# Style Transfer Matrix Learning for Writer Adaptation

Xu-Yao Zhang Cheng-Lin Liu
National Laboratory of Pattern Recognition (NLPR)
Institute of Automation, Chinese Academy of Sciences, Beijing, China
xyz@nlpr.ia.ac.cn liucl@nlpr.ia.ac.cn

## Abstract

*In this paper, we propose a novel framework of style transfer matrix (STM) learning to reduce the writing style variation in handwriting recognition. After writer-specific style transfer learning, the data of different writers is projected onto a style-free space, where a writer independent classifier can yield high accuracy. We combine STM learning with a specific nearest prototype classifier: learning vector quantization (LVQ) with discriminative feature extraction (DFE), where both the prototypes and the subspace transformation matrix are learned via online discriminative learning. To adapt the basic classifier (trained with writer-independent data) to particular writers, we first propose two supervised models, one based on incremental learning and the other based on supervised STM learning. To overcome the lack of labeled samples for particular writers, we propose an unsupervised model to learn the STM using the self-taught strategy (also known as self-training). Experiments on a large-scale Chinese online handwriting database demonstrate that STM learning can reduce recognition errors significantly, and the unsupervised adaptation model performs even better than the supervised models.*

## 1. Introduction

Many classification models assume that the training and test data are drawn from the same distribution. However, many applications observe changing distributions, such as the accents change in speech recognition, writing style change in handwriting recognition, and viewing conditions change in image classification. The difference between the distributions of the training and test data is known as the concept drift [11]. To improve the generalization performance in situations of changing distribution, we should transfer the classifier learned on the training data to the new distribution of the test data, which is known as transfer learning [20].

The large variability of handwriting styles across individuals makes handwriting recognition a challenging problem. Writer adaptation is the process of converting a generic (writer-independent) classifier into a personalized (writer-dependent) one to improve the individual recognition accuracy. While training the generic classifier needs a large amount of writer-independent data, the adaptation process uses only a few labeled (supervised writer adaptation) or unlabeled (unsupervised writer adaptation) samples from a single writer. Writer adaptation is closely related to other topics like speaker adaptation [13] in speech recognition, domain adaptation in natural language processing, multi-task learning [6] and transfer learning [20].

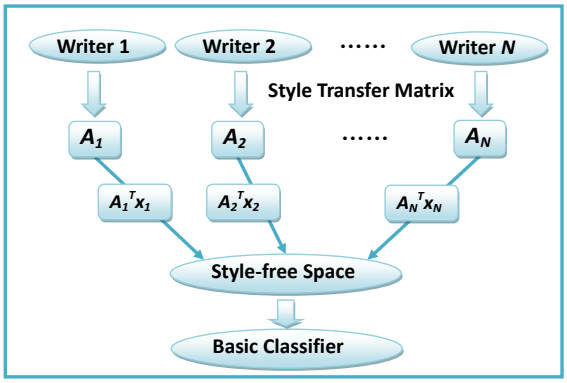



Figure 1. The framework of style transfer matrix (STM) learning for writer adaptation. The STM is a writer-specific class-independent feature transformation matrix, which can be combined with different types of classifiers, for both supervised and unsupervised writer adaptation.

In this paper, we propose a novel framework to learn a style transfer matrix (STM), which can be combined with different types of classifiers, for both supervised and unsupervised writer adaptation.

To alleviate the influence of writing style variation among individuals, a writer-specific style transfer matrix (STM) is learned for each writer (Figure 1). The samples of each writer are projected onto a style-free space by its own STM. As a result, the transformed samples of different writers are expected to lie in a uniform style space. Then

we can use a style-free classifier on the transformed data for high accuracy classification. To get a balance between style transfer and non-transfer, we pose a regularization term to constrain the change of the STM from the identity matrix, which can avoid over-transfer. Depending on whether the new adaptation samples for STM learning are labeled or not, we propose two models to supervised STM learning and unsupervised STM learning, respectively.

We combine our STM learning strategy with an efficient nearest prototype classifier called DFE+LVQ: learning vector quantization (LVQ) with discriminative feature extraction (DFE). Nearest prototype classifier is a good choice for classification of large category set, and yields high accuracy via discriminative prototype learning and DFE. Our experiments demonstrate the effectiveness of STM learning to improve the accuracy of DFE+LVQ.

The rest of this paper is organized as follows. Section 2 gives a brief review of related works; Section 3 introduces the basic DFE+LVQ classifier and Section 4 introduces writer adaptation using DFE+LVQ based on online learning; Section 5 describes the proposed framework of style transfer matrix (STM) learning; Section 6 specifies the STM to supervised and unsupervised writer adaptation; Section 7 presents our experimental results on Chinese online handwriting recognition and Section 8 offers concluding remarks.

## 2. Related Works

The methods of writer adaptation can be divided into two main categories: supervised writer adaptation and unsupervised writer adaptation.

There have been many works of supervised writer adaptation, which can be summarized into four main categories.

