

# Incremental learning of LDA model for Chinese writer adaptation

Lianwen Jin<sup>\*</sup>, Kai Ding, Zhibin Huang

School of Electronic and Information Engineering, South China University of Technology, Guangzhou, Guangdong 510640, China

## ARTICLE INFO

Available online 12 March 2010

### Keywords:

Incremental LDA  
WILDA  
Writer adaptation  
Handwritten Chinese character recognition

## ABSTRACT

A new writer adaptation method based on incremental linear discriminant analysis (ILDA) is presented in this paper. We first provide a more general solution for ILDA and then present a Weighted ILDA (WILDA) approach. Based on ILDA or WILDA, the writer adaptation is performed by updating the LDA transformation matrix and the classifier prototypes in the discriminative feature space. Experimental results show that both ILDA and WILDA are very effective to improve the recognition accuracy for writer adaptation, and WILDA outperforms ILDA. The proposed WILDA based writer adaptation method can reduce as much as 47.88% error rate on the writer-dependent dataset while it only has as less as 0.85% accuracy loss on the writer-independent dataset. It indicates that writer adaptation using WILDA can significantly increase the recognition accuracy for the particular writer while having limited impact on the accuracy for general writers.

© 2010 Elsevier B.V. All rights reserved.

## 1. Introduction

The ability to transcribe handwritten characters to a computerized text format is a great benefit to inputting, organizing and annotating data in various applications such as the input, storage and distribution of notes or messages [3,22,32]. The successes of products, such as PDA, smart cell phone, and Tablet PC, are the evidence that users have interest in such capabilities. However, the large variability of handwriting styles across individuals makes handwriting recognition a challenging problem. Although great progress has been achieved in the field of online handwritten Chinese character recognition (OHCCR) during the past 40 years [8,12,22,25,32], recent researches on unconstrained cursive online handwriting recognition show that this problem is far from having been completely solved. Jin et al. reported that for the recognition of 6763 categories of unconstrained handwritten characters from the SCUT-COUCH dataset [21], the best recognition accuracy was only 92.43% using the state-of-the-art feature extraction method plus linear discriminant analysis (LDA) classifier [5]. Wang et al. [39] reported that using a state-of-the-art recognizer for the classification of the samples from a new unconstrained online handwritten dataset CASIA-OLHWDB1, the highest accuracies achieved on average, 92.44% for 4037 categories and 92.91% for 3866 categories, respectively, were achieved, which are much lower than those reported on other previous databases, where the handwritten samples were much more regularly written (e.g., 98.56% on HCL2000 [24], 97.84% and

98.24% on Japanese Kanji [23]). On the other hand, the required recognition rate for the recognizer by ordinary users is very high. For example, tests with keyboard typing have shown that the writers tolerate random errors up to 1% while 0.5% is unnoticeable and 2% is intolerable [9]. All of these indicate that the unconstrained OHCCR problem is far from being completely solved, especially when using some recently available challenging datasets, such as SCUT-COUCH [21] or CASIA-OLHWDB1 [39].

One challenge of unconstrained OHCCR we face is that there are too many different writing styles to be handled when designing a general purpose handwriting recognizer. Writer-independent systems trained from examples require large training sets from many writers to deal with this variability, but it cannot yet achieve very high performance for unconstrained OHCCR [21,39]. In contrast, writer-dependent systems can be trained on a specific user's handwritten samples to achieve higher accuracy [33]. It is generally agreed that, for a given handwriting recognition task, a writer-dependent system usually outperforms a writer-independent system [38]. The writer adaptation is the process of converting a writer-independent system learned from the writer-independent dataset to a writer-dependent system, which is turned for a particular writer using a specific incremental data. As the writer-adaptation is an online incremental learning process to learn the particular writing behavior and adaptively updating the classification model, we usually need an initial classification model trained on general-purpose writer-independent dataset and then conduct the adaptation learning processing. This adaptation has the potential advantage of significantly increasing recognition accuracies for a particular writer, which is very useful for a real world application, such as building a high performance, writer adaptive (personalized) online handwritten character input

<sup>\*</sup> Corresponding author.

E-mail address: [lianwen.jin@gmail.com](mailto:lianwen.jin@gmail.com) (L. Jin).

method. In the past, a number of writer adaptation handwriting recognition methods have been proposed [3,7,16,20,33,38]. Szummer and Bishop [33] proposed a discriminative writer adaptation method through clustering the writing styles, training a set of corresponding classifiers and then choosing an appropriate combination of classifiers for a particular writer. Connel and Jain [3] proposed an adaptive online handwriting recognition model, where another writer-dependent model was used to identify the styles present in a particular writer's training data, and then these models are retrained using the writer's data. But the adaptation in this way depends on the correctly clustering users' writing styles by classification confidence. Vuori [38] proposed a simple prototype based adaptation system using  $k$  nearest neighbor (KNN) classifier. The whole adaptation consisted of three steps, i.e. adding new prototypes, deactivating confusing prototypes, and reshaping existing prototypes. Kienzle and Chellapilla [16] presented a personalized handwriting recognition approach by minimizing a regularized risk function of SVM. LaViola and Zeleznik [20] proposed a practical technique of using a writer-independent recognizer to improve the accuracy of writer-dependent symbol recognizer based on the AdaBoost learning algorithm. Unfortunately, all these methods are designed for small scale handwriting recognition problem (for example, English letter, digit or symbol recognition), where the class number is relatively small; thus many of the adaptation methods are not practically applicable (such as SVM or AdaBoost based adaptation methods) for handling large datasets with many classes, such as Chinese handwriting recognition problem involving thousands of classes and hundreds of thousands of training/testing handwritten samples.

On the other hand, as a well known scheme for feature extraction and dimension reduction, linear discriminant analysis (LDA), also known as Fisher discriminant analysis (FDA), has been widely used in OLHCCR [5,22,24,25] and other pattern classification tasks [10,13,15,31]. The LDA seeks the best linear projections of data for discrimination, under the assumption that the classes have equal covariance Gaussian structure [6]. However, recent researches demonstrate that the classical LDA has some problems [34]. The first is heteroscedastic problem [27] that is the LDA models different classes with identical covariance matrices. Therefore, it fails to take account of any variations in the covariance matrices between different classes. The second is multimodal problem [11] that is the samples in each class cannot be approximated by a single Gaussian in many applications. Instead, a Gaussian mixture model (GMM) [2] is required. However, the LDA models each class by a single Gaussian distribution. The third is class separation problem [35]. In applications, distances between different classes are different and the LDA tends to merge classes that are close together in the original feature space. The fourth is the small sample size (SSS) problem [37,40], i.e. the number of training samples is less than the dimension of the feature space, which is also known as singularity problem [18].

To solve the heteroscedastic problem, Decell and Mayekar [4] proposed a method to obtain a subspace to maximize the average interclass divergence, which measured the separations between the classes. This criterion takes into account the discriminative information preserved in the covariances of different classes. Kumar and Andreou [19] developed the heteroscedastic discriminant analysis (HDA) by dropping the identical class covariances assumption. Jelinek [14] proposed a different way to deal with the heteroscedastic problem in subspace selection by the gradient steepest ascent method to find the projection matrix. Loog and Duin [27] introduced the Chernoff criterion to heteroscedasticize LDA.

A straightforward approach to solving the multi-model problem is to employ the GMM approach. Hastie and Tibshirani [11]

combined GMM with LDA by directly replacing the original single Gaussian in each class by a Gaussian mixture model.

To deal with the class separation problem, Lotlikar and Kothari [28] developed the fractional-step LDA (FS-LDA) by introducing a weighted function. Loog et al. [26] developed another weighted method for LDA, namely the approximate pairwise accuracy criterion (aPAC). The advantage of aPAC is that the projection matrix can be obtained by the eigenvalue decomposition. Lu et al. [29] combined the FS-LDA and the direct LDA for very high dimensional problems. However, both FS-LDA and aPAC do not use the discriminative information in different class covariances. To reduce this problem, Tao et al. [36] proposed the general averaged divergences analysis framework by using geometric mean for subspace selection.

To deal with the SSS problem, many approaches have been proposed, such as pseudo-inverse LDA, PCA+LDA and regularized LDA [18]. In recent years, Ye et al. [40] proposed the two dimensional LDA (2DLDA). Motivated by the successes of the 2DLDA, Tao et al. [37] proposed the general tensor discriminant analysis (GTDA) to solve the SSS problems.

