,Y|X^i;\Lambda)\tilde{\ell}((q,Y_q),(S^i,Y^i))\\
&\quad=\sum_{(q,Y_q)\in\mathcal{H}(S,Y)\in\mathcal{H}:(q,Y_q)\in(S,Y)}\ \sum P(S,Y|X^i;\Lambda)\tilde{\ell}((q,Y_q),(S^i,Y^i))\\
&\quad=\sum_{(q,Y_q)\in\mathcal{H}}\tilde{\ell}((q,Y_q),(S^i,Y^i))\sum_{(S,Y)\in\mathcal{H}:(q,Y_q)\in(S,Y)} P(S,Y|X^i;\Lambda)\\
&\quad=\sum_{(q,Y_q)\in\mathcal{H}}\tilde{\ell}((q,Y_q),(S^i,Y^i))P(q,Y_q|X^i;\Lambda)
\end{aligned}\tag{A.1}$$

From the above per-sample loss, we can derive Eq. (10) computed on the whole training set.

Appendix B. Derivation of Eq. (12)

From the definition of $P(S,Y|X^i;\Lambda)$ (cf. Eq. (1)), the marginal probability $P(q,Y_q|X^i;\Lambda)$ formulated in Eq. (11) can be rewritten as

$$P(q,Y_q|X^i;\Lambda)=\sum_{(S,Y)\in\mathcal{H}:(q,Y_q)\in(S,Y)}\frac{\exp\{-E(X^i,S,Y;\Lambda)\}}{Z(X^i;\Lambda)},\tag{B.1}$$

thus the partial derivatives of $P(q,Y_q|X^i;\Lambda)$ with respect to the model parameters can be calculated by

$$\begin{aligned}
\frac{\partial P(q,Y_q|X^i;\Lambda)}{\partial\lambda_k}&=-P(q,Y_q|X^i;\Lambda)\frac{\partial\log Z(X^i;\Lambda)}{\partial\lambda_k}\\
&\quad+\sum_{(S,Y)\in\mathcal{H}:(q,Y_q)\in(S,Y)}\ \sum_{c\in S} f_k(X_c^i,Y_c)P(S,Y|X^i;\Lambda),
\end{aligned}\tag{B.2}$$

in which

$$\begin{aligned}
&\sum_{(S,Y)\in\mathcal{H}:(q,Y_q)\in(S,Y)}\ \sum_{c\in S} f_k(X_c^i,Y_c)P(S,Y|X^i;\Lambda)\\
&\quad=\sum_{(c,Y_c)\in\mathcal{H}(S,Y)\in\mathcal{H}:(c,Y_c)\in(S,Y)\wedge(q,Y_q)\in(S,Y)}\ \sum f_k(X_c^i,Y_c)P(S,Y|X^i;\Lambda)\\
&\quad=\sum_{(c,Y_c)\in\mathcal{H}} f_k(X_c^i,Y_c)\sum_{(S,Y)\in\mathcal{H}:(c,Y_c)\in(S,Y)\wedge(q,Y_q)\in(S,Y)} P(S,Y|X^i;\Lambda)\\
&\quad=\sum_{(c,Y_c)\in\mathcal{H}} f_k(X_c^i,Y_c)P(c,Y_c,q,Y_q|X^i;\Lambda)
\end{aligned}\tag{B.3}$$

From the definition of $Z(X^i;\Lambda)$ (cf. Eq. (4)), we can calculate the partial derivatives of $\log Z(X^i;\Lambda)$ in Eq. (B.2) by

$$\begin{aligned}
\frac{\partial\log Z(X^i;\Lambda)}{\partial\lambda_k}&=\sum_{(S,Y)\in\mathcal{H}}\ \sum_{c\in S} f_k(X_c^i,Y_c)P(S,Y|X^i;\Lambda)\\
&=\sum_{(c,Y_c)\in\mathcal{H}} f_k(X_c^i,Y_c)P(c,Y_c|X^i;\Lambda)
\end{aligned}\tag{B.4}$$

By substituting Eqs. (B.3) and (B.4) into Eq. (B.2), we can derive Eq. (12).