- Writer-specific class-independent feature transformation learning. For example, maximum likelihood linear regression (MLLR) estimates a regression matrix to maximize the likelihood on the adaptation data, and then the means of the Gaussians in the HMM (hidden Markov model) are re-estimated by transformation with the regression matrix. MLLR was firstly proposed for speaker adaptation [13], and has been used for writer adaptation as well [2, 24]. Another example is the incremental learning of Fisher linear discriminant analysis (ILDA) [10], where the LDA transformation matrix is updated with the new labeled samples, and then a nearest class mean classifier is used for classification.

- Incremental or online classifier learning. For neural network adaptation, Matic et al. proposed to retrain the last layer of a 5-layer time delay neural network by one-vs-all linear SVM [18]. Platt and Matic place an output adaptation module of radial basis function (RBF) network on the top of a standard neural networks for writer adaptation [19]. For SVM adaptation, Kienzle and Chellapilla apply a biased regularization (a tradeoff between the new classifier and the original one) to retrain the classifier using the new writer-specific data [12]. Vuori and Korkeakoulu propose three methods for template classifier adaptation: adding new prototypes, reshaping existing prototypes and inactivating poorly performing prototypes [26].

- Multiple classifier systems. An example is the combination of a writer-dependent recognizer with a writer-independent one [14].

- Writer style clustering. By grouping the writing styles into different clusters [5, 23], different classifiers can be trained corresponding to different style clusters.

Unsupervised writer adaptation is closely related to semi-supervised transfer learning. In this situation, we should learn from both the writer-independent labeled data and writer-dependent unlabeled data, and style transfer exists between the two datasets. There have not been many works in this direction. Ball and Srihari use a self-training strategy for HMM model retraining for handwriting recognition [4], and Frinken and Bunke use self-training for adapting a neural network classifier for handwritten words and sentence recognition [8]. A related method is called field classification [25] [27], which learns the style context (inter-pattern style dependence) in a group of patterns. By utilizing style consistency, classifying groups (or fields) of patterns is often more accurate than classifying single patterns.

The above supervised adaptation methods do not apply to unsupervised situation very well, while the unsupervised adaptation methods (e.g. self-training for HMM and neural network) were designed for specific classifiers. For Chinese handwriting recognition of large category set, other types of classifiers are more suitable. For the transformation based methods, ILDA is a supervised method only suitable for incremental training of LDA; and MLLR is designed for probabilistic models like Gaussians in HMM, and there is no strategy (like regularization) to avoid over-transfer in MLLR.

There has not been a general framework for writer adaptation of different classifiers. In this paper, the proposed style transfer matrix (STM) is a writer-specific class-independent feature transformation matrix, which can be combined with different types of classifiers, for either supervised or unsupervised writer adaptation.

## 3. DFE+LVQ Classifier

We use an efficient nearest prototype classifier, DFE+LVQ, for large category set Chinese character recog-

nition. The nearest prototype classifier has few prototypes for each class and is therefore efficient in computation. And via discriminative prototype learning and feature extraction (dimensionality reduction), the nearest prototype classifier can yield high classification accuracy. Discriminative feature extraction (DFE) [1] is different from the popular linear feature extraction methods (e.g. PCA, LDA, ICA, NMF, LPP, etc.) in that the feature subspace parameters are jointly trained with the classifier parameters to optimize a classification objective. Combining DFE with prototype learning by learning vector quantization (LVQ) has yielded high accuracies in character recognition [15].

The DFE+LVQ classifier learns a feature subspace and class prototypes on the subspace simultaneously. Given a dataset $\{x_i, y_i\}_{i=1}^N$, where $x_i \in \mathbb{R}^d$ and $y_i \in \{1, \cdots, C\}$ ($C$ is the number of classes), the task is to learn a feature transformation matrix $W \in \mathbb{R}^{d \times d'}$ ($d' < d$) and prototypes $\{m_k, y_k\}_{k=1}^K$, where $m_k \in \mathbb{R}^{d'}$ and $y_k \in \{1, \cdots, C\}$. The decision rule in classifying a test sample $x \in \mathbb{R}^d$ is that: assigning $x$ to class $y_{k_0}$ (the label of $m_{k_0}$), where

$$k_0 = \arg\min_{k=1}^{K} \left\| W^T x - m_k \right\|_2^2. \tag{1}$$

For discriminative learning of feature transformation matrix and classifier parameters, a loss function $\ell(\mu)$ is defined based on a misclassification measure $\mu(x)$, and the empirical loss on training data is minimized using stochastic gradient descent.

