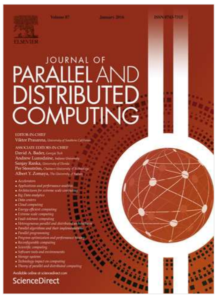


Heterogeneous domain adaptation network based on autoencoder

Xuesong Wang, Yuting Ma, Yuhu Cheng, Liang Zou, Joel.
J.P.C. Rodrigues

PII: S0743-7315(17)30192-2
DOI: http://dx.doi.org/10.1016/j.jpdc.2017.06.003
Reference: YJPDC 3693

To appear in: *J. Parallel Distrib. Comput.*

Received date : 28 February 2017
Revised date : 3 May 2017
Accepted date : 3 June 2017

Heterogeneous Domain Adaptation Network Based on Autoencoder

Xuesong Wang[1], Yuting Ma[1], Yuhu Cheng[1]*, Liang Zou[2], Joel. J. P. C. Rodrigues[3,4,5,6]

1. School of Information and Control Engineering, China University of Mining and Technology, 221116, Xuzhou, Jiangsu, China;

2. Department of Electrical and Computer Engineering, University of British Columbia, V6T 1Z4, Vancouver, British Columbia, Canada;

3. National Institute of Telecommunications (Inatel), 37540-000 Santa Rita do Sapucaí - MG, Brazil;

4. Instituto de Telecomunicações, Universidade da Beira Interior, 6201-001 Covilhã, Portugal;

5. ITMO University, 191002 St. Petersburg, Russia

6. University of Fortaleza (UNIFOR), 60811-905 Fortaleza-CE, Brazil

Abstract: Heterogeneous domain adaptation is a more challenging problem than homogeneous domain adaptation. The transfer effect is not ideal caused by shallow structure which cannot adequately describe the probability distribution and obtain more effective features. In this paper, we propose a heterogeneous domain adaptation network based on autoencoder, in which two sets of autoencoder networks are used to project the source-domain and target-domain data to a shared feature space to obtain more abstractive feature representations. In the last feature and classification layer, the marginal and conditional distributions can be matched by empirical maximum mean discrepancy metric to reduce distribution difference. To preserve the consistency of geometric structure and label information, a manifold alignment term based on labels is introduced. The classification performance can be improved further by making full use of label information of both domains. The experimental results of 16 cross-domain transfer tasks verify that HDANA outperforms several state-of-the-art methods.

Keywords: heterogeneous domain adaptation; autoencoder; maximum mean discrepancy; manifold alignment

1 Introduction

In various recognition and classification applications, it is unrealistic or even impossible to collect sufficient labeled data [1-2]. Domain adaptation (DA), which aims at digging the potential knowledge in the source domain to model the data in target domain, has attracted increasing attention in the field of machine learning. By knowledge transfer, the prediction model for the target domain lacking of labeled data is guided by the extracted knowledge from the source domain which owns plenty of labeled data. Nowadays there are a lot of domain adaptation methods [3-7], all of which assume that the data in different domains are represented in the same feature space with the same dimensionality. However, for more complex situations where the feature dimensionality and distribution are different across the source and target domains, these DA methods fail to work. In the light of this, the more challenging heterogeneous domain adaptation (HDA) is investigated in this paper.

