Deep Transfer Mapping for Unsupervised Writer Adaptation

***Hong-Ming Yang*[1,2]**, Xu-Yao Zhang[1,2], Fei Yin[1,2], Jun Sun[4], Cheng-Lin Liu[1,2,3]

[1]NLPR, Institute of Automation, Chinese Academy of Sciences

[2]University of Chinese Academy of Sciences

[3]CAS Center for Excellence in Brain Science and Intelligence Technology

[4]Fujitsu Research & Development Center

Aug. 8, 2018

Outline

- Introduction
- Style Transfer Mapping
- Motivation and the Proposed Method
- Experiments and Analysis
- Conclusions

Introduction

- **A main challenge for handwriting recognition:**

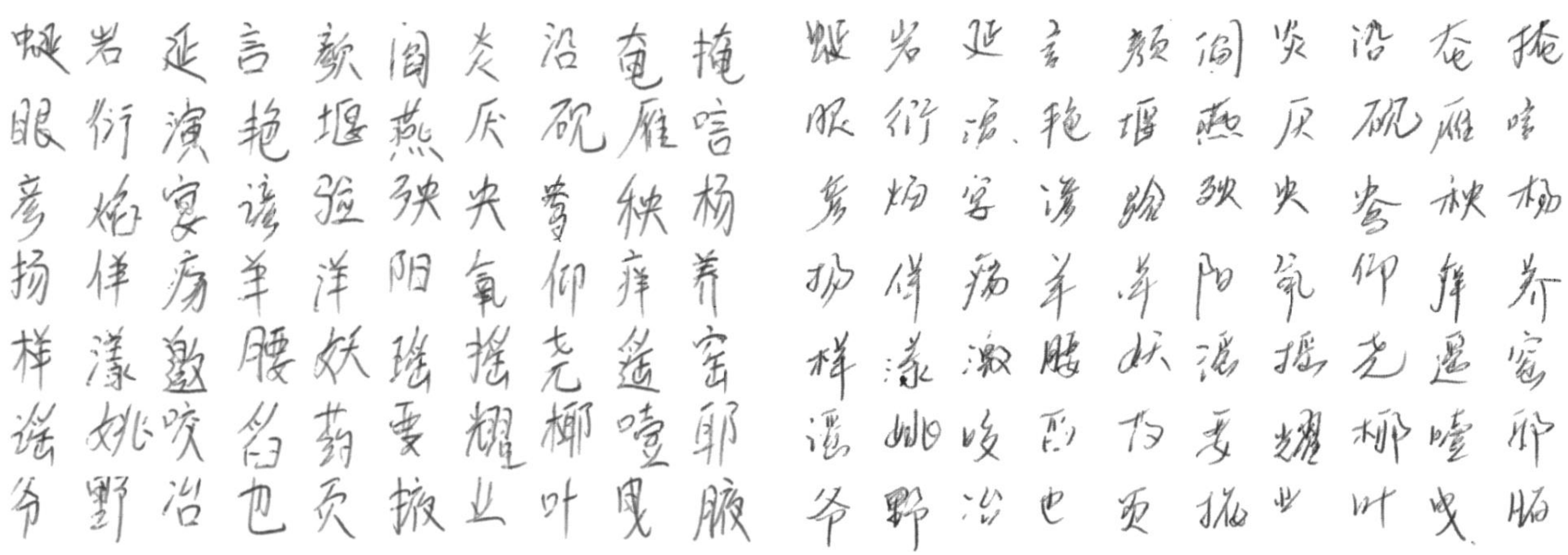
The large variability of distributions across training and different test data

– Different writing styles of different writers

– Different writing tools (e.g. different pens or electronic writing devices)

– Different writing environments (e.g. normal or emergency situations)

