Pattern Recognition 46 (2013) 2313–2322

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

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

A discriminative linear regression approach to adaptation of multi-prototype based classifiers and its applications for Chinese OCR

Jun Du [a,*], Qiang Huo [b]

[a] National Engineering Laboratory for Speech and Language Information Processing (NEL-SLIP), University of Science and Technology of China, No. 96, JinZhai Road, Hefei, Anhui, PR China
[b] Microsoft Research Asia, 13/F, Building 2, No. 5 Danling Street, Haidian District, Beijing, PR China

ARTICLE INFO

Article history:
Received 18 April 2012
Received in revised form
9 January 2013
Accepted 14 January 2013
Available online 23 January 2013

Keywords:
Discriminative linear regression
Sample separation margin
Minimum classification error
Rprop
Adaptation
OCR

ABSTRACT

This paper presents a new discriminative linear regression approach to adaptation of a discriminatively trained prototype-based classifier for Chinese OCR. A so-called sample separation margin based minimum classification error criterion is used in both classifier training and adaptation, while an Rprop algorithm is used for optimizing the objective function. Formulations for both model-space and feature-space adaptation are presented. The effectiveness of the proposed approach is confirmed by a series of experiments for adaptation of font styles and low-quality text, respectively.

1. Introduction

With the fast development of mobile internet, OCR-based applications are becoming increasingly more popular (e.g., [1–4]). However, the most off-the-shelf OCR engines were trained on scanned documents, and they may not work well for new application scenarios where the properties of the captured character images are significantly different from the ones in the training data set (e.g., [5]). One of the solutions to address this problem is to adapt a pre-trained classifier to deal with the new scenario by using the document to be recognized itself via an *unsupervised adaptation* strategy, or by using a small amount of adaptation data collected in the target scenario via a *supervised adaptation* strategy.

Using adaptation to improve the OCR accuracy has been a research topic for several decades. Some ingenious ideas specific to OCR have been tried. For example, Nagy et al. [6,7] demonstrated that a character classifier trained on many typefaces can be adapted effectively to text in a single unknown typeface by using a *self-adaptation* strategy. Hong [8] showed how to use an adaptation strategy that alternates between applying "visual constraints" and "linguistic constraints" to reduce errors for recognizing books printed in a single typeface. Authors in [9–13] investigated a family of style-conscious algorithms to improve the recognition accuracy on documents which contain only a few typefaces and limited variations in image qualities and other variabilities. More recently, Xiu and Baird [14] demonstrated how to use a self-adaptation technique to improve the whole-book recognition, where a so-called iconic model and a linguistic model are mutually leveraged and adapted.

In the past several decades, many effective adaptation techniques have also been invented for speaker adaptation in the area of speech recognition (see an overview paper [15] and the references therein) and for writer adaptation in the area of handwriting recognition (see a recent work in [16,17] and the references therein). Some of them have been applied to OCR adaptation. For example, in [18], a continuous density hidden Markov model (CDHMM) based English character recognizer was adapted to deal with unseen fonts with two popular adaptation algorithms, namely maximum-likelihood linear regression (MLLR) [19,20] and maximum *a posteriori* (MAP) estimation [21], which were developed originally in speech recognition area for speaker adaptation. Apparently, the style transfer mapping (STM) technique proposed in [16,17] for writer adaptation can also be used for OCR adaptation.

In this paper, we study the adaptation techniques for Chinese OCR. One of the state-of-the-art techniques to build a Chinese OCR engine is to use a discriminatively trained prototype-based classifier as reported in [22]. In spite of the large vocabulary of Chinese characters, such a classifier can be made both compact

* Corresponding author. Tel.: +86 551 63607863; fax: +86 551 63607863.
E-mail addresses: jundu@ustc.edu.cn (J. Du), qianghuo@microsoft.com (Q. Huo).

0031-3203/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2013.01.021

(e.g., [22,23]) and efficient in the recognition stage (e.g., [24]). A high recognition accuracy can be achieved by using Gabor features, LDA (linear discriminant analysis) or MCE (minimum classification error) based discriminative feature extraction, and MCE-based classifier parameter training [22]. Recently, a so-called sample separation margin (SSM) based MCE training approach was proposed in [25] for training prototype-based classifiers, which performs better than the MCE training approach in [22]. In [23], the SSM-MCE training approach has been used to construct a state-of-the-art compact prototype-based handwritten Chinese character recognizer where a batch-mode Quickprop algorithm [26] is used for optimizing the SSM-MCE objective function. In [27], the SSM-MCE formulation has been extended to training pattern classifiers with quadratic discriminant functions (QDF), including the "modified quadratic discriminant function (MQDF)" [28] popular in the areas of OCR and handwriting recognition for East Asian languages. In this study, we have built our baseline classifier for Chinese OCR by using the techniques described in [22,25,23] with a minor difference: we used an Rprop algorithm (e.g., [29,30]) to optimize the SSM-MCE objective function because the setting of control parameters is much easier than the Quickprop algorithm used in [23].

The main contribution of this paper is to propose a new SSM-MCE linear regression (LR) approach to adaptation of an SSM-MCE trained prototype-based classifier and demonstrate its effectiveness for Chinese OCR as an illustrative example. Formulations for both model-space and feature-space adaptation are presented. In terms of general concept, our work is related to the MCE-LR approach reported in [31] for speaker adaptation of CDHMM-based speech recognizer, where a traditional MCE objective function is used. Our work is also relevant to the STM work on writer adaptation for handwritten Chinese character recognition reported in [16,17], where a similar MCE training approach as in [22] is used to train a prototype-based classifier, but a least regularized weighted squared error approach is used to estimate a global feature transform (a.k.a. style transfer mapping (STM)) for writer adaptation. The adaptation capability of the STM approach is similar to our feature-space adaptation approach, but is inferior to our model-space adaptation approach because multiple transforms can be used for model adaptation. Even for feature-space approach, our experimental results show that our approach performs significantly better than the original STM approach in [16] for both supervised and unsupervised adaptation of font styles and low-quality text, respectively, which confirms that SSM-MCE is a better objective function to learn the feature transform.

Fig. 1 illustrates an overall system development flow of our work in this paper. In training stage, after feature extraction of training samples, an LBG clustering algorithm [32] is used to construct multiple prototypes for each character class. Then a baseline classifier is constructed by using the SSM-MCE training. For model-space adaptation, an adapted classifier is constructed by using the linear regression transform(s) estimated from the adaptation data and the baseline classifier, which will be used to recognize unknown characters in the target scenario. For feature-space adaptation, a global feature transform is estimated from the adaptation data, which will be used to transform the feature vector of the unknown character back into the feature space of the training data so that the baseline classifier can be used in recognition stage.

It is noted that the preliminary results of this study have been published in [33]. The current paper is an extended version of the above report by including more detailed descriptions of relevant procedures, reporting additional experimental results and findings, and adding new figures and references to make the presentation more readable and accessible.

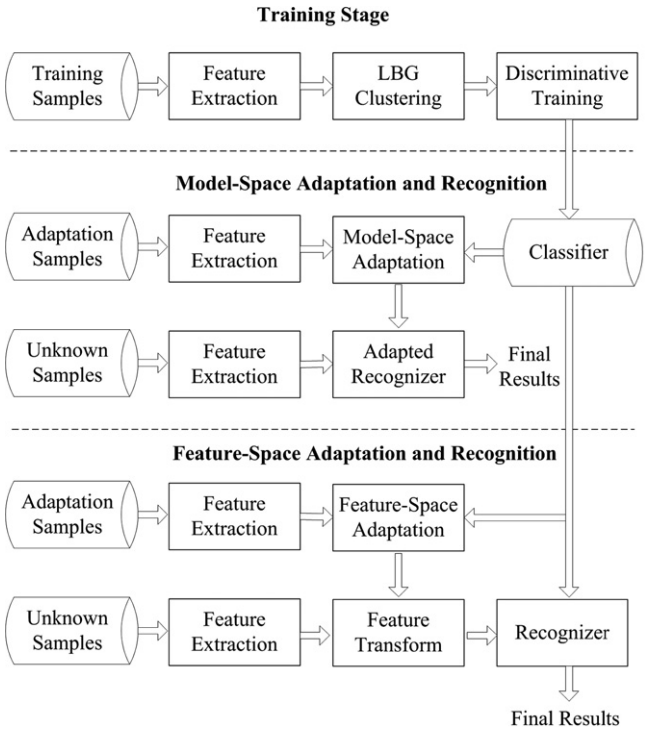

Fig. 1. Overall flow of system development.

