IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Marginalized Stacked Denoising Autoencoder with Adaptive Noise Probability for Cross domain Classification

YUHONG ZHANG[1,2], SHUAI YANG[1,2], PEIPEI LI[1,2], XUEGANG HU[1,2,3], AND HAO WANG[1,2]

[1] Key Laboratory of Knowledge Engineering with Big Data (Hefei University of Technology), Ministry of Education
[2] School of Computer Science and Information Engineering, Hefei University of Technology, Anhui 230009, China
[3] Anhui Province Key Laboratory of Industry Safety and Emergency Technology, Anhui 230009, China

Corresponding author: Shuai Yang (e-mail: yangs@mail.hfut.edu.cn).

This work is supported in part by the National Key Research and Development Program of China under grant 2016YFB1000901, the Natural Science Foundation of China under grants (61673152,91746209) and the Key Laboratory of Data Science and Intelligence Application, Fujian Province University (NO. 1902)

ABSTRACT Cross-domain classification is a challenging problem, in which, how to learn domain invariant features is critical. Recently, significant improvements to this problem have emerged with the wide application of deep learning models, which have been proposed to learn higher level and robust feature representation. Marginalized stacked denoising autoencoder model (mSDA) has proved to be effective to address this problem. However, the performance of mSDA is sensitive to the noise probability. In previous works, the noise probability is usually set as a constant value by cross-validation in the source domain. There is few work focus on the relationship between the noise probability and cross-domain task. In this paper, we try to compute the value of noise probability adaptively. Thus, an approach called Marginalized Stacked Denoising Autoencoders with Adaptive noise Probability (mSDA-AP) is proposed. Firstly, we extract an informative feature space by an improved index, weighted log-likehood ratio (IWLLR), then aggregate these informative features by weighting. Secondly, we compute the value of noise probability adaptively according to the distance between source domain and target domain, and then with the adaptive noise probability, we disturb the input data to learn a stronger feature space with mSDA. Finally, experimental results show the effectiveness of our proposed approach.

INDEX TERMS Domain adaptation, mSDA, noise probability, adaptive parameter.

I. INTRODUCTION

CROSS-domain classification aims to train a classifier from the source domain, in which the labeled instances are relatively sufficient, to learn a classifier for unseen or unlabeled data in a target domain. Both domains are assumed to be related, but not identical. And cross-domain text classification is thought as a challenging problem, because there are some domain-specific words, which only occur in the source domain and do not occur in the target domain [1]–[4], [39]. Cross-domain classification is also called transfer learning, domain adaptation.

Learning an invariant and shared feature space is a common and efficient way to address this problem. Most works can be divided into two sub-categories: raw feature space based on approaches [5], [6] and the latent or higher feature space based on approaches [7], [8], [41]. These works extract a shared feature set and build the mapping relation between the specific features of two domains with the shared features as a bridge.

Recently, deep learning models attract attention from researchers widely. Deep learning models try to learn a feature space which is suitable for source and target domains to address cross-domain classification. Specially, deep learning models learn a stronger feature space by yielding multiple layers of intermediate concepts between raw input and target. These intermediate concepts can been seen as higher-level feature representation. In recent years, deep learning has been used as a generic solution in Natural Language Processing and achieved significant effect. Models such as Convolutional Neural Network [9]–[11], [32], Recurrent Neural Network

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2925811, IEEE Access

[12]–[15] and Autoencoders [17], [19], [31] have been used in cross-domain classification.

As an unsupervised model, stacked denoising autoencoders (SDA) [17] has shown great success in cross-domain classification, which stacked multiple layers of denoising autoencoders to form a deep learning model and learned a strong latent space for the cross-domain classification. However, SDA optimized the network structure iteratively, which led to a higher computational cost. Chen et al. [19] proposed the marginalized stacked denoising autoencoders (mSDA), it optimized the structure and parameters with a liner approach instead of the expensive iteration process. In addition, Clinchant et al. [20] suggested a more appropriate regularization for denoising autoencoders (MDA-TR) based on the work of [21], which made the invariant features can be focused more easily. However, MDA-TR only performed well in one-layer model. In addition, Csurka et al. [22] proposed an extended framework for marginalized domain adaptation with a domain regularization that may be either a domain prediction loss or a maximum mean discrepancy between the source and target data, aimed at addressing unsupervised, supervised and semi-supervised scenarios. Jiang et al. [23], [33] used $\ell_{2,1}$-norm to measure the reconstruction error to learn powerful representation for domain adaptation tasks.

Although these SDA and mSDA can learn good feature representation, their performances were sensitive to the noise probability, which were used to disturb the input data. Most works set the noise probability as a constant value through experimental results. In this paper, we find that the value of noise probability should be varied with the difference of cross-domain tasks. Therefore, an adaptive computing way for the optimal value of noise probability is proposed for cross-domain classification. Our main contributions are summarized below:

- We propose a mechanism to compute the value of noise probability adaptively for mSDA according to the divergence between the source and target domain, with which, we can get a more robust feature space. Moreover, this mechanism can make sure that the adaptation is not sensitive to different datasets because of the value of parameter.
- We propose an improved Weighted Log-likehood Ratio (IWLLR) index, which is improved from WLLR (Weighted Log-likehood Ratio), to distinguish the shared features and special features. Compared with WLLR, IWLLR makes the important features have higher polarity values. Thus it is more suitable for the cross-domain task.

The remainder of this paper is organized as follows. Section 2 briefly reviews related work. Section 3 gives the details of our mSDA-AP algorithm. Section 4 shows the effectiveness of our approach with experimental results. Section 5 summarizes the paper.

II. RELATED WORK

The goal of cross-domain classification is to reduce the performance degradation of classifier between source domain and target domain. Recent works have mainly investigated two techniques for alleviating the difference: learning joint feature representations [1], [2], [24], and learning a robust feature space with deep learning models [17], [19], [25]. This paper belongs to the second one.

In view of raw words, Blitzer et al. [1] proposed a structural correspondence learning (SCL) method to find the correspondence between features from different domains via pivot features. Pan et al. [2] proposed a spectral feature alignment (SFA) method to align domain-specific words from different domains into unified clusters. Bollegala et al. [16] grouped different features expressing the same sentiment into one thesaurus, and proposed an asymmetric related measure to compute the similarity of features. III et al. [5] augmented instances with features differentiating source and target domains to improve a nonlinear kernel mapping between domains. Lan et al. [40] proposed a multiple sparse representation framework for visual tracking which jointly utilized the shared and feature-specific properties of different features by decomposing multiple sparsity patterns.

