

Incremental Adaptive Learning Vector Quantization for Character Recognition with Continuous Style Adaptation

Yuan-Yuan Shen¹ · Cheng-Lin Liu¹

Received: 31 March 2017 / Accepted: 18 July 2017 / Published online: 4 August 2017
© Springer Science+Business Media, LLC 2017

Abstract Incremental learning enables continuous model adaptation based on a constantly arriving data stream. It is a way relevant to human cognitive system, which learns to predict objects in a changing world. Incremental learning for character recognition is a typical scenario that characters appear sequentially and the font/writing style changes irregularly. In the paper, we investigate how to classify characters incrementally (i.e., input patterns appear once at a time). A reasonable assumption is that adjacent characters from the same font or the same writer share the same style in a short period while style variation occurs in characters printed by different fonts or written by different persons during a long period. The challenging issue here is how to take advantage of the local style consistency and adapt to the continuous style variation as well incrementally. For this purpose, we propose a continuous incremental adaptive learning vector quantization (CIALVQ) method, which incrementally learns a self-adaptive style transfer matrix for mapping input patterns from style-conscious space onto style-free space. After style transformation, this problem is casted into a common character recognition task and an incremental learning vector quantization (ILVQ) classifier is used. In this framework, we consider two learning

modes: supervised incremental learning and active incremental learning. In the latter mode, samples receiving low confidence from the classifier are requested class labels. We evaluated the classification performance of CIALVQ in two scenarios, interleaved test-then-train and style-specific classification on NIST hand-printed data sets. The results show that local style consistency improves the accuracies of both two test scenarios, and for both supervised and active incremental learning modes.

Keywords Continuous incremental adaptive learning vector quantization · Style transfer mapping · Local style consistency · Active learning

Abbreviation

NP: Nearest prototype

STM: Style transfer mapping

LVQ: Learning vector quantization

ILVQ: Incremental learning vector quantization

IALVQ: Incremental adaptive learning vector quantization

CIALVQ: Continuous incremental adaptive learning vector quantization

Active CIALVQ: Active continuous incremental adaptive learning vector quantization

Introduction

In the community of machine learning, there have been many research efforts in human-like learning via studying the cognitive processes of human beings [1, 2]. Humans usually learn much better when the examples are organized in a meaningful order. Curriculum learning [3–6], which

✉ Cheng-Lin Liu
liucl@nlpr.ia.ac.cn

Yuan-Yuan Shen
shenyuanyuan2015@ia.ac.cn

¹ National Laboratory of Pattern Recognition,
Institute of Automation of Chinese Academy of Sciences,
University of Chinese Academy of Sciences, Beijing 100190,
People's Republic of China

iteratively selects the simplest examples in each step, is proposed based on such training strategies. The cognitive process of human beings is usually increasing gradually: while sensing the environment, pay attention to the objects in the sensed data, find new concepts (classes) or update the representation of memorized concepts continuously. Inspired by this, many incremental learning algorithms [7–10], which are relevant to human-like learning in the sense that the classification model is self-adaptive continuously based on data sequence, have been researched extensively.

In this paper, we combine incremental learning with a practical pattern classification task and introduce an incremental character recognition [11, 12] method, taking into account the sequential nature of character samples appearance and the periodical consistency of font or writing style (characters in a period are usually in the same style). The main difficulties of this task are twofold. The first is to model the style mathematically to enable continuous adaptation, and the second is to build an on-the-fly classifier which balances between plasticity and stability [13].

Character patterns are presented as groups frequently. The patterns from the same group (e.g., characters in a paragraph or a text line are usually printed in the same font or written by the same person) usually have the consistent style. In the long period, the style changes from time to time when moving from group to group. Classification model is supposed to adapt to this kind of style variation simultaneously. Inspired by the method [14] in writer adaptation, we learn a kind of linear projection (style transfer matrix) to formulate style model. Meanwhile, we bring in the forget mechanism in which old styles are discarded or weakened. In this way, the style model is always calculated in a window and the window size depends on the speed of the decay factor. Thus, the style model is always estimated from the latest several patterns without assuming known style transition point, and so, it is highly adaptive to style change and local style consistency.

“Stability-plasticity dilemma” is a fundamental problem in incremental learning. Plasticity refers to the ability of incorporating new acquired knowledge into model, while stability considers the capability of preserving the learned knowledge. For the practical application of incremental character recognition, we hope that the incremental learning can not only achieve a good instant performance by means of incorporating new knowledge from sequence of patterns, but also memorize all the styles in the previously learned patterns to enhance the generalization performance. To cope with the situation stated above, we proposed continuous incremental adaptive learning vector quantization (CIALVQ). Learning vector quantization (LVQ) [15–17] is a kind of prototype-based classifier, which is akin to human learning in that prototypes (templates) are memorized in the brain to recognize new patterns. CIALVQ simultaneously

learns two kinds of prototypes: one is style-free prototypes, which are independent of specific style and good for plasticity; the other is style-conscious prototypes, which memorize all the learned styles and benefit the stability.

Humans perceive, cognize, and comprehend the world on the fly, then respond to the surrounding environment in real time. Thus, learning occasionally interacts with a teacher (e.g., inquires a teacher for labels when facing unknown characters). Inspired by this, we also introduce the active mode of CIALVQ. In this case, we attach labels to the training samples only when the confidence assigned by the classifier is low. Our experimental results show that in the case of active incremental learning, utilizing local style consistency is also beneficial to classification.

The method proposed in this paper is capable of maintaining style consistency in a short period and adapting to the continuous style variation during a long period as well. The major contributions of this paper are threefold:

1. We propose a continuous self-adaptive character recognition method which does not require prior tagging of style-transitions in the data stream.
2. The intra-class distance constraint is explicitly adopted in CIALVQ to learn a more compact representation in style-free space.
3. An active learning mode is explored in our paper.

CIALVQ is an extended version of the IALVQ in our previous conference paper [18]. Similar to the IALVQ, CIALVQ models style with linear transformation based on local style consistency. The framework of our incremental learning algorithm is illustrated in Fig. 1. In addition to considering local style consistency, CIALVQ also minimizes intra-class distance in style-free space so that the style-free patterns from the same class are closer. The distinction between IALVQ and CIALVQ is illustrated in Fig. 2, where CIALVQ is shown to learn more compact intra-class prototypes in the style-free space.

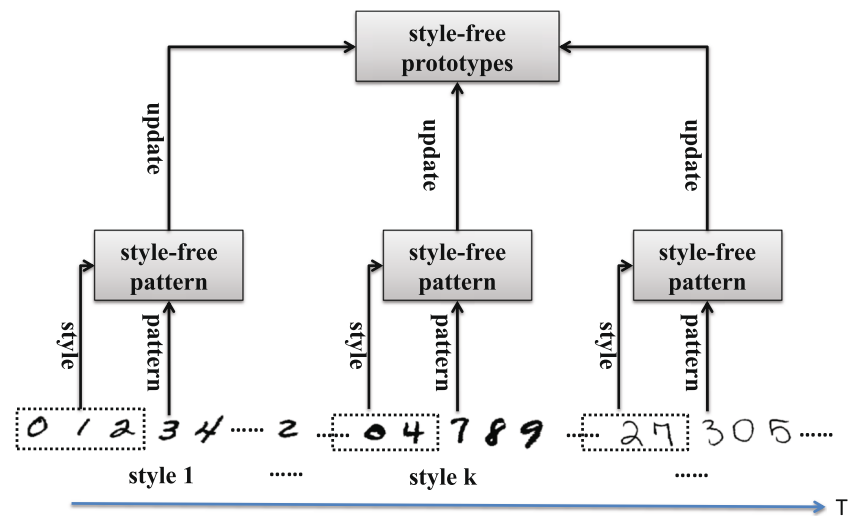
The rest of the paper is organized as follows. We review the related work in “[Related Work](#)” section. The detailed CIALVQ method for both supervised and active mode is present in “[Proposed Methods](#)” section. Experimental results are reported and discussed in “[Experiments and Results](#)” section, and we finally conclude the paper in “[Discussion](#)” section.