Although the LDA and its extensions have been widely used in pattern recognition field, the typical implementation of these techniques assumes that a complete dataset for training is given in advance and it is often beneficial to learn the LDA model from large training sets, which may not be available initially. This motivates techniques for incrementally updating the LDA model when more data become available [17,30]. Several incremental versions of LDA (ILDA) have been suggested [17,30,41], which have successfully been applied to online learning tasks such as the classification of data streams [30], face image retrieval [17] and face recognition [41].

Although a number of researches on writer adaptation or ILDA were conducted, the ILDA based writer adaptation handwriting recognition remains unexploited. Motivated by this problem, we investigate how to adapt a writer independent recognizer to make it writer dependent based on the incremental learning of LDA model under the LDA based OLHCCR classification framework for the first time in this paper. We first provide a general incremental learning solution for LDA, and then propose a weighted incremental linear discriminant analysis (WILDA) approach for writer adaptive handwriting recognition by taking into account the issue of uncertain number of incremental data for writer adaptation in an online handwriting recognition application. Based on the incremental learning of the LDA model using ILDA or WILDA, the writer adaptation is performed by updating the LDA transformation matrix and the classifier prototypes in the discriminative feature space. Experimental results show that both ILDA and WILDA are very effective to improve the recognition accuracy for particular writers, and WILDA outperforms ILDA. The experimental results indicate that the writer adaptation using the WILDA approach can not only significantly increase the recognition accuracy for the particular writers but also have limited impact on the accuracy for the general writers.

The rest of this paper is organized as follows. Section 2 presents a general solution for the ILDA algorithm, and then proposes a new weighted incremental linear discriminant analysis (WILDA) approach. Classifier design based on LDA and the writer adaptation based on ILDA/WILDA are given in Section 3. Section 4 presents the experimental results and discussion. Finally, the conclusions are summarized in Section 5.

## 2. Incremental learning of linear discriminant analysis (LDA) model

Pang et al. [30] proposed an incremental LDA for classification of data streams. However, the final solution of this method is too

complex, since the sequential incremental learning condition and the chunk incremental learning condition are considered separately. In addition, for each case, the solution is divided into two cases depending on whether the new class sample is added or not. In this section, we first give a brief introduction to LDA, and then present a general incremental learning solution for LDA based on the research in [30]. Furthermore, considering the problem of uncertain number of incremental data for writer adaption in an online handwriting recognition application, we present a Weighted ILDA (WILDA) method in Section 2.3.

## 2.1. Introduction to LDA

LDA [6] is a supervised learning method, which utilizes the category information associated with each sample. The goal of LDA is to seek directions for efficient discrimination through maximizing the between-class scatter while minimizing the within-class scatter. Mathematically speaking, the within-class scatter matrix  $\mathbf{S}_w$  and between-class scatter matrix  $\mathbf{S}_b$  are defined as

$$\mathbf{S}_w = \sum_{j=1}^M \Sigma_j = \sum_{j=1}^M \sum_{i=1}^{N_j} (\mathbf{x}_i^{(j)} - \mathbf{m}_j)(\mathbf{x}_i^{(j)} - \mathbf{m}_j)^T \quad (1)$$

$$\mathbf{S}_b = \sum_{j=1}^M N_j (\mathbf{m}_j - \mathbf{m})(\mathbf{m}_j - \mathbf{m})^T \quad (2)$$

where  $\mathbf{x}_i^{(j)}$  is the  $i$ th sample of class  $j$ ,  $\mathbf{m}_j = (1/N_j) \sum_{i=1}^{N_j} \mathbf{x}_i^{(j)}$  is the mean of class  $j$ ,  $\mathbf{m} = (1/N) \sum_{j=1}^M N_j \mathbf{m}_j$  is the mean vector of all classes,  $M$  is the number of classes,  $N_j$  is the number of samples of class  $j$ , and  $N = \sum_{j=1}^M N_j$  is the total sample number.

For LDA transformation matrix,  $\mathbf{W}_{lda}$  can be derived by maximizing the following object function:

$$J(\mathbf{W}_{lda}) = \frac{|\mathbf{W}_{lda}^t \mathbf{S}_b \mathbf{W}_{lda}|}{|\mathbf{W}_{lda}^t \mathbf{S}_w \mathbf{W}_{lda}|} \quad (3)$$

This solution can be shown to correspond to the generalized eigenvectors of the following equation:

$$\mathbf{S}_b \mathbf{w}_i = \lambda_i \mathbf{S}_w \mathbf{w}_i \quad (4)$$

If  $\mathbf{S}_w$  is a nonsingular matrix then the objection function of Eq. (3) is maximized when the transformation matrix  $\mathbf{W}_{lda}$  consists of  $D$  generalized eigenvectors corresponding to the  $D$  largest eigenvalues of  $\mathbf{S}_w^{-1} \mathbf{S}_b$  [6]. In other words, by sorting the eigenvalues in descending order, we can then use the corresponding first  $D$  eigenvectors to form the columns of the LDA matrix  $\mathbf{W}_{lda}$ . In practical application, eigenvectors with low eigenvalues can be discarded to compress a high dimensional feature to a low-dimensional feature with an enhanced discrimination. Note that there are at most  $M-1$  nonzero generalized eigenvalues, so an upper bound on  $D$  is  $M-1$ .

## 2.2. A general solution for incremental LDA

The problem of Incremental LDA (ILDA) can be described as follows: when new samples are being presented, how can we update the corresponding LDA model parameters, including the class mean vector  $\mathbf{m}_j$ ,  $j=1,2,\dots,M$ , mean vector  $\mathbf{m}$  of all classes, within-class scatter matrix  $\mathbf{S}_w$ , and between-class scatter matrix  $\mathbf{S}_b$ . With these updated parameters, the new updated LDA transformation matrix  $\mathbf{W}_{lda}$  can be computed accordingly.

Suppose we have  $L$  incremental samples  $\mathbf{Y} = \{\mathbf{y}_i\}$  ( $i=1, \dots, L$ ) in  $P$  classes. Without loss of generality, we assume that  $l_i$  of  $L$  incremental samples belong to class  $C_i$  ( $i=1, \dots, P$ ). It is worthwhile to note that some of the class  $C_i$  may be newly introduced classes. Let  $\mathbf{m}_i^y$  and  $\mathbf{m}^y$  represent the mean vector of class  $C_i$  and all

incremental samples, respectively, which are computed by

$$\mathbf{m}_i^y = \frac{1}{l_i} \sum_{j=1}^{l_i} \mathbf{y}_j^{(i)} \quad (5)$$

$$\mathbf{m}^y = \frac{1}{L} \sum_{i=1}^L \mathbf{y}_i = \frac{1}{L} \sum_{i=1}^P l_i \mathbf{m}_i^y \quad (6)$$

The incremental within-class scatter matrix  $\mathbf{S}_w^y$  and between-class scatter matrix  $\mathbf{S}_b^y$  of the incremental samples are given as follows:

$$\mathbf{S}_w^y = \sum_{i=1}^P \Sigma_i^y = \sum_{i=1}^P \sum_{j=1}^{l_i} (\mathbf{y}_j^{(i)} - \mathbf{m}_i^y)(\mathbf{y}_j^{(i)} - \mathbf{m}_i^y)^T \quad (7)$$

$$\mathbf{S}_b^y = \sum_{i=1}^P l_i (\mathbf{m}_i^y - \mathbf{m}^y)(\mathbf{m}_i^y - \mathbf{m}^y)^T \quad (8)$$

Since some classes may or may not be updated by the new samples and some new classes containing only new samples may be introduced, the merged class set  $\Omega$  can be therefore divided into three parts: updated class set  $\Psi$ , no updated class set  $\Phi$ , and newly introduced class set  $\Gamma$ . We assume the class number is updated from  $M$  to  $T$  ( $T \geq M$ ,  $T \geq P$ ), and the sample number of each class is updated as  $N_i' = N_i + l_i$ , where  $i=1, \dots, T$ . It is obvious that, if  $C_i \in \Phi$ ,  $l_i=0$ , and if  $C_i \in \Gamma$ ,  $N_i=0$ .