Many researchers have used autoencoders as a powerful tool for automatic extraction of nonlinear features. Glorot et al. [17] trained a SDA to reconstruct the input vectors on the union of the source and target data, and extracted features for domain adaptation from partial and random corruption. The denoiseres can be stacked into deep learning architectures. Chen et al. [19] proposed the marginalized stacked denoising autoencoders (mSDA) that addressed two crucial limitations of SDA: high computational cost and the lack of scalability to high-dimensional features. mSDA marginalized noise and adopted the liner denoiser to learn parameters instead of the stochastic gradient descent algorithm. Yang et al. [25] proposed marginalized structured dropout, which exploited the feature structure to obtain a remarkably simple and efficient feature projection. They also proposed two alternative noising techniques: feature scrambling and structured dropout. And the structured dropout can make marginalization easier and obtain dramatic speedups without sacrificing accuracy. Deep nonlinear feature coding (DNFC) [35] was presented to address two main limitations of mSDA: the divergence between source and target domains in the new feature space was not taken into consideration and the nonlinear relationship in the data with the new feature representation may not be captured because mSDA injected the nonlinearity after feature learning. DNFC minimized domain divergence with empirical maximum mean discrepancy (MMD) metric and used kernelization for nonlinear coding. Feature analysis of marginalized stacked denoising autoenconder (DTFC) [31] disturbed the input data with multinomial dropout noise to obtain richer feature representation. Unsupervised learning has a major disadvantage that is easy to overfit on the source training data. Yang et al. [36] presented a representation learning framework via serial autoencoders (SEAE), which

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2925811, IEEE Access

learned richer feature representation by serially connecting two different types of autoencoders. Based on the work of Ganin and Lempitsky [21], Clinchant et al. [20] proposed an unsupervised regularization method for mSDA to make the adaptation easier. Ziser et al. [26] proposed a neural network model that marries two ideas together: SCL [1] and autoencoder neural networks. Their model was a three-layer neural network that encoded the non-pivot features of an input example into a low dimensional representation, so that the existence of pivot features in the example can be decoded. Zhuang et al. [11] proposed transfer learning with deep autoencoders (TLDA). It was a supervised representation learning method based on deep autoencoders, and consisted of the methods of distance minimization between source and target domains and label encoding of the data in source domain, the main drawback of TLDA was that the common autoencoder was used for feature learning, and it did not consider the sparse and over-complete features in learning feature representation. In addition, there are some variant models based on SDA and mSDA [28], [29]. Jiang et al. [23] proposed $\ell_{2,1}$-norm stacked robust autoencoders to learn useful representations for domain adaptation tasks. It was based on a loss and regularizer framework, each layer of $\ell_{2,1}$ -SRA contained two steps: a robust linear reconstruction step which was based on $\ell_{2,1}$ robust regression and a non-linear squashing transformation step, which made their method easier to implement. Stacked reconstruction independent component analysis (SRICA) [27] minimized the KL-Divergence between source and target domains and used the softmax regression to encode label information of the data in source domain. Wasserstein distance guided representation learning (WDGRL) [34] obtained domain invariant feature representations in an adversarial manner by minimizing the wasserstein distance between source and target domains. Zhou et al. [37] proposed a transfer learning model DATNet for low-resource NER to address the problems of representation difference and resource data imbalance. Zhou et al. [38] presented a framework named transfer hashing with privileged information to solve the data sparsity issue in hashing. Zhou et al. [42] proposed a method of multi-class heterogeneous domain adaptation to obtain a sparse feature transformation between domains with multiple classes.

III. PROPOSED ALGORITHM

In this section, we first give some basic concepts used in this paper, and then give the details of our mSDA-AP approach.

Given a labeled source domain D_s and an unlabeled target domain D_t, where $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ and $D_t = \{(x_j^t)\}_{i=1}^{n_t}$. Among them, n_s and n_t are the number of instances in the source and target domain respectively; x_i^s and x_j^t are the i-th, j-th instance in the source and target domain respectively; y_i^s is the label of the i-th instance in the source domain D_s. The goal of this paper is to learn a robust feature representation F from $D_s \cup D_t$. $D_s \cup D_t$ can be denoted as $\Re^{(n_s+n_t)\times d}$, where d is the dimension of feature space of $D_s \cup D_t$. And with the learnt F, we will train a classifier for target domain.

In this paper, we propose an adaptive approach for noise probability based on the Marginalized Stacked Denoising Autoencoders (mSDA-AP). Firstly, we select the shared features using our proposed IWLLR and weight this feature space to get an informative input data. Then, we compute the value of noise probability according to the distance between two domains, and disturb the weighted data with this noise probability to learn the stronger feature space using mSDA. Lastly, a classifier based on the stronger feature space is trained for the target domain classification. The framework of our mSDA-AP is shown in Figure 1.

A. SELECT SHARED FEATURES AND WEIGHT THEM

Shared features are critical for cross-domain tasks. In this subsection, we select some informative shared features according to the frequency and polarity index, and then highlight the shared features by weighting.

In SCL [1], the frequency is used to select shared features, and some features with lower polarities will be selected. In SFA [2], both mutual information (MI) and frequency are used, and some features with higher polarities in D_s but not in D_t will be selected. In this subsection, we expect to select the features with higher polarities and frequency in both domains. So we select some candidate words that appear more than 3 times in both source and target domains as a set $CW = \{w_1, w_2, \cdots\}$, based which, IWLLR index is designed.