Related Work

Incremental Learning

Incremental learning is a well-studied branch of machine learning. Many effective methods have been developed for this. Unlike the process of off-line learning, which assumes that training data are prepared in advance, incremental

Fig. 1 The framework of our algorithm



learning is to incrementally learn prediction model on a stream of patterns. Thus, the essential problem in incremental learning is how to update the model on the latest data. Depending on the discrepancy among different update modes, some representative works are summarized as follows.

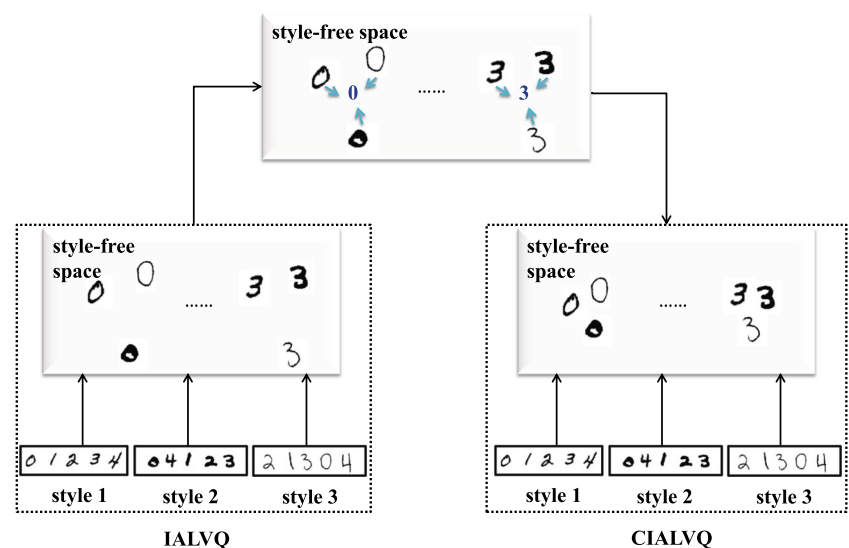
One of the best known incremental learning approach is the Perceptron algorithm [19] proposed in 1950s. Perceptron updates the weights of single-layer neural network when the incoming pattern is misclassified. Online ensemble algorithms, including online bagging [20], online boosting [21], and online random forest [22], selectively update multiple classifiers in ensemble models on an incoming pattern. In incremental prototype-based classifier [23, 24], two nearest prototypes from the genuine class and the rival class of input pattern are adjusted. Using covariance matrix as confidence information of different dimensions, a few second-order incremental algorithms [25, 26] have been

proposed. A self-adaptive clustering algorithm for incremental unsupervised learning is formulated in [9]. Imitating the topographic organization of the visual cortex, a biologically inspired neural architecture for incremental learning is presented in [10]. In [27], three guidelines of incremental learning for large-scale visual recognition are concluded. A comprehensive survey about adaptation on data stream can be found in [28].

Style Consistent Learning

Character recognition has been widely studied in visual information processing and pattern classification research. In the task, characters appearing together usually come from the same source with consistent style. Such local style consistency is common in document images where the characters in a paragraph or text line are printed in the same font or written by the same person. Previous works have shown that

Fig. 2 The distinction between IALVQ and CIALVQ



exploiting style consistency effectively enhance the performance of character recognition. Some related works are below.

Sarkar and Nagy model style shared within a group by a class-style conditional mixture of Gaussians. In [12], they propose a style constrained classifier which processes entire fields of characters rendered in a consistent style. Veeramachaneni and Nagy [29, 30] propose an adaptive Gaussian quadratic discriminant field classifier which initializes model parameters from the training set and adapts parameters on the test set. In [31], Huang et al. learn a writer-specific (or style-specific) LDA transformation matrix with the new labeled data in incremental handwriting recognition. By learning a style transfer mapping matrix which projects patterns from writer-specific space onto writer-independent space, a style adaptation method is posed by Zhang and Liu [14], which can be applied to various classifier models.

Relationship with Other Incremental Character Recognition

To the best of our knowledge, the closest methods to our current work are those in [31, 32]. However, there are three main differences between other incremental handwriting recognition methods and our method:

1. The data employed for capturing the style consistency. Our method takes the style consistency both in training data and test data, while the existing methods generally exploit the consistency only in test data.
2. The pre-trained classifier. Existing incremental handwriting recognition methods commonly adapt a pre-trained classifier with a large number of samples, while our method initializes model with only a small number of samples, and then updates model incrementally.
3. The evaluation protocol. While many previous methods adapt and evaluate on a batch of test data, our method gives the instant performance of the model in interleaved test-then-train classification, which reflects the continuous self-adaptive capacity in recognition.

Proposed Methods

In this section, we first briefly introduce the LVQ/ILVQ method, then present the CIALVQ method and extend to the active mode. Last, two paradigms for evaluating the proposed methods are outlined.

Brief Description of LVQ/ILVQ

For M -class classification, prototype learning is to design a set of prototype vectors m_{ij} ($i = 1, 2, \dots, M, j = 1, \dots, n_i$, n_i is the number of prototypes in class i), usually

by minimizing the empirical loss on a training set. An input pattern $x \in \mathbb{R}^d$ is classified to the class of the nearest prototype:

$$k = \arg \min_{i=1}^M \min_{j=1}^{n_i} \|x - m_{ij}\|_2^2 = G(x, m), \quad (1)$$

There are many variations of LVQ algorithm [15–17]. In this paper, we use the one of LOG-likelihood of Margin (LOGM) [17].

Specifically, given that m_1 and m_2 are two nearest prototypes to pattern x from the positive class and the rival class respectively, the posterior probability of x belonging to genuine class i (i.e., the probability of correct classification) can be approximated by the sigmoid function σ :

$$\begin{aligned} d(x) &= \|x - m_2\|^2 - \|x - m_1\|^2, \\ P(C_i|x) &= \sigma(\xi d(x)), \end{aligned} \quad (2)$$

where ξ ($\xi > 0$) is a constant for tuning the smoothness of sigmoid function and the conditional log-likelihood loss of pattern x is $\phi(x) = -\log P(C_i|x)$.

As new pattern x is arriving, LVQ updates the two prototypes m_1 and m_2 by stochastic gradient descent [33]:

$$\begin{aligned} m_1 &= m_1 - \eta \frac{\partial \phi(x)}{\partial m_1}, \\ m_2 &= m_2 - \eta \frac{\partial \phi(x)}{\partial m_2}. \end{aligned} \quad (3)$$

where η is the learning rate.