It can be easily derived that the updated mean  $\mathbf{m}_i'$  for class  $C_i$  is updated by

$$\mathbf{m}_i' = \frac{N_i \mathbf{m}_i + l_i \mathbf{m}_i^y}{N_i'}, \quad i=1, \dots, T \quad (9)$$

and the mean vector  $\mathbf{m}'$  of total samples is

$$\mathbf{m}' = \frac{N\mathbf{m} + L\mathbf{m}^y}{N+L} \quad (10)$$

Then the updated between-class scatter matrix  $\mathbf{S}_b'$  after incremental data has been presented as follows:

$$\mathbf{S}_b' = \sum_{i=1}^T N_i' (\mathbf{m}_i' - \mathbf{m}')(\mathbf{m}_i' - \mathbf{m}')^T \quad (11)$$

The updated within-class scatter matrix  $\mathbf{S}_w'$  is

$$\mathbf{S}_w' = \sum_{j=1}^T \sum_{i=1}^{N_j'} (\mathbf{x}_i^{(j)} - \mathbf{m}_j')(\mathbf{x}_i^{(j)} - \mathbf{m}_j')^T = \sum_{j=1}^T \Sigma_j' \quad (12)$$

where the training samples set is given by  $\mathbf{X}' = \{\mathbf{x}_i\}_{i=1}^{N+L} = \{\mathbf{x}_i\}_{i=1}^N + \{\mathbf{y}_i\}_{i=1}^L$ . It can be derived (see [30] for details) that

$$\begin{aligned} \Sigma_j' &= \sum_{i=1}^{N_j'} (\mathbf{x}_i^{(j)} - \mathbf{m}_j')(\mathbf{x}_i^{(j)} - \mathbf{m}_j')^T \\ &= \Sigma_j + \frac{N_j l_j^2}{(N_j + l_j)^2} (\mathbf{m}_j^y - \mathbf{m}_j)(\mathbf{m}_j^y - \mathbf{m}_j)^T \\ &\quad + \frac{N_j^2}{(N_j + l_j)^2} \sum_{i=1}^{l_j} (\mathbf{y}_i^{(j)} - \mathbf{m}_j)(\mathbf{y}_i^{(j)} - \mathbf{m}_j)^T \\ &\quad + \frac{l_j(l_j + 2N_j)}{(N_j + l_j)^2} \sum_{i=1}^{l_j} (\mathbf{y}_i^{(j)} - \mathbf{m}_j^y)(\mathbf{y}_i^{(j)} - \mathbf{m}_j^y)^T \end{aligned} \quad (13)$$

The sum for the last three terms of Eq. (13) is rewritten as follows (see Proof A in the Appendix):

$$\begin{aligned} &\frac{N_j l_j^2}{(N_j + l_j)^2} (\mathbf{m}_j^y - \mathbf{m}_j)(\mathbf{m}_j^y - \mathbf{m}_j)^T \\ &\quad + \frac{N_j^2}{(N_j + l_j)^2} \sum_{i=1}^{l_j} (\mathbf{y}_i^{(j)} - \mathbf{m}_j)(\mathbf{y}_i^{(j)} - \mathbf{m}_j)^T \\ &\quad + \frac{l_j(l_j + 2N_j)}{(N_j + l_j)^2} \sum_{i=1}^{l_j} (\mathbf{y}_i^{(j)} - \mathbf{m}_j^y)(\mathbf{y}_i^{(j)} - \mathbf{m}_j^y)^T \end{aligned}$$

$$\begin{aligned}
& + \frac{l_j(l_j + 2N_j)}{(N_j + l_j)^2} \sum_{i=1}^{l_j} (\mathbf{y}_i^{(j)} - \mathbf{m}_j^y)(\mathbf{y}_i^{(j)} - \mathbf{m}_j^y)^T \\
& = \Sigma_j^y + \frac{N_j l_j}{(N_j + l_j)} (\mathbf{m}_j^y - \mathbf{m}_j)(\mathbf{m}_j^y - \mathbf{m}_j)^T
\end{aligned} \quad (14)$$

From Eqs. (13) and (14), the updated within-class matrix (12) can be rewritten as

$$\begin{aligned}
\mathbf{S}'_w & = \sum_{j=1}^T \Sigma'_j = \sum_{j \in \Omega} \Sigma'_j \\
& = \sum_{j \in \Omega} \Sigma'_j + \sum_{j \in \Omega} \Sigma_j^y + \sum_{j \in \Omega} \frac{N_j l_j}{(N_j + l_j)} (\mathbf{m}_j^y - \mathbf{m}_j)(\mathbf{m}_j^y - \mathbf{m}_j)^T \\
& = \sum_{j \in (\Omega - \Gamma)} \Sigma_j + \sum_{j \in \Gamma} \Sigma_j + \sum_{j \in (\Omega - \Phi)} \Sigma_j^y + \sum_{j \in \Phi} \Sigma_j^y \\
& \quad + \sum_{j \in \Psi} \frac{N_j l_j}{(N_j + l_j)} (\mathbf{m}_j^y - \mathbf{m}_j)(\mathbf{m}_j^y - \mathbf{m}_j)^T \\
& \quad + \sum_{j \in (\Omega - \Psi)} \frac{N_j l_j}{(N_j + l_j)} (\mathbf{m}_j^y - \mathbf{m}_j)(\mathbf{m}_j^y - \mathbf{m}_j)^T
\end{aligned} \quad (15)$$

It is obvious that

$$\begin{aligned}
\sum_{j \in (\Omega - \Psi)} \frac{N_j l_j}{(N_j + l_j)} (\mathbf{m}_j^y - \mathbf{m}_j)(\mathbf{m}_j^y - \mathbf{m}_j)^T & = 0 \\
\sum_{j \in \Gamma} \Sigma_j & = 0 \\
\sum_{j \in \Phi} \Sigma_j^y & = 0
\end{aligned} \quad (16)$$

Using the conclusions of Eq. (16) in (15), the updated within-class matrix  $\mathbf{S}'_w$  is finally given as follows:

$$\mathbf{S}'_w = \mathbf{S}_w + \mathbf{S}_w^y + \sum_{j \in \Psi} \frac{N_j l_j}{(N_j + l_j)} (\mathbf{m}_j^y - \mathbf{m}_j)(\mathbf{m}_j^y - \mathbf{m}_j)^T \quad (17)$$

Once we have updated the between-class scatter matrix  $\mathbf{S}'_b$  according to Eq. (11) and the within-class matrix  $\mathbf{S}'_w$  according to Eq. (17), we can now get the updated LDA transformation matrix  $\mathbf{W}'_{lda}$  by conducting the eigenvalue decomposition of  $\mathbf{S}'_w^{(-1)} \mathbf{S}'_b$ .

Compared with the method proposed in [30], our approach provided a general solution for the ILDA. In [30], the solution was separated into two situations depending on whether the incremental data were sequence data or chunk data. Furthermore, in either situation, the solution was separated into two cases again according to whether the new class is introduced or not. If the new class was introduced, the number of the new class must be 1. In contrast, our approach can solve all of the above situations using a uniform framework without restricting the number of newly introduced classes.

### 2.3. Weighted Incremental LDA for writer adaption for online handwritten Chinese character recognition

For the problem of writer adaption, the handwritten character samples of a particular writer serve as the incremental sample set  $Y = \{\mathbf{y}_i\}_{i=1}^{l_j}$  under the ILDA framework. From Eqs. (9)–(12) and (17), we can see that the performance of adaptation could be affected by the number of new samples used. In general, if the new samples of a particular writer for learning the ILDA model are sufficient, it can be expected that the updated ILDA model would give an improved accuracy for the specific writer, but may significantly decrease the accuracy for general writers. Otherwise, if the updating samples only make up a small proportion of the total training data for updating the LDA model, the performance improvement might not be so significant in such situation though the accuracy loss for the general writer may be very little.

However, in practical application, the amount of data that a particular writer provides is uncertain and various for different characters. On the other hand, we do not expect to make the adaptation to a specific writer's handwriting style at the cost of losing too much generality for other writer styles or to only have little improvement for a particular writer. In other words, a good trade-off between writer-dependent and writer-independent handwriting recognition is expected. To achieve the trade-off, we induce a weighted update mechanism to the ILDA.