We propose IWLLR index by introducing $\overline{y}$ into WLLR, which is a polarity index of features. IWLLR makes the important features present higher polarity values, and is more suitable for the cross-domain task. The process is shown in Eq. (1). Besides, we consider both the importance of features in the target domain and the polarities of features (as shown in Eq. (2)) to select shared features.

$$IWLLR(w_i, y^s) = p_{w_i}^{y^s}(1 - p_{w_i}^{\overline{y^s}})log(\frac{p_{w_i}^{y^s}(1 - p_{w_i}^{\overline{y^s}})}{p_{w_i}^{\overline{y^s}}(1 - p_{w_i}^{y^s})}) \quad (1)$$

$$r(w_i|y^s) = IWLLR(w_i, y^s)\frac{p(w_i|D_t)}{p(w_i|D_s)}. \quad (2)$$

where w_i is a word in the set of CW, y^s is the sentiment label, $\overline{y^s}$ is the opposite label of y^s, and $p_{w_i}^{y^s}$ is the probability of feature w_i in sentence labeled with y^s, and $p_{w_i}^{\overline{y^s}}$ means the probability of w_i in sentences those labels are not y^s. $p(w_i|D_s)$ and $p(w_i|D_t)$ is the probability of feature w_i in the source and target domain respectively. $p(w_i|D_t)/p(w_i|D_s)$ is used to measure the dependency of feature w_i with the target domain. If $p(w_i|D_t)/p(w_i|D_s) > 1$, it means that w_i is more dependent to the target domain than that to the source domain, and vice versa. Therefore, some features with higher polarities in D_s but not in D_t will not be selected. Then, we rank the candidate features CW in a decreasing order of $r(w_i|y^s)$. Finally, the former k words were selected to form the shared feature set of SW.

In order to achieve a better performance in domain adaptation, a feature w_i should have a higher weight if it has a

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2925811, IEEE Access

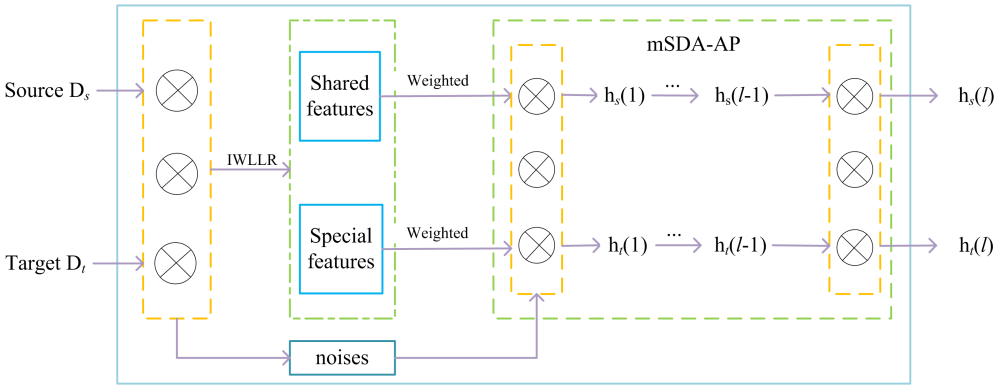


FIGURE 1. The whole framework of our proposed mSDA-AP.

higher polarity in the source domain and a higher frequency in the target domain. In this subsection, we will weight the feature space with α, as shown in Eq. (3).

$$\alpha = \frac{\pi}{|SW|}\sum_{i=1}^{|SW|}\frac{|p_{w_i}^{y^s} - p_{w_i}^{\overline{y^s}}|p(w_i|D_t)}{max\{p_{w_i}^{y^s}, p_{w_i}^{\overline{y^s}}\}max\{p(w_i|D_t), p(w_i|D_s)\}}. \tag{3}$$

And then the shared feature set SW is weighted by α, and the special feature set is weighted by $1-\alpha$. The process is shown in Eq. (4).

$$w_i = \begin{cases} \sin(\alpha)w_i & , w_i \in SW \\ (1-\sin(\alpha))w_i & , w_i \notin SW. \end{cases} \tag{4}$$

B. LEARN STRONGER FEATURE SPACE

In this subsection, we first compute the noise probability adaptively, and then disturb the feature space with the adaptive noise probability to learn stronger feature representation using mSDA.

It is known that the performance of mSDA is sensitive to the noise probability. Obviously, it is not suitable to give a constant value for the noise probability. For different cross-domain tasks, the divergence between source and target domains differs. Therefore, the noise disturbing should also vary. In this paper, we think the value of noise probability is related to the divergence of the source domain and target domain. Thus we compute a distance between source and target domains as the value of noise probability.

More specifically, the distance between source and target domains is measured by the distribution distance of CW between two domains. We define a third mapping space M_i, which is shown in Eq. (5), and then we calculate the KL distance from the source domain D_s to M_i, and the distance from the target domain D_t to the M_i. We take the distance (as shown in Eq. (6)) as the distribution difference value between the two domains. Then our optimal noise probability is computed according to Eq. (7).

$$M_i = \frac{f_{w_i}^s + f_{w_i}^t}{2}; \tag{5}$$

$$JSD = \sum_{i=1}^{|CW|}(f_{w_i}^s log\frac{f_{w_i}^s}{M_i} + f_{w_i}^t log\frac{f_{w_i}^t}{M_i}); \tag{6}$$

$$noises = JSD + \gamma. \tag{7}$$

where $f_{w_i}^s$ and $f_{w_i}^t$ indicate the frequency of candidate word w_i in source and target domains respectively, and γ is a penalty factor. Thus, we can get the value of noise probability according to different cross-domain tasks.

With this noise probability, we disturb the input data (including the source domain and target domain), and then learn a more robust feature space using mSDA. Then we train a classifier based on the robust feature space.

Specially, we take the weighted instances $X \epsilon \Re^{(n_s+n_t)\times d}$ in both source and target domains as the input, where d is the dimension of the feature space. These input data is corrupted with $noises$, which means each feature is set to 0 with the probability $noises$. The corrupted version of X_i is denoted as $\widetilde{X_i}$. Then, the corruption inputs $\widetilde{X_i}$ is reconstructed by minimizing the squared reconstruction loss, as shown in Eq. (8), in which, K is the time of corruption.

$$min\zeta(U) = min\frac{1}{K}\sum_{i=1}^{K}\|X - U\widetilde{X_i}\|^2. \tag{8}$$

According to [19], a unique and optimal solution is yield, and the mapping U can be expressed in closed form as $U = PQ^{-1}$, and P and Q is shown in Eq. (9) and Eq. (10).

$$Q_{ij} = \begin{cases} S_{ij}q_iq_j & , if \quad i \neq j \\ S_{ij}q_i & , if \quad i = j; \end{cases} \tag{9}$$

$$P_{ij} = S_{ij}q_j. \tag{10}$$

In Eq. (9) and Eq. (10), $q = [1-noises, \cdots, 1-noises, 1] \epsilon \Re^{d+1}$, and $S = XX^T$ is the covariance matrix of the uncorrupted data X. S_{ij} and Q_{ij} is the i-th row, j-th column element in matrix S and Q respectively.

Following the same strategy adopted by other autoencoders, mSDA-AP learns the new representations layer by layer

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2925811, IEEE Access

Algorithm 1: mSDA-AP.