ILVQ is the incremental one-pass version of LVQ, i.e., whenever an input pattern x arrives, the prototypes are updated according to Eq. 3.

CIALVQ

In our proposed method, we learn two types of prototypes for each class: one is style-conscious prototypes m_{ij} ($i = 1, \dots, M, j = 1, 2, \dots, L_1$) which are treated completely the same as conventional ILVQ; the other is style-free prototypes \hat{m}_{ij} ($i = 1, \dots, M, j = 1, 2, \dots, L_2$), which are irrelevant to specific style. However, the two types of prototypes are learned independently.

How to learn the style-free prototypes from the online stream of data is the key difficulty in our method, which is detailed in the following.

Assume that x and A are the observation and the style transfer matrix at time t , respectively, we can obtain the style-free pattern Ax after projecting the pattern onto style-free space. Here, Ax is expected to be close to the nearest style-free prototype \hat{m} with the same label as pattern x . Simultaneously, it should be as close as possible to the style-free patterns seen before. The former goal can easily be realized by minimizing the difference between Ax and the target prototype \hat{m} . For the latter one, intuitively, all the past style-free patterns should be saved and then compared with the current observation. However, this is not feasible

in incremental learning due to the cost of memory and computation. Instead, we minimize the distance between the observation and the calculated class mean from the past style-free patterns, which is easily proved to be equal to optimize the distance among the style-free patterns from the same class. So, the cost function of learning the style transfer matrix from the current observation x is given by:

$$\Delta F(t) = \|Ax - \hat{m}\|_2^2 + \|Ax - \hat{\mu}\|_2^2. \quad (4)$$

where $\hat{\mu}$ is the accumulated mean of the style-free pattern with the same class label as the observation x .

Obviously, estimating style transfer matrix only from the current observation is unstable. Considering the existence of the local style consistency in character recognition, we estimate the style transfer matrix from the accumulated patterns of the past. Meanwhile, in order to reduce the influence caused by the past patterns and adapt to the current pattern, we decay the influence at every moment. Thus, the style transfer matrix is computed by minimizing the cost:

$$F(t) = DecayWeight * F(t-1) + \Delta F(t), \quad (5)$$

where $F(0) = 0$ and *DecayWeight* is the decay parameter. The weights of samples received from the time stream for calculating the style transfer matrix of the current observation are visualized in Fig. 3.

Furthermore, we also add a regularization term to avoid overtransfer as in [14]. The final objective function is defined as follows:

$$H(t) = F(t) + \beta \|A - I\|_F^2, \quad (6)$$

where the hyperparameter β controls the trade-off between style transfer and nontransfer. I is the identity matrix.

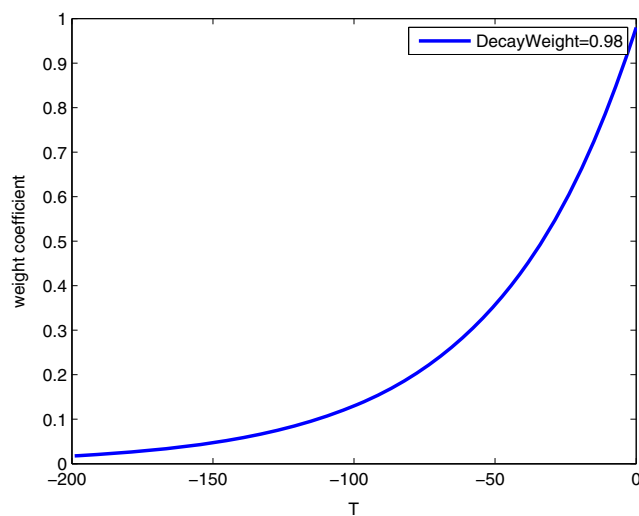


Fig. 3 Weights of patterns with *DecayWeight* = 0.98 for calculating the style transfer matrix of the current observation

Let

$$S(t) = DecayWeight * S(t-1) + \Delta S, \\ s.t. \Delta S = 2 * xx^T, S(0) = 0, \quad (7)$$

$$T(t) = DecayWeight * T(t-1) + \Delta T, \\ s.t. \Delta T = \hat{m}x^T + \hat{\mu}x^T, T(0) = 0, \quad (8)$$

then the computation of A has a closed-form solution:

$$A = QP^{-1}, \\ Q = T(t) + \beta I, \\ P = S(t) + \beta I. \quad (9)$$

Please refer to [14] for more details.

With the mapping matrix A , we can map pattern x onto a style-free space by:

$$\hat{x} = Ax. \quad (10)$$

And then, we update the style-free prototypes similar to Eq. 3:

$$\hat{m}_1 = \hat{m}_1 - \eta \frac{\partial \phi(\hat{x})}{\partial \hat{m}_1}, \\ \hat{m}_2 = \hat{m}_2 - \eta \frac{\partial \phi(\hat{x})}{\partial \hat{m}_2}. \quad (11)$$

We summarize the process of learning style-free prototypes in Algorithm 1.

Algorithm 1 Learning of style-free prototypes

Input: Patterns X , initial Prototypes \hat{m} ,

Output: new Prototypes \hat{m}

1: Initial style transfer matrix $A = I$

2: **while** Receive new pattern x **do**

3: Compute style-free pattern by Equation (10)

4: Update style-free prototypes \hat{m} by Equation (11)

5: Compute new style transfer matrix A by Equation (9)

6: **end while**

7: **return** \hat{m}

Active CIALVQ

In character recognition, active incremental learning is very helpful. By partial interaction between the learner and the environment, we label for patterns that are assigned low confidence by the classifier. Due to the important effect of patterns that are assigned low confidence, active learning can boost the performance of classifier by requiring labels from a small number of patterns.

In active incremental learning, how to evaluate the confidence for coming pattern is the key to the problem. We use a simple but effective strategy that the confidence f for pattern x is calculated by equation (2) (posterior probability). Because the genuine class is unknown, we compute two nearest prototypes from the top two classes instead of the positive class and the rival class. Actually, the closer the

sample to the classification boundary, the lower confidence is assigned in our method. This is similar to [34] which selects a query on the borderline of the actual classification. Only when the confidence f is smaller than a predefined threshold p , the input pattern requests a label. The larger the predefined threshold p is set, the more labels are requested. The active incremental adaptive learning vector quantization is summarized in Algorithm 2.

Algorithm 2 Active CIALVQ

Input: Patterns X , initial Prototypes \hat{m} , predefined Threshold p
Output: new Prototypes \hat{m}
 1: Initial style transfer matrix $A = I$
 2: **while** Receive new pattern x **do**
 3: Compute confidence f by Equation (2)
 4: Decide whether to query the label ($Z = 1$) or not ($Z = 0$)
 5: **if** $Z = 1$
 6: Query label y
 7: Compute style-free pattern \hat{x} by Equation (10)
 8: Update style-free prototypes \hat{m} using style-free pattern \hat{x} by Equation (11)
 9: Update style-conscious prototypes m using pattern x by Equation (3)
 10: **end if**
 11: Compute new style transfer matrix A by Equation (9)
 12: **end while**
 13: **return** \hat{m}

Evaluation

We evaluate our proposed algorithm in two scenarios: interleaved test-then-train and style-specific classification.