Suppose we have  $l_i$  of  $L$  incremental samples belonging to class  $C_i$  ( $i=1, \dots, P$ ), and the original training number of the class  $C_i$  is  $N_i$ , a weighted parameter  $r$  is introduced to compute the weighted within-class scatter matrix  $\tilde{\mathbf{S}}_w^y$  of the incremental samples defined as follows:

$$\begin{aligned}
\tilde{\mathbf{S}}_w^y & = \sum_{i=1}^P Q_i \sum_{j=1}^{l_i} (\mathbf{y}_j^{(i)} - \mathbf{m}_i^y)(\mathbf{y}_j^{(i)} - \mathbf{m}_i^y)^T, \\
\text{where } Q_i & = \begin{cases} \frac{r \times N_i}{l_i} & \text{when } N_i \neq 0, l_i \neq 0 \\ l_i & \text{when } N_i = 0, l_i \neq 0 \\ 0 & \text{when } l_i = 0 \end{cases}
\end{aligned} \quad (18)$$

Similarly, the updated mean vector  $\mathbf{m}'_i$  of each class and the mean vector  $\mathbf{m}'$  of total samples are modified accordingly as follows:

$$\tilde{\mathbf{m}}'_i = \begin{cases} \frac{N_i \mathbf{m}_i + r N_i \mathbf{m}_i^y}{N_i(1+r)} = \frac{\mathbf{m}_i + r \mathbf{m}_i^y}{(1+r)} & \text{when } N_i \neq 0, l_i \neq 0 \\ \mathbf{m}_i & \text{when } N_i \neq 0, l_i = 0, \\ \mathbf{m}_i^y & \text{when } N_i = 0, l_i \neq 0 \end{cases} \quad i = 1, \dots, T \quad (19)$$

$$\tilde{\mathbf{m}}' = \frac{N \mathbf{m} + (rN) \mathbf{m}^y}{(1+r)N} = \frac{\mathbf{m} + r \mathbf{m}^y}{1+r} \quad (20)$$

The updated weighted between-class scatter matrix  $\tilde{\mathbf{S}}'_b$  is given as follows:

$$\begin{aligned}
\tilde{\mathbf{S}}'_b & = \sum_{i=1}^T Q_i (\mathbf{m}'_i - \mathbf{m}')(\mathbf{m}'_i - \mathbf{m}')^T, \\
\text{where } Q_i & = \begin{cases} l_i & \text{when } N_i = 0 \\ N_i & \text{when } N_i = 0 \\ (1+r)N_i & \text{when } N_i \neq 0 \end{cases}
\end{aligned} \quad (21)$$

It is worth noting that the WILDA approach is equal to the ILDA approach under the situation  $l_i=0$  or  $N_i=0$ . Since no incremental samples are added to class  $C_i$  when  $l_i=0$ ,  $N_i=0$  means that the class  $C_i$  is a newly introduced class.

According to the above equations, the weighted within-class scatter matrix  $\tilde{\mathbf{S}}'_w$  can be derived by updating (18)–(20) to Eq. (17) accordingly. Then the weighted ILDA transformation matrix  $\tilde{\mathbf{W}}'_{lda}$  can be computed by conducting the eigenvalue decomposition of  $\mathbf{S}'_w^{(-1)} \tilde{\mathbf{S}}'_b$ . We refer to this modified ILDA as WILDA in this paper.

In general, the parameter of  $r$  is for purpose of controlling the ratio of the partial training data, which are used for updating LDA parameters to the whole training data. In other words, larger weighted parameter  $r$ , which indicates a larger proportion of writer-specific incremental data for updating the LDA model parameters, means that the WILDA model turned out to be much more adapted for a particular writer's styles. This would result in higher recognition accuracy on the particular writer's handwritten samples, but much more accuracy loss for the general writer, and vice versa. Therefore, the parameter of  $r$  should be carefully chosen to achieve a good trade-off between writer-dependent and writer-independent handwriting recognition.

Later in the experimental Section 4.5, we will design a set of experiments to see how this parameter influences the performance of writer-dependent dataset and writer-independent dataset.

### 3. Classifier design and writer adaptation based on ILDA/WILDA

Suppose there are  $M$  character classes  $\{C_i\}_{i=1}^M$ , each modeled by a prototype  $\lambda_i = \{\mathbf{m}_i^p\}$ , where prototype  $\mathbf{m}_i^p$  is a  $D$  dimensional vector in some feature space. We use  $\Lambda = \{\mathbf{m}_i^p\}_{i=1}^M$  to denote the set of prototype parameters for the classifier. In this paper, we use the 8-directional feature extraction method proposed by Bai and Huo [1] to extract  $D_1$  raw feature vector  $\mathbf{x}$  for a given online handwritten Chinese character sample for which the original feature dimension is 512, i.e.  $D_1=512$ . The  $D_1$  raw features are then transformed into a new feature vector  $\mathbf{y}$  of dimension  $D$  in the LDA space by using a  $D_1 \times D$  LDA transformation matrix  $\mathbf{W}_{lda}$ , i.e.  $\mathbf{y} = \mathbf{W}_{lda}^t \mathbf{x}$ , where  $D \leq D_1$ . If  $D < D_1$ , the dimension reduction is achieved by the LDA. The diagram of our proposed writer adaptation for online handwritten Chinese character recognition using ILDA/WILDA is shown in Fig. 1, which consists of the training phase to train a general baseline classifier, the writer adaptation phase using ILDA/WILDA, and the classification phase.

In the training phase, suppose we have a training dataset  $\{\mathbf{x}_i^{(j)}\}_{i=1,2,\dots,N_j}^{j=1,2,\dots,M}$  of  $N$  training samples belonging to  $M$  classes, contributed by a large number of writers. The LDA transformation matrix  $\mathbf{W}_{lda}$  is first learned using the training data, and then the class prototype  $\mathbf{m}_i^p$  is given by

$$\mathbf{m}_j^p = \frac{1}{N_j} \sum_{i=1}^{N_j} \mathbf{W}_{lda}^t \mathbf{x}_i^{(j)} = \frac{1}{N_j} \mathbf{W}_{lda}^t \sum_{i=1}^{N_j} \mathbf{x}_i^{(j)} = \mathbf{W}_{lda}^t \mathbf{m}_j \quad (22)$$

In the classification phase, the feature vector  $\mathbf{y}$  in the LDA space is compared with each of the  $M$  character prototypes, and a discriminant function is computed for each class  $C_i$  as follows:

$$g_i(\mathbf{x}, \Lambda, \mathbf{W}_{lda}) = -\min_i \|\mathbf{y} - \mathbf{m}_i^p\| = -\min_i \|\mathbf{W}_{lda}^t \mathbf{x} - \mathbf{m}_i^p\| \quad (23)$$

The class that gives the maximum discriminant function is considered to be the recognized class, i.e.

$$\mathbf{x} \in C_k, \text{ if } k = \operatorname{argmax}_i g_i(\mathbf{x}, \Lambda, \mathbf{W}_{lda}) \quad (24)$$

In the writer adaptation phase using the ILDA or WILDA approach, when new handwritten character samples of a particular writer are presented, the LDA transformation matrix  $\mathbf{W}_{lda}$  is updated, respectively, through updating the between- and within-class scatter matrixes according to Eqs. (12) and (17) for ILDA, and Eqs. (18) and (21) for WILDA, respectively. Then the classifier prototype parameter set  $\Lambda = \{\mathbf{m}_i\}_{i=1}^M$  is updated according to Eqs. (19) and (22).

## 4. Experimental results

### 4.1. Data preparation and experimental setup

The benchmark dataset used in this paper comes from the SCUT-COUCH database. It is a revision of SCUT-COUCH2008 [21], which is now contributed by more than 168 participants. One characteristic of the SCUT-COUCH dataset is that all the samples were collected in a natural way without any guidance or constraint for the writing styles, therefore, some of the samples were written cursively. All characters were written in an unconstrained manner. This database is a comprehensive dataset composed of 8 subsets: GB1 (3755 level 1 GB2312-80) simplified Chinese character, GB2 (3038 level 2 GB2312-80) simplified Chinese character, traditional Chinese character (5041 classes), word (8888 classes), Pinyin (2010 classes), digit (10 classes), alphabet (52 classes) and symbol (122 classes). The SCUT-COUCH database is available at <http://www.hcii-lab.net/data/SCUTCOUCH/>.