Input:
labeled source domain $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$, unlabeled target domain data $D_t = \{(x_j^t)\}_{i=1}^{n_t}$, parameter l and k

Output:
the adaptive classifier $f : X \rightarrow Y$;

Initialize $X^0 = [x_1^s, \cdots x_{n_s}^s, x_1^t, \cdots x_{n_t}^t]$;
Select shared features and weight the X^0 using Eq. (3-4);
Compute the optimal value of noise probability adaptively using Eq. (5-7);
For $r \leftarrow 1$ to l do
    Solve U^r according to Eq.(9-10);
    Computer $h^r = tanh(U^r X^{r-1})$;
    Define $X^r = [X^{r-1}; h^r]$;
end
Build the classifier f on the final union source feature space X_s^r and predict in the target domain X_t^r;
return the adaptive classifier f

greedily. Moreover, mSDA-AP uses the same strategy as Chen et al. [19], mSDA-AP does not need an end-to-end fine-tuning with the BP algorithm and can be computed in a closed form with a less time cost. To apply mSDA-AP to domain adaptation, we first learn feature representation in an unsupervised way on source domain and target domain data. Then the output of all layers, after squashing function $tanh(\cdot)$, are combined with original features h^0 to form new representations. Finally a linear SVM is trained on the new features, it only uses one epoch to train the last layer using the linear SVM. The whole process of our mSDA-AP model is summarized in Algorithm 1.

IV. EXPERIMENTS

In this section, we perform a comprehensive experimental study on domain adaption problem to evaluate both the effectiveness and scalability of the proposed mSDA-AP models, including sentiment polarity prediction, email spam filtering and Office-Caltech10 dataset.

A. DATASETS

Amazon Review Dataset[1]. The Amazon review Dataset has been widely adopted as the benchmark dataset for domain adaptation and sentiment analysis. It contains a collection of product reviews from Amazon.com about four product domains: Books (B), DVDS (D), Electronics (E) and Kitchen appliances (K). And we can construct 12 cross-domain tasks including B→D, E→K and so on. In this dataset, each review is assigned a sentiment label, -1 (negative review) or +1 (positive review), based on the rating score. Reviews with rating 4 or 5 are labeled as positive, and reviews with rating 1 or 2 are labeled as negative. For each domain, there are 1000 positive and 1000 negative reviews. As mSDA-AP and mSDA focus on feature learning, we use the raw bag-of-words (bow) features as their input, and adopt the setting of paper [19] with the 5000 most frequent common features selected for each adaptation task with TF-IDF as weight.

[1]http://www.cs.jhu.edu/ mdredze/datasets/sentiment/.

TABLE 1. Description of Office-Caltech10 dataset.

Dataset	Type	Instances	Features	Classes
Caltech-256	Object	1123	800	10
AMAZON	Object	958	800	10
Webcam	Object	295	800	10
DSLR	Object	157	800	10

Email Spam Filtering Dataset[2]. It is from the ECML/PKDD 2006 discovery challenge. In this dataset, 4000 labeled training samples were collected from publicly available sources (source domain), with half of them are spam and the other half are non-spam. The testing samples were collected from 3 different user inboxes U0, U1 and U2, which is treated as target domain, each of which consists of 2500 samples. Thus, we construct 3 cross-domain tasks including Pub→U0, Pub→U1 and Pub→U2. As on Amazon review dataset, we also chose the 5000 most frequent terms as features.

Office-Caltech10 dataset. Office-Caltech10 contains 10 object categories from an office environment in 4 image domains: Amazon (Am), Webcam (We), DSLR (DS), and Caltech256 (Ca). There are 8 to 151 samples per category per domain, and there are 2,533 images in total. Details of the Office-Caltech10 dataset are described in Table 1. In our experiment, we construct 3 cross-domain tasks Ca→Am, Ca→We, Ca→DS. Because our algorithm is designed for binary classification problems, on Office-Caltech10 dataset, we computer α by treating categories 1-5 as positive samples and categories 6-10 as negative samples.

B. COMPARED METHODS

We compare our approach with several state-of-the-art methods to test the effectiveness of our work.

(1) Support Vector Machine (SVM). It is the traditional SVM classifier without transfer learning. We train a linear SVM on the raw TF-IDF representation of the labeled source data and test it on the target domain.

(2) Marginalized Stacked Denoising Autoencoders (mSDA)[3] [19]. It is a transfer learning algorithm based on stacked denoising autoencoders. Since mSDA is better than SDA, we only provide our comparisons with mSDA. The performance metric is classification accuracy.

(3) Regularization Denosing Autoencoders (MDA-TR)[4] [20]. It is an appropriate regularization for the denoising autoencoders, in particular for MDA. With the aim to make

[2]http://www.ecmlpkdd2006.org/challenge.html.
[3]http://www.cse.wustl.edu/∼mchen.
[4]http://github.com/sclincha/xrce_msda_da_regularization.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2925811, IEEE Access

TABLE 2. Performance (accuracy %) on Amazon review dataset.

task	SVM	MDA-TR	mSDA	$\ell_{2,1}$-SRA	DNFC	WDGRL	DTFC	mSDA-AP
B→D	80.65	83.32	84.36	84.72	83.83	83.05	56.50	**84.77**
B→E	72.58	75.29	76.43	76.06	81.17	**83.28**	60.01	79.00
B→K	75.44	82.86	82.46	84.32	84.29	85.45	60.56	**86.21**
D→B	78.59	83.20	83.49	**83.65**	83.07	80.72	55.45	83.58
D→E	72.82	80.00	82.05	80.64	83.54	**83.58**	58.85	83.47
D→K	76.40	86.14	87.38	87.55	87.52	86.24	58.32	**88.26**
E→B	70.48	78.66	79.37	78.99	78.90	77.22	56.28	**80.47**
E→D	72.36	79.14	79.34	80.12	79.56	78.28	56.25	**80.28**
E→K	86.19	86.16	88.46	88.07	**88.73**	88.16	75.04	88.36
K→B	71.49	79.06	79.13	79.35	78.84	77.16	62.37	**80.20**
K→D	74.40	79.39	79.39	79.95	80.15	79.89	61.52	**80.56**
K→E	84.49	87.33	87.38	87.64	**87.94**	86.29	74.46	87.45
Avg	76.32	81.71	82.44	82.59	83.13	82.43	61.30	**83.55**

TABLE 3. Performance (accuracy %) on Spam dataset.

task	SVM	MDA-TR	mSDA	$\ell_{2,1}$-SRA	DNFC	WDGRL	DTFC	mSDA-AP
Public→U0	72.79	82.55	78.00	81.99	83.84	85.67	81.96	90.48
Public→U1	73.94	85.87	85.12	85.99	85.20	88.62	85.12	92.67
Public→U2	78.64	85.92	90.44	91.36	85.24	**95.76**	80.56	93.64
Avg	75.12	84.78	84.52	86.45	84.76	89.90	82.41	**92.26**

TABLE 4. Performance (accuracy %) on Office-Caltech10 dataset.