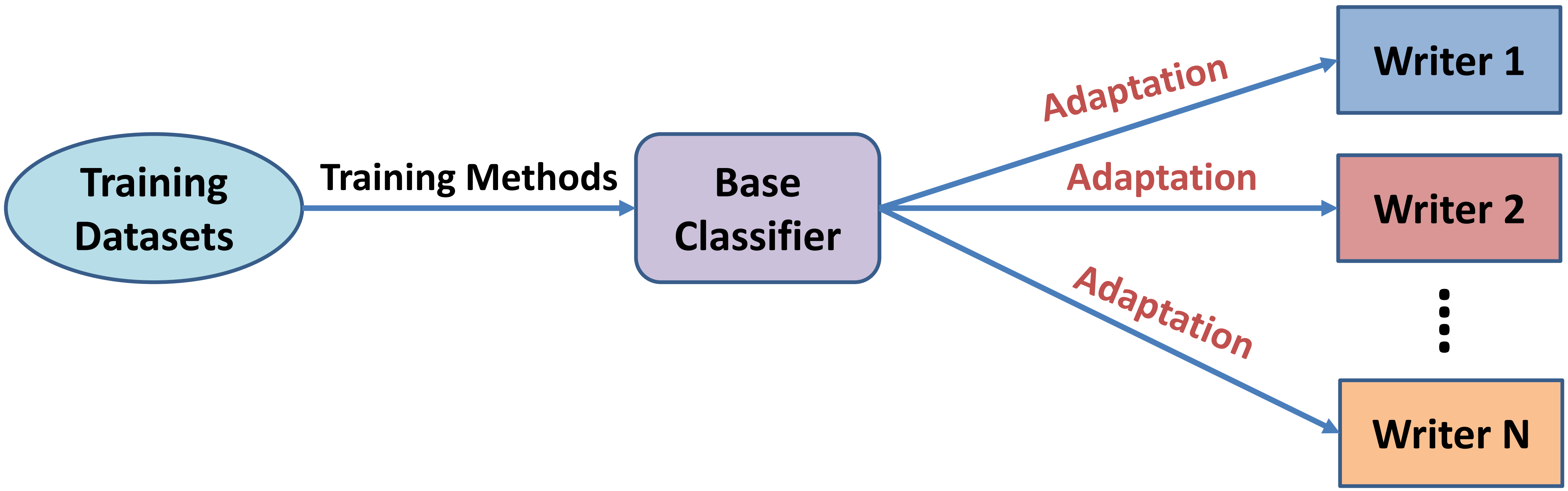
– …………



Written characters of two writers.

Introduction

- Domain adaption: a form of transfer learning
  Adapt the base classifier to each domain in the test dataset



- Recent methods: mainly based on deep learning
  – Fine tuning with target domain data
  – Learning domain invariant representations (features)
  – Project source or target domain data to align the distribution

Style Transfer Mapping

Style transfer mapping (STM)

Main idea: project the target domain (test) data to balance the data distribution

$$p(x_s) \neq p(x_t)$$
$$\hat{x}_t = A_t x_t + b_t$$
$$p(x_s) \approx p(\hat{x}_t)$$

Learning classifier on x_s and apply $\hat{x}_t$ to the base classifier

Learning of the projection (A_t and b_t)

$$min_{A \in R^{d \times d}, b \in R^d} \sum_{i=1}^{n} f_i \|As_i + b - t_i\|_2^2 + \beta \|A - I\|_F^2 + \gamma \|b\|_2^2$$

Source points s_i: features in the target domain, i.e., x_t

Target points t_i: prototype (LVQ) or mean (MQDF) for class y_i (y_i is the label of sample s_i)

[Xu-Yao. Zhang et al., writer adaptation with style transfer mapping. TPAMI'13]

Style Transfer Mapping

Solution: a convex quadratic programming problem, has a closed-form solution

$$A = QP^{-1}, b = \frac{1}{\hat{f}}(\hat{t} - A\hat{s})$$

$$Q = \sum_{i=1}^{n} f_i t_i s_i^{\mathrm{T}} - \frac{1}{\hat{f}}\hat{t}\hat{s}^{\mathrm{T}} + \beta I \quad P = \sum_{i=1}^{n} f_i s_i s_i^{\mathrm{T}} - \frac{1}{\hat{f}}\hat{s}\hat{s}^{\mathrm{T}} + \beta I \quad \hat{s} = \sum_{i=1}^{n} f_i s_i \quad \hat{t} = \sum_{i=1}^{n} f_i t_i \quad \hat{f} = \sum_{i=1}^{n} f_i + \gamma$$

- **Dealing with unsupervised adaptation**

– Using the pseudo labels, predicted by the base classifier

– Iteration method: base classifier → pseudo label → adaptation → better pseudo label → adaptation → ……

- **Extend to convolutional neural networks (CNNs)**

Main idea: perform adaptation on the deep features

$$f(x): CNN\ feature\ extractor$$

$$x_s = f(x_s), \qquad x_t = f(x_t)$$

[Xu-Yao. Zhang et al., writer adaptation with style transfer mapping. TPAMI'13]