The remainder of the paper is organized as follows. In Section 2, we describe briefly how to construct a multi-prototype based classifier by using the SSM-MCE training. In Section 3, we present formulations of SSM-MCE LR for both model-space and feature-space adaptation. Several important implementation issues are discussed in Section 4. In Section 5, we report experimental results for adaptation of font styles and low-quality text, respectively. Finally we conclude the paper in Section 6.

2. SSM-MCE training of a multi-prototype based classifier

Suppose our classifier can recognize M character classes denoted as $\{C_i | i=1,\ldots,M\}$. For a multi-prototype based classifier, each class C_i is represented by K_i prototypes, $\boldsymbol{\lambda}_i=\{\boldsymbol{m}_{ik}\in\mathcal{R}^D | k=1,\ldots,K_i\}$, where $\boldsymbol{m}_{ik}$ is the kth prototype of the ith class. Let us use $\boldsymbol{\Lambda}=\{\boldsymbol{\lambda}_i\}$ to denote the set of prototypes. In the classification stage, a feature vector $\boldsymbol{x}\in\mathcal{R}^D$ is first extracted. Then $\boldsymbol{x}$ is compared with each of the M classes by evaluating a Euclidean distance based discriminant function for each class C_i as follows:

$$g_i(\boldsymbol{x};\boldsymbol{\lambda}_i)=-\min_k \|\boldsymbol{x}-\boldsymbol{m}_{ik}\|^2. \tag{1}$$

The class with the maximum discriminant function score is chosen as the recognized class $r(\boldsymbol{x};\boldsymbol{\Lambda})$, i.e.,

$$r(\boldsymbol{x};\boldsymbol{\Lambda})=\arg\max_i g_i(\boldsymbol{x};\boldsymbol{\lambda}_i). \tag{2}$$

In the training stage, given a set of training data $\boldsymbol{X}=\{\boldsymbol{x}_r\in\mathcal{R}^D | r=1,\ldots,R_1\}$, first we initialize $\boldsymbol{\Lambda}$ by LBG clustering [32]. Then $\boldsymbol{\Lambda}$ can be re-estimated by minimizing the following MCE objective function:

$$l(\boldsymbol{X};\boldsymbol{\Lambda})=\frac{1}{R_1}\sum_{r=1}^{R_1}\frac{1}{1+\exp[-\alpha d(\boldsymbol{x}_r;\boldsymbol{\Lambda})+\beta]}, \tag{3}$$

where α, β are two control parameters, and $d(\boldsymbol{x}_r;\boldsymbol{\Lambda})$ is a misclassification measure defined by using a so-called sample

separation margin (SSM) as follows [25]:

$$d(\boldsymbol{x}_r;\boldsymbol{\Lambda})=\frac{-g_p(\boldsymbol{x}_r;\boldsymbol{\lambda}_p)+g_q(\boldsymbol{x}_r;\boldsymbol{\lambda}_q)}{2\|\boldsymbol{m}_{p\hat{k}}-\boldsymbol{m}_{q\bar{k}}\|}, \tag{4}$$

where

$$\hat{k}=\arg\min_k\|\boldsymbol{x}_r-\boldsymbol{m}_{pk}\|^2, \tag{5}$$

$$q=\arg\max_{i\in\mathcal{M}_r}g_i(\boldsymbol{x}_r;\boldsymbol{\lambda}_i), \tag{6}$$

$$\bar{k}=\arg\min_k\|\boldsymbol{x}_r-\boldsymbol{m}_{qk}\|^2, \tag{7}$$

and $\mathcal{M}_r$ is the hypothesis space for the rth sample, excluding the true label p.

To optimize the objective function, in [23], a modified Quickprop procedure is used. In this work, an Rprop algorithm described in [30] is adopted. The detailed formulation is given in Appendix A.

3. SSM-MCE linear regression

To adapt an OCR engine, we can adapt the classifier to the new scenario (i.e., model-space method) or adapt the observed features in the new scenario back to the original feature space (i.e., feature-space method).

3.1. Model-space method

Suppose we are given a set of labeled adaptation data $\boldsymbol{Y}=\{\boldsymbol{y}_r\in\mathcal{R}^D|r=1,\ldots,R_2\}$ collected in the target application scenario. For model-space method, we transform the parameters of the original classifier as follows:

$$\hat{\boldsymbol{m}}_{ik}=\mathcal{F}(\boldsymbol{m}_{ik};\boldsymbol{\Theta})=\boldsymbol{A}_{e_i}\boldsymbol{m}_{ik}+\boldsymbol{b}_{e_i}, \tag{8}$$

where i and k are indices of class and prototype, respectively; and e_i is the transform index for the ith class. Let us use $\boldsymbol{\Theta}=\{(\boldsymbol{A}_e,\boldsymbol{b}_e)|e=1,\ldots,E\}$ to denote the set of transform parameters, where $\boldsymbol{A}_e$ is a $D\times D$ nonsingular matrix and $\boldsymbol{b}_e$ is a D-dimensional bias vector. The SSM-MCE objective function is defined as follows:

$$l(\boldsymbol{Y};\boldsymbol{\Lambda},\boldsymbol{\Theta})=\frac{1}{R_2}\sum_{r=1}^{R_2}\frac{1}{1+\exp[-\alpha d(\boldsymbol{y}_r;\boldsymbol{\Lambda},\boldsymbol{\Theta})+\beta]}, \tag{9}$$

where

$$d(\boldsymbol{y}_r;\boldsymbol{\Lambda},\boldsymbol{\Theta})=\frac{-g_p(\boldsymbol{y}_r;\hat{\boldsymbol{\lambda}}_p)+g_q(\boldsymbol{y}_r;\hat{\boldsymbol{\lambda}}_q)}{2\|\hat{\boldsymbol{m}}_{p\hat{k}}-\hat{\boldsymbol{m}}_{q\bar{k}}\|}, \tag{10}$$

$\hat{\boldsymbol{\lambda}}_p$ and $\hat{\boldsymbol{\lambda}}_q$ represent the prototype set after model-space transformation using Eq. (8) for the pth and qth class, respectively.

We use again an Rprop algorithm to optimize $\boldsymbol{\Theta}$ with a procedure detailed in Appendix B.

3.2. Feature-space method

For feature-space method, the following global feature transformation function is used:

$$\boldsymbol{x}_r=\mathcal{F}(\boldsymbol{y}_r;\boldsymbol{\Theta})=\boldsymbol{A}\boldsymbol{y}_r+\boldsymbol{b}, \tag{11}$$

where $\boldsymbol{A}$ is a $D\times D$ nonsingular matrix, $\boldsymbol{b}$ is a D-dimensional bias vector, $\boldsymbol{y}_r$ and $\boldsymbol{x}_r$ are the rth D-dimensional input and transformed feature vectors, respectively.

The SSM-MCE objective function is defined as follows:

$$l(\boldsymbol{Y};\boldsymbol{\Lambda},\boldsymbol{\Theta})=\frac{1}{R_2}\sum_{r=1}^{R_2}\frac{1}{1+\exp[-\alpha d(\boldsymbol{y}_r;\boldsymbol{\Lambda},\boldsymbol{\Theta})+\beta]}, \tag{12}$$

where

$$d(\boldsymbol{y}_r;\boldsymbol{\Lambda},\boldsymbol{\Theta})=\frac{-g_p(\boldsymbol{x}_r;\boldsymbol{\lambda}_p)+g_q(\boldsymbol{x}_r;\boldsymbol{\lambda}_q)}{2\|\boldsymbol{m}_{p\hat{k}}-\boldsymbol{m}_{q\bar{k}}\|}. \tag{13}$$

The optimization procedure for $\boldsymbol{\Theta}$ is almost the same as Appendix B except the derivatives of the objective function, which are listed in Appendix C. In recognition stage, the estimated transform $\{\boldsymbol{A},\boldsymbol{b}\}$ is used to transform the feature vector of each unknown character first, which is then fed to baseline classifier for recognition.