Interleaved Test-Then-Train Each pattern is used to test the classifier before it is used for training, so the classifier is always being tested on patterns it has not seen before. This evaluation method can make maximum use of the available data.

Style-Specific Classification After training an incremental classifier, we test the generalization performance of the classifier. During test time, a batch of b patterns with the same style are given once. The process of testing is reported in Algorithm 3. Note that we use style-conscious prototypes for first-round classification to obtain the initial labels of patterns. Then in style adaptive classification, style-free prototypes are used because the patterns are mapped toward style-free space.

Algorithm 3 Style-specific classification

Input: Patterns X , Style-conscious Prototypes m , Style-free Prototypes \hat{m} Style-specific unlabeled data $\{x_k\}_{k=1}^b$
Output: Predicted labels $\hat{y}_k, k = 1 \dots b$
 1: Initial style transfer matrix $A = I$
 2: **for** $k=1:b$ **do**
 3: Compute label $y_k = G(x_k, m)$ using Equation (1)
 4: Find nearest style-free prototype $\hat{m}_{y_k d_k}$ ($d_k = \arg \min_{j=1}^{L_2} \|x_k - \hat{m}_{y_k j}\|_2^2$)
 5: **end for**
 6: **for** iter = 1:iterNum **do**
 7: Learn style transfer matrix A using equation (9)
 8: **for** $k=1:b$ **do**
 9: Predict label $\hat{y}_k = G(Ax_k, \hat{m})$ using Equation (1)
 10: Find nearest style-free prototype $\hat{m}_{y_k d_k}$ ($d_k = \arg \min_{j=1}^{L_2} \|x_k - \hat{m}_{y_k j}\|_2^2$)
 11: **end for**
 12: **end for**

Experiments and Results

We evaluate the performance of the proposed CIALVQ method on NIST handwritten digit data. Two learning modes are considered:

- (i) supervised incremental learning;
- (ii) active incremental learning.

Database

To test our proposed method on realistic data, we experiment with the datasets SD3 and SD7, which are contained in the NIST Special Database SD19 [35]. The datasets contain patterns of handwritten numerals labeled by writer. From SD3, we use samples of 400 writers for training and 399 writers for testing, and from SD7, samples of 100 writers for training and 100 writers for testing. The patterns of each writer are assumed to have the same writing style. The statistics of patterns for SD3 and SD7 are listed in Table 1. The patterns from some writers are shown in Fig. 4. Our choice of data is similar to that of [29] for adaptive classification.

Table 1 Handwritten numeral datasets

Dataset	Writers	Number of samples
SD3-Train	No.0-No.399 (400)	42,969
SD7-Train	No.2100-No.2199 (100)	11,585
SD3-Test	No.400-No.799 (399)	42,821
SD7-Test	No.2200-No.2299 (100)	11,660

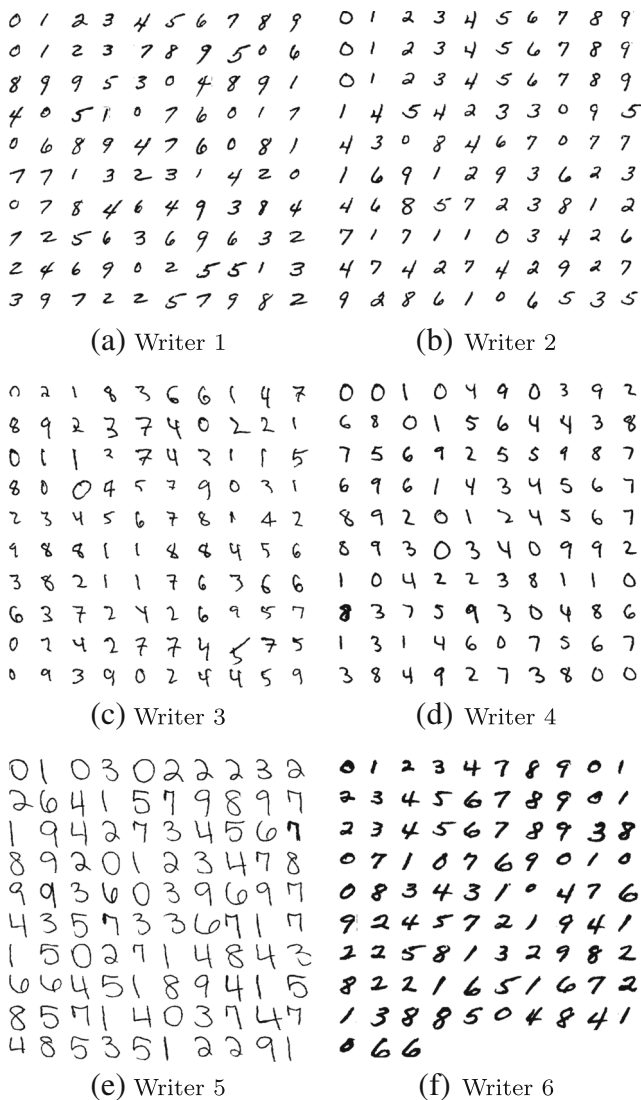


Fig. 4 Samples of handwritten digits from six different writers

Experimental Setting

All these patterns are arranged by writer order; however, all writers and all patterns within one writer are permuted randomly. In the experiments, we choose all patterns of each writer or randomly choose a small portion of patterns from each writer but the number of patterns (5 patterns, 10 patterns, 15 patterns, 20 patterns) for each writer is equal. In order to yield stable average results, each experiment runs multiple times. Experiments are repeated 10 times when all patterns from each writer are used or otherwise 20 times. We extract 100 blurred directional (chain-code) features from each pattern [36]. A small number of patterns are used to construct the initial prototypes. After getting initial classifier, new coming patterns are used for both prototypes updating and estimating style transfer matrix. The

classification performance is evaluated using interleaved test-then-train and style-specific classification.

For all the datasets, we use very small portion of patterns (200 patterns) for initialization. The trade-off parameter β in style adaptation is set as in [14]: $\beta = \hat{\beta} \frac{1}{d} \text{Tr}(\sum_{i=1}^n x_i x_i^T)$, where n is the total number of patterns that have been seen in model and $\hat{\beta}$ is set as 3 in our experiments. The smoothing parameter ξ is initialized as $2/\text{cov}$ like [17], where cov is the average covariance estimated from training data. The initial rate of gradient descent is set as 1 and the learning rate of the n th pattern is calculated by adagrad algorithm [37]. The *DecayWeight* is set as 0.98.

Results

Interleaved Test-then-Train Classification

Interleaved test-then-train is the most frequently-used method in the evaluation of incremental learning. In this mode, the data used for training and testing are completely the same. By testing pattern before being used for training, we can make maximum use of patterns. The results assessed by this method give some quantitative insight into the instant performance of model.