Appendix C. Derivation of Eq. (20)

By substituting the HD cost (cf. Eq. (19)) into Eq. (7), the per-sample loss can be calculated by

$$\begin{aligned}
&\sum_{(S,Y)\in\mathcal{H}} P(S,Y|X^i;\Lambda)\sum_u(1-\delta(Y_u,Y_u^i))\\
&\quad=\sum_u\sum_{(S,Y)\in\mathcal{H}} P(S,Y|X^i;\Lambda)(1-\delta(Y_u,Y_u^i))\\
&\quad=\sum_u\left(1-\sum_{(S,Y)\in\mathcal{H}}\delta(Y_u,Y_u^i)P(S,Y|X^i;\Lambda)\right)\\
&\quad=\sum_u(1-P(Y_u^i|X^i;\Lambda))
\end{aligned}\tag{C.1}$$

in which

$$\begin{aligned}
P(Y_u^i|X^i;\Lambda)&=\sum_{(S,Y)\in\mathcal{H}}\delta(Y_u,Y_u^i)P(S,Y|X^i;\Lambda)\\
&=\sum_{(S,Y)\in\mathcal{H}q\in S:u\in q}\ \sum\delta(Y_u,Y_u^i)P(S,Y|X^i;\Lambda)\\
&=\sum_{(S,Y)\in\mathcal{H}_{q\in S:u\in q\wedge Y_u^i=Y_q}}\ \sum P(S,Y|X^i;\Lambda)\\
&=\sum_{(q,Y_q)\in\mathcal{H}:u\in q\wedge Y_u^i=Y_q}\ \sum_{(S,Y)\in\mathcal{H}:(q,Y_q)\in(S,Y)} P(S,Y|X^i;\Lambda)\\
&=\sum_{(q,Y_q)\in\mathcal{H}:u\in q\wedge Y_u^i=Y_q} P(q,Y_q|X^i;\Lambda)
\end{aligned}\tag{C.2}$$

From the above per-sample loss, we can derive Eq. (20) computed on the whole training set.

Appendix D. Edge selection in Eq. (12)

In Eq. (12), considering that

$$\begin{aligned}
&|P(c,Y_c,q,Y_q|X^i;\Lambda)-P(c,Y_c|X^i;\Lambda)P(q,Y_q|X^i;\Lambda)|\\
&\quad\le\max\{P(c,Y_c,q,Y_q|X^i;\Lambda),P(c,Y_c|X^i;\Lambda)P(q,Y_q|X^i;\Lambda)\}\\
&\quad\le P(q,Y_q|X^i;\Lambda),
\end{aligned}\tag{D.1}$$

we can prove that when $P(q,Y_q|X^i;\Lambda)$ approach zero, the left side of the above inequality will also approach zero. On the other hand, considering that

$$\begin{aligned}
P(c,Y_c,q,Y_q|X^i;\Lambda)&=P(c,Y_c|X^i;\Lambda)+P(q,Y_q|X^i;\Lambda)\\
&\quad-P((c,Y_c)\cup(q,Y_q)|X^i;\Lambda),
\end{aligned}\tag{D.2}$$

where $P((c,Y_c)\cup(q,Y_q)|X^i;\Lambda)$ denotes the probability that (c,Y_c) or (q,Y_q) is on the desired path, we can derive that

$$\begin{aligned}
&|P(c,Y_c,q,Y_q|X^i;\Lambda)-P(c,Y_c|X^i;\Lambda)P(q,Y_q|X^i;\Lambda)|\\
&\quad\le P((c,Y_c)\cup(q,Y_q)|X^i;\Lambda)-P(q,Y_q|X^i;\Lambda)\\
&\quad\quad+P(c,Y_c|X^i;\Lambda)(1-P(q,Y_q|X^i;\Lambda)).
\end{aligned}\tag{D.3}$$

Thus, when $P(q,Y_q|X^i;\Lambda)$ approach 1, the left side of the above inequality will approach zero. Based on the above conclusions, we can prove that when $P(q,Y_q|X^i;\Lambda)$ approaches 0 or 1, its derivatives will become close to zero.