3.3. Defining regression classes

In the previous formulation of model-space method, the LR transforms are tied across character classes, where each transform is associated with a set of character classes. To design a fully automatic adaptation procedure for any given amount of labeled adaptation data, we use a regression class tree to group the character classes, just like what has been done in MLLR [20]. As shown in Fig. 2, a binary regression tree is constructed to cluster similar character classes. Each leaf node holds a bucket of character classes, while each internal node holds the set of character classes from its descendants. Starting from the root node which holds all the character classes, the regression tree is constructed as follows until the specified number of leaf nodes is reached:

- *Step* 1: Select a leaf node to split. All the prototypes of character classes in this leaf node are pooled together as samples.
- *Step* 2: Use LBG algorithm [32] to cluster the above samples into two clusters with a centroid calculated for each cluster.
- *Step* 3: For each character class in the parent node, it is classified to one of the above two clusters which gives the smaller total Euclidean distance between the prototypes of this character and the centroid of the corresponding cluster.
- *Step* 4: Go to Step 1 if the specified number of leaf nodes is not reached; Stop, otherwise.

In the adaptation stage, given a fixed amount of adaptation data, a maximum rooted subtree is obtained from the above regression class tree such that each leaf node in the subtree has at least N_T adaptation samples. For each leaf node of the subtree, a regression class will be assigned and an LR transform will be trained. In this way, the number of regression classes E can be determined automatically. N_T is a control parameter set empirically.

3.4. Discussion

For notational convenience, we refer to hereinafter our SSM-MCE based feature-space and model-space LR approaches

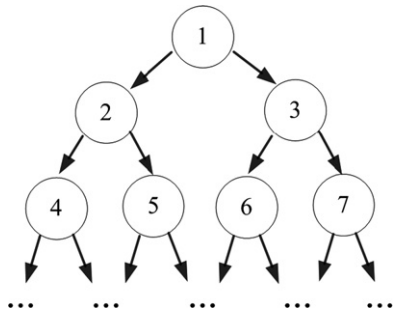

Fig. 2. A binary regression tree.

as F-DLR (feature-space discriminative linear regression) and M-DLR (model-space discriminative linear regression), respectively.

Because only a single global transform is used in feature-space method, this adaptation scheme is good for adaptation scenario with a very limited amount of adaptation data. Compared with the STM approach in [16], our feature transformation function is more general than STM by having an additional bias term. The main advantage of the STM approach is to have a closed-form solution for estimating the feature transform, which can be used, as discussed in the next section, for initialization of the Rprop optimization of our DLR objective function. As for the objective function, SSM-MCE is more directly related to the objective of minimizing the recognition error than the weighted mean squared error objective function used in STM approach.

When more adaptation data are available, the M-DLR approach is preferred because more LR transforms can be used to achieve a better adaptation effect. When large amount of adaptation data are available, for the extreme case of using a bias transform for each prototype, SSM-MCE-LR adaptation becomes equivalent to SSM-MCE re-training of the prototype-based classifier using the adaptation data. It is hoped that the M-DLR adaptation will approach the upper bound of re-training when enough flexible LR transforms are used in practice.

4. Implementation issues

4.1. STM-based initialization

In our F-DLR approach, we initialize the bias vector as $\boldsymbol{b}=0$, and use the STM approach in [16] to initialize the $\boldsymbol{A}$ matrix as follows:

- Define the source point set as the set of feature vectors of adaptation samples, and the target point set as the set of the corresponding prototypes with the smallest Euclidean distances to those feature vectors.
- Find a style transfer matrix $\boldsymbol{A}$ to solve the following optimization problem:

$$\min_{\boldsymbol{A}} \sum_r f_r \|\boldsymbol{A}\boldsymbol{s}_r-\boldsymbol{t}_r\|_2^2+\beta_1\|\boldsymbol{A}-\boldsymbol{I}\|_2^2, \tag{14}$$

where the rth source point $\boldsymbol{s}_r$ is transformed to the target point $\boldsymbol{t}_r$ with the confidence f_r, which is set as 1 for supervised adaptation, and a value between 0 and 1 for unsupervised adaptation as described in the next subsection. The hyper-parameter β_1 is set as

$$\beta_1=\frac{\tilde{\beta}_1}{2D}\mathrm{tr}\left(\sum_r f_r(\boldsymbol{s}_r+\boldsymbol{t}_r)\boldsymbol{s}_r^\top\right), \tag{15}$$

where $\mathrm{tr}(\cdot)$ is the trace of a matrix and $\tilde{\beta}_1$ takes a value between 0 and 3. The closed-form solution of the above problem is as follows:

$$\boldsymbol{A}=\left[\sum_r f_r\boldsymbol{t}_r\boldsymbol{s}_r^\top+\beta_1\boldsymbol{I}\right]\left[\sum_r f_r\boldsymbol{s}_r\boldsymbol{s}_r^\top+\beta_1\boldsymbol{I}\right]^{-1}. \tag{16}$$

In our M-DLR approach, we initialize the bias vector as $\boldsymbol{b}_{e_i}=0$, and initialize the $\boldsymbol{A}_{e_i}$ matrix separately for each regression class e_i in a similar way as described above by using the adaptation samples of the corresponding regression class, and exchanging the role of $\boldsymbol{s}_r$ and $\boldsymbol{t}_r$ in Eqs. (14)–(16), respectively.

4.2. Data selection for unsupervised adaptation

In unsupervised adaptation, the class labels of adaptation data are obtained by recognition results using the baseline classifier. To mitigate the negative effect of the possible wrong labels, in STM-based unsupervised adaptation, the following confidence measure is used as suggested in [16]:

$$f_r=\frac{\exp\{-\tau g_p(\boldsymbol{x}_r;\boldsymbol{\lambda}_p)\}}{\sum_{i=1}^{M}\exp\{-\tau g_i(\boldsymbol{x}_r;\boldsymbol{\lambda}_i)\}},\quad \tau=\frac{R_1}{\sum_{r=1}^{R_1} g_i(\boldsymbol{x}_r;\boldsymbol{\lambda}_i)}, \tag{17}$$

where p is the label assigned to adaptation sample $\boldsymbol{x}_r$ by the recognizer and τ is estimated from the training set. This trick works only when the recognition accuracy of the baseline classifier is relatively high. To play safe, in this study, we propose to only use those samples which achieve high enough confidence for adaptation as follows:

$$\boldsymbol{Y}_{\mathrm{sub}}=\{\boldsymbol{y}_r \,|\, f_r>\beta_2 f_{\mathrm{train}},\ \boldsymbol{y}_r\in\boldsymbol{Y}\}, \tag{18}$$

where f_{train} is the averaged confidence calculated using Eq. (17) on the training set and $\beta_2\in[0,1]$ is a weighting coefficient. The new STM approach using this data selection trick is referred to as "STM-Sub" approach hereinafter. Experimental results show that the STM-Sub outperforms the original STM, especially for those cases with very low recognition accuracy.

For DLR-based unsupervised adaptation, the above data selection strategy is also used in each iteration of Rprop optimization. The label "p" and the most competing label "q" in Eqs. (10) and (13) are replaced by the best and the second best recognition results, respectively.