For a sample $x_i$, let

$$\begin{aligned} + \ &\text{denotes} \ \ \arg\min_{k:\ y_k = y_i} \left\| W^T x_i - m_k \right\|_2^2, \\ - \ &\text{denotes} \ \ \arg\min_{k:\ y_k \neq y_i} \left\| W^T x_i - m_k \right\|_2^2, \\ d_+(x_i) &= \left\| W^T x_i - m_+ \right\|_2^2, \\ d_-(x_i) &= \left\| W^T x_i - m_- \right\|_2^2, \end{aligned} \tag{2}$$

which mean that $+$ is the nearest genuine prototype (from the labeled class of $x_i$), and $-$ is the nearest rival prototype (from negative classes). Following the generalized LVQ (GLVQ) method [21], a normalized misclassification measure is defined as

$$\mu(x_i) = \frac{d_+(x_i) - d_-(x_i)}{d_+(x_i) + d_-(x_i)}. \tag{3}$$

It is easy to see that $x_i$ is misclassified when $\mu(x_i) > 0$ and correctly classified when $\mu(x_i) < 0$. The normalized misclassification measure $\mu(x_i)$ is ranged in $[-1, +1]$, and functions like a normalized margin. It has recently been used for kernel classifier learning as well [22].

Based on the normalized misclassification measure, we adopt a convex loss function which has been widely used in logistic regression [3] and recently for prototype learning [9]. This method approximates the posterior probability of genuine class by sigmoidal function:

$$P(y_i|x_i) = \frac{1}{1 + e^{\xi \mu(x_i)}}, \tag{4}$$

where the scaling parameter $\xi$ can be set to 1 in the case of normalized misclassification measure. Then maximizing the posterior likelihood $\max \prod_{i=1}^N P(y_i|x_i)$ is equal to $\min \sum_{i=1}^N -\log P(y_i|x_i)$. This defines the loss function as

$$\ell(x_i) = -\log P(y_i|x_i) = \log(1 + e^{\xi \mu(x_i)}). \tag{5}$$

This loss function is a differentiable convex function of misclassification measure $\mu(x_i)$, and it's curve shape is close to the hinge loss. The learning objective is to minimize the empirical loss on all training samples:

$$\min_{W, m_1, \cdots, m_K} \ \sum_{i=1}^{N} \ell(x_i). \tag{6}$$

By stochastic gradient descent, the feature transformation matrix and prototypes are updated iteratively on the coming sample $x_t$:

$$\begin{aligned} W^{t+1} &= W^t - \eta_t \frac{\partial \ell(x_t)}{\partial W}, \\ m^{t+1} &= m^t - \eta_t \frac{\partial \ell(x_t)}{\partial m}, \end{aligned} \tag{7}$$

here we omit the subscript of $m$ for simplification, and $\eta_t$ is the learning rate. We can use LDA or random projection to initialize $W$ and $k$-means to initialize $m_k, k = 1, \cdots, K$ (i.e. the initial prototypes of each class are the cluster centers of class-wise training data).

## 4. Online Learning for Adaptation

To adapt the basic DFE+LVQ classifier (trained with writer-independent data) to particular writers, we may have labeled samples $\{x_j, y_j\}_{j=1}^L$ from a particular writer. A simple method is to retrain the parameters on the newly coming writer-specific data. Since the training process of DFE+LVQ is already in an online fashion, it is straightforward for adaptation. We can simply re-train the classifier based on (7) using the writer-specific data $\{x_j, y_j\}_{j=1}^L$. To balance the confidence between the initial writer-independent data and the additional writer-specific data, we can set a much smaller learning rate ($\eta' << \eta$) in the adaptation process. We call this method "online adaptation LVQ (on-LVQ)".

Although on-LVQ is simple and straightforward, it needs a large number of writer-specific labeled samples to achieve

stable performance. Very often, we have only a small number of labeled samples (not covering all classes). In extreme case, we may have no labeled samples for a particular writer, but instead, a lot of unlabeled samples are available. How to make full use of these unlabeled samples is known as unsupervised writer adaptation. In the following, we propose a novel quick and efficient style transfer matrix learning framework to handle the problem of lack of labeled samples in adaptation, which can be adopted for both supervised and unsupervised adaptation.

## 5. Style Transfer Matrix Learning

The objective of style transfer matrix learning is to map a point set from a source domain to a target domain. Suppose we have a target point set $T = \{t_j \in \mathbb{R}^D \mid j = 1, \cdots, M\}$. Because of the style variation during handwriting generation, the target point set $T$ is changed to a source point set $S = \{s_j \in \mathbb{R}^D \mid j = 1, \cdots, M\}$. This change is also known as concept drift. To model this change, we have the one-to-one correspondence between the points in $S$ and $T$. Suppose $t_j$ is transformed to $s_j$ with confidence $f_j \in [0, 1]$. Now we want to learn the inverse transformation function to transform $S$ back to $T$. For simplification, we assume that the inverse style transformation between $S$ and $T$ is linear, which can embrace rotation, scaling and shear transformation.