[Xu-Yao. Zhang et al., online and offline handwritten Chinese character recognition: a comprehensive study and new benchmark. PR'17]

Motivations & Methods

- **Traditional adaptation methods with CNN**

– Consider only the fully connected layers

– Perform adaptation only on one layer

- **Motivations**

– Adaptation on both fully connected layers and convolutional layers

– Perform adaptation on multiple (or all) layers of the base CNN

- **Adaptation method for fully connected layers**

STM based on the deep features of the layer (unsupervised adaptation)

- **Adaptation methods for convolutional layers**

– Use a linear transformation to project the target domain data for aligning the data distributions

– Propose four variations of linear transformation, which are based on different assumptions of the space relation in the feature maps

Motivations & Methods

➢ Fully associate adaptation (FAA):

– Output of a convolutional layer for an input x_i

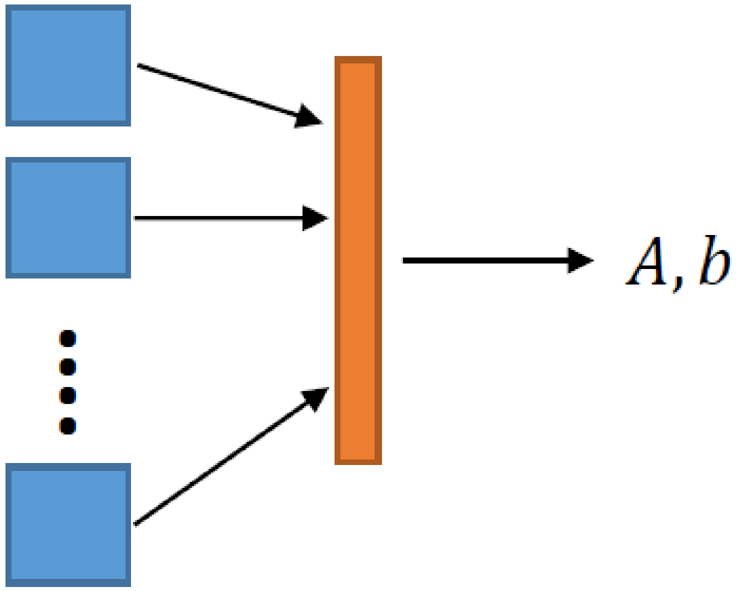
$$o_i = \{d_{cjk}\}_{c=1,j=1,k=1}^{c=C,j=H,k=W}$$

c, j, k: index of the feature maps, rows, and columns in each feature map

– Assumption: all positions of (c, j, k) are related to each other

– Method: expand o_i to a long vector v_i with dimension CHW, and learn a transformation $A \in R^{CHW \times CHW}$, $b \in R^{CHW}$ by STM

– $v_i' = Av_i + b$, $(v_i')_j = \sum_{k=1}^{CHW} A_{jk}(v_i)_k + b_j$, each position j in v_i' are related to all positions in v_i



Motivations & Methods

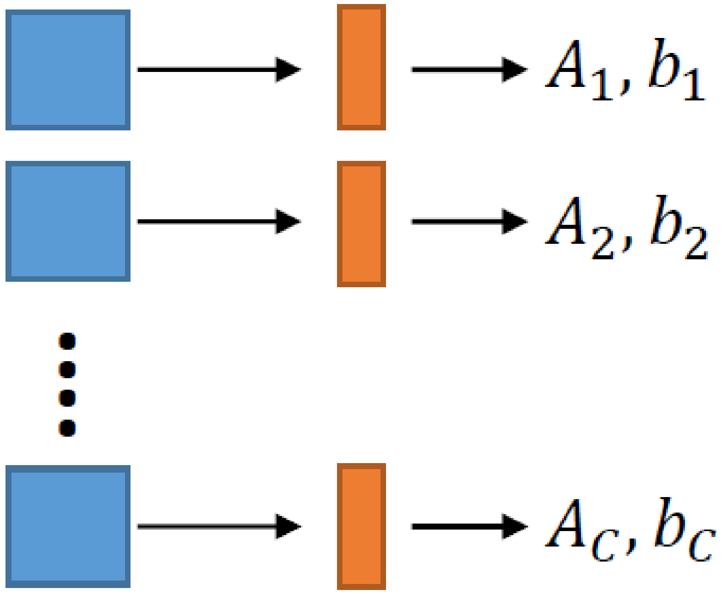
- **Partly associate adaptation (PAA):**