Table 2 and Fig. 5 show the results of ILVQ, IALVQ, and CIALVQ in supervised interleaved test-then-train classification. Different number of patterns for each writer are used to evaluate the influence of the length of the patterns. For fair comparison, we use three prototypes for every class in all three methods. Best performance of the compared methods are given in italics. We compare the models in respect of test error rate (Table 2) and error reduction rate (Fig. 5) (calculated as in [14]):

$$\text{Error reduction rate} = \frac{\text{Error}_{\text{initial}} - \text{Error}_{\text{adapted}}}{\text{Error}_{\text{initial}}}. \quad (12)$$

In this experiment, the error rate of the ILVQ method is computed as $\text{Error}_{\text{initial}}$ and the error rate of the IALVQ/CIALVQ method as $\text{Error}_{\text{adapted}}$. The higher error reduction rate is better.

To evaluate the discriminative performance of our method in active interleaved test-then-train classification, we run experiments in sd3-train, sd3-test, sd7-train, sd7-test. We vary the confidence threshold to acquire different number of request labels. The experiment results are given in Fig. 6. Each graph shows the error rate score (on the y-axis) achieved over the entire online dataset by requesting a given number of patterns (on the x-axis). We still experiment on two cases, including that all patterns are used and only small portion patterns for each writer are randomly chosen. In graph (a) and graph (b), the results for all patterns from sd7-train and sd7-test are given respectively. In graph (c)

Table 2 Error rates on four datasets using different number of patterns for each writer in interleaved test-then-train

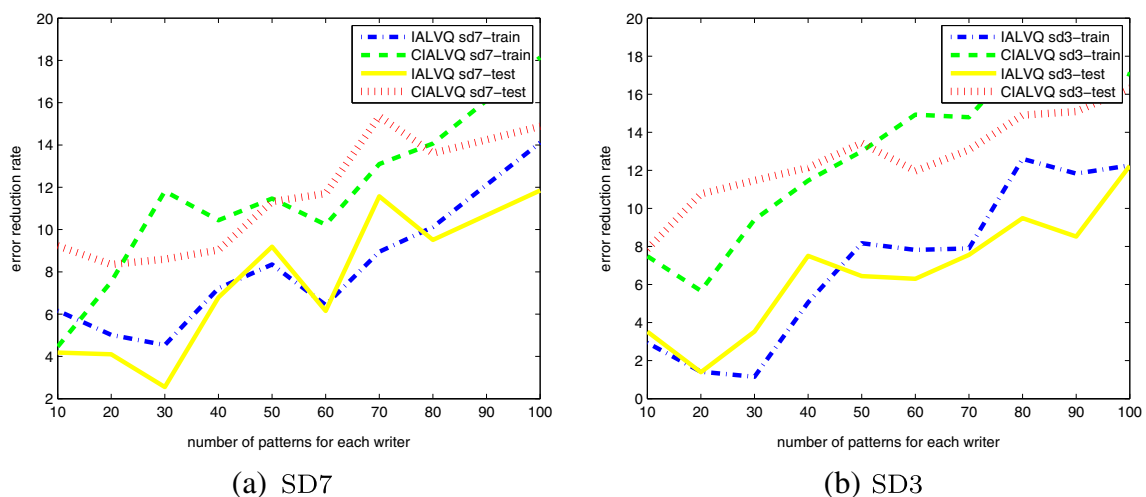
Dataset	5 patterns		10 patterns		15 patterns		20 patterns		All patterns	
SD3-Train	ILVQ	4.87%	ILVQ	3.74%	ILVQ	3.39%	ILVQ	2.82%	ILVQ	1.56%
	IALVQ	4.80%	IALVQ	3.63%	IALVQ	3.26%	IALVQ	2.78%	IALVQ	1.34%
	CIALVQ	4.61%	CIALVQ	3.46%	CIALVQ	3.07%	CIALVQ	2.66%	CIALVQ	1.29%
SD7-Train	ILVQ	11.93%	ILVQ	9.42%	ILVQ	8.22%	ILVQ	7.17%	ILVQ	3.97%
	IALVQ	11.84%	IALVQ	8.84%	IALVQ	7.86%	IALVQ	6.81%	IALVQ	3.41%
	CIALVQ	11.11%	CIALVQ	9.00%	CIALVQ	7.56%	CIALVQ	6.63%	CIALVQ	3.25%
SD3-Test	ILVQ	5.08%	ILVQ	3.69%	ILVQ	3.11%	ILVQ	2.88%	ILVQ	1.47%
	IALVQ	4.97%	IALVQ	3.56%	IALVQ	3.03%	IALVQ	2.84%	IALVQ	1.29%
	CIALVQ	4.74%	CIALVQ	3.40%	CIALVQ	2.86%	CIALVQ	2.57%	CIALVQ	1.23%
SD7-Test	ILVQ	11.13%	ILVQ	9.32%	ILVQ	7.63%	ILVQ	6.83%	ILVQ	3.63%
	IALVQ	11.03%	IALVQ	8.93%	IALVQ	7.29%	IALVQ	6.55%	IALVQ	3.20%
	CIALVQ	10.93%	CIALVQ	8.46%	CIALVQ	7.14%	CIALVQ	6.26%	CIALVQ	3.09%

and graph (d), we randomly choose five samples of each writer from sd3-train and sd3-test.

Style-Specific Classification

Different from the interleaved test-then-train classification, style-specific classification exploits the additional datasets for testing. In our experiments, A/B represents that A is adopted in training stage and B is adopted in testing stage. In testing stage, NP refers to nearest prototype classifier without considering style consistency, while STM utilizes style consistency of writer-specific data in [14]. For fair comparison, we use the best prototypes parameters respectively, that is, five prototypes are used in ILVQ and five style-conscious prototypes and two style-free prototypes are used in IALVQ and CIALVQ.

In the supervised incremental learning setting, we implement our method with CIALVQ for training and STM strategy for testing. We compare seven different methods when all patterns of each writer are utilized. The detailed information of compared methods are indicated in Table 3. Table 4 gives the corresponding misclassification rates. On the other hand, if the patterns from each writer are only a few, the style evaluated from the few patterns is extremely inaccurate. Hence, we evaluate our method by another mode in this case. Specifically, we continue to learn the style transfer matrix based on the result of training stage. After projecting patterns from style-conscious space onto style-free space with the style transfer matrix, labels are predicted in style-free space for patterns. Accordingly, the compared methods includes LVQ(1)/NP, LVQ(30)/NP, ILVQ/NP, and IALVQ/STM. Table 5 gives the corresponding results. Best