The style transfer matrix (STM) $A \in \mathbb{R}^{D\times D}$ can be learned form the correspondence between $S$ and $T$ by minimizing the weighted squared error:

$$\min_{A\in\mathbb{R}^{D\times D}} \sum_{j=1}^{M} f_j \left\|A^T s_j - t_j\right\|_2^2. \tag{8}$$

In the case of identical points in $S$ and $T$, which means no style change, we obtain a STM $A = I$ ($I$ is the identity matrix). When $S$ and $T$ are different, in order to obtain better generalization performance, we can also add a regularization term to avoid over-transfer. This is to constrain the deviation of $A$ from the identity matrix:

$$\min_{A\in\mathbb{R}^{D\times D}} \sum_{j=1}^{M} f_j \left\|A^T s_j - t_j\right\|_2^2 + \beta \left\|A^T - I\right\|_F^2. \tag{9}$$

This is a multiple regression model with a biased regularization. In this formulation, the computation of $A$ is a quadratic programming problem, and has a closed-form solution:

$$A^T = \left[\sum_{j=1}^{M} f_j t_j s_j^T + \beta I\right] \left[\sum_{j=1}^{M} f_j s_j s_j^T + \beta I\right]^{-1}. \tag{10}$$

Since a identity matrix is added to the second term on the right hand of the equation, it is always invertible.

The hyper-parameter $\beta$ acts as a tradeoff between transfer and non-transfer. A large value of $\beta$ results in a matrix close to the identity matrix in favor of non-transfer, and a small value of $\beta$ will lead to over-transfer which may deteriorate the generalization performance. Considering the influence of data scaling, we suggest to set $\beta$ as

$$\beta = \frac{\tilde{\beta}}{2D}\left[||\mathrm{diag}(\sum_{j=1}^{M} f_j s_j s_j^T)||_1 + ||\mathrm{diag}(\sum_{j=1}^{M} f_j t_j s_j^T)||_1\right], \tag{11}$$

where $\tilde{\beta}$ can be selected from $[0, 3]$ effectively via cross-validation or other methods.

## 6. Adaptation Based on STM

The proposed STM model can be combined with many different types of classifiers. The key problem is how to define the source point set $S$ , the target point set $T$, and the corresponding confidence $f_j$. For handwriting recognition, we can define the source point set as the writer-specific data, and the target point set as some corresponding parameters in the basic classifier (trained with writer-independent data). In this way, we can learn a style transfer matrix to transform the writer-specific data toward writer-independent style, while the basic classifier needs no change to classify the transformed data. Specifically, we combine the STM model with the DFE+LVQ classifier for both supervised and unsupervised adaptation.

### 6.1. Supervised Adaptation

For supervised adaptation, we have extra labeled samples (adaptation samples) $\{x_j, y_j\}_{j=1}^{L}$ from a particular writer. In this situation, the source point set is defined as the projected data of adaptation samples onto the DFE subspace:

$$S = \left\{s_j = W^T x_j \mid j = 1, \cdots, L\right\}, \tag{12}$$

and the target point set is defined as the set of the corresponding genuine prototypes in the DFE subspace:

$$T = \left\{t_j = m_{k_j} \mid k_j = \arg\min_{k:\ y_k = y_j} \left\|W^T x_j - m_k\right\|_2^2\right\}. \tag{13}$$

Now that the class labels of adaptation samples are known, we simply set the correspondence confidence $f_j = 1, \forall j$, since all the labeled samples are provided by the particular writer with high confidence.

Based on the above definitions of source and target point sets, we can learn a style transform matrix (STM) $A$ according to equation (10). On obtaining $A$, the feature transformation matrix in the DFE+LVQ classifier is adapted to

$$W_{\mathrm{new}} = W \cdot A, \tag{14}$$

**Algorithm 1** *u-STM*

**Input:** unlabeled dataset $\{x_j\}_{j=1}^{U}$
generic DFE+LVQ model $\{W, m_k\}$
hyper-parameter $\beta$
**repeat**
  Classify each sample to get the label (1)
  Set up the source and target point set (12)(13)
  Calculate the correspondence confidence (15)
  Learn a style transfer matrix $A$ (10)
  Update the transformation matrix (14)
**until** convergence or exceeding a pre-defined iteration number
**Return:** the adapted DFE+LVQ classifier

and the decision rule in equation (1) is used to classify new samples from this particular writer. This adaptation process is very fast, and we do not need a large number of labeled samples to cover all the classes, because $A$ is class-independent and the change of $W$ will spread its influence to all the classes. What's more, the adaptation model based on STM will not affect other un-adapted writers, because we can simply set the style transfer matrix as $A = I$ for all the other writers. We call this model "adaptation based on supervised style transfer matrix (s-STM)".

### 6.2. Unsupervised Adaptation

For unsupervised adaptation, we only need some unlabeled samples $\{x_j\}_{j=1}^{U}$ from a particular writer. We adopt a self-taught strategy (also known as self-training) to learn the style transfer matrix. We first deduce the label $y_j$ of each sample using the old classifier, and then update the transformation matrix in the DFE+LVQ model based on STM learning using the method for supervised adaptation (equations (12,13,14)). On the updated model, the unlabeled samples are re-classified. This process repeats until convergence (stability of labels) or when a pre-defined maximum number of iterations reaches. We call this model "adaptation based on unsupervised style transfer matrix (u-STM)".