Dataset	SVM	MDA-TR	mSDA	$\ell_{2,1}$-SRA	DNFC	WDGRL	DTFC	mSDA-AP
Ca→Am	51.98	44.15	54.70	54.80	52.19	55.22	37.68	**57.52**
Ca→We	34.92	41.69	37.29	37.29	42.37	42.37	32.54	**43.05**
Ca→DS	43.31	45.68	45.86	47.13	49.04	48.41	33.76	**52.87**
Avg	43.40	43.90	45.95	46.41	47.87	48.67	34.66	**51.14**

TABLE 5. Running time (seconds) of training the last layer using linear SVM on Amazon review dataset.

task	SVM	MDA-TR	mSDA	$\ell_{2,1}$-SRA	DNFC	WDGRL	DTFC	mSDA-AP
B→D	4.10	35.71	227.52	274.23	152.61	35.44	35.42	195.08
B→E	4.20	31.36	188.34	265.83	154.13	35.54	35.68	158.49
B→K	4.22	31.01	197.05	274.59	152.60	35.35	35.40	146.92
D→B	4.02	31.29	194.79	255.38	144.66	35.03	35.18	153.00
D→E	3.97	32.32	189.44	269.88	144.42	35.20	35.20	159.85
D→K	3.94	31.84	193.21	277.39	156.64	35.29	35.26	158.75
E→B	3.06	31.71	180.22	256.97	141.51	34.86	34.22	120.04
E→D	3.02	31.50	185.02	254.86	140.86	34.43	34.80	114.11
E→K	3.02	27.39	195.26	244.88	140.52	32.33	32.42	141.93
K→B	2.78	31.34	184.09	251.19	141.12	35.19	34.65	144.20
K→D	2.74	31.80	192.08	259.03	137.18	35.09	35.34	130.91
K→E	2.73	25.62	200.86	238.67	133.63	33.61	32.59	107.70

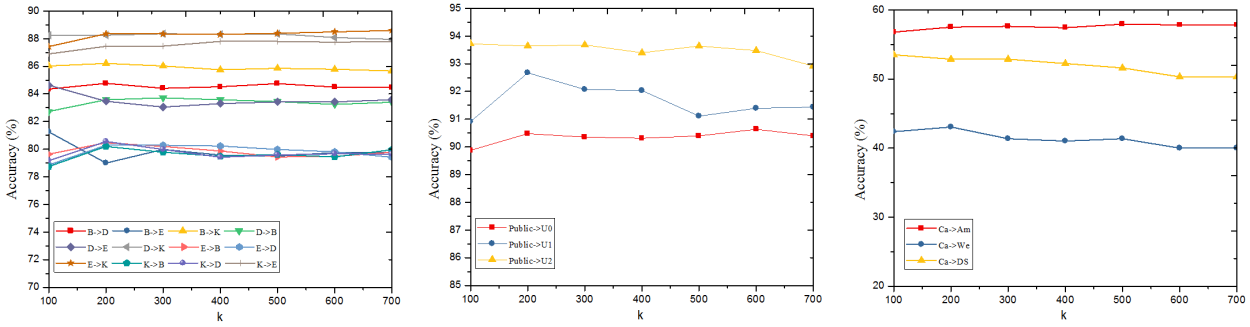
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2925811, IEEE Access

TABLE 6. Running time (seconds) of training the last layer using linear SVM on Spam dataset.

task	SVM	MDA-TR	mSDA	$\ell_{2,1}$-SRA	DNFC	WDGRL	DTFC	mSDA-AP
Public→U0	6.63	44.13	218.41	167.51	157.33	88.24	191.53	178.38
Public→U1	6.52	35.67	177.38	164.37	158.17	83.72	180.79	143.11
Public→U2	6.52	31.16	161.39	137.54	144.08	70.95	158.08	153.00

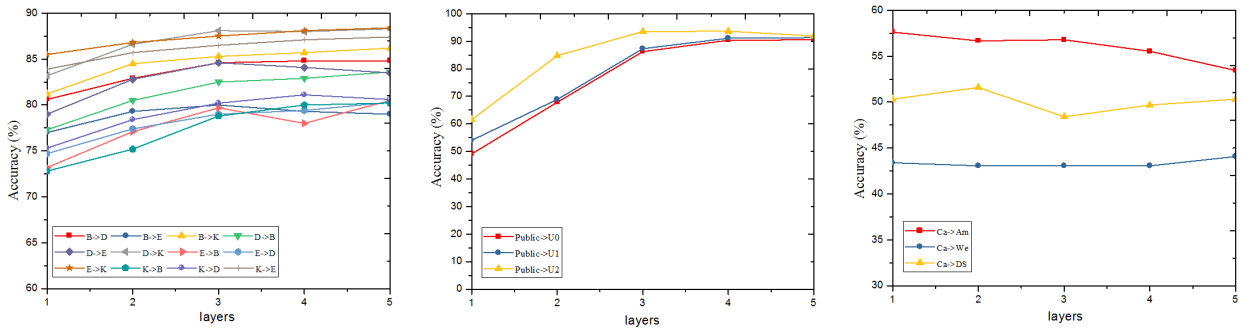
TABLE 7. Running time (seconds) of training the last layer using linear SVM on Office-Caltech10 dataset.

Dataset	SVM	MDA-TR	mSDA	$\ell_{2,1}$-SRA	DNFC	WDGRL	DTFC	mSDA-AP
Ca→Am	6.63	1.91	15.83	12.01	8.12	8.83	7.91	5.83
Ca→We	6.52	1.93	15.03	12.83	8.11	8.81	7.84	6.04
Ca→DS	6.52	1.97	14.84	12.63	8.18	8.80	7.88	6.17



(a) Amazon review (b) Spam (c) Office-Caltech10

FIGURE 2. Accuracy of our mSDA-AP with different numbers of shared features on 3 datasets.