The main difficulty of HDA is that the source and target-domain data are in different feature spaces, and only a small number of labeled samples in target domain are available. Therefore, the prediction model trained in source domain cannot be directly applied to the target domain. In order to solve this problem, many HDA methods have been proposed, which can be divided into two categories: the domain transformation and the shared subspace learning. The domain transformation method transforms samples of one domain to another through the study of asymmetric transformation matrices, such as the asymmetric regularized cross-domain transformation (ARC-t) by Kulis et al. [8], in which the non-symmetric nonlinear matrices are learned. This kind of methods has a couple of limitations. On one hand, in feature learning, the objective function of a discriminative classifier is not optimized directly. On another hand, it is not suitable for domains with large number of samples due to the concern of high computational complexity. Therefore, Hoffman et al. [9] proposed the method of max-margin domain transforms (MMDT) to enhance computational efficiency, in which the classifier is trained by transforming the target-domain samples as one part to source domain through linear transformation. Meanwhile, both the transformation and classifier parameters are optimized jointly. Considering the fact that small amount of labeled samples in target domain may contain noise and it may trigger the estimation failure of ARC-t, the sparse heterogeneous feature representation (SHFR) was proposed by Zhou et al. [10], in which the weight of classifier in source domain is transformed to target domain by constructing the sparse feature transformation matrix. The semi-supervised kernel matching for heterogeneous domain adaptation (SSKMDA) method was proposed by Xiao et al. [11], in which a prediction function is learned on the labeled source-domain data while the target-domain data points are transformed to similar source points by kernel matching methods. Tsai et al. [12] proposed a cross-domain landmark selection (CDLS) method to identify the contributions of each landmark when matching cross-domain class-conditional data distributions in the derived feature space. However, the manifold alignment term does not introduced into the objective function of CDLS. Further as an improvement of CDLS, Tsai et al. [13] proposed label and structure-consistent unilateral projection (LS-UP) that transforms source-domain data to the target domain, with the goal of matching cross-domain data distribution and preserving data structure after projection by imposing a class-wise locality constraint on the projected source-domain data. However, the structure consistency between target-domain data does not be taken into consideration in LS-UP. As well, the label consistency between source-domain and target-domain samples is ignored.

The shared subspace learning method projects domain data from different feature spaces to one shared latent subspace. An example is heterogeneous spectral mapping (HeMap) by Shi et al. [14] with no label information considered. The source and target feature domains are unified by spectral embedding. Relevant samples in source domain are selected and relationships between different output spaces are modeled by Bayesian method. Chen et al. [15] proposed a HDA method of transfer neural trees (TNT) which jointly solves cross-domain feature mapping, adaptation, and classification in a neural network-based architecture. The structure consistency between target-domain data is preserved by introducing an embedding loss in the layer of feature mapping. But on the other hand, the manifold and label relationships of data in both source and target domains do not be taken into consideration in TNT. Moreover, the distribution difference between domains is not minimized. The domain adaptation using manifold alignment (DAMA) was proposed by Wang and Mahadevan [16] by projecting both domain data to a new feature space. In the new space, samples with the same labels are close to while simultaneously those of different labels are away from each other, and the topology of each input domain is preserved. However, this method is only applicable for data that have strong manifold structures. The heterogeneous feature augmentation (HFA) was proposed by Duan et al. [17], in which samples of both domains are projected to a shared subspace to measure data similarity, and the transformed data are augmented by two new feature mapping functions. However, the complicated semi-definite program problem should be solved. Further HFA was extended by Li et al. [18] into the semi-supervised HFA (SHFA), where the unlabeled training data in target domain are integrated for SHFA, and the target classifier is learned while simultaneously the labels of unlabeled target-domain samples are predicted. The transfer discriminant analysis of canonical correlations (HTDCC) method was proposed by Wu et al. [19], in which a couple of optimal projection matrices are obtained by respectively minimizing and maximizing typical correlations of the inter-class and intra-class samples, and then data of both domains are projected to the shared feature space to realize knowledge transfer. By exploiting both source-domain and target-domain data and simultaneously learning the projection matrices and the prediction models, Xiao et al. [20] proposed the semi-supervised subspace co-projection (SCP). In recent years, considering the fact that the maximum mean discrepancy (MMD) can effectively measure the distribution difference between domains, Hsieh et al. [21] proposed the generalized joint distribution adaptation (G-JDA). A couple of feature transformation matrices are learned, and the domain data are projected to the domain-invariant feature space while the marginal and conditional distributions are matched by MMD metric to have an improved classification performance.

However, all the aforementioned HDA methods are based on the shallow structure. Therefore, they cannot effectively perform curve fitting and cannot get better feature representation. In addition, they do not take into account the distribution matching of the cross-domain data or the consistency of geometric structure and label. Motivated by recent progress of deep learning and manifold learning in many fields, a heterogeneous domain adaptation network based on autoencoder (HDANA) is proposed in this paper. Specifically, our main contributions in this work are outlined below: 1) We applied two autoencoders (DAs) to project the source-domain and target-domain data to a shared feature space. Thus, a more abstractive feature representation can be obtained through the multi-layer nonlinear mapping mechanism; 2) We used MMD metric to simultaneously match the marginal and conditional distributions, which can effectively narrow the distribution difference; 3) We introduced the manifold alignment term containing geometric, similarity and dissimilarity terms into the objective function. The geometric term is used to preserve the structure consistency while the similarity and dissimilarity terms are used to preserve the label consistency; 4) We conducted comprehensive experiments on 16 groups of cross-domain transfer tasks. The experimental results demonstrate the effectiveness and superiority of the proposed HDANA to other alternatives.