**Fig. 5** The error reduction rate for interleaved test-then-train classification in different datasets

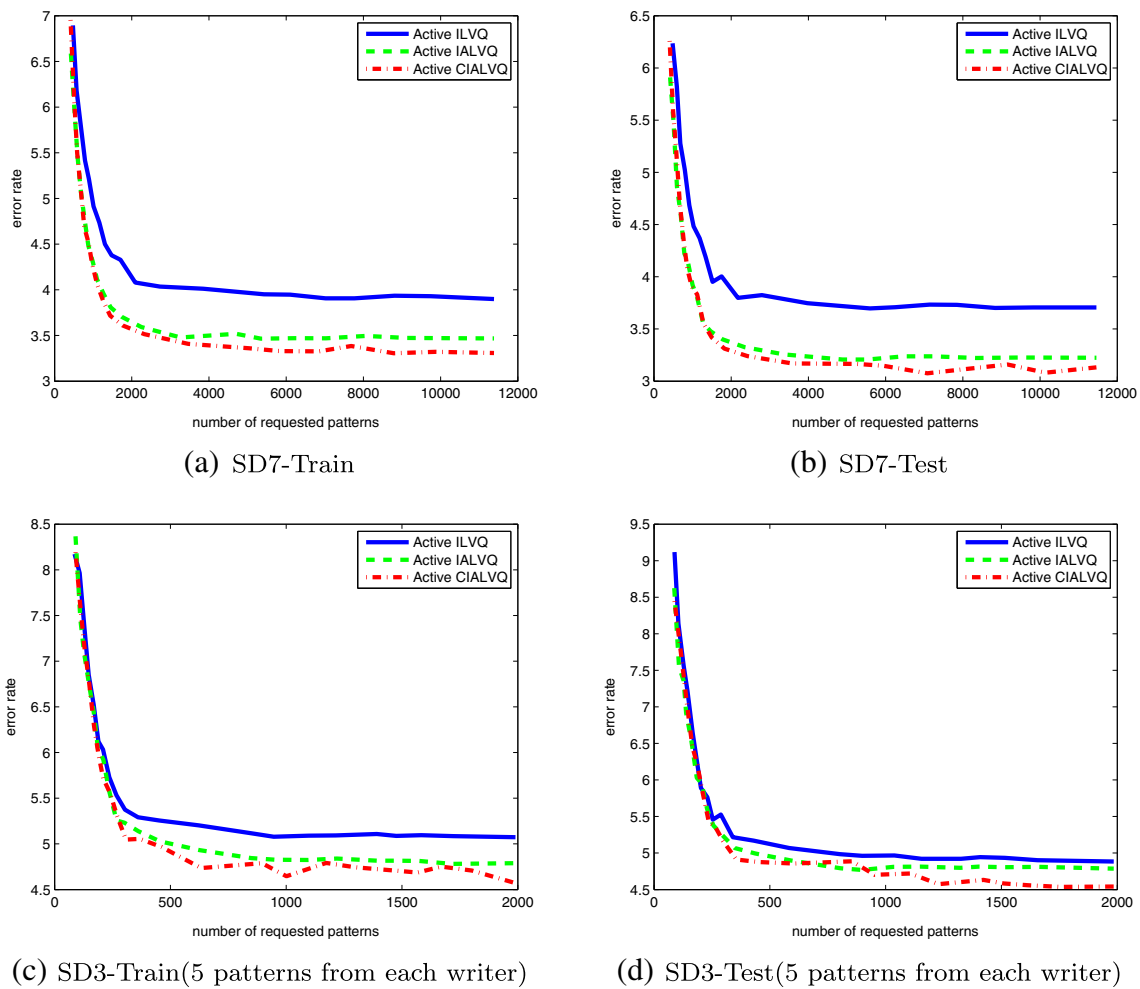


Fig. 6 Active interleaved test-then-train classification result in different dataset

performance in offline mode and online mode are given in *italics* respectively.

Figure 7 shows the curve of the active classification generalization error rates as different number of labeled samples are requested. Similarly, graph (a) and graph (b) show the results for all patterns from sd7-train as training set and all patterns from sd7-test or from sd3-test as test set, respectively. In graph (c) and graph (d), we randomly choose five samples of each writer from sd3-train as training

set and randomly choose five patterns from sd7-test or from sd3-test as test set.

Discussion

CIALVQ is an incremental character recognition method which takes advantage of the local style consistency and meanwhile continuously adapts to the style variation. For

Table 3 The detail information of the compared methods

Methods	Training method	Testing method	Iteration time
LVQ(1)/NP	Offline LVQ	Nearest prototypes	1
LVQ(30)/NP	Offline LVQ	Nearest prototypes	30
LVQ(1)/STM	Offline LVQ	Style transfer mapping	1
LVQ(30)/STM	Offline LVQ	Style transfer mapping	30
ILVQ/NP	Incremental LVQ	Nearest prototypes	1
ILVQ/STM	Incremental LVQ	Style transfer mapping	1
IALVQ/STM	Incremental adaptive LVQ	Style transfer mapping	1

Table 4 Error rates in style-specific classification

Dataset	LVQ(1)/NP	LVQ(30)/NP	LVQ(1)/STM	LVQ(30)/STM	ILVQ/NN	ILVQ/STM	IALVQ/STM	CIALVQ/STM
SD3-Train								
SD3-Test	1.21%	0.82%	1.09%	0.77%	1.47%	1.12%	1.01%	1.01%
SD3-Train								
SD7-Test	5.23%	3.92%	4.84%	3.36%	5.52%	4.42%	4.33%	4.26%
SD7-Train								
SD3-Test	2.36%	1.86%	1.75%	1.33%	3.31%	2.45%	2.05%	2.05%
SD7-Train								
SD7-Test	3.05%	2.40%	2.58%	2.01%	3.73%	2.90%	2.54%	2.51%

each coming pattern, a style transfer matrix is learned from the latest few patterns. By considering the local style consistency, the style transfer matrix is expected to be nearly invariant within patterns from the same writer. At the same time, the style transfer matrix between the current writer and the former writer are different through forgetting the style knowledge learned from the patterns in a longer past. In this study, we evaluate our method by interleaved test-then-train, which tests for the ability of the style adaptation, and style-specific classification, which tests for the ability of the generalization performance.

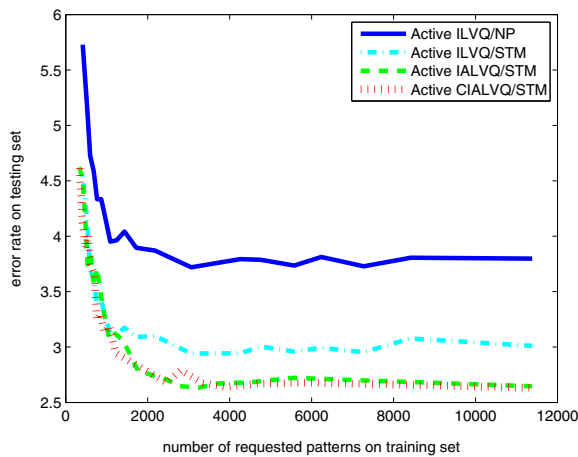
From the results in Table 2, we can see that CIALVQ has the best performance for style adaptation in the interleaved test-then-train scenario. Compared with ILVQ, IALVQ and CIALVQ can effectively reduce the error rate by means of

utilizing the local style consistency. Furthermore, CIALVQ further lowers the error rate by minimizing the inner-class distance in style-free space explicitly when estimating the style transfer matrix. As shown in Fig. 5, the error reduction rate gradually increases as the number of writer-specific patterns increases. This is consistent with the intuition that the local style consistency performs better when there are more same-style samples.