References

[1] M. Cheriet, N. Kharma, C.L. Liu, C.Y. Suen, Character Recognition Systems: A Guide for Students and Practitioners, John Wiley & Sons, Inc., 2007.
[2] Q.F. Wang, F. Yin, C.L. Liu, Handwritten Chinese text recognition by integrating multiple contexts, IEEE Trans. Pattern Anal. Mach. Intell. 34 (8) (2012) 1469–1481.
[3] D.H. Wang, C.L. Liu, X.D. Zhou, An approach for real-time recognition of online Chinese handwritten sentences, Pattern Recognit. 45 (10) (2012) 3661–3675.
[4] B. Zhu, X.D. Zhou, C.L. Liu, M. Nakagawa, A robust model for on-line handwritten Japanese text recognition, Int. J. Doc. Anal. Recognit. 13 (2) (2010) 121–131.

[5] X.D. Zhou, D.H. Wang, F. Tian, C.L. Liu, M. Nakagawa, Handwritten Chinese/Japanese text recognition using semi-Markov conditional random fields, IEEE Trans. Pattern Anal. Mach. Intell. 35 (10) (2013) 2413–2426.
[6] S. Sarawagi, W. Cohen, Semi-Markov conditional random fields for information extraction, Neural Inf. Process. Syst. 17 (2005) 1185–1192.
[7] J. Lafferty, A. McCallum, F. Pereira, Conditional random fields: probabilistic models for segmenting and labeling sequence data, in: Proceedings of the 18th ICML, 2001, pp. 282–289.
[8] F. Kimura, K. Takashina, S. Tsuruoka, Y. Miyake, Modified quadratic discriminant functions and its application to Chinese character recognition, IEEE Trans. Pattern Anal. Mach. Intell. 9 (1) (1987) 149–153.
[9] R. Duda, P. Hart, D. Stork, Pattern Classification, 2nd edition, Wiley, New York, 2001.
[10] T.H. Su, T.W. Zhang, D.J. Guan, H.J. Huang, Off-line recognition of realistic Chinese handwriting using segmentation-free strategy, Pattern Recognit. 42 (1) (2009) 167–182.
[11] V. Vapnik, The Nature of Statistical Learning Theory, 2nd edition, Springer-Verlag, New York, 1999.
[12] L.R. Bahl, P.F. Brown, P.V. DeSouza, R.L. Mercer, Maximum mutual information estimation of hidden Markov model parameters for speech recognition, in: Proceedings of the ICASSP, 1986, pp. 49–52.
[13] B.H. Juang, S. Katagiri, Discriminative learning for minimum error classification, IEEE Trans. Signal Process. 40 (12) (1992) 3043–3054.
[14] D. Povey, Discriminative training for large vocabulary speech recognition (Ph.D. dissertation), Cambridge University, 2003.
[15] J. Li, M. Yuan, C.H. Lee, Approximate test risk bound minimization through soft margin estimation, IEEE Trans. Audio, Speech, Lang. Proc. 15 (8) (2007) 2393–2404.
[16] D. Yu, L. Deng, Large-margin discriminative training of hidden Markov models for speech recognition, in: Proceedings of the 1st ICSC, 2007, pp. 429–438.
[17] B. Taskar, C. Guestrin, D. Koller, Max-margin Markov networks, Neural Inf. Process. Syst. 16 (2003).
[18] I. Tsochantaridis, T. Hofmann, T. Joachims, Y. Altun, Support vector machine learning for interdependent and structured output spaces, in: Proceedings of the 21th ICML, 2004, pp. 104–112.
[19] F. Sha, F. Pereira, Shallow parsing with conditional random fields, in: Proceedings of the NAACL-HLT, 2003, pp. 213–220.
[20] X.D. Zhou, F. Tian, C.L. Liu, Minimum risk training for handwritten Chinese/Japanese text recognition using semi-Markov conditional random fields, in: Proceedings of the 12th ICDAR, 2013, pp. 940–944.
[21] X. He, L. Deng, W. Chou, Discriminative learning in sequential pattern recognition – a unifying review for optimization-oriented speech recognition, IEEE Signal Process. Mag. 25 (5) (2008) 14–36.
[22] T.M.T. Do, T. Artières, Maximum margin training of Gaussian HMMs for handwriting recognition, in: Proceedings of the 10th ICDAR, 2009, pp. 976–980.
[23] G. Heigold, T. Deselaers, R. Schlüter, H. Ney, Modified MMI/MPE: a direct evaluation of the margin in speech recognition, in: Proceedings of the 25th ICML, 2008, pp. 384–391.
[24] D. Yu, L. Deng, X. He, A. Acero, Large-margin minimum classification error training: a theoretical risk minimization perspective, Comput. Speech Lang. 22 (4) (2008) 415–429.
[25] K. Gimpel, N.A. Smith, Softmax-margin CRFs: training log-linear models with loss functions, in: Proceedings of the NAACL-HLT, 2010, pp. 733–736.
[26] M.Y. Kim, Large margin cost-sensitive learning of conditional random fields, Pattern Recognit. 43 (10) (2010) 3683–3692.