4.3. A hybrid adaptation approach

To achieve the best possible adaptation effect for different amount of adaptation data, we propose to use the following hybrid adaptation approach:

- If the amount of adaptation data is very small, i.e., $R_2\le N_T$, use adaptive STM approach with $\beta_1^{\mathrm{new}}=\beta_1(N_T)/R_2$.
- If more adaptation data are available but not enough to estimate multiple transforms in M-DLR, i.e., $N_T<R_2\le N_M$, use single-transform based F-DLR or M-DLR approach.
- If enough adaptation data are available, i.e., $R_2>N_M$, use multi-transform based M-DLR approach.

Two control parameters N_T and N_M are set empirically to $D^2/16$ and $2D^2$, respectively. Hereinafter, we use M-Hybrid and F-Hybrid to refer to the above hybrid adaptation approach for model-space and feature-space, respectively.

In the following experiments, we will show that the adaptive STM can outperform significantly the original STM for the case of very limited adaptation data because the control parameter β_1 is adjusted dynamically according to the amount of available adaptation data.

5. Experiments and results

5.1. Experimental setup

The experiments are conducted on a task of recognizing isolated printed Chinese characters. The vocabulary of our baseline classifier consists of 9252 Chinese characters. For SSM-MCE training of the baseline classifier, we use about 150 gray-scale image samples per character. These image samples are mostly from the scanned documents with several commonly used fonts,

and normalized to a fixed size of 64×64. Fig. 3 shows samples with multiple fonts in the training set. For feature extraction, a 512-dimensional raw Gabor feature vector is extracted first from each gray-scale character image as described in [22], where the spatial sampling points are 8×8, the number of orientations is 8, and the wavelength is 8. A new 512-dimensional feature vector is then calculated by dividing each raw Gabor feature by the maximum element value in the raw Gabor feature vector for intensity normalization. A 513-dimensional feature vector is formed by extending the above 512-dimensional feature vector with an aspect-ratio feature, which is followed by LDA transformation to obtain a 128-dimensional feature vector (i.e., $D=128$) [22]. As for the number of prototypes for each character, we use four prototypes for 3755 most frequently used Chinese characters and two prototypes for the rest of character classes. For Rprop-based SSM-MCE training and SSM-MCE-LR adaptation, the control parameters are set as follows: $\alpha=7$; $\beta=0$; $T_1=100$; $T_2=50$; $\Delta_0=0.0125$; $\Delta_{\max}=50$; $\Delta_{\min}=0$; $\eta^+=1.2$; $\eta^-=0.5$. It is noted that all the control parameters related to Rprop are set empirically as suggested in [30] without tuning. For STM based adaptation, $\tilde{\beta}_1$ is set to 0.1. For unsupervised adaptation, β_2 is set to 0.02.

5.2. Supervised adaptation to font style

The first set of experiments is designed to examine the effectiveness of the proposed approach for supervised adaptation to font style. We use sets of new font library for experiments. Fig. 4 shows how a Chinese character looks like in different fonts. For each font library, there are 6823 character classes and one sample per character class. We divide 6823 samples per font into two equal subsets as adaptation set and testing set. In this case, our hybrid adaptation approach in Section 4.3 has used the DLR approach.

Fig. 6 summarizes a performance (character recognition error rate in %) comparison of the baseline classifier and different approaches for supervised adaptation to each font style on testing

Fig. 3. Samples with multiple fonts in the training set.

Fig. 4. A Chinese character in different new fonts.

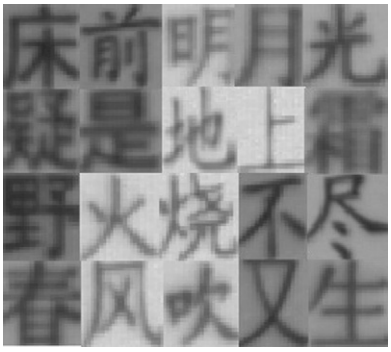

Fig. 5. Samples of low-quality Chinese characters.

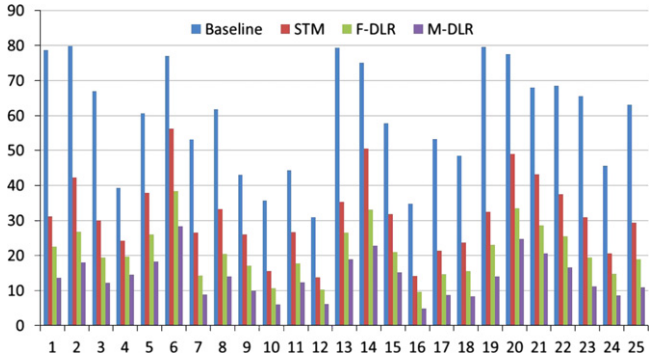

Fig. 6. Performance (character recognition error rate in % on each testing set) comparison of the baseline classifier and different approaches for supervised adaptation to *each of*25 *new font styles*.

sets of 25 new font styles. Several observations can be made. First, all methods for supervised adaptation outperform the baseline classifier without adaptation, which demonstrates that a linear transformation is reasonable as a mapping function for font adaptation. Second, both F-DLR and M-DLR achieve consistently significant improvements in recognition accuracy compared to STM, which indicates that the SSM-MCE objective function of DLR is indeed better than the least weighted squared error criterion used in STM. Third, M-DLR performs much better than F-DLR.

Fig. 7 gives a similar performance comparison as in Fig. 6 for mixed-font adaptation scenario, where the adaptation data of all 25 font styles are pooled together. One interesting observation is that the performance of STM in Fig. 7 is much worse than that in Fig. 6, which means single transformation cannot deal with mixed font styles by STM approach. But for DLR, by using a discriminative criterion, significant improvements can also be achieved compared to the baseline classifier.

As the above adaptation experiments are performed on data sets with high error rates achieved by the baseline classifier, Table 1 gives another comparison on data sets with different ranges of baseline error rates. As the baseline performance becomes better and better, all the adaptation methods can still achieve significant improvements, yet the gap between F-DLR and STM is narrowed. M-DLR always achieves the best performance among all the adaptation methods. The experiment labeled as F-DLR-NB in Table 1 used a new version of F-DLR without the bias term in Eq. (11). This experiment is designed for a fair comparison between STM and F-DLR as now the type of transformation is exactly the same. We can see that there are only slight differences in performance between F-DLR-NB and F-DLR compared with STM, which means that the matrix plays a much more important role than the bias term for the transformation in F-DLR.

Fig. 8 shows the learning curves of Rprop algorithms for our adaptation methods. We select three representative font styles with diverse baseline performance, namely A, E, and J in Table 1 for F-DLR and M-DLR adaptation. From those curves, several observations can be made. First, for F-DLR adaptation, the main performance gain is achieved in the first several iterations. Second, for M-DLR adaptation, even only one iteration can get the most performance gain. Third, for both F-DLR and M-DLR adaptation to each font style, the best performance is achieved after 50 iterations, which is used for all the other experiments.

Another concern of our proposed adaptation methods is their computational complexity. Although F-DLR and M-DLR adaptation can be done offline, we still want to give readers an idea of how the User CPU Time looks like. To handle large-scale data processing, our algorithms are implemented based upon Microsoft Research Asia's MPI-based machine learning platform [34]. This platform is developed on top of Microsoft Windows HPC Server, and optimized for various machine learning algorithms. With this high-performance parallel computing platform, experiments can be run very efficiently for large-scale task. In our font style adaptation experiments, we use 200 CPU cores with a clock rate of 2.5 GHz to run 25 sets of font style adaptation experiments with 3411 adaptation samples for each set. The number of Rprop iterations for F-DLR and M-DLR is 50. The total User CPU Time for F-DLR and M-DLR adaptation are 108 and 267 s, respectively.