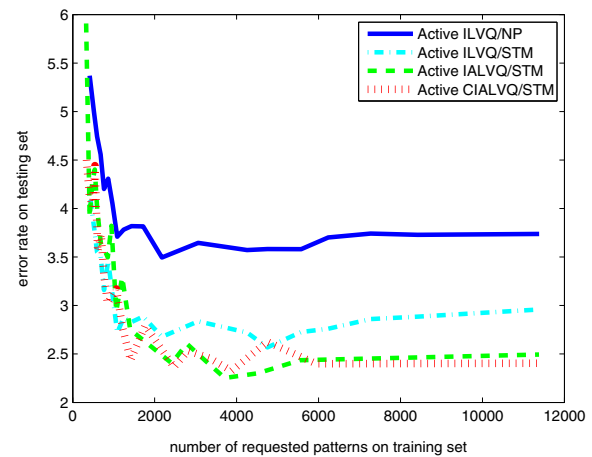
The results in Fig. 6 show the advantage of active CIALVQ over active ILVQ and active IALVQ. Better classification performance can be obtained when more labels are requested. However, after requesting a few important patterns, which have a lower confidence and around the classification boundary, the classification performance will tend to be stable. With the reduction of the needed tagged

Table 5 Error rates on four datasets using different number of patterns for each writer in style-specific classification

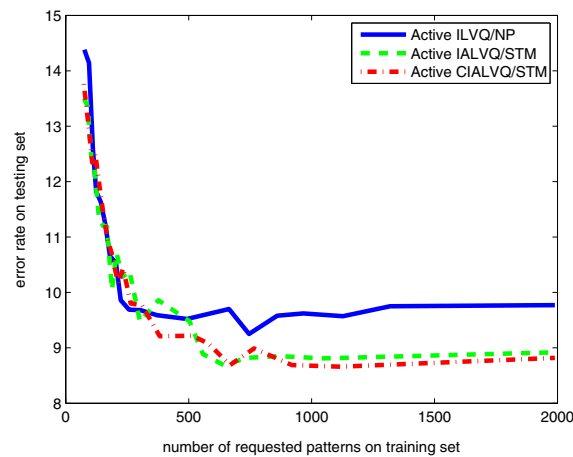
Training set	Test set	5 patterns		10 patterns		15 patterns		20 patterns	
SD3-Train	SD3-Test	LVQ (1)/NP	3.36%	LVQ (1)/NP	2.72%	LVQ (1)/NP	2.47%	LVQ (1)/NP	2.26%
		LVQ (30)/NP	2.31%	LVQ (30)/NP	1.86%	LVQ (30)/NP	1.59%	LVQ (30)/NP	1.47%
		ILVQ/NP	3.76%	ILVQ/NP	3.03%	ILVQ/NP	2.54%	ILVQ/NP	2.35%
		IALVQ/STM	3.26%	IALVQ/STM	2.60%	IALVQ/STM	2.19%	IALVQ/STM	2.05%
		CIALVQ/STM	3.20%	CIALVQ/STM	2.63%	CIALVQ/STM	2.18%	CIALVQ/STM	2.03%
SD3-Train	SD7-Test	LVQ (1)/NP	8.17%	LVQ (1)/NP	8.10%	LVQ (1)/NP	7.85%	LVQ (1)/NP	7.47%
		LVQ (30)/NP	7.10%	LVQ (30)/NP	6.53%	LVQ (30)/NP	6.00%	LVQ (30)/NP	5.74%
		ILVQ/NP	9.26%	ILVQ/NP	8.67%	ILVQ/NP	7.85%	ILVQ/NP	7.26%
		IALVQ/STM	8.58%	IALVQ/STM	7.71%	IALVQ/STM	7.19%	IALVQ/STM	6.72%
		CIALVQ/STM	8.47%	CIALVQ/STM	7.78%	CIALVQ/STM	7.19%	CIALVQ/STM	6.76%
SD7-Train	SD3-Test	LVQ (1)/NP	8.00%	LVQ (1)/NP	6.20%	LVQ (1)/NP	5.00%	LVQ (1)/NP	4.53%
		LVQ (30)/NP	5.71%	LVQ (30)/NP	4.57%	LVQ (30)/NP	3.72%	LVQ (30)/NP	3.57%
		ILVQ/NP	10.77%	ILVQ/NP	8.34%	ILVQ/NP	6.51%	ILVQ/NP	6.39%
		IALVQ/STM	8.68%	IALVQ/STM	6.88%	IALVQ/STM	5.26%	IALVQ/STM	5.20%
		CIALVQ/STM	8.38%	CIALVQ/STM	6.88%	CIALVQ/STM	5.44%	CIALVQ/STM	5.06%
SD7-Train	SD7-Test	LVQ (1)/NP	7.48%	LVQ (1)/NP	6.35%	LVQ (1)/NP	5.29%	LVQ (1)/NP	5.08%
		LVQ (30)/NP	6.20%	LVQ (30)/NP	4.74%	LVQ (30)/NP	3.88%	LVQ (30)/NP	3.72%
		ILVQ/NP	9.31%	ILVQ/NP	7.66%	ILVQ/NP	6.32%	ILVQ/NP	6.04%
		IALVQ/STM	7.89%	IALVQ/STM	6.66%	IALVQ/STM	5.27%	IALVQ/STM	4.98%
		CIALVQ/STM	7.85%	CIALVQ/STM	6.51%	CIALVQ/STM	5.22%	CIALVQ/STM	4.97%



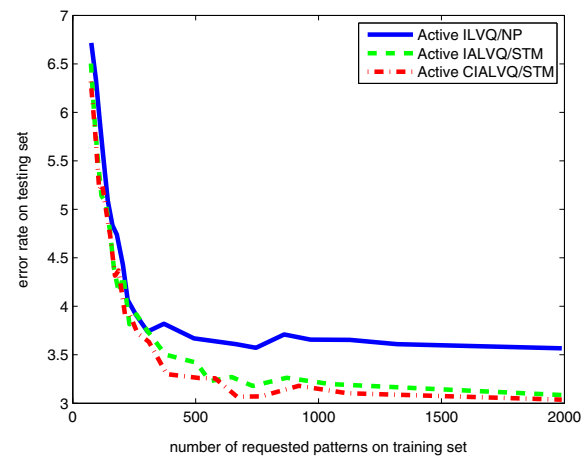
(a) SD7-Train/SD7-Test



(b) SD7-Train/SD3-Test



(c) SD3-Train/SD7-Test(5 patterns from each writer)



(d) SD3-Train/SD3-Test(5 patterns from each writer)

Fig. 7 Active style-specific classification result in different dataset

samples in active learning, the labor of labeling samples are saved significantly. So, finding a good balance between the number of the requested labels and classification performance are very important in active learning.

In the supervised style-specific classification mode, the results from Tables 4 and 5 show that when evaluating generalized classification performance using nearest prototype (NP) rule, the incremental LVQ results in higher error rate than supervised LVQ which treats all training samples iteratively. When all patterns from the writers are utilized, STM reduces the error rate of both LVQ and ILVQ considerably. This observation conforms with previous results in [14]. Due to the utilization of local style consistency we can see that IALVQ/STM, CIALVQ/STM can further reduce the error rate of ILVQ/STM. It is noted that ILVQ/STM projects the patterns from style-specific space onto style-conscious space while IALVQ/STM and CIALVQ/STM project the

patterns from style-specific space onto style-free space. With only a small proportion of patterns from the writers, learning an accurate style transfer matrix is impossible due to the underfitting. So, we just finetune the style transfer matrix based on a small quantity of the patterns from the specific writer and the linear projection learned in training stage. Again IALVQ, CIALVQ reduce the error rate of ILVQ when local style consistency is considered in training stage. Figure 7 leads to a similar conclusion for the active style-specific classification.