Two subsets of SCUT-COUCH dataset are used in our experiments. One is the GB1 subset, which contains 168 writers' samples of 3755 categories of Chinese characters, and the other is the Word8888 subset, which consists of 30 writers' samples of 8888 categories of most frequently used handwritten words. Fig. 2 shows some typical handwritten Chinese character samples from the SCUT-COUCH GB1 subset and Word8888 subset. It is worthwhile to notice that a number of characters appear in different places of different words for many times, such as the character “我” appears in the words “我看，我国，我家，我爱，给我，让我”，etc. This indicates that when the Word8888 data were collected from a particular writer, the same character was written for many times in accordance with the context of the word corpus. Table 1 gives the statistics on the 36 most frequently reused characters to show the frequency of a character repeated in different words. Data collected in this way provide us with particularly realistic writer-independent incremental handwritten samples.

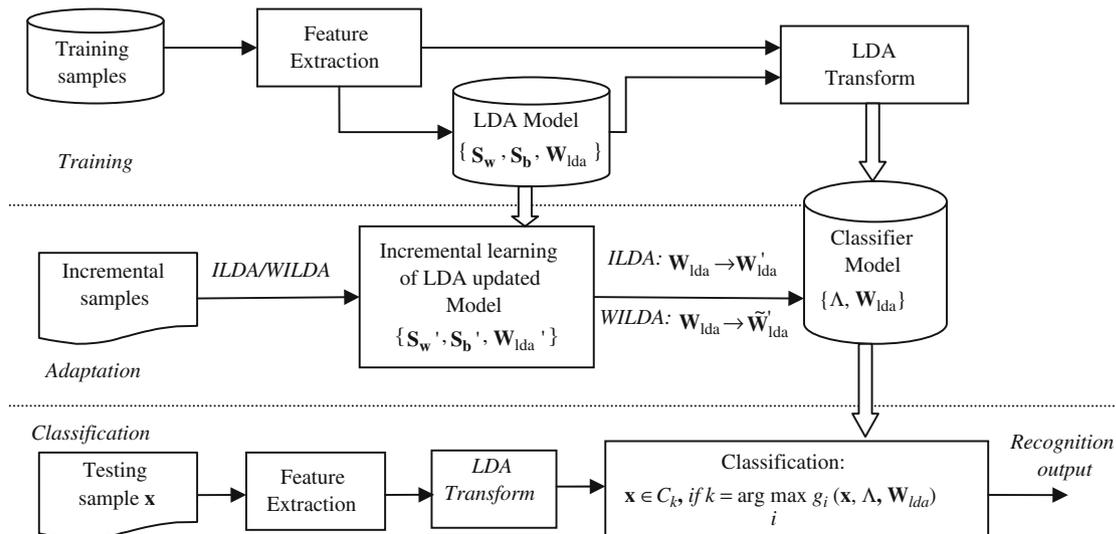


Fig. 1. Diagram of the writer adaptive handwriting recognition system.



Fig. 2. Some handwritten Chinese character samples from the SCUT-COUCH subset GB1 and Word8888: (a) 50 handwritten samples of three Chinese characters from SCUT-Couch GB1 subset and (b) the handwritten word samples contributed by three different writers that contain the corresponding three Chinese character “我” “能” “学”, respectively.

Table 1  
The statistics of the top 36 most frequently reused characters.

Character	frequency of repetition						
一	277	上	98	这	83	生	69
不	251	出	95	要	82	为	69
人	189	会	95	说	77	理	68
大	151	你	95	用	77	能	68
有	150	中	95	可	73	本	66
我	141	到	87	好	72	地	66
是	117	个	87	下	72	子	66
了	109	来	87	他	71	时	65
在	101	发	84	国	69	家	64

All of the handwritten word samples are manually segmented into isolated characters, which results in 2078 categories of 19,595 isolated Chinese characters, to form a new dataset which we name it as *IncCouchDB*. In other words, we have a general dataset that contains 168 sets of 3755 classes of GB1 Chinese character (we refer to it as *CouchGB1* thereafter) and a writer-dependent incremental dataset *IncCouchDB* that contains 30 sets of 2078 classes within GB1 level Chinese characters. The two datasets do not share any common writers. The dataset *CouchGB1* is used to train/test a baseline general purpose writer-independent LDA classifier, and then the incremental dataset *IncCouchDB* is used to train/test the Incremental LDA model for writer adaptation. It should be noted that in our experiment, new writers are introduced in the *IncCouchDB* dataset.

To build a general purpose classifier, we randomly select 134 (or 79.16%) sets of data from the *CouchGB1* to build a writer independent baseline classifier, and then use the remaining 34 (or 20.84%) sets to test the performance of the baseline classifier, as well as to evaluate the influence after the adaptation has been conducted for specific writer. For each particular writer's handwritten samples from the *IncCouchDB* dataset, we randomly select 50% of the data in each category for learning the Incremental LDA model (ILDA or WILDA), and then use the remaining 50% data to test the writer adaption performance.

#### 4.2. Baseline performance on *CouchGB1* and *IncCouchDB* before writer adaption with different LDA dimension parameters

After the classifier is trained by the 134 sets of *CouchGB1* data, its performance is evaluated on the remaining 34 sets of *CouchGB1* data and on the new *IncCouchDB* dataset (30 sets). Table 2 shows the average recognition rate of the testing sample of *CouchGB1*, with different LDA dimensions  $D$ . Table 3 shows the recognition accuracy on the dataset of *IncCouchDB*. Due to the limited length

**Table 2**  
Classification accuracy (%) for the *CouchGB1* dataset with different LDA dimension  $D$ .

LDA dimension	Recognition rate (%)			
	Top 1	Top 5	Top 10	Top 20
512	94.00	99.05	99.51	99.73
256	93.97	99.05	99.50	99.73
160	93.83	99.03	99.50	99.73
96	93.25	98.93	99.44	99.70

**Table 3**  
Classification accuracy (%) for the *IncCouchDB* dataset with different LDA dimensions  $D$ .

Writer #	$D=512$			$D=256$			$D=160$			$D=96$		
	Top 1	Top 5	Top 10	Top 1	Top 5	Top 10	Top 1	Top 5	Top 10	Top 1	Top 5	Top 10
1	88.51	97.69	98.58	88.53	97.70	98.58	88.33	97.69	98.60	87.63	97.50	98.57
2	90.45	97.95	98.70	90.39	97.95	98.70	90.08	97.94	98.68	89.43	97.80	98.68
...	...	...	...	...	...	...	...	...	...	...	...	...
5	75.33	90.30	93.39	75.29	90.35	93.40	75.01	90.19	93.28	73.89	89.67	92.91
13	69.87	87.42	91.39	69.88	87.40	91.38	69.50	87.25	91.33	68.33	86.82	91.01
16	67.44	85.39	89.89	67.39	85.33	89.84	67.04	85.11	89.66	65.87	84.34	89.12
18	60.32	81.87	87.54	60.29	81.93	87.50	59.94	81.77	87.39	58.66	80.87	86.80
22	43.83	66.23	73.95	43.88	66.21	73.92	43.43	66.00	73.67	42.16	65.06	72.99
23	91.37	98.41	98.91	91.29	98.40	98.92	91.13	98.40	98.88	90.41	98.25	98.82
...	...	...	...	...	...	...	...	...	...	...	...	...
30	85.16	96.15	97.68	85.07	96.14	97.68	84.83	95.98	97.60	83.57	95.57	97.39
Average	82.77	94.28	96.18	82.75	94.28	96.17	82.52	94.21	96.13	81.66	93.87	95.94

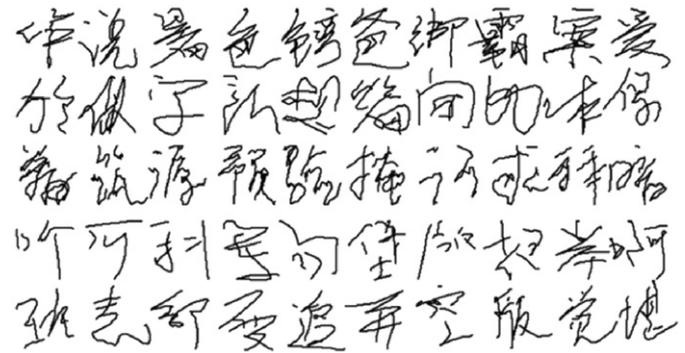
of the paper, we list only the accuracies for some typical writers, especially the writers whose samples are hard to be recognized.