5.3. Supervised adaptation to low-quality text

The second set of experiments is designed to examine the effectiveness of the proposed approach for supervised adaptation to low-quality text. We use a database of low-quality character images captured by a camera with a resolution of 640×480 pixels. Fig. 5 shows some samples of low-quality Chinese characters. There are 7915 character classes with dozens of samples per character class. First, 15 samples per character are randomly selected from the database to form the testing set. The remaining samples are used for adaptation with different amount of data. The character recognition error rate of the baseline classifier on testing set is 46.98%. Fig. 9 compares the performance of STM and hybrid adaptation approach in Section 4.3 on testing set. If adaptation data are very limited ($R_2 < 256$), the performance of STM is even worse than that of the baseline classifier. The performance improvement for STM saturates beyond a certain point ($R_2 > 1024$). As expected, our hybrid adaptation approach can reduce error rates consistently for different amount of adaptation data, and outperforms significantly the STM approach across the board, especially when more data are used for adaptation. It is observed again that M-Hybrid approach performs better than F-Hybrid approach. For the reader's interest, Table 2 lists the number of transformations (regression classes) for each M-Hybrid adaptation in Fig. 9. The dramatic increase of the number of transformations from the case of "32,768" to "65,536" is due to the specific thresholds we used in the hybrid adaptation procedure, which by no means of being optimal for other possible adaptation tasks and scenarios.

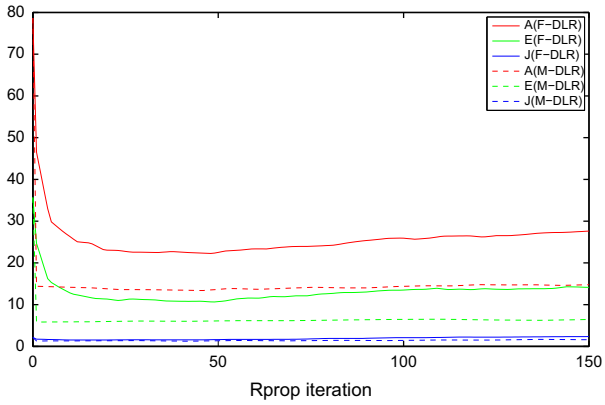

Fig. 8. The learning curves (i.e., character recognition error rates on each testing set as a function of number of iterations) of Rprop algorithms for F-DLR and M-DLR adaptation to font styles A, E, and J in Table 1.

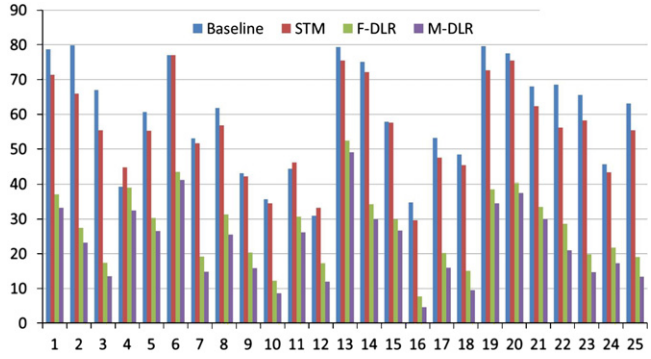

Fig. 7. Performance (character recognition error rate in % on each testing set) comparison of the baseline classifier and different approaches for supervised adaptation to *a merged set of* 25 *new font styles*.

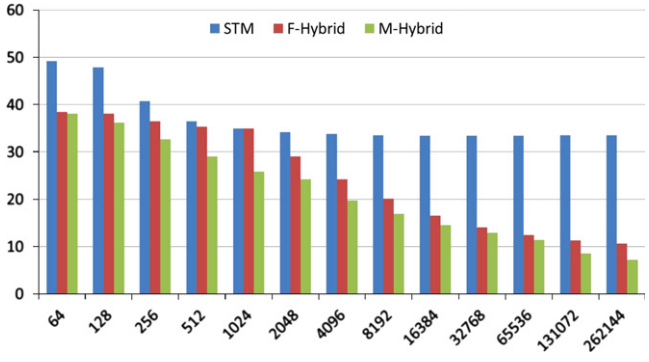

Fig. 9. Performance (character recognition error rate in % on testing set) comparison of different approaches for supervised adaptation with different number of adaptation samples of low-quality text (baseline recognition error rate is 46.98%).

Table 1
Performance (character recognition error rate in % on each testing set) comparison of the baseline classifier and different approaches for supervised adaptation to each of new font styles with different ranges of baseline error rates.

Font style ID	A	B	C	D	E	F	G	H	I	J
Baseline	78.66	66.96	57.78	45.62	35.68	28.76	17.65	12.58	8.85	2.14
STM	31.19	30.05	31.19	20.61	15.60	11.02	13.49	7.15	5.19	1.50
F-DLR-NB	22.46	19.48	21.52	14.61	10.75	9.09	10.63	6.13	4.82	1.61
F-DLR	22.52	19.37	20.99	14.81	10.73	8.71	10.24	6.16	4.85	1.63
M-DLR	13.66	12.23	15.19	8.59	6.10	4.34	5.39	3.69	2.90	1.35

Table 2
The number of transformations (regression classes) used for M-Hybrid in Fig. 9.

No. of samples	64	128	256	512	1024	2048	4096	8192	16,384	32,768	65,536	131,072	262,144
No. of transforms	1	1	1	1	1	1	1	1	1	1	39	80	137

Table 3
Performance (character recognition error rate in % on testing set) comparison of SSM-MCE based M-DLR adaptation and SSM-MCE based re-training by using different number of adaptation samples of low-quality text (baseline recognition error rate is 46.98%).

No. of adaptation samples	32,768	65,536	131,072	262,144
Re-training	31.57	19.61	12.22	9.34
Adaptation	12.87	11.41	8.54	7.18

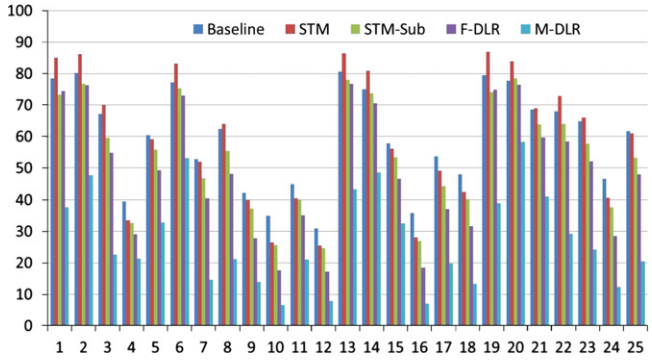

Fig. 10. Performance (character recognition error rate in % on *each adaptation set*) comparison of the baseline classifier and different approaches for unsupervised adaptation to each of 25 new font styles.

In order to understand the effectiveness of using adaptation versus re-training by using adaptation data, Table 3 compares the performance (character recognition error rate in % on testing set) of SSM-MCE based M-DLR adaptation and SSM-MCE based re-training by using different number of adaptation samples of low-quality text. For all the cases we experimented with SSM-MCE based M-DLR adaptation performs much better than re-training. The power of the M-DLR adaptation approach is demonstrated clearly.

5.4. Effects of unsupervised adaptation

To verify the effectiveness of the proposed adaptation approaches for unsupervised adaptation, we repeat the above two sets of adaptation experiments by running the relevant adaptation procedures in unsupervised adaptation mode.

Figs. 10 and 11 give a performance comparison of the baseline classifier and different approaches for unsupervised adaptation to each of 25 new font styles by measuring the character recognition error rate in % on each adaptation set and testing set, respectively. It is observed that for most font styles with very low baseline performance (over 60% character recognition error rate), the performance of STM is worse than that of the baseline classifier. However, our proposed adaptation approaches can achieve significant improvement of recognition accuracy over both the STM and the baseline classifier. M-DLR still achieves the best performance in all cases, and the performance gap between M-DLR and F-DLR in unsupervised adaptation is larger than that in supervised adaptation. For those cases with better baseline performance, the performance gap between supervised and unsupervised M-DLR based adaptation is much smaller than other more difficult cases with a worse baseline performance.