[27] J. Suzuki, E. McDermott, H. Isozaki, Training conditional random fields with multivariate evaluation measures, in: Proceedings of the COLING-ACL, 2006, pp. 217–224.
[28] S. Gross, O. Russakovsky, C. Do, S. Batzoglou, Training conditional random fields for maximum labelwise accuracy, Neural Inf. Process. Syst. 19 (2006).
[29] V. Stoyanov, J. Eisner, Minimum-risk training of approximate CRF-based NLP systems, in: Proceedings of the NAACL-HLT, 2012, pp. 120–130.
[30] S. Kakade, Y.W. Teh, S.T. Roweis, An alternate objective function for Markovian fields, in: Proceedings of the 19th ICML, 2002, pp. 275–282.
[31] Y. Altun, M. Johnson, T. Hofmann, Investigating loss functions and optimization methods for discriminative learning of label sequences, in: Proceedings of the EMNLP, 2003.
[32] Y. Altun, T. Hofmann, Large margin methods for label sequence learning, in: Proceedings of the EuroSpeech, 2003.
[33] S. Kim, S. Yun, C.D. Yoo, Large margin discriminative semi-Markov model for phonetic recognition, IEEE Trans. Audio, Speech, Lang. Proc. 19 (7) (2011) 1999–2012.
[34] Q. Shi, L.Wang, L. Cheng, A.J. Smola, Discriminative human action segmentation and recognition using semi-Markov model, in: Proceedings of the CVPR, 2008, pp. 1–8.
[35] J. Kaiser, B. Horvat, Z. Kacic, Overall risk criterion estimation of hidden Markov model parameters, Speech Commun. 38 (3–4) (2002) 383–398.
[36] G. Heigold, P. Dreuw, S. Hahn, R. Schlüter, H. Ney, Margin-based discriminative training for string recognition, IEEE J. Sel. Top. Signal Process. – Stat. Learn. Methods Speech Lang. Process. 4 (6) (2010) 917–925.
[37] D. Povey, B. Kingsbury, Evaluation of proposed modifications to MPE for large scale discriminative training, in: Proceedings of the ICASSP, 2007, pp. 321–324.
[38] M. Gibson, T. Hain, Error approximation and minimum phone error acoustic model estimation, IEEE Trans. Audio, Speech, Lang. Proc. 18 (6) (2010) 1269–1279.
[39] D. Smith, J. Eisner, Minimum risk annealing for training log-linear models, in: Proceedings of the COLING-ACL, 2006, pp. 787–794.
[40] Z. Li, J. Eisner, First- and second-order expectation semirings with applications to minimum-risk training on translation forests, in: Proceedings of the EMNLP, 2009.
[41] Y. Xiong, J. Zhu, H. Huang, H. Xu, Minimum tag error for discriminative training of conditional random fields, Inf. Sci. 179 (1–2) (2009) 169–179.
[42] G. Heigold, W. Macherey, R. Schluter, H. Ney, Minimum exact word error training, in: Proceedings of the ASRU, 2005, pp. 186–190.
[43] J. Zheng, A. Stolcke, Improved discriminative training using phone lattices, in: Proceedings of the Interspeech, 2005, pp. 2125–2128.
[44] F. Wessel, R. Schlüter, H. Ney, Explicit word error minimization using word hypothesis posterior probabilities, in: Proceedings of the ICASSP, 2001, pp. 33–36.
[45] B. Chen, S.H. Liu, F.H. Chu, Training data selection for improving discriminative training of acoustic models, Pattern Recognit. Lett. 30 (13) (2009) 1228–1235.
[46] H. Jiang, X.W. Li, C.J. Liu, Large margin hidden Markov models for speech recognition, IEEE Trans. Audio, Speech, Lang. Proc. 14 (5) (2006) 1584–1595.
[47] F. Sha, L.K. Saul, Large margin hidden Markov models for automatic speech recognition, Proc. Neural Inf. Process. Syst. 19 (2007) 1249–1256.
[48] N. Ratliff, J.A. Bagnell, M. Zinkevich, (Online) subgradient methods for structured prediction, in: Proceedings of the 11th AISTATS, 2007, pp. 2:380–387.
[49] C.L. Liu, F. Yin, D.H. Wang, Q.F. Wang, CASIA online and offline Chinese handwriting databases, in: Proceedings of the 11th ICDAR, 2011, pp. 37–41.
[50] C.L. Liu, F. Yin, Q.F. Wang, D.H. Wang, ICDAR 2011 Chinese handwriting recognition competition, in: Proceedings of the 11th ICDAR, 2011, pp. 1464–1469.
[51] C.L. Liu, X.D. Zhou, Online Japanese character recognition using trajectory-based normalization and direction feature extraction, in: Proceedings of the 10th IWFHR, 2006, pp. 217–222.
[52] C.L. Liu, R. Mine, M. Koga, Building compact classifier for large character set recognition using discriminative feature extraction, in: Proceedings of the 8th ICDAR, 2005, pp.846–850.
[53] C.L. Liu, High accuracy handwritten Chinese character recognition using quadratic classifiers with discriminative feature extraction, in: Proceedings of the 18th ICPR, 2006, pp.942–945.