From Tables 2 and 3, it can be seen that for *CouchGB1* testing dataset, we can achieve as high as 94% top 1 recognition accuracy and 99.51% top 10 accuracy when  $D=512$ . However, for the 30 sets of *IncCouchDB* data, the top 1 and top 10 average recognition rates are only 82.77% and 96.18%, respectively. We can see that several sets of samples (#5, 13, 16, 18, and 22) are particularly hard to be recognized (with very low top 1 recognition accuracies). This may be because of large deformations and the unconstrained cursive styles of these handwritten samples. Fig. 3 gives some samples taken from them.

From Tables 2 and 3, it can also be seen that the recognition accuracies do not drop down significantly when the dimension of LDA space is reduced from 512 to 160 (as less as 0.17% and 0.05% for *CouchGB1* and *IncCouchDB*, respectively). Therefore, in the following experiments, we set the dimension  $D$  for the reduced LDA space as 160.

#### 4.3. Performance on *IncCouchDB* after writer adaption using WILDA

From Table 3, we can see that the accuracy for the 30 sets of *IncCouchDB* is not good enough. This may be due to the fact that many of the writing styles of *IncCouchDB* are unseen in the training dataset. It is expected that through the incremental LDA learning on a part of specific writer training data, the baseline classifier is trained to be adapted to the writer. In this experiment, we update the LDA model using the incremental LDA algorithm on the *IncCouchDB* data. For each writer's handwritten data from the *IncCouchDB* dataset, one half the handwritten samples are



**Fig. 3.** Some handwritten samples taken from #5, 13, 16, 18, and 22 of the *IncCouchDB* dataset.

**Table 4**

Performance comparison between with and without adaptations on the IncCouchDB.

Writer #	Before WILDA (%)			After WILDA (%)			Error rate reduction (%)		
	Top 1	Top 5	Top 10	Top 1	Top 5	Top 10	Top 1	Top 5	Top 10
1	88.33	97.69	98.60	95.09	99.21	99.54	57.89	66.03	66.88
2	90.08	97.94	98.68	96.65	99.33	99.54	66.19	67.29	64.96
...	...	...	...	...	...	...	...	...	...
5	75.01	90.19	93.28	90.44	97.09	97.86	61.73	70.34	68.07
13	69.50	87.25	91.33	82.29	93.91	95.93	41.93	52.22	53.09
16	67.04	85.11	89.66	84.63	95.50	97.13	53.36	69.77	72.26
18	59.94	81.77	87.39	78.59	92.93	95.45	46.55	61.24	63.91
22	43.43	66.00	73.67	75.47	90.78	93.66	56.63	72.88	75.91
23	91.13	98.40	98.88	97.03	99.19	99.37	66.52	49.74	43.24
...	...	...	...	...	...	...	...	...	...
30	84.83	95.98	97.60	95.15	99.11	99.38	68.02	77.97	73.98
Average	82.52	94.21	96.13	92.39	98.09	98.77	56.47	67.04	68.22

**Table 5**

Performance comparison of WILDA against ILDA on writer adaptation.

Adaptation method	Recognition rate (%)		
	Top 1	Top 5	Top 10
Without adaption	82.52	94.21	96.13
ILDA adaptation	86.92	97.04	98.42
WILDA ( $r=0.1$ )	86.49	95.99	97.40
WILDA ( $r=0.2$ )	89.43	97.10	98.16
WILDA ( $r=0.3$ )	90.89	97.58	98.45
WILDA ( $r=0.4$ )	91.96	97.93	98.69
WILDA ( $r=0.5$ )	92.39	98.09	98.77
WILDA ( $r=0.6$ )	92.82	98.20	98.84

used as training data for learning the WILDA model, and then the remaining half samples are used to test the performance after the WILDA adaptation.

The performance comparison between with and without adaptations on the *IncCouchDB* is given in Table 4 in which we set the incremental weighted parameter  $r=0.5$  in this experiment. From Table 4, it can be seen that the recognition accuracies for all sets are improved significantly. In other words, the error rate is reduced dramatically. In general, the average top 1 recognition rate is improved from 82.52% to 92.39%, indicating a very high error reduction of  $56.47\% = ((17.48\% - 7.619\%) / 17.48\%)$ .

Taking the five writers (#5, 13, 16, 18, and 22) as examples, the recognition rate is improved from 75.01% to 90.44% after WILDA model adaption for writer #5, whereas the error rate decreases from 24.99% to 9.56%, resulting in a reduction as large as  $61.73\% = (24.99\% - 9.56\%) / 24.99\%$ .

Similarly, it can be seen that the recognition rate is improved from 69.50% to 82.29% for writer #13, 67.04% to 84.63% for writer #16, 59.94% to 78.59% for writer #18, and 43.43% to 75.47% for writer #22, respectively. Overall speaking, the results in Table 4 clearly indicate that the WILDA learning algorithm is very useful and effective for the performance improvement for writer-specific adaptation.

#### 4.4. Performance comparison of writer adaption using WILDA against ILDA

This experiment is designed to examine the performance of writer adaptation using the proposed WILDA algorithm against the ILDA algorithm. The experimental results on *IncCouchDB* dataset are given in Table 5.

From Table 5, it can be seen that by applying the ILDA writer adaption learning, the top 1 recognition rate is improved from 82.52% to 86.92%, while by applying the proposed WILDA writer

**Table 6**Performance comparison of IncCouchDB dataset with different weighting parameters  $r$ .

$R$	Before adaptation (%)	After adaptation (%)	Error rate reduction (%)
0.05	82.52	84.68	12.36
0.1	82.52	86.49	22.71
0.2	82.52	89.43	39.53
0.3	82.52	90.89	47.88
0.4	82.52	91.96	54.00
0.5	82.52	92.39	56.47
0.6	82.52	92.82	58.92

adaption learning algorithm, it is improved to higher accuracies ranging from 86.49% to 92.82% according to different values of  $r$ . It can also be seen that when the weighted parameter  $r$  is no less than 0.2, the recognition accuracy of the WILDA writer adaption approach is significantly higher than the ILDA approach. This clearly indicates that the proposed WILDA method is better than the ILDA method for writer adaption in the application of online Chinese handwritten character recognition.

#### 4.5. Performance comparison on IncCouchDB using WILDA with different weighted parameters $r$

This experiment is designed to examine the effect of different weighted parameters  $r$  on the recognition performance. We set the parameter  $r$  from 0.05 to 0.6, and the comparison of average recognition rate of the 30 sets in *IncCouchDB* dataset before and after adaptation is shown in Table 6.

Clearly, it is observed from Table 6 that our writer adaptation method using WILDA can significantly reduce the error rate for the 30 sets of writer-dependent dataset; meanwhile, the recognition accuracy is increased in accordance with the increase of the weighted parameter  $r$ . This is reasonable due to the fact that larger updating ratio  $r$  means that the new writer dependent incremental data will have more contributions to updating the WILDA model, resulting in better performance of the corresponding updated classifier on the writer-specific data. From Table 6, it can be seen that when the updating ratio is 0.6, the recognition accuracy is improved from original 82.52% to 92.82%.

#### 4.6. Performance comparison on CouchGB1 using WILDA and ILDA

It seems that the recognition performance after using the ILDA/WILDA-based adaptation always achieves significant improvement

**Table 7**  
Performance comparison of writer-independent *CouchGB1* dataset.

	Without adaptation (%)	With adaptation (%)	Accuracy loss (%)
ILDA adaptation	93.83	93.62	0.21
WILDA ( $r=0.05$ )	93.83	93.63	0.2
WILDA ( $r=0.1$ )	93.83	93.58	0.25
WILDA ( $r=0.2$ )	93.83	93.32	0.51
WILDA ( $r=0.3$ )	93.83	92.98	0.85
WILDA ( $r=0.4$ )	93.83	92.50	1.33
WILDA ( $r=0.5$ )	93.83	91.96	1.87
WILDA ( $r=0.6$ )	93.83	91.33	2.5

for all the writer-specific datasets in *IncCouchDB*. However, a question may be whether such kind of adaptation has dramatic negative impact on the general purpose writer-independent dataset *CouchGB1*. This is particularly an important issue to be taken for consideration, because we do not expect the adaptation to specific writer's handwriting style at the cost of losing much generality for other writer styles. To examine how the writer adaptation using the ILDA/WILDA affects the performance on general writer-independent dataset, the recognition results of ILDA/WILDA based writer adaptation approach on the general purpose writer-independent dataset *CouchGB1* after the classifier has been updated using the specific user-dependent *IncCouchDB* have been demonstrated in Table 7. This table also shows how the parameter of  $r$  influences the performance of the WILDA based adaptation approach on the writer-independent dataset *CouchGB1*.