(a) Amazon review (b) Spam (c) Office-Caltech10

FIGURE 3. Accuracy of our mSDA-AP with different numbers of layers on 3 datasets.

source data resemble the target data, it is easy to address domain adaptation.

(4) $\ell_{2,1}$-Norm Stacked Robust Autoencoders ($\ell_{2,1}$-SRA) [23]. It is a simple combination of statistics tool and deep arcitecture. And it can learn effective representations.

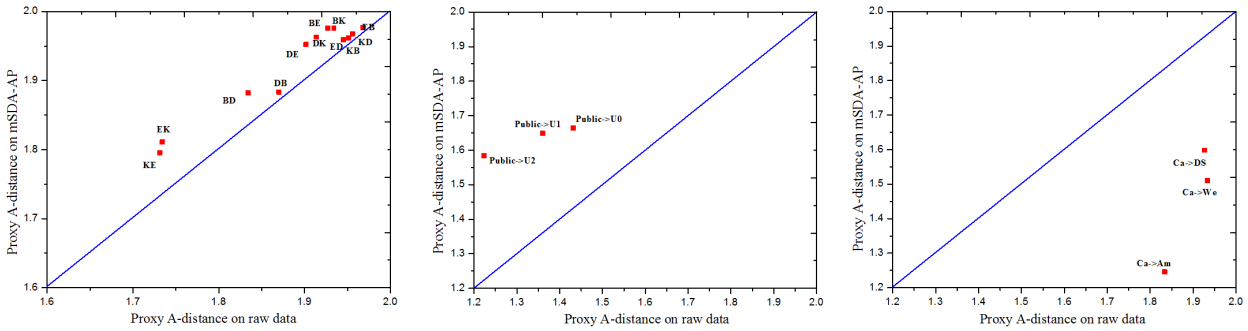
(5) Deep Nonlinear Feature Coding framework (DNFC) [35] introduce kernelization and MMD into mSDA to obtain nonlinear deep feature representation. In DNFC, we use SVM as the basic classifier.

(6) Wasserstein Distance Guided Representation Learning (WDGRL)[5] [34], which is an adversarial method and it learns domain invariant feature representations by minimizing the wasserstein distance between source domain and target domain.

(7) Feature analysis of marginalized stacked denoising autoenconder (DTFC) [31], which extracted effective feature representations by corrupting the raw input data with multinomial dropout noise.

[5]https://github.com/RockySJ/WDGRL.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2925811, IEEE Access



(a) Amazon review (b) Spam (c) Office-Caltech10

FIGURE 4. Proxy-A-distance on different datasets.

Parameter setting: In our experiments, we set the number of shared features k as 200 on all datasets, and set the number of layers l as 5, 4, 2 Amazon, Spam and Office-Caltech10 datasets, γ are 0.5, 0.7, 0.4 for Amazon Review, Spam and Office-Caltech10 dataset respectively. In the method of mSDA, the best parameters will be shown in the experiment. For MDA-TR and WDGRL, we use the default parameters as reported in [20] and [34] respectively. For $\ell_{2,1}$-SRA, the number of layers are 5, 3 and 2 for Amazon Review, Spam and Office-Caltech10. The parameter α is 2 for Amazon Review, 20 for spam, and 10 for Office-Caltech10. And the parameter λ are 0.5, 10, 10 for Amazon Review, Spam and Office-Caltech10. In the method of DNFC, we set θ as 1000 on all datasets, and set layer as 3, 2, 3 for Amazon, Spam, Office-Caltech10 respectively. For DTFC, we set θ as 1000 on all datasets, and set layer as 3, 3, 3 for Amazon, Spam, Office-Caltech10 respectively. All experimental results are conducted on a PC with Intel(R) i7-7700T, 2.9 GHz CPU, and 16GB memory.

C. CLASSIFICATION ACCURACY

Tables 2 3 4 show the accuracy of results on three datasets. The best results in each task have been marked in bold. And SVM is a traditional approach, the accuracy is the lower limits. We have the following observations from experimental results.

The first one is that our mSDA-AP performs better than MDA-TR and mSDA, DNFC, DTFC, which shows the superiority of applying adaptive noise probability. In mSDA and DNFC, the best value of noise probability is set by experiment results, MDA-TR sets noise as 0.9, and DTFC used the second-order statistic of features to obtain the noise probabilities. If the noise probability is equal to 1, which means the feature is removed. In DTFC, the performance is depressing because of our data dimension is very high and the noise probability is closed to 1 for each feature. While mSDA-AP computes the noise probability adaptively. Additionally, our mSDA-AP also has a little superior to $\ell_{2,1}$-SRA and WDGRL on three datasets, which also indicates the importance of weighting the feature space to learn a good representation feature in transfer learning. It indicates that the invariant features is important for cross-domain classification. In a word, our mSDA-AP is superior to all the other baselines in these three datasets.

Tables 5 6 7 display the running time of training the last layer using the linear SVM. Because the original input and output from all layers are concatenated to form the final feature representations, thus we use the final feature representations to train the linear SVM. With the number of layer increases, the feature dimension of the final feature representations increases, and the training time will increase. From Tables 5 6 7, we observe that mSDA-AP is faster than mSDA and $\ell_{2,1}$-SRA when using the same stacked layers. The stacked layers are 4 and 3 in DTFC and mSDA-AP respectively, we find mSDA-AP is faster than DTFC, which can be seen from Table 6. Since the proposed mSDA-AP adopts the weighting strategy, the weight of some unimportant features will be close to 0, so mSDA-AP is relatively faster when using the same number of stacked layers.

D. PARAMETER SENSITIVITY

In this subsection, we conduct empirical parameter sensitivity analysis, which validates that mSDA-AP can achieve optimal performance under wide range of parameter values. There are two parameters in our method: the number of domain shared number k and the number of layers l. We study the effects of two parameters on three datasets. When we change one parameter, the rest one parameter is fixed.