Xiang-Dong Zhou is an associate professor at the Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences. He received the B.S. degree in Applied Mathematics, the M.S. degree in Management Science and Engineering from National University of Defense Technology, Changsha, China, the Ph.D. degree in pattern recognition and artificial intelligence from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 1998, 2003 and 2009, respectively. He was a postdoctoral fellow at Tokyo University of Agriculture and Technology from March 2009 to March 2011. From May 2011 to October 2013, he was a research assistant and later an associate professor at the Institute of Software, Chinese Academy of Sciences. His research interests include handwriting recognition and ink document analysis.

Yan-Ming Zhang is currently an assistant professor at National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences. He received the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences in 2011. He got his bachelor degree from the Beijing University of Posts and Telecommunications. His research interests include machine learning and pattern recognition.

Feng Tian is a professor in Institute of Software, Chinese Academy of Sciences where he manages the pen-based and multimodal user interface research group in Intelligence Engineering Lab. He earned his Ph.D. degree in Institute of Software, Chinese Academy of Sciences in 2003. His research interests include theories, interaction techniques and tools in Natural User Interface, Pen-based UI, Multi-modal UI, and other new UI styles. He has published over 60 papers in HCI field, including ACM CHI, ACM CSCW, ACM IUI, ACM TIST, etc. He also serves as the chair of ACM SIGCHI China chapter now.

Hong-An Wang received the Ph.D. degree from the Institute of Software, Chinese Academy of Sciences, Beijing, China, in 1999. He is a Professor of the Institute of Software, Chinese Academy of Science. His research interests include real-time intelligence and user interface.

Cheng-Lin Liu is a Professor at the National Laboratory of Pattern Recognition (NLPR), Institute of Automation of Chinese Academy of Sciences, Beijing, China, and is now the deputy director of the laboratory. He received the B.S. degree in electronic engineering from Wuhan University, Wuhan, China, the M.E. degree in electronic engineering from Beijing Polytechnic University, Beijing, China, the Ph.D. degree in pattern recognition and intelligent control from the Chinese Academy of Sciences, Beijing, China, in 1989, 1992 and 1995, respectively. He was a postdoctoral fellow at Korea Advanced Institute of Science and Technology (KAIST) and later at Tokyo University of Agriculture and Technology from March 1996 to March 1999. From 1999 to 2004, he was a research staff member and later a senior researcher at the Central Research Laboratory, Hitachi, Ltd., Tokyo, Japan. His research interests include pattern recognition, image processing, neural networks, machine learning, and especially the applications to character recognition and document analysis. He has published over 170 technical papers at prestigious international journals and conferences. He is on the editorial board of journals Pattern Recognition, Image and Vision Computing, and International Journal on Document Analysis and Recognition. He is a fellow of the IAPR, and a senior member of the IEEE.