As shown in Table 7, although the recognition accuracies for the general purpose dataset decrease after the adaptation, the loss is very small ( $< 3\%$ ), especially for ILDA and WILDA with small values of  $r$ . This indicates that the proposed writer adaptation methods can significantly reduce the error rate for writer-dependent dataset. In the mean time, it has little negative impact on a writer-independent testing dataset.

From Table 7, it can be observed that when  $r < 0.4$ , the accuracy loss on writer-independent dataset is much smaller ( $< 1\%$ ). When  $r \geq 0.4$ , the proposed method may lose more than 1% accuracy on writer-independent testing dataset. However, such quantity of loss is acceptable (less than 3% even for large updating ratio  $r$ ). Comparing Table 6 with Table 7, it can be found that: (1) the improvement of recognition accuracy for writer-specific dataset is much more significant than the accuracy loss for general writer-independent dataset; (2) the larger the updating ratio  $r$  is, the higher recognition accuracy is for writer-specific data, and the lower but acceptable recognition accuracy is for general writer-independent dataset; and (3) a trade-off should be found between the performance for a specific user dataset and that of the general dataset.

In a practical view, we suggest that the reasonable range of  $r$  should be taken from 0.1 to 0.3. Under such settings, the proposed adaptation method can reduce about 20.44–46.35% error rate on the writer-dependent dataset while it has only less than 0.85% accuracy loss on the writer-independent general dataset.

## 5. Conclusion

Writer adaptation converts a writer-independent system, which is trained from the data contributed by a large group of writers, to a writer-dependent system, which is turned for a particular writer using a specific incremental data. This adaption has potential advantage of significantly increasing recognition accuracies for a particular writer, so it is very useful for a real world application, such as building a *personalized* online handwritten character input method. In this paper, a general solution

for incremental linear discriminant analysis (ILDA) is presented, and a weighted incremental linear discriminant analysis (WILDA) method by considering the issue of uncertain number of incremental data for writer adaptation is proposed for online handwritten Chinese character recognition. Based on the incremental learning of the LDA model using ILDA or WILDA, the writer adaptation is performed by updating the LDA transformation matrix and the classifier prototypes in the feature space. From the experimental results on general purpose writer-independent dataset *CouchGB1* and writer-dependent dataset *IncCouchDB*, we can draw the following conclusions:

- (1) Both ILDA and WILDA are very effective to improve the recognition accuracy for different particular writers.
- (2) The WILDA outperforms the ILDA for writer adaptation. The proposed WILDA method can reduce as much as 47.88% error rate on the *IncCouchDB* dataset while it only has as little as 0.85% accuracy loss on the *CouchGB1* dataset when the weighted parameter  $r$  is set to 0.3, showing the effectiveness of the proposed writer adaptation approach.
- (3) One good property of our writer adaptation approach based on WILDA is that it can significantly increase the recognition accuracy for the specific writer. In the meantime, it has little negative impact on the performance of general writers.

## Acknowledgments

We would like to thank all anonymous reviewers for their valuable suggestions. This work is supported in part by NSFC (Grant no. U0735004, 60772116), GDSFC (no. 07118074) and Microsoft Research Asia (MSAR) fund (Grant no. FY08-RES-THEME-157).

## Appendix

### Proof A.

$$\begin{aligned}
& \frac{N_j l_j^2}{(N_j + l_j)^2} (\mathbf{m}_j^y - \mathbf{m}_j) (\mathbf{m}_j^y - \mathbf{m}_j)^T + \frac{N_j^2}{(N_j + l_j)^2} \sum_{i=1}^{l_j} (\mathbf{y}_i^{(j)} - \mathbf{m}_j) (\mathbf{y}_i^{(j)} - \mathbf{m}_j)^T \\
& + \frac{l_j (l_j + 2N_j)}{(N_j + l_j)^2} \sum_{i=1}^{l_j} (\mathbf{y}_i^{(j)} - \mathbf{m}_j^y) (\mathbf{y}_i^{(j)} - \mathbf{m}_j^y)^T \\
& = \frac{1}{(N_j + l_j)^2} \left\{ N_j^2 \left( \sum_{i=1}^{l_j} \mathbf{y}_i^{(j)} \mathbf{y}_i^{(j)T} - l_j \mathbf{m}_j^y \mathbf{m}_j^{yT} \right) \right. \\
& + l_j N_j (l_j + N_j) (\mathbf{m}_j \mathbf{m}_j^T + \mathbf{m}_j^y \mathbf{m}_j^{yT} - \mathbf{m}_j^y \mathbf{m}_j^T - \mathbf{m}_j \mathbf{m}_j^{yT}) \\
& \left. + l_j (l_j + 2N_j) \sum_{i=1}^{l_j} (\mathbf{y}_i^{(j)} - \mathbf{m}_j^y) (\mathbf{y}_i^{(j)} - \mathbf{m}_j^y)^T \right\} \\
& = \frac{1}{(N_j + l_j)^2} \left\{ N_j^2 \left( \sum_{i=1}^{l_j} (\mathbf{y}_i^{(j)} - \mathbf{m}_j^y) (\mathbf{y}_i^{(j)} - \mathbf{m}_j^y)^T \right) \right. \\
& + (l_j^2 + 2N_j l_j) \sum_{i=1}^{l_j} (\mathbf{y}_i^{(j)} - \mathbf{m}_j^y) (\mathbf{y}_i^{(j)} - \mathbf{m}_j^y)^T \\
& \left. + l_j N_j (l_j + N_j) (\mathbf{m}_j \mathbf{m}_j^T + \mathbf{m}_j^y \mathbf{m}_j^{yT} - \mathbf{m}_j^y \mathbf{m}_j^T - \mathbf{m}_j \mathbf{m}_j^{yT}) \right\} \\
& = \frac{1}{(N_j + l_j)^2} \left\{ (l_j + N_j)^2 \Sigma_j^y + l_j N_j (l_j + N_j) (\mathbf{m}_j^y - \mathbf{m}_j) (\mathbf{m}_j^y - \mathbf{m}_j)^T \right\} \\
& = \Sigma_j^y + \frac{N_j l_j}{(N_j + l_j)} (\mathbf{m}_j^y - \mathbf{m}_j) (\mathbf{m}_j^y - \mathbf{m}_j)^T \quad \square
\end{aligned}$$

## References

- [1] Z. Bai, Q. Huo, A study on the use of 8-directional features for online handwritten Chinese character recognition, in: International Conference on Document Analysis and Recognition (ICDAR), 2005, pp. 232–236.