A major difference from supervised adaptation is that we can not set the corresponding confidence $f_j = 1$ for unlabeled samples, because the labels assigned by the old classifier are not reliable. So, we set the confidence as the posterior class probabilities approximated by soft-max:

$$f_j = \frac{e^{-\tau d_{y_j}(x_j)}}{\sum_{c=1}^{C} e^{-\tau d_c(x_j)}}, \quad \tau = \frac{N}{\sum_{i=1}^{N} d_{y_i}(x_i)}, \qquad (15)$$

where $d_c(x_j) = \min_{k:\, y_k=c} \left\| W^T x_j - m_k \right\|_2^2$, and the scaling hyper-parameter $\tau$ is estimated as the reciprocal of the average within-class distance from the original training dataset. The complete algorithm for the u-STM model is listed in Algorithm 1.

## 7. Experiments

We used a new online handwritten Chinese character database CASIA-OLHWDB [17] to evaluate the performance of our adaptation models. Specifically, we used the datasets OLHWDB1.1 (isolated characters) and OLHWDB2.1 (handwritten texts), both containing samples of 300 writers (denotes as 1001-1300). Different writers have great handwriting style variations (Figure 2), therefor writer adaptation is necessary and effective.

(a) (b)

Figure 2. Different handwriting styles of two writers (a) and (b).

We consider a classification problem of 3,755 classes (level-1 set of GB2312-80). For each writer, we denote the isolated characters as $D_1$ and the text data as $D_2$. And for each writer, $D_1$ contains 3,755 character samples (one per class), and $D_2$ contains about 1,200 (variable depending on text templates) characters segmented from five pages of handwritten texts.

For representing a character sample, we use a benchmark feature extraction method [16]: 8-direction histogram feature extraction combined with pseudo 2D bi-moment normalization (P2DBMN). We also add the direction values of off-strokes (pen lifts) to real strokes with a weight of 0.5 [7]. The feature dimensionality is 512.

The writer-independent DFE+LVQ classifier was trained using the isolated character data $D_1$ of 240 writers (no.1001-1240), where we set the DFE subspace dimensionality as $d' = 150$ and use one prototype per class. We evaluate the basic classifier and adaptation models on the isolated character data $D_1$ and text data $D_2$ of the remaining 60 writers (no.1241-1300).

### 7.1. Supervised Adaptation on Isolated Characters

To evaluate the performance of supervised adaptation on isolated characters, we partitioned $D_1$ of each writer randomly into two equal subsets (Table 1): $D_1^a$ for adaptation

| | supervised | | unsupervised |
|---|---|---|---|
| | adaptation set | test set | test set |
| isolated | $D_1^a$ | $D_1^t$ | $D_1^t$ |
| text | $D_1$ | $D_2$ | $D_2$ |

Table 1. The partition of dataset for supervised and unsupervised adaptation, on the isolated characters and the text data.

| | isolated characters | | | | | text data | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Writer no. | DFE+LVQ | on-LVQ | s-STM | u-STM | reduction | DFE+LVQ | on-LVQ | s-STM | u-STM | reduction |
| 1241 | 5.27 | 4.58 | 3.94 | 3.89 | 26.19 | 10.59 | 6.71 | 7.68 | 5.83 | 44.95 |
| 1242 | 42.81 | 42.97 | 41.85 | 43.35 | -1.26 | 44.17 | 40.50 | 43.87 | 41.67 | 5.66 |
| 1244 | 3.90 | 3.53 | 3.26 | 3.10 | 20.51 | 18.52 | 16.39 | 15.63 | 13.19 | 28.78 |
| 1249 | 11.73 | 10.83 | 10.29 | 10.40 | 11.34 | 21.55 | 17.09 | 22.21 | 18.33 | 14.94 |
| 1253 | 2.29 | 2.35 | 1.65 | 1.44 | 37.12 | 7.91 | 6.20 | 6.74 | 4.19 | 47.03 |
| 1255 | 5.06 | 4.42 | 3.78 | 3.67 | 27.47 | 14.23 | 7.95 | 10.37 | 8.40 | 40.97 |
| 1256 | 31.78 | 28.43 | 30.00 | 30.65 | 3.56 | 47.05 | 43.60 | 45.54 | 46.21 | 1.79 |
| 1258 | 3.75 | 3.11 | 3.11 | 2.04 | 45.60 | 5.43 | 3.45 | 4.70 | 3.60 | 33.70 |
| 1259 | 12.42 | 10.76 | 10.92 | 9.80 | 21.10 | 29.55 | 19.44 | 22.39 | 19.28 | 34.75 |
| 1261 | 30.31 | 24.92 | 25.83 | 25.67 | 15.31 | 38.38 | 18.52 | 27.44 | 29.55 | 23.01 |
| 1274 | 4.83 | 4.94 | 4.24 | 4.35 | 9.94 | 2.62 | 2.10 | 2.10 | 1.20 | 54.20 |
| 1277 | 16.28 | 14.47 | 13.93 | 13.77 | 15.42 | 17.80 | 12.85 | 14.88 | 11.71 | 34.21 |
| 1278 | 6.05 | 5.46 | 5.52 | 5.25 | 13.22 | 8.81 | 6.71 | 6.71 | 5.29 | 39.95 |
| 1284 | 6.75 | 6.00 | 5.73 | 5.20 | 22.96 | 18.88 | 16.87 | 16.21 | 12.49 | 33.85 |
| 1285 | 10.23 | 8.90 | 8.58 | 9.06 | 11.44 | 21.86 | 17.41 | 18.28 | 16.54 | 24.34 |
| 1286 | 8.20 | 7.51 | 8.52 | 8.09 | 1.34 | 15.88 | 15.16 | 14.84 | 9.34 | 41.18 |
| 1289 | 27.92 | 23.01 | 23.97 | 23.87 | 14.51 | 31.78 | 18.95 | 24.42 | 24.85 | 21.81 |
| 1291 | 9.16 | 8.90 | 9.00 | 8.26 | 9.83 | 6.13 | 5.52 | 5.67 | 4.67 | 23.82 |
| 1295 | 6.88 | 6.45 | 6.29 | 5.81 | 15.55 | 8.93 | 7.07 | 8.04 | 5.73 | 35.83 |
| 1299 | 7.12 | 6.69 | 6.75 | 6.69 | 6.04 | 10.90 | 7.80 | 8.47 | 6.54 | 40.00 |
| average | 10.56 | 9.86 | 9.76 | 9.58 | 9.30 | 16.39 | 13.07 | 14.55 | 12.64 | 22.87 |