The number of domain shared features k: We run mSDA-AP varying with the values of k. Theoretically, k should be neither too small nor too large. If k is too small, some shared features with a higher polarity are not chosen, which will result in the sparsity in classification. However, if k is too large, some features those are not important in classification will be selected. We plot the classification accuracy with regard to different values of k in Figure 2. As shown in Figure 2, the best value of k falls into $k \in$ [200,400] on Amazon review dataset, $k \in$ [200,500] on Spam dataset, $k \in$ [100,200] on Office-Caltech10 dataset respectively.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2925811, IEEE Access

The number of layers: We run mSDA-AP varying with the values of l, the number of layers for mSDA model to validate the sensitiveness of our mSDA-AP to the parameter l. We plotted the accuracies with different numbers of layers on the same datasets as above, which are shown in Figure 3. We can see that the accuracy of our mSDA-AP increases as the number of layers increases totally. On Amazon review and Spam datasets, the number of layers usually performs best when $l = 5$. On Office-Caltech10 dataset, the number of layers usually performs best when $l = 2$.

E. TRANSFER DISTANCE

Ben-David et al. [30] suggested a proxy-A-distance to measure the similarity between source domain and target domain. In practice, it is impossible to [30] suggest the proxy-A-distance as a measure of how different two domains are from each other. A proxy-A-distance is defined as $d_A = 2(1 - 2\epsilon)$, where ϵ is the generalization error of a classifier (a linear SVM in our case) trained on the binary classification problem to discriminate source domain and target domain. Here we use this proxy-A-distance to measure the performance of cross-domain task. The value of proxy-A-distance increasing after cross-domain task means that the new representations are benefit to domain classification tasks. As we show in Figure 4, the points of Amazon and spam datsets increase according to the blue line, while it decreases on Office-Caltech10 dataset. Although the proxy-A-distance with new representation decreases on Office-Caltech10 dataset, mSDA-AP achieves promising results on Office-Caltech10 dataset. We can get the same result as mentioned in reference [23], the proxy-A-distance might become smaller or bigger after feature learning.

V. CONCLUSION

In this paper, we proposed a transfer learning algorithm for learning feature representation with a deep learning architecture. mSDA is a remarkable model to learn feature representation. In order to avoid the sensitiveness of noise probability in mSDA, we proposed a marginalized stacked denoising autoencoder with adaptive noise probability for domain adaptation (mSDA-AP), in which, the value of noise probability is calculated according to the distance between the source and target domains. Our algorithm can compute the right noise probability according to different transfer learning tasks, thus it will benefit the classification for target domain. And a series of experimental results demonstrate the effectiveness of our proposed mSDA-AP. In the future, we will focus on the change of invariant features in disturbing step.

REFERENCES

[1] J. Blitzer, R.T. McDonald, and F. Pereira, "Domain Adaptation with Structural Correspondence Learning," in Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, Jul. 2006, pp. 120-128.

[2] S. J. Pan, X. Ni, J. Sun, Q. Yang, and Z. Chen, "Cross-domain sentiment classification via spectral feature alignment," in Proceedings of the 19th International Conference on World Wide Web, Apr. 2010, pp. 751-760.

[3] Y. H. Zhang, X. G. Hu, P. P. Li, L. Li, and X. Wu, "Cross-domain sentiment classification-feature divergence, polarity divergence or both?," Pattern Recognition Letters., vol. 65, pp. 44-50, 2015.

[4] J. Y. Chen, Y. T Yang, K. K. Hu, Q. X, Y. L, and C. Yang, "Multiview Transfer Learning for Software Defect Prediction," IEEE Access., vol. 7, pp. 8901-8916,2019.

[5] H. D. III, "Frustratingly Easy Domain Adaptation," CoRR, abs/0907.1815, 2009.

[6] H. D. III, D. Marcu, "Domain Adaptation for Statistical Classifiers," CoRR, abs/1109.6341, 2011.

[7] S. Gao, H. Z. Li, "A cross-domain adaptation method for sentiment classification using probabilistic latent analysis," in Proceedings of the 20th ACM Conference on Information and Knowledge Management, Oct. 2011, pp. 1047-1052.

[8] L. H. Li, X. M. Jin, and M. S. Long, "Topic Correlation Analysis for Cross-Domain Text Classification," in Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, Jul. 2012, pp. 998-1004.

[9] Y. Kim, "Convolutional Neural Networks for Sentence Classification," in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Oct. 2014, pp. 1746-1751.

[10] B. Du, W. Xiong, J. Wu, L. F. Zhang, L. P. Zhang, and D. C Tao, "Stacked Convolutional Denoising Auto-Encoders for Feature Representation," IEEE Trans.Cybernetics., vol. 47, no. 4, pp. 1017-1027, 2017.

[11] F. Z. Zhuang, X. H. Cheng, P. Luo, S. J. Pan, and Q. He, "Supervised Representation Learning: Transfer Learning with Deep Autoencoders," in Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, Jul. 2015, pp. 4119-4125.

[12] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, "Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions," in Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Jul. 2011, pp. 151-161.

[13] L. Dong, F. Wei, C. Q. Tan, D. Y. Tang, M. Zhou, and K. Xu, "Adaptive Recursive Neural Network for Target-dependent Twitter Sentiment Classification," in Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Jun. 2017, pp. 49-54.

[14] D. Y. Tang, B. Qin, and T. Liu, "Document Modeling with Gated Recurrent Neural Network for Sentiment Classification," in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Sep. 2015, pp. 1422-1432.

[15] W. Rawat, Z. H. Wang, "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review," Neural Computatio., vol. 29, no. 9, pp. 2352-2449, 2017.

[16] D. Bollegala, D. J. Weir, and J. A. Carroll, "Cross-Domain Sentiment Classification Using a Sentiment Sensitive Thesaurus," IEEE Trans. Knowl. Data Eng., vol. 25, no. 8, pp. 1719-1731, 2013.

[17] X. Glorot, A. Bordes, and Y. Bengio, "Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach," in Proceedings of the 28th International Conference on Machine Learning, Jul. 2011, pp. 513-520.

[18] S. X. Cao, N. Yang, and Z. Z. Liu, "Online news recommender based on stacked auto-encoder," in IEEE/ACIS International Conference on Computer and Information Science, May. 2017, pp. 721-726.

[19] M. M. Chen, Z. X. E. Xu, K. L. Q. Weinberger, and F. Sha, "Marginalized Denoising Autoencoders for Domain Adaptation," in Proceedings of the 29th International Conference on Machine Learning, Jul. 2012.

[20] S. Clinchant, G. Csurka, and B. Chidlovskii, "A Domain Adaptation Regularization for Denoising Autoencoders," in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Aug. 2016, pp. 26-31.

[21] Y. Ganin, V. S. Lempitsky, "Unsupervised Domain Adaptation by Backpropagation," in Proceedings of the 32nd International Conference on Machine Learning, Jul. 2015, pp. 1180-1189.