- [2] C.M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, 2006.
- [3] S.D. Connel, A.K. Jain, Writer adaptation of online handwriting models, *IEEE Trans. Pattern Anal. Mach. Intell.* 24 (3) (2002) 329–346.
- [4] H.P. Decell, S.M. Mayekar, Feature combinations and the divergence criterion, *Comput. Math. Appl.* 3 (1977) 71–76.
- [5] K. Ding, G. Deng, L. Jin, An investigation of imaginary stroke technique for cursive online handwriting Chinese character recognition, in: *International Conference on Document Analysis and Recognition (ICDAR)*, 2009, pp. 531–535.
- [6] R.O. Duda, P.E. Hart, D.G. Stork, *Pattern Classification*, Wiley, New York, 2001.
- [7] H. Fu, H. Chang, Y. Xu, H. Pao, User adaptive handwriting recognition by self-growing probabilistic decision-based neural networks, *IEEE Trans. Neural Networks* 11 (6) (2000) 1373–1384.
- [8] H. Fujisawa, Forty years of research in character and document recognition—an industrial perspective, *Pattern Recognition* 41 (2008) 2435–2446.
- [9] I. Guyon, C. Warwick, Joint EC-US survey of the state-of-the-art in human language technology, <<http://www.cse.ogi.edu/CSLU/HLTsurvey.htm>>, 1996.
- [10] J. Han, B. Bhanu, Individual recognition using gait energy image, *IEEE Trans. Pattern Anal. Mach. Intell.* 28 (2) (2006) 316–322.
- [11] T. Hastie, R. Tibshirani, Discriminant analysis by gaussian mixtures, *J. R. Statist. Soc. Ser. B: Methodological* 58 (1996) 155–176.
- [12] Q. Huo, T. He, A minimax classification approach to HMM-based online handwritten Chinese character recognition robust against affine distortions, in: *International Conference on Document Analysis and Recognition (ICDAR)*, 2007, pp. 43–47.
- [13] A.K. Jain, S. Prabhakar, L. Hong, A multichannel approach to fingerprint classification, *IEEE Trans. Pattern Anal. Mach. Intell.* 21 (4) (1999) 348–359.
- [14] B. Jelinek, Review on heteroscedastic discriminant analysis, unpublished report, Center for Advanced Vehicular Systems, Mississippi State University, 2001.
- [15] X. Jing, D. Zhang, A face and palmprint recognition approach based on discriminant DCT feature extraction, *IEEE Trans. Syst. Man Cybern. Pt. B* 34 (6) (2004) 2,405–2,415.
- [16] W. Kienzle, K. Chellapilla, Personalized handwriting recognition via biased regularization, in: *International Conference on Machine Learning (ICML)*, 2006, pp. 457–464.
- [17] T.K. Kim, S.F. Wong, B. Stenger, J. Kittler, R. Cipolla, Incremental linear discriminant analysis using sufficient spanning set approximations, in: *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007, pp. 1–7.
- [18] W.J. Krzanowski, P. Jonathan, W.V. McCarthy, M.R. Thomas, Discriminant analysis with singular covariance matrices: methods and applications to spectroscopic data, *Appl. Stat.* 44 (2005) 101–115.
- [19] N. Kumar, A.G. Andreou, Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition, *Speech Commun.* 26 (1998) 283–297.
- [20] J.J. LaViola, R.C. Zeleznik, A practical approach for writer-dependent symbol recognition using a writer-independent symbol recognizer, *IEEE Trans. Pattern Anal. Mach. Intell.* 29 (11) (2007) 1917–1926.
- [21] Y. Li, L. Jin, X. Zhu, T. Long, SCUT-COUCH2008: a comprehensive online unconstrained Chinese handwriting dataset, in: *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2008, pp. 165–170, <<http://www.hcii-lab.net/data/SCUTCOUCH/>>.
- [22] C. Liu, S. Jaeger, M. Nakagawa, Online recognition of Chinese characters: the state-of-the-art, *IEEE Trans. Pattern Anal. Mach. Intell.* 26 (2) (2004) 198–213.
- [23] C. Liu, X. Zhou, Online Japanese character recognition using trajectory-based normalization and direction feature extraction, a comprehensive online unconstrained Chinese handwriting dataset, in: *International Workshop on Frontiers in Handwriting Recognition (IWFHR)*, 2006, pp. 217–222.
- [24] H. Liu, X. Ding, Handwritten character recognition using gradient feature and quadratic classifier with multiple discrimination schemes, in: *International Conference on Document Analysis and Recognition (ICDAR)*, 2005, pp. 19–23.
- [25] T. Long, L. Jin, A novel orientation free method for online unconstrained cursive handwritten Chinese word recognition, in: *International Conference on Pattern Recognition (ICPR)*, 2008, pp. 1–4.
- [26] M. Loog, R.P.W. Duin, R. Haeb-Umbach, Multiclass linear dimension reduction by weighted pairwise Fisher criteria, *IEEE Trans. Pattern Anal. Mach. Intell.* 23 (7) (2001) 762–766.
- [27] M. Loog, R.P.W. Duin, Linear dimensionality reduction via a heteroscedastic extension of LDA: the Chernoff criterion, *IEEE Trans. Pattern Anal. Mach. Intell.* 26 (6) (2004) 732–739.
- [28] R. Lotlikar, R. Kothari, Fractional-step dimensionality reduction, *IEEE Trans. Pattern Anal. Mach. Intell.* 22 (6) (2000) 623–627.
- [29] J. Lu, K.N. Plataniotis, A.N. Venetsanopoulos, Face recognition using LDA based algorithms, *IEEE Trans. Neural Networks* 14 (1) (2003) 195–200.
- [30] S. Pang, S. Ozawa, N. Kasabov, Incremental linear discriminant analysis for classification of data streams, *IEEE Trans. Syst. Man Cybern.* 35 (5) (2005) 905–914.
- [31] P.J. Phillips, H. Moon, S.A. Rizvi, P.J. Rauss, The FERET evaluation methodology for face-recognition algorithms, *IEEE Trans. Pattern Anal. Mach. Intell.* 22 (10) (2000) 1090–1104.
- [32] *IEEE Trans. Pattern Anal. Mach. Intell.* 22 (1) (2000) 63–84.
- [33] M. Szummer, C.M. Bishop, Discriminative writer adaptation, in: *International Workshop on Frontiers in Handwriting Recognition (IWFHR)*, 2006, pp. 281–287.
- [34] D. Tao, Discriminative linear and multilinear subspace methods, Ph.D. Thesis, School of Computer Science and Information Systems, Birkbeck College, University of London, 2006.
- [35] D. Tao, X. Li, X. Wu, S.J. Maybank, General averaged divergences analysis, in: *IEEE International Conference on Data Mining*, 2007, pp. 302–311.
- [36] D. Tao, X. Li, X. Wu, S.J. Maybank, Geometric mean for subspace selection, *IEEE Trans. Pattern Anal. Mach. Intell.* 31 (2) (2009) 260–274.
- [37] D. Tao, X. Li, X. Wu, S.J. Maybank, General tensor discriminant analysis and Gabor features for gait recognition, *IEEE Trans. Pattern Anal. Mach. Intell.* 9 (10) (2007) 1700–1715.
- [38] V. Vuori, Adaptive methods for on-line recognition of isolated handwritten characters, Helsinki University of Technology, Helsinki, 2002.
- [39] D. Wang, C. Liu, J. Yu, X. Zhou, CASIA-OLHWDB1: a database of online handwritten Chinese characters, in: *International Conference on Document Analysis and Recognition (ICDAR)*, 2009, pp. 1206–1210.
- [40] J. Ye, R. Janardan, L. Qi, Two-dimensional linear discriminant analysis, *Neural Inform. Process. Syst.* 17 (2005) 1569–1576.
- [41] H. Zhao, P.C. Yuen, Incremental linear discriminant analysis for face recognition, *IEEE Trans. Syst. Man Cybern.* 38 (2008) 210–221.



**Lianwen Jin** received a B.S. degree from the University of Science and Technology of China and a Ph.D. degree from South China University of Technology in 1991 and 1996, respectively. He visited Motorola China Research Center in 2000 and the University of Hong Kong in 2002 and 2006 as a research fellow, respectively. He is now a professor at the School of Electronic and Information Engineering, South China University of Technology. He received the New Century Excellent Talent Program Award of China MOE in 2006. He has published more than 90 papers in the areas of handwritten Chinese character recognition, image processing and pattern recognition. His current research

interests include character recognition, pattern analysis and recognition, image processing, machine learning and intelligent systems. He is a member of the IEEE and IEEE Computer Society. He served as Program Committee member for a number of international conferences, including IWFHR '06, ICFHR '08, ICDAR '09, ICMLC '07, ICMLC '08, ICMLC '09, ICFHR '10.



**Kai Ding** received his B.S. degree in 2006, and is a Ph.D. candidate in Communication and Information System at South China University of Technology, Guangzhou, China. His research interests include pattern recognition, machine learning and digital image processing.



**Zhibin Huang** received his B.S. degree in 2007, and is pursuing his Master's degree, in Communication and Information System at South China University of Technology, Guangzhou, China. His research interests include pattern recognition, machine learning and intelligent system.