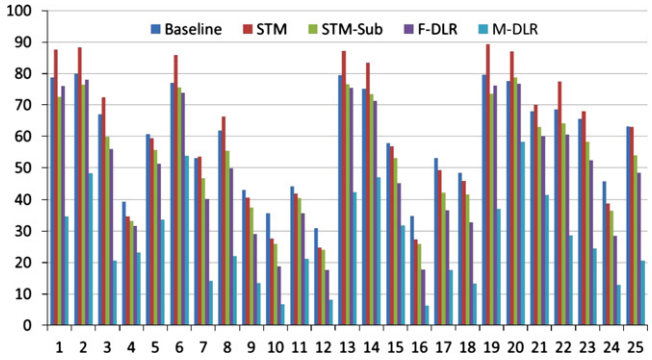

Fig. 11. Performance (character recognition error rate in % on *each testing set*) comparison of the baseline classifier and different approaches for unsupervised adaptation to each of 25 new font styles.

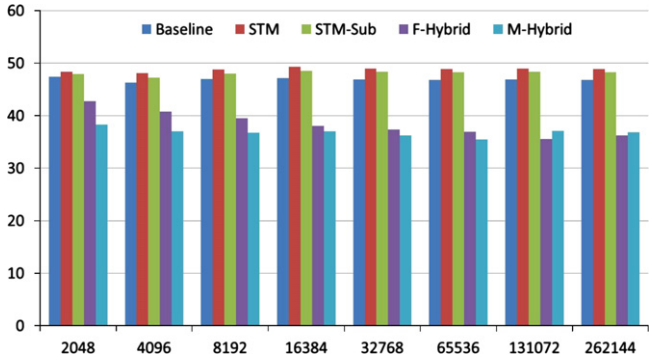

Fig. 12. Performance (character recognition error rate in % on *adaptation set*) comparison of the baseline classifier and different approaches for unsupervised adaptation to low-quality text on adaptation sets with different number of samples.

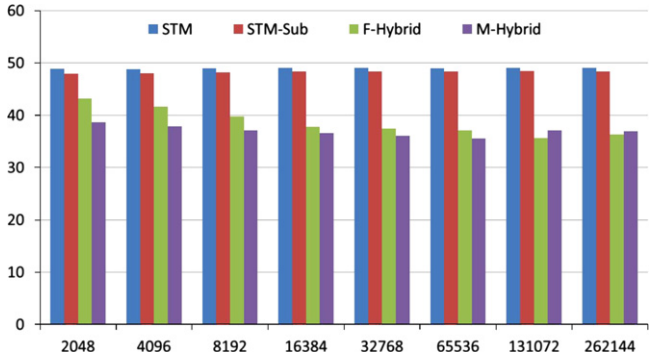

Fig. 13. Performance (character recognition error rate in % on *testing set*) comparison of different approaches for unsupervised adaptation with different number of adaptation samples of low-quality text (baseline recognition error rate is 46.98%).

Similar experiments are conducted for unsupervised adaptation to low-quality text. Performance comparisons of different approaches for unsupervised adaptation with different amount of

adaptation data to low-quality text on both adaptation set and testing set are shown in Figs. 12 and 13, respectively. Compared with unsupervised adaptation to font style, several new observations can be made. First, with adequate adaptation data, the performance of F-DLR is comparable to that of the M-DLR. Second, the performance improvement of all the unsupervised adaptation approaches saturates quickly with increasingly more adaptation data. The best performance achieved by unsupervised adaptation is far away from that of the supervised adaptation.

6. Conclusion

In this paper, we have proposed a new SSM-MCE linear regression approach to adaptation of an SSM-MCE trained prototype-based classifier and demonstrated its application for Chinese OCR. In real-world application, the feature-space adaptation method can be used for fast adaptation with a small amount of adaptation data, while the model-space adaptation method can be used to upgrade the performance of the classifier by using increasingly more adaptation data. The proposed hybrid adaptation approach offers a good practical solution for cases with different amount of adaptation data. In this study, we have confirmed the effectiveness of the proposed approach for supervised adaptation of font styles and low-quality text, respectively. As future work, we will:

- study more adaptation scenarios with mismatched training and recognition conditions;
- study how to further improve the effectiveness of unsupervised adaptation for those difficult recognition tasks; and
- apply the proposed approach to writer adaptation for handwriting recognition.

Results will be reported elsewhere once they become available.

Conflict of interest statement

None declared.

Acknowledgments

The authors would like to thank their colleagues, Ivan Stojiljkovic, Magdalena Vukosavljevic and David Nister for their help in accessing Microsoft's OCR resources, Professor Lianwen Jin for his sharing with us Chinese character image corpora, and Kai Chen for his help on some experiments.

Appendix A. Rprop optimization procedure for SSM-MCE training

Step 1: Let $t=0$. η^+ and η^- $(0<\eta^-<1<\eta^+)$ are the increase factor and decrease factor, respectively. Δ_0 is the initial step-size. $\Delta_{\max}$ and $\Delta_{\min}$ are the upper limit and lower limit of step-size, respectively. Calculate the derivative of $l(\boldsymbol{X};\boldsymbol{\Lambda})$ w.r.t. each m_{ikd} and update the prototype parameters as follows:

$$m_{ikd}^{(t+1)} = m_{ikd}^{(t)} + \Delta m_{ikd}^{(t)}, \tag{19}$$

$$\Delta m_{ikd}^{(t)} \triangleq -\mathrm{sign}\left(\frac{\partial l(\boldsymbol{X};\boldsymbol{\Lambda}^{(t)})}{\partial m_{ikd}}\right)\Delta_{ikd}^{(t)}, \tag{20}$$

where m_{ikd} is the dth element of $\boldsymbol{m}_{ik}$, $m_{ikd}^{(t)} = m_{ikd}$, $\Delta_{ikd}^{(t)} = \Delta_0$, and

$$\frac{\partial l(\boldsymbol{X};\boldsymbol{\Lambda}^{(t)})}{\partial m_{ikd}} \triangleq \left.\frac{\partial l(\boldsymbol{X};\boldsymbol{\Lambda})}{\partial m_{ikd}}\right|_{\boldsymbol{\Lambda}=\boldsymbol{\Lambda}^{(t)}}. \tag{21}$$

Step 2: Let $t=t+1$. Define

$$S = \frac{\partial l(\boldsymbol{X};\boldsymbol{\Lambda}^{(t-1)})}{\partial m_{ikd}} \cdot \frac{\partial l(\boldsymbol{X};\boldsymbol{\Lambda}^{(t)})}{\partial m_{ikd}}. \tag{22}$$

Then, the updating formulas are

$$\Delta_{ikd}^{(t)} = \begin{cases} \min(\eta^+\Delta_{ikd}^{(t-1)}, \Delta_{\max}) & \text{if } S>0, \\ \max(\eta^-\Delta_{ikd}^{(t-1)}, \Delta_{\min}) & \text{if } S<0, \\ \Delta_{ikd}^{(t-1)} & \text{else.} \end{cases} \tag{23}$$

$$\text{If } S<0, \quad \frac{\partial l(\boldsymbol{X};\boldsymbol{\Lambda}^{(t)})}{\partial m_{ikd}} = 0, \tag{24}$$

$$m_{ikd}^{(t+1)} = m_{ikd}^{(t)} + \Delta m_{ikd}^{(t)}. \tag{25}$$

Step 3: Repeat Step 2 (T_1-1) times.

In the above procedure, the relevant derivative can be calculated as follows:

$$\frac{\partial l_r}{\partial m_{ikd}} = \alpha l_r(1-l_r) \left[\frac{\delta(i,p)\delta(k,\hat{k})(m_{p\hat{k}d}-x_{rd}) - \delta(i,q)\delta(k,\bar{k})(\hat{m}_{q\bar{k}d}-x_{rd})}{\|\boldsymbol{m}_{p\hat{k}}-\boldsymbol{m}_{q\bar{k}}\|} - d(\boldsymbol{x}_r;\boldsymbol{\Lambda})\frac{(\delta(i,p)\delta(k,\hat{k})-\delta(i,q)\delta(k,\bar{k}))(m_{p\hat{k}d}-m_{q\bar{k}d})}{\|\boldsymbol{m}_{p\hat{k}}-\boldsymbol{m}_{q\bar{k}}\|^2}\right],$$

where

$$l_r = \frac{1}{1+\exp[-\alpha d(\boldsymbol{x}_r;\boldsymbol{\Lambda})+\beta]}, \tag{26}$$

and δ is the Kronecker delta function.