[22] G. Csurka, B. Chidlovskii, S. Clinchant, and S. Michel, "An Extended Framework for Marginalized Domain Adaptation," in CoRR, abs/1702.05993, 2017.

[23] W. H. Jiang, H. C. Gao, F. Chung, and H. Huang, "The l2, 1-Norm Stacked Robust Autoencoders for Domain Adaptation," in Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Feb. 2016, pp. 1723-1729.

[24] G. Xue, W. Y. Dai, Q. Yang, and Y. Yu, "Topic-bridged PLSA for cross-domain text classification," in Proceedings of the 31st Annual International

ACM SIGIR Conference on Research and Development in Information Retrieval, Jul. 2008, pp. 627-634.
[25] Y. Yang, J. Eisenstein, "Fast Easy Unsupervised Domain Adaptation with Marginalized Structured Dropout," in Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Jun. 2014, pp. 538-544.
[26] Y. Ganin, V. S. Lempitsky, "Neural Structural Correspondence Learning for Domain Adaptation," in Proceedings of the 21st Conference on Computational Natural Language Learning, Aug. 2017, pp. 400-410.
[27] Y. Zhu, X. G. Hu, Y. H. Zhang, P. P. Li, "Transfer learning with stacked reconstruction independent component analysis," Knowl.-Based Syst., vol. 152, pp. 100-106, 2018.
[28] D. P. Kingma, M. Welling, "Auto-Encoding Variational Bayes," in CoRR, abs/1312.6114, 2013.
[29] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic Backpropagation and Approximate Inference in Deep Generative Models," in Proceedings of the 31th International Conference on Machine Learning, Jun. 2014, pp. 1278-1286.
[30] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of Representations for Domain Adaptation," in Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Dec. 2006, pp. 137-144.
[31] P. F. Wei, Y. Y. Ke and C. K. Goh, "Feature Analysis of Marginalized Stacked Denoising Autoenconder for Unsupervised Domain Adaptation," IEEE Transactions on Neural Networks and Learning Systems., vol. 99, pp. 1-14, 2018.
[32] P. Cao, S. L. Zhang, and J. Tang, "reprocessing-Free Gear Fault Diagnosis Using Small Datasets With Deep Convolutional Neural Network-Based Transfer Learning," IEEE Access., vol. 6, pp. 26241-26253, 2018.
[33] W. H. Jiang, H. C. Gao, W. Lu, W. Liu, F. L. Chung, and H Huang, "Stacked Robust Adaptively Regularized Auto-Regressions for Domain Adaptation," IEEE Trans. Knowl. Data Eng., vol. 31, pp. 561-574, 2019.
[34] J. Shen, Y. R. Qu, W. N. Zhang, and Y. Yu, "Wasserstein Distance Guided Representation Learning for Domain Adaptation," in Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), Feb. 2018, pp. 4058-4065.
[35] P. F. Wei, Y. Y. Ke, and C. K. Goh, "Deep Nonlinear Feature Coding for Unsupervised Domain Adaptation," in Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, Jul. 2016, pp. 2189-2195.
[36] S. Yang, Y. H. Zhang, Y. Zhu, P. P. Li and X. G. Hu, "Representation learning via serial autoencoders for domain adaptation," Neurocomputing., vol. 351, pp. 1-9, 2019.
[37] J. T. Zhou, H. Zhang, D. Jin, H. Zhu, R. S. M. Goh and K. Kenneth, "DATNet: Dual Adversarial Transfer for Low-resource Named Entity Recognition," 2019.
[38] J. T. Zhou, H. Zhao, X. Peng, M. Fang, Z. Qin, and R. S. M. Goh, "Transfer hashing: From shallow to deep," IEEE transactions on neural networks and learning systems, vol. 99, pp. 1-11, 2019.
[39] J. T. Zhou, S. J. Pan, I. W. Tsang and S. S. Ho, "Transfer learning for cross-language text categorization through active correspondences construction," in Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Feb. 2018, pp. 2400-2406.
[40] X. Lan, S. Zhang, P. C. Yuen and R. Chellappa, "Learning Common and Feature-Specific Patterns: A Novel Multiple-Sparse-Representation-Based Tracker," IEEE Trans. Image Processing, vol. 27, pp. 2022-2037, 2018.
[41] X. Lan, M. Ye, R. Shao, B. Zhong, P. C. Yuen and H. Zhou, "Learning Modality-Consistency Feature Templates: A Robust RGB-Infrared Tracking System," IEEE Transactions on Industrial Electronics, DOI:10.1109/TIE.2019.2898618.
[42] J. T. Zhou, I. W. Tsang, S. J. Pan and M. Tan, "Multi-class Heterogeneous Domain Adaptation," Journal of Machine Learning Research, vol. 20, pp. 57:1–57:31, 2019.

YUHONH ZHANG is an associate professor at Hefei University of Technology, China. She received her B.S., M.S. and Ph.D. degrees from Hefei University of Technology in 2001, 2004, 2011 respectively. Her research interests are in transfer learning, data stream classification and data mining.

SHUAI YANG is currently working toward the PhD degree at Hefei University of Technology, China. He received his B.S. and M.S. degrees from Hefei University of Technology in 2016 and 2019 respectively. His research interests are in data mining, transfer learning.

PEIPEI LI is currently an associate professor at Hefei University of Technology, China. She received her B.S., M.S. and Ph.D. degrees from Hefei University of Technology in 2005, 2008, 2013 respectively. She was a research fellow at Singapore Management University from 2008 to 2009. She was a student intern at Microsoft Research Asia between Aug. 2011 and Dec. 2012. Her research interests are in data mining and knowledge engineering.

XUEGANG HU is a professor at the School of Computer Science and Information Engineering, Hefei University of Technology, China. He received his B.S. degree from the Department of Mathematics at Shandong University, China, and his M.S. and Ph.D. degrees in Computer Science from the Hefei University of Technology, China. He is engaged in research in data mining and knowledge engineering.

HAO WANG received the B.S. degree from the Department of Electrical Engineering and Automation, Shanghai Jiao Tong University, Shanghai, China, and the M.S. and Ph.D. degrees in computer science from the Hefei University of Technology, Hefei, China. He is a Professor with the School of Computer Science and Information Engineering, Hefei University of Technology. His current research interests include artificial intelligence and robotics and knowledge engineering.