– Assumption: positions within the same feature map are related to each other, but the feature maps are mutually independent

– Method: expand each feature map to a vector with dimension HW, and learn a transformation $A_c \in R^{HW \times HW}$, $b_c \in R^{HW}$ for each feature map c separately by STM

– Transformation A_c, b_c ensures the relation of positions within a feature map, learn A_c, b_c separately ensures the independence between the feature maps

Separate STM for each feature map



Motivations & Methods

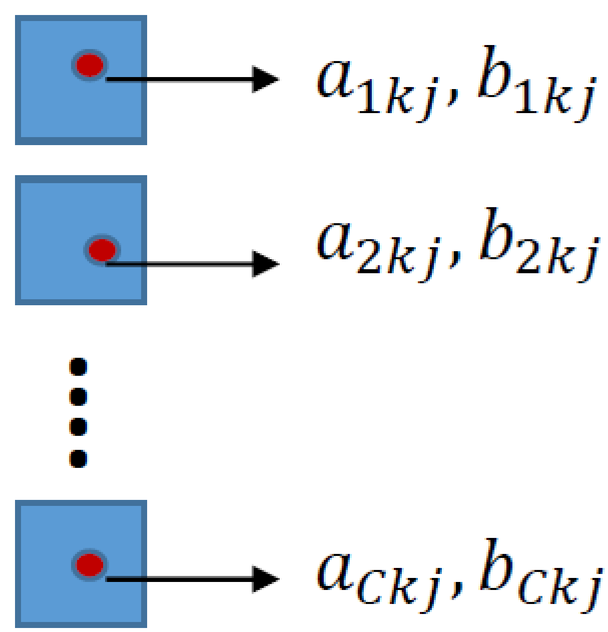
- **Weakly independent adaptation (WIA):**

– Assumption: all positions (c, j, k) in o_i are independent to each other

– Learn a transformation $a, b \in R$ for each position (c, j, k) separately by STM

– $(o_i')_{c_0,j_0,k_0} = a(o_i)_{c_0,j_0,k_0} + b$

– $min_{a,b\in R} \sum_{i=1}^{N_t} f_i\left(a(o_i)_{c_0,j_0,k_0} + b - (t_i)_{c_0,j_0,k_0}\right)^2 + \beta(a-1)^2 + \gamma b^2$



Motivations & Methods

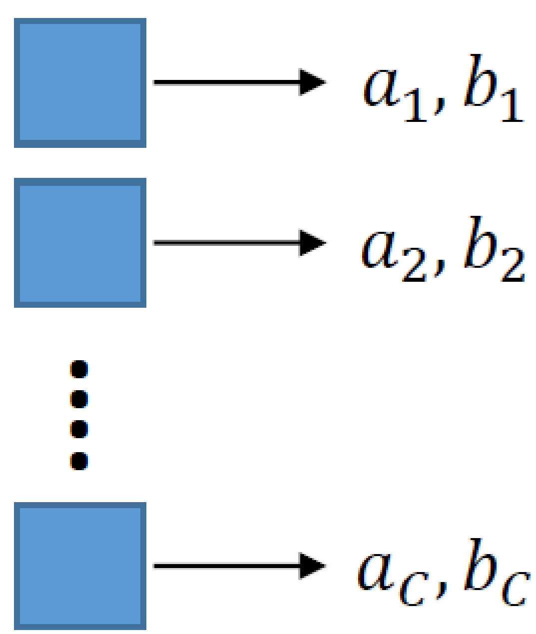
- **Strong independent adaptation (SIA):**

– Assumption: all positions are independent to each other and the positions within the same feature map share a same linear transformation

– Similar to the linear projection in the batch normalization (BN) layer

– Learn a transformation $a, b \in R$ for each feature map separately by STM

$$min_{a,b \in R} \sum_{i=1}^{N_t} \sum_{j=1}^{H} \sum_{k=1}^{W} f_i\left(a(o_i)_{c_0,j,k} + b - (t_i)_{c_0,j,k}\right)^2 + \beta(a-1)^2 + \gamma b^2$$



Motivations & Methods

➢ Analysis and comparison

Adaptation Methods	FAA	PAA	WIA	SIA
Assumption	All positions related	Inner feature map related	All positions independent	All positions independent & parameter sharing
Feature Dimension	CHW	HW	1	1
Matrix Size	$CHW \times CHW$	$HW \times HW$	1×1	1×1
Transformation Number	1	C	CHW	C
Total Parameters	$CHW(CHW+1)$	$CHW(HW+1)$	$2CHW$	$2C$