Appendix B. Rprop optimization procedure for model-space adaptation

Step 1: Let $t=0$. Calculate the derivative of $l(\boldsymbol{Y};\boldsymbol{\Lambda},\boldsymbol{\Theta})$ w.r.t. each A_{edj} and b_{ed}, where A_{edj} is the (d,j)th element of the matrix $\boldsymbol{A}_e$ and b_{ed} is the dth element of the bias vector $\boldsymbol{b}_e$. Then update the transform parameters as follows:

$$A_{edj}^{(t+1)} = A_{edj}^{(t)} + \Delta A_{edj}^{(t)}, \tag{27}$$

$$\Delta A_{edj}^{(t)} \triangleq -\mathrm{sign}\left(\frac{\partial l(\boldsymbol{Y};\boldsymbol{\Lambda},\boldsymbol{\Theta}^{(t)})}{\partial A_{edj}}\right)\Delta_{edj}^{(t)}, \tag{28}$$

$$b_{ed}^{(t+1)} = b_{ed}^{(t)} + \Delta b_{ed}^{(t)}, \tag{29}$$

$$\Delta b_{ed}^{(t)} \triangleq -\mathrm{sign}\left(\frac{\partial l(\boldsymbol{Y};\boldsymbol{\Lambda},\boldsymbol{\Theta}^{(t)})}{\partial b_{ed}}\right)\Delta_{ed}^{(t)}, \tag{30}$$

where $b_{ed}^{(t)}=0$, $\Delta_{edj}^{(t)} = \Delta_{ed}^{(t)} = \Delta_0$, and

$$\frac{\partial l(\boldsymbol{Y};\boldsymbol{\Lambda},\boldsymbol{\Theta}^{(t)})}{\partial A_{edj}} \triangleq \left.\frac{\partial l(\boldsymbol{Y};\boldsymbol{\Lambda},\boldsymbol{\Theta})}{A_{edj}}\right|_{\boldsymbol{\Theta}=\boldsymbol{\Theta}^{(t)}}, \tag{31}$$

$$\frac{\partial l(\boldsymbol{Y};\boldsymbol{\Lambda},\boldsymbol{\Theta}^{(t)})}{\partial b_{ed}} \triangleq \left.\frac{\partial l(\boldsymbol{Y};\boldsymbol{\Lambda},\boldsymbol{\Theta})}{\partial b_{ed}}\right|_{\boldsymbol{\Theta}=\boldsymbol{\Theta}^{(t)}}. \tag{32}$$

Step 2: Let $t=t+1$. Define

$$S_A = \frac{\partial l(\boldsymbol{Y};\boldsymbol{\Lambda},\boldsymbol{\Theta}^{(t-1)})}{\partial A_{edj}} \cdot \frac{\partial l(\boldsymbol{Y};\boldsymbol{\Lambda},\boldsymbol{\Theta}^{(t)})}{\partial A_{edj}}, \tag{33}$$

$$S_b = \frac{\partial l(\boldsymbol{Y};\boldsymbol{\Lambda},\boldsymbol{\Theta}^{(t-1)})}{\partial b_{ed}} \cdot \frac{\partial l(\boldsymbol{Y};\boldsymbol{\Lambda},\boldsymbol{\Theta}^{(t)})}{\partial b_{ed}}. \tag{34}$$

Then, the updating formulas are

$$\Delta_{edj}^{(t)} = \begin{cases} \min(\eta^+ \Delta_{edj}^{(t-1)}, \Delta_{\max}) & \text{if } S_A > 0, \\ \max(\eta^- \Delta_{edj}^{(t-1)}, \Delta_{\min}) & \text{if } S_A < 0, \\ \Delta_{edj}^{(t-1)} & \text{else}, \end{cases} \tag{35}$$

$$\Delta_{ed}^{(t)} = \begin{cases} \min(\eta^+ \Delta_{ed}^{(t-1)}, \Delta_{\max}) & \text{if } S_b > 0, \\ \max(\eta^- \Delta_{ed}^{(t-1)}, \Delta_{\min}) & \text{if } S_b < 0, \\ \Delta_{ed}^{(t-1)} & \text{else}. \end{cases} \tag{36}$$

$$\text{If } S_A < 0, \quad \frac{\partial l(\boldsymbol{Y};\boldsymbol{\Lambda},\boldsymbol{\Theta}^{(t)})}{\partial A_{edj}} = 0, \tag{37}$$

$$\text{If } S_b < 0, \quad \frac{\partial l(\boldsymbol{Y};\boldsymbol{\Lambda},\boldsymbol{\Theta}^{(t)})}{\partial b_{ed}} = 0, \tag{38}$$

$$A_{edj}^{(t+1)} = A_{edj}^{(t)} + \Delta A_{edj}^{(t)}, \tag{39}$$

$$b_{ed}^{(t+1)} = b_{ed}^{(t)} + \Delta b_{ed}^{(t)}. \tag{40}$$

Step 3: Repeat Step 2 (T_2-1) times.

In the above procedure, the relevant derivatives can be calculated as follows:

$$\frac{\partial l_r}{\partial A_{edj}} = \alpha l_r(1-l_r) \left[\frac{\delta(e,e_p) m_{p\hat{k}j}(\hat{m}_{p\hat{k}d} - y_{rd}) - \delta(e,e_q) m_{q\bar{k}j}(\hat{m}_{q\bar{k}d} - y_{rd})}{\|\hat{\boldsymbol{m}}_{p\hat{k}} - \hat{\boldsymbol{m}}_{q\bar{k}}\|} - d(\boldsymbol{y}_r;\boldsymbol{\Lambda},\boldsymbol{\Theta}) \frac{(\delta(e,e_p) m_{p\hat{k}j} - \delta(e,e_q) m_{q\bar{k}j})(\hat{m}_{p\hat{k}d} - \hat{m}_{q\bar{k}d})}{\|\hat{\boldsymbol{m}}_{p\hat{k}} - \hat{\boldsymbol{m}}_{q\bar{k}}\|^2} \right],$$

$$\frac{\partial l_r}{\partial b_{ed}} = \alpha l_r(1-l_r) \left[\frac{\delta(e,e_p)(\hat{m}_{p\hat{k}d} - y_{rd}) - \delta(e,e_q)(\hat{m}_{q\bar{k}d} - y_{rd})}{\|\hat{\boldsymbol{m}}_{p\hat{k}} - \hat{\boldsymbol{m}}_{q\bar{k}}\|} - d(\boldsymbol{y}_r;\boldsymbol{\Lambda},\boldsymbol{\Theta}) \frac{(\delta(e,e_p) - \delta(e,e_q))(\hat{m}_{p\hat{k}d} - \hat{m}_{q\bar{k}d})}{\|\hat{\boldsymbol{m}}_{p\hat{k}} - \hat{\boldsymbol{m}}_{q\bar{k}}\|^2} \right],$$

where

$$l_r = \frac{1}{1+\exp[-\alpha d(\boldsymbol{y}_r;\boldsymbol{\Lambda},\boldsymbol{\Theta}) + \beta]}. \tag{41}$$

Appendix C. Derivatives of objective function for feature-space adaptation

The relevant derivatives can be calculated as follows:

$$\frac{\partial l_r}{\partial A_{dj}} = \frac{\alpha l_r(1-l_r)(m_{q\bar{k}d} - m_{p\hat{k}d}) y_{rj}}{\|\boldsymbol{m}_{p\hat{k}} - \boldsymbol{m}_{q\bar{k}}\|}, \tag{42}$$

$$\frac{\partial l_r}{\partial b_d} = \frac{\alpha l_r(1-l_r)(m_{q\bar{k}d} - m_{p\hat{k}d})}{\|\boldsymbol{m}_{p\hat{k}} - \boldsymbol{m}_{q\bar{k}}\|}, \tag{43}$$

where A_{dj} is the (d,j)th element of the matrix $\boldsymbol{A}$, b_d is the dth element of the bias vector $\boldsymbol{b}$, and

$$l_r = \frac{1}{1+\exp[-\alpha d(\boldsymbol{y}_r;\boldsymbol{\Lambda},\boldsymbol{\Theta}) + \beta]}. \tag{44}$$