Table 2. Error rates (%) of DFE+LVQ classifier and adaptation models on isolated characters and text data.

and $D_1^t$ for testing. For each of the 60 writers, the on-LVQ and s-STM models were trained on $D_1^a$ and the performance was evaluated on $D_1^t$.

Table 2 [1] show the error rates of representative writers and average error rates over 60 writers. The 2-6th columns show the results of isolated character recognition, with the 2nd column for the basic writer-independent classifier (initial error rates), the 3-5th columns for adaptation by on-LVQ, s-STM and u-STM, and the 6th column for the error reduction percentage of u-STM compared to the initial rates. The results of unsupervised adaptation will be detailed in Section 7.3.

From the results in the 3th and 4th columns of Table 2, we can see that both on-LVQ and s-STM can reduce the initial error rate for most writers. The supervised adaptation mode s-STM performs a little better than the online learning model on-LVQ on isolated characters, but the difference between them is not significant.

For each writer, there was only one sample per class in $D_1$ and we randomly partitioned them into two equal subsets $\{D_1^a, D_1^t\}$, so the adaptation data and the test data were from completely different classes. In this situation, we can affirm that the improvement were resulted from the style information of the writers, which is independent of classes.

## 7.2. Supervised Adaptation on Text Data

The characters in handwritten texts are more cursive because people write more fluently in the context of natural language. Hence, the style change from isolated characters to texts is more significant, and we expect that writer adaptation of classifier can produce greater improvement for character recognition in texts.

In our experiments of supervised adaptation, we used for each of the 60 writers the data of $D_1$ for adaptation and the data of $D_2$ for testing (Table 1). In this case, the writer-specific data of $D_1$ was not partitioned, and there is one sample per class for adaptation. The adaptation data was used to train the on-LVQ and s-STM models.

The test results are shown in the 8th and 9th columns of Table 2. We can see that both on-LVQ and s-STM can reduce the initial error rates more significantly than the experiments in Section 7.1. This is because handwritten text data has greater change of style from isolated characters. And the performance of the adaptation model s-STM is inferior to the online learning model on-LVQ in this case. This can be attributed to the large number of labeled adaptation samples (available for all classes) for online learning, which is critically dependent on the training sample size. We can expect even larger improvement for online learning when the adaptation dataset is increased.

## 7.3. Unsupervised Adaptation

For unsupervised adaptation on isolated characters, the u-STM model was trained and evaluated on $D_1^t$ of each writer (Table 1), and the class labels of samples were not used in adaptation. The evaluation results thus can be fairly compared with that of supervised adaptation (section 7.1). In Table 2, we can see that the unsupervised adaptation model u-STM results in a lower error rate than the super-

[1] For lack of space, we only show the results of some representative writers here, and the average error rates are over all the 60 writers (no.1241-1300). The representative writers were selected to cover different interval of initial error rates.

vised adaptation models (on-LVQ, s-STM).

For unsupervised adaptation on text data, we directly adapted the u-STM model on the text data $D_2$ (without the label information) for each of the 60 writers and evaluated on the same text data. The 10th column of Table 2 show that unsupervised adaptation on text data yields significantly lower error rate (average 12.64%) than supervised adaptation models. This is because the u-STM model was adapted directly on the text data (much like transductive learning), while the supervised models (on-LVQ, s-STM) were adapted on isolated character data and evaluated on text data.