– Complexity & Flexibility: FAA > PAA > WIA > SIA

– Computation & Memory efficiency: SIA > WIA > PAA > FAA

Motivations & Methods

● Deep transfer mapping (DTM)

Perform adaptation on multiple layers in a deep manner

➢ **Algorithm**

1. Select a group of layers L on which to perform adaptation
2. From bottom to top layers in L, perform adaptation on the specific layer with the proposed adaptation methods, but keep the other layers unchanged
3. After adaptation on each layer, insert an linear layer after it and set the weights and bias of the linear layer as the solved A and b

➢ **Advantages of DTM**

– More powerful for flexibly aligning the distributions between the domains

– Captures more comprehensive information and minimize the discrepancy of distributions under different abstract levels

Experiments & Analysis

Datasets

Dataset	Dataset Info	#Sample (in 3755-class)	#Writers (domains)
Training Set	CASIA OLHWDB 1.0-1.2	2,697,673	1020
Test Set	On-ICDAR2013 Competition	224,590	60
Adaptation Set	Unlabeled samples from each domain (writer) in test set		

Online handwritten Chinese characters. The samples of one writer stored in one single file, can be viewed as a domain.

writer 1258

writer 1242

Experiments & Analysis

Layer ID	Layer Type	Parameter	Pooling	Drop Rate
0	input	$8(32 \times 32)$	#	0.0
1	conv	$50(3 \times 3)$	#	0.0
2	conv	$100(3 \times 3)$	2×2	0.1
3	conv	$150(3 \times 3)$	#	0.1
4	conv	$200(3 \times 3)$	2×2	0.2
5	conv	$250(3 \times 3)$	#	0.2
6	conv	$300(3 \times 3)$	2×2	0.3
7	conv	$350(3 \times 3)$	#	0.3
8	conv	$400(3 \times 3)$	2×2	0.4
9	FC	900	#	0.5
10	FC	200	#	0.0
11	softmax	3755	#	#

Base classifier: 11 layers
97.55% accuracy on test set

- **Different adaptation methods for Convolutional layers**

Four adaption methods on the same convolutional layer #8

Methods	without	FAA	PAA	WIA	SIA
Test Acc (%)	97.55	97.91	97.71	97.69	97.62
ERR (%)	0	14.69	6.53	5.71	2.86

Experiments & Analysis

Adaptation property of different layers in CNN

TABLE III
ADAPTATION PERFORMANCE FOR DIFFERENT LAYERS IN CNN.

Layer ID	1	2	3	4	5
Test Acc (%)	97.51	97.57	97.61	97.61	97.63
ERR (%)	-1.63	0.82	2.45	2.45	3.27
Layer ID	6	7	8	9	10
Test Acc (%)	97.67	97.67	97.69	97.85	97.91
ERR (%)	4.90	4.90	5.71	12.24	14.69

Adaptation method: WIA

- From bottom to top layers, the adaptation performance increases
- Bottom layers extract general features, which are applicable across different domains, thus the promotion are not obvious after adaptation
- Top layers occupy abstract features, which are more domain specific, thus adaptation is helpful for such layers

Experiments & Analysis

Deep transfer mapping

TABLE IV
ADAPTATION PERFORMANCE FOR DEEP TRANSFER MAPPING.

Layer ID	without	8	$8 \to 9$	$8 \to 9 \to 10$
Test Acc (%)	97.55	97.91	98.00	98.02
ERR (%)	0	14.69	18.37	19.18

- DTM can further boost the performance of the base classifier
- DTM still has some limitations, the promotion is not obvious when adopt overmuch adaptations

Conclusions

- Unsupervised domain adaptation to alleviate the writing style variation, assuming each writer has an consistent style
- Four variations of adaptation methods for convolutional layers, assuming different space relations in the output of convolutional layers
- Deep transfer mapping (DTM) method to conduct adaptation on multiple (or all) layers of CNN, to better align the data distributions of different styles
- Remaining Problems
  - What is the best way of adaptation for deep neural networks
  - How to adapt in the case of small sample in adaption/testing (currently 3,755 samples per writer)
  - Theoretical modeling of within/between-writer style variation
  - Continuous adaptation of classifier

Thanks for your attention!