Conclusion

In this paper, we have proposed a continuous adaptive prototype-based model for incremental character recognition. The proposed method models the writing styles with

a linear transformation matrix which projects patterns from style-conscious space onto style-free space incrementally. Two kinds of prototypes (i.e., style-conscious prototypes and style-free prototypes) are learned simultaneously. The style-conscious prototypes are updated by the original patterns while the style-free prototypes are updated by the projected patterns. Finally, the interleaved test-then-train performance is tested on style-free prototypes. Union of style-conscious prototypes and style-free prototypes are used for the style-specific classification.

Experimental results have shown that the proposed method is effective in both supervised incremental learning and active incremental learning. It is important to note that our method also has the advantage even when the patterns from the same style are only a few, which is a very common setting in practical applications (such as in signature and email address). Active incremental learning is akin to human learning which occasionally interacts with a teacher for inquiring labels of unknown patterns. In our experiments, however, the proportion of inquired samples is considerable and implies high cost of interactive learning. As part of future work, we will consider efficient unsupervised incremental learning or interactive learning with very small number of samples inquired.

Compliance with Ethical Standards

Funding This work has been supported in part by the Strategic Priority Research Program of the CAS Grant XDB02060009 and the National Natural Science Foundation of China (NSFC) Grant 61411136002.

Conflict of Interests The authors declare that they have no conflict of interest.

Ethical Approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

- Bransford JD, Brown AL, Cocking RR. How people learn: brain, mind, experience, and school. National Academy Press. 2000.
- Gros C. Cognitive computation with autonomously active neural networks: an emerging field[J]. *Cogn Comput*. 2009;1(1):77–90.
- Gong C, Tao D, Liu W, Liu L, Yang J. Label propagation via teaching-to-learn and learning-to-teach[J]. *IEEE Trans Neural Netw Learn Syst*. 2017;28(6):1452–65.
- Gong C, Tao D, Maybank SJ, Liu W, Kang G, Yang J. Multi-modal curriculum learning for semi-supervised image classification[J]. *IEEE Trans Image Process*. 2016;25(7):3249–60.
- Gong C, Tao D, Yang J, Liu W. Teaching-to-learn and learning-to-teach for multi-label propagation[C]. *AAAI*. 2016. p. 1610–16.
- Gong C. Exploring commonality and individuality for multi-modal curriculum learning[C]. *AAAI*. 2017. p. 1926–33.
- Syed N, Liu H, Sung K. Incremental learning with support vector machines[C]. In: *International joint conference on artificial intelligence*. Sweden: Morgan Kaufmann Publishers. 1999. p. 352–6.
- Hoi SCH, Wang J, Zhao P. Libol: A library for online learning algorithms[J]. *J Mach Learn Res*. 2014;15(1):495–9.
- Ding S, Zhang J, Jia H, et al. An adaptive density data stream clustering algorithm[J]. *Cogn Comput*. 2016;8(1):30–38.
- Gepperth A, Karaoguz C. A bio-inspired incremental learning architecture for applied perceptual problems[J]. *Cogn Comput*. 2016;8(5):924–34.
- Trier ØD, Jain AK, Taxt T. Feature extraction methods for character recognition—a survey[J]. *Pattern Recogn*. 1996;29(4):641–62.
- Sarkar P, Nagy G. Style consistent classification of isogenous patterns[J]. *IEEE Trans Pattern Anal Mach Intell*. 2005;27(1):88–98.
- Abraham WC, Robins A. Memory retention the synaptic stability versus plasticity dilemma. *Trends Neurosci*. 2005;28(2):73–8.
- Zhang XY, Liu CL. Writer adaptation with style transfer mapping[J]. *IEEE Trans Pattern Anal Mach Intell*. 2013;35(7):1773–87.
- Kohonen T. Improved versions of learning vector quantization[C]. In: *International joint conference on neural networks*. 1990. p. 545–550.
- Kohonen T. The self-organizing map[J]. *Proc IEEE*. 1990;78(9):1464–80.
- Jin XB, Liu CL, Hou X. Regularized margin-based conditional log-likelihood loss for prototype learning[J]. *Pattern Recogn*. 2010;43(7):2428–38.
- Shen YY, Liu CL. Incremental learning vector quantization for character recognition with local style consistency[C]. In: *Proceeding of the 8th international conference in brain inspired cognitive systems*. 2016. p. 228–39.
- Rosenblatt F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychol Rev*. 1958;65(6):386–408.
- Oza NC. Online bagging and boosting[C]. *IEEE International Conference on Systems, Man and Cybernetics*:2340–45. 2005.
- Liu X, Yu T. Gradient feature selection for online boosting[C]. *ICCV*. 2007. p. 18.
- Saffari A, Leistner C, Santner J, et al. On-line random forests[C]. In: *Computer vision workshops (ICCV Workshops)*. 2009. p. 1393–400.
- Kirstein S, Wersing H, Körner E. A biologically motivated visual memory architecture for online learning of objects[J]. *Neural Netw*. 2008;21(1):65–77.
- Xu Y, Shen F, Zhao J. An incremental learning vector quantization algorithm for pattern classification[J]. *Neural Comput Appl*. 2012;21(6):1205–15.
- Cesa-Bianchi N, Conconi A, Gentile C. A second-order perceptron algorithm[J]. *SIAM J Comput*. 2005;34(3):640–68.
- Crammer K, Dredze M, Kulesza A. Multi-class confidence weighted algorithms[C]. In: *Proceedings of the conference on empirical methods in natural language processing*. 2009. p. 496–504.
- Ushiku Y, Hidaka M, Harada T. Three guidelines of online learning for large-scale visual recognition[C]. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014. p. 3574–81.
- Gama J, Žilobaitė I, Bifet A, Pechenizkiy M, Bouchachia A. A survey on concept drift adaptation[J]. *ACM Comput Surv*. 2014;46(4).
- Veeramachaneni S, Nagy G. Adaptive classifiers for multisource OCR[J]. *Int J Doc Anal Recogn*. 2003;6(3):154–66.
- Veeramachaneni S, Nagy G. Style context with second-order statistics[J]. *IEEE Trans Pattern Anal Mach Intell*. 2005;27(1):14–22.
- Huang Z, Ding K, Jin L, et al. Writer adaptive online handwriting recognition using incremental linear discriminant analysis[C]. In: *IEEE Proceedings of the conference on document analysis and recognition*. 2009. p. 91–5.

32. Ding K, Jin L. Incremental MQDF learning for writer adaptive handwriting recognition[C]. In: Proceedings of the conference on frontiers in handwriting recognition (ICFHR). 2010. p. 559–64.
33. Bishop CM. Pattern recognition and machine learning. New York: Springer; 2006.
34. Schleif FM, Hammer B, Villmann T. Margin-based active learning for LVQ networks[J]. *Neurocomputing*. 2007;70(7):1215–24.
35. Grother PJ. Handprinted forms and character database, NIST special database 19. Technical Report and CDROM. 1995.
36. Liu CL, Sako H, Fujisawa H. Performance evaluation of pattern classifiers for handwritten character recognition[J]. *Int J Doc Anal Recogn*. 2002;4(3):191–204.
37. Duchi J, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization[J]. *J Mach Learn Res*. 2011;12:2121–59.