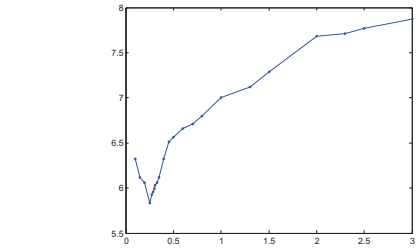
| | | Error reduction (%) | |
|---|---|---|---|
| Initial error rate | #writer | on-LVQ | u-STM |
| $[0, 10]$ | 17 | 23.71 | 32.65 |
| $[10, 15]$ | 14 | 23.59 | 30.25 |
| $[15, 20]$ | 12 | 19.34 | 27.90 |
| $[20, 25]$ | 8 | 15.76 | 17.83 |
| $[25, 30]$ | 5 | 14.99 | 16.77 |
| $> 30$ | 4 | 24.67 | 11.83 |

Table 3. Error reduction rates on different partitions of writers.

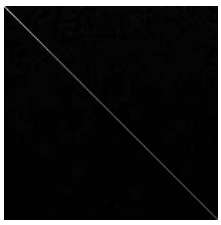
We also compared the influences of different initial error rates on the unsupervised adaptation model (u-STM) and the supervised online learning model (on-LVQ). We partition the 60 writers into different groups according to the initial error rates of DFE+LVQ classifier. Table 3 shows that for the writers with initial error rates not very high ($< 30\%$), the unsupervised model u-STM yields better performance than the supervised model on-LVQ. But for the writers with initial error rates over 30%, the supervised model on-LVQ performs better. The error rates of the basic DFE+LVQ, on-LVQ and u-STM models on the text data of 60 writers (no.1241-1300) are depicted in Figure 3, which shows again that the u-STM model is more suitable for "good" writers (those with lower initial error rates). Take for example the writer no.1274, the initial error rate 2.62% is already low, but is further reduced to 1.20% by u-STM. But for "bad" writers (those with higher initial error rates), on-LVQ performs better than u-STM (e.g. writer 1242, 1256). Low initial error rate is important for unsupervised adaptation because the adaptation process relies on the class labels assigned by the initial classifier.

### 7.4. On Model Parameter Selection

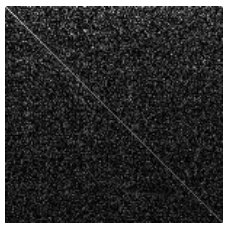
A key problem in the style transfer matrix based models (s-STM, u-STM) is how to set the hype-parameters $\beta$, which acts as a balance between transfer and non-transfer. We suggest to set $\beta$ via equation (11) and select $\tilde{\beta}$ from $[0, 3]$. We evaluated the s-STM and u-STM models on the training dataset (writer 1001-1240) and selected the $\tilde{\beta}$ with the best average performance. The selected $\tilde{\beta}$ was fixed in the experiments on writer 1241-1300.[2] Take writer no.1241 as a example, Figure 4(a) shows the effects of different $\tilde{\beta}$ values, where $\tilde{\beta} = 0.25$ favors the u-STM model best. At this point, the style transfer matrix (STM) is shown in Figure 4(b). We can see that the learned STM is close to the identity matrix, which helps avoid over-transfer. In Figure 4(c), the non-diagonal values are amplified to show that the style information is well coded in the STM.




(a)



(b)



(c)

Figure 4. For adaptation on text data of writer no.1241, (a) error rates after adaptation with respect to $\tilde{\beta}$; (b) the style transfer matrix with $\tilde{\beta} = 0.25$; (c) the style transfer matrix with non-diagonal values multiplied by 10.

## 8. Conclusion

In this paper, we propose a novel model to learn a style transfer matrix (STM) for writer adaptation. Via the STM, we can transfer the character data of different writers into a style-free space, where we can reduce the classification error rates significantly using a writer-independent classifier. We combine the STM model with the efficient DFE+LVQ classifier, in both supervised and unsupervised fashion. Experiments of writer adaptation on online handwritten Chinese character data demonstrate the superiority of the unsupervised adaptation model u-STM, which even outperforms the supervised adaptation model (s-STM) and the supervised online learning model (on-LVQ). The framework of STM learning can be also applied to many other types of classifiers for adaptation when the source and target point sets are appropriately defined. Our future work will extend the STM model to the Bayesian classification model and kernel classifiers.

[2] A better way is to choose $\tilde{\beta}$ dynamically for different writers.