References

[1] Google Goggles 〈http://www.google.com/mobile/goggles/#text〉.
[2] Word Lens 〈http://questvisual.com/〉.
[3] Translator App for Windows phone 〈http://www.windowsphone.com/en-us/store/app/translator〉.
[4] J. Du, Q. Huo, L. Sun, J. Sun, Snap and translate using Windows phone, in: Proceedings of the ICDAR-2011, 2011, pp. 809–813.
[5] I. Marosi, Industrial OCR approaches: architecture, algorithms and adaptation techniques, in: Proceedings of the DRR-2007, 2007, pp. 1–10.
[6] G. Nagy, G.L. Shelton, Self-corrective character recognition system, IEEE Transactions on Information Theory IT-12 (2) (1966) 215–222.
[7] G. Nagy, H.S. Baird, A self-correcting 100-font classifier, in: Proceedings of the IS&T/SPIE Symposium on Electronic Imaging: Science & Technology, 1994.
[8] T. Hong, Degraded Text Recognition Using Visual and Linguistic Context, in: Ph.D. Thesis, State University of New York at Buffalo, 1995.
[9] P. Sarkar, Style Consistency in Pattern Fields, Ph.D. Thesis, Rensselaer Polytechnic Institute, 2000.
[10] P. Sarkar, An iterative algorithm for optimal style conscious field classification, in: Proceedings of the ICPR-2002, 2002, pp. IV-40–43.
[11] P. Sarkar, G. Nagy, Style consistent classification of isogenous patterns, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (1) (2005) 88–98.
[12] S. Veeramachaneni, G. Nagy, Style context with second order statistics, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (1) (2005) 14–22.
[13] S. Veeramachaneni, G. Nagy, Analytical results on style-constrained Bayesian classification of pattern fields, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (7) (2007) 1280–1285.
[14] P. Xiu, H.S. Baird, Whole-book recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (12) (2012) 2467–2480.
[15] C.-H. Lee, Q. Huo, On adaptive decision rules and decision parameter adaptation for automatic speech recognition, Proceedings of the IEEE 88 (8) (2000) 1241–1269.
[16] X.-Y. Zhang, C.-L. Liu, Style transfer matrix learning for writer adaptation, in: Proceedings of the CVPR-2011, 2011, pp. 393–400.
[17] X.-Y. Zhang, C.-L. Liu, Writer adaptation with style transfer mapping, IEEE Transactions on Pattern Analysis and Machine Intelligence, 29 Oct. 2012. IEEE computer Society Digital Library. IEEE Computer Society, 〈http://doi.ieeecomputersociety.org/10.1109/TPAMI.2012.239〉.
[18] K. Ait-Mohand, L. Heutte, T. Paquet, N. Ragot, Font adaptation of an HMM-based OCR system, in: Proceedings of the DRR-2010, 2010, pp. 1–8.
[19] C.J. Leggetter, P.C. Woodland, Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models, Computer Speech and Language 9 (2) (1995) 171–185.
[20] C.J. Leggetter, P.C. Woodland, Flexible speaker adaptation for large vocabulary speech recognition, in: Proceedings of the EUROSPEECH-1995, 1995, pp. 1155–1158.
[21] J.L. Gauvain, C.H. Lee, Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains, IEEE Transactions on Speech, and Audio Processing 2 (2) (1994) 291–298.
[22] Q. Huo, Y. Ge, Z.-D Feng, High performance Chinese OCR based on Gabor features, discriminative feature extraction and model training, in: Proceedings of the ICASSP-2001, 2001, pp. 1517–1520.
[23] Y.-Q. Wang, Q. Huo, A study of designing compact recognizers of handwritten Chinese characters using multiple-prototype based classifiers, in: Proceedings of the ICPR-2010, 2010, pp. 1872–1875.
[24] Z.-D. Feng, Q. Huo, Confidence guided progressive search and fast match techniques for high performance Chinese/English OCR, in: Proceedings of the ICPR-2002, 2002, pp. III-89–92.
[25] T. He, Q. Huo, A study of a new misclassification measure for minimum classification error training of prototype-based pattern classifiers, in: Proceedings of the ICPR-2008, 2008.
[26] S.E. Fahlman, An Empirical Study of Learning Speed in Back-propagation Networks, Technical Report CMU-CS-88-162, Carnegine Mellon University, 1988.
[27] Y.-Q. Wang, Q. Huo, Sample-separation-margin based minimum classification error training of pattern classifiers with quadratic discriminant functions, in: Proceedings of the ICASSP-2010, 2010, pp. 1866–1869.
[28] F. Kimura, K. Takashina, S. Tsuruoka, Y. Miyake, Modified quadratic discriminant functions and the application to Chinese character recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 9 (1) (1987) 149–153.

[29] M. Riedmiller, H. Braun, A direct adaptive method for faster backpropagation learning: the Rprop algorithm, in: Proceedings of the International Conference on Neural Networks, 1993, pp. 586–591.
[30] C. Igel, M. H ü sken, Improving the Rprop learning algorithm, in: International Symposium on Neural Computation, 2000, pp. 115–121.
[31] J. Wu, Q. Huo, A study of minimum classification error (MCE) linear regression for supervised adaptation of MCE-trained continuous-density hidden Markov models, IEEE Transactions on Audio, Speech, and Language Processing 15 (2) (2007) 478–488.
[32] Y. Linde, A. Buzo, R. Gray, An algorithm for vector quantizer design, IEEE Transactions on Communications 28 (1) (1980) 84–95.
[33] J. Du, Q. Huo, A discriminative linear regression approach to OCR adaptation, in: Proceedings of the ICPR-2012, 2012, pp. 629–632.
[34] Z.-J. Yan, T. Gao, Q. Huo, Designing an MPI-based parallel and distributed machine learning platform on large-scale HPC clusters, in: 2012 International Workshop on Statistical Machine Learning for Speech Processing, Kyoto, Japan, March 31, 2012 〈http://www.ism.ac.jp/IWSML2012/〉.

Jun Du received the B.Eng. and Ph.D. degrees from the Department of Electronic Engineering and Information Science, University of Science and Technology of China (USTC), Hefei, China, in 2004 and 2009, respectively. From 2004 to 2009, he was with the iFlytek Speech Lab of USTC, where he conducted research on speech recognition. During the above period, he worked as an Intern twice for nine months at Microsoft Research Asia (MSRA), Beijing, China, doing research on discriminative training and noise-robust front-end for speech recognition, and speech enhancement. In 2007, he also worked as a Research Assistant for six months at the Department of Computer Science, The University of Hong Kong, doing research on robust speech recognition. From July 2009 to June 2010, he worked at iFlytek Research on speech recognition. In July 2010, Dr. Du joined Visual Computing Group of MSRA as an Associate Researcher. Currently he works on handwriting recognition and OCR.

Qiang Huo is a Research Manager in Microsoft Research Asia (MSRA), Beijing, China. Prior to joining MSRA in August 2007, he had been a faculty member at the Department of Computer Science, The University of Hong Kong since 1998, where he also did his Ph.D. research on speech recognition during 1991–1994. From 1995 to 1997, Dr. Huo worked at Advanced Telecommunications Research Institute (ATR), Kyoto, Japan. In the past 25 years, he has been doing research and making contributions in the areas of speech recognition, handwriting recognition, OCR, gesture recognition, biometric-based user authentication, hardware design for speech and image processing. Dr. Huo received the B.Eng. degree from the University of Science and Technology of China (USTC), Hefei, China, in 1987, the M.Eng. degree from Zhejiang University, Hangzhou, China, in 1989, and the Ph.D. degree from the USTC in 1994, all in electrical engineering.