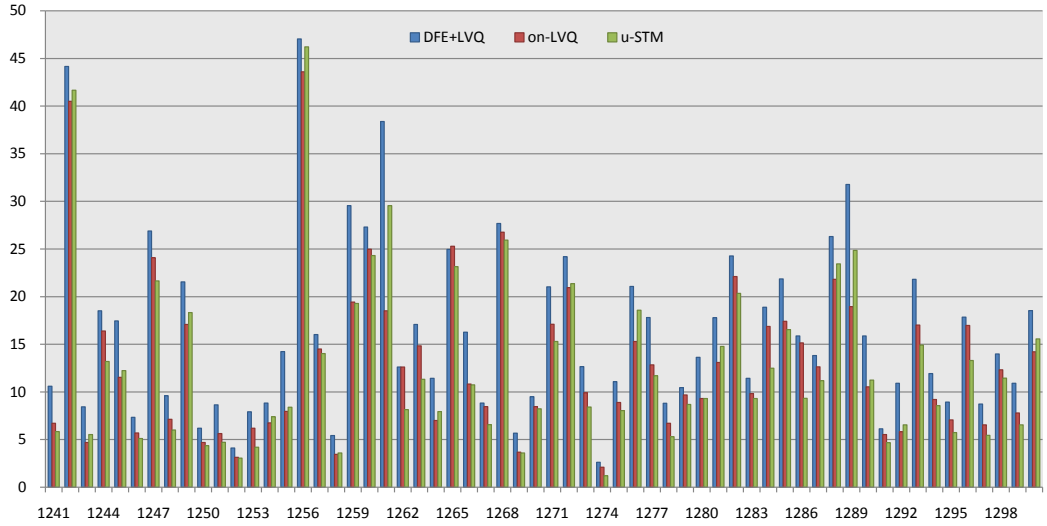



Figure 3. Error rates of different adaptation models on the text data of 60 writers.

## Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC) under grants no.60825301 and no.60933010.

## References

[1] A. Biem, S. Katagiri, B.-H. Juang, Pattern recognition using discriminative feature extraction, *IEEE Trans. Signal Processing*, 1997.

[2] A. Brakensiek, A. Kosmala, G. Rigoll, Comparing adaptation techinieuqs for on-line handwriting recognition, *ICDAR* 2001.

[3] C.M. Bishop, Pattern Recognition and Machine Learning, *Springer* 2006.

[4] G.R. Ball, S.N. Srihari, Semi-supervised learning for handwriting recognition, *ICDAR* 2009.

[5] S.D. Connel, A.K. Jain, Writer adaptation for online handwriting recognition, *IEEE Trans. PAMI*, 2002.

[6] R. Caruana, Multitask learning, *Machine Learning*, 1997.

[7] K. Ding, G. Deng, L. Jin, An investigation of imaginary stroke technique for cursive online handwriting Chinese character recognition, *ICDAR* 2009.

[8] V. Frinken, H. Bunke, Evaluating retraining rules for semi-supervised learning in neural network based cursive word recognition, *ICDAR* 2009.

[9] X.-B. Jin, C.-L. Liu, X. Hou, Regularized margin-based conditional log-likelihood loss for prototype learning, *Pattern Recognition*, 2010.

[10] L. Jin, K. Ding, Z. Huang, Incremental learning of LDA model for Chinese writer adaptation, *Neuralcomputing*, 2010.

[11] M.G. Kelly, D.J. Hand, N.M. Adams, The impact of changing populations on classifier performance, *KDD* 1999.

[12] W. Kienzle, K. Chellapilla, Personalized handwriting recognition via biased regularization, *ICML* 2006.

[13] C.J. Leggetter, P.C. Woodland, Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models, *Computer Speech and Language*, 1995.

[14] J.J. Laviola, R.C. Zeleznik, A practical approach for writer-dependent symbol recognition using a writer-independent symbol recognizer, *IEEE Trans. PAMI*, 2007.

[15] C.-L. Liu, R. Mine, M. Koga, Building compact classifier for large character set recognition using discriminative feature extraction, *ICDAR* 2005.

[16] C.-L. Liu, X.-D. Zhou, Online Japanese character recognition using trajectory-based normalization and direction feature extraction, *IWFHR* 2006.

[17] C.-L. Liu, F. Yin, D.-H. Wang, Q.-F. Wang, CASIA online and offline Chinese handwriting databases, submitted to *ICDAR* 2011.

[18] N. Matic, I. Guyon, J. Denker, V. Vapnik, Writer adaptation for on-line handwritten character recognition, *ICDAR* 1993.

[19] J.C. Platt, N.P. Matic, A constructive RBF netwrok for writer adaptation, *NIPS* 1997.

[20] S.J. Pan, Q. Yang, A survey on transfer learning, *IEEE Trans. KDE*, 2010.

[21] A. Sato, K. Yamada, Generalized learning vector quantization, *NIPS* 1995.

[22] A. Sato, A new learning formulation for kernel classifier design, *ICPR* 2010.

[23] M. Szummer, C.M. Bishop, Discriminative writer adaptation, *IWFHR* 2006.

[24] A. Vinciarelli, S. Bengio, Writer adaptation techniques in HMM based off-line cursive script recognition, *Pattern Recognition Letters*, 2002.

[25] S. Veeramachameni, G. Nagy, Analytical results on style-constrained Bayesian classification of pattern fields, *IEEE Trans. PAMI*, 2007.

[26] V. Vuori, T. Korkeakoulu, Adaptive methods for online recognition of isolated handwritten characters, *Helsinki University of Technology*, 2002.

[27] X.-Y. Zhang, K. Huang, C.-L. Liu, Pattern field classification with style normalized transformation, to appear in *IJCAI* 2011.