The rest of this paper is organized as follows. The details of the proposed HDANA are described in Section 2. Experimental results on 16 groups of cross-domain transfer tasks constructed on the cross-domain object recognition dataset Office+Caltech-256 and the cross-lingual text categorization dataset are reported in Section 3, followed by a conclusion in Section 4.

2 Heterogeneous Domain Adaptation Network Based on Autoencoder

2.1 HDANA network structure

Assume source-domain dataset is $D_s = \{\boldsymbol{X}_s, \boldsymbol{Y}_s\} = \{\boldsymbol{x}_{si}, y_{si}\}|_{i=1}^{n_s}$ and target-domain dataset is $D_t = \{\boldsymbol{X}_t, \boldsymbol{Y}_t\} = \{\boldsymbol{x}_{tj}, y_{tj}\}|_{j=1}^{n_t}$, where $\boldsymbol{x}_{si} \in \mathrm{R}^{d_s^{(1)}}$, $\boldsymbol{x}_{tj} \in \mathrm{R}^{d_t^{(1)}}$, $d_s^{(1)} \neq d_t^{(1)}$. Define $L = \{1, \cdots, C\}$ as the label dataset and $y_{si}, y_{tj} \in L$ holds true. For semi-supervised HDA, D_t are composed of labeled data subset $D_l = \{\boldsymbol{X}_l, \boldsymbol{Y}_l\} = \{\boldsymbol{x}_{lj}, y_{lj}\}|_{j=1}^{n_l}$ and unlabeled data subset $D_u = \{\boldsymbol{X}_u, \boldsymbol{Y}_u\} = \{\boldsymbol{x}_{uj}, y_{uj}\}|_{j=1}^{n_u}$, where $\boldsymbol{Y}_l$ is known and $\boldsymbol{Y}_u$ are labels to be predicted. Only a small number of labeled target-domain data are available, and each class in both domains is assumed to contain the labeled data.

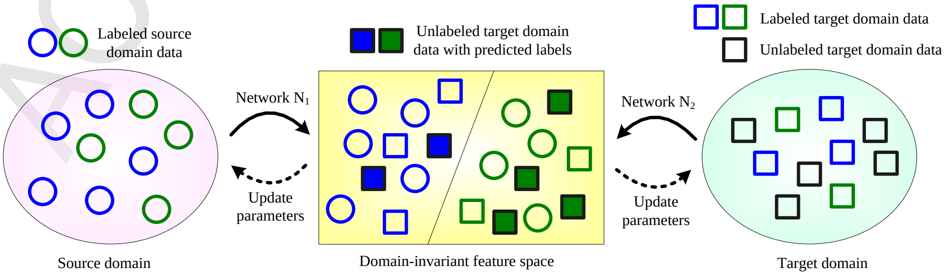

Labeled source domain data
Unlabeled target domain data with predicted labels
Labeled target domain data
Unlabeled target domain data
Network N$_1$
Update parameters
Network N$_2$
Update parameters
Source domain
Domain-invariant feature space
Target domain

Fig. 1 The working flow of the proposed heterogeneous domain adaptation network

According to the idea in [21], the working flow of the proposed heterogeneous domain adaptation network is shown in Fig. 1. We aim to classify unlabeled data in target domain by projecting the source-domain and target-domain data from different feature spaces to the domain-invariant feature space with the same dimensionality, respectively via the heterogeneous domain adaptation network based on autoencoders N_1 and N_2.

For the heterogeneous domain adaptation network, different feature projections for source-domain and target-domain data need to be learned. Therefore, a couple of networks N_1 and N_2 are included. Fig. 2 shows the topology of HDANA network. There are altogether n+1 layers for each network, of which the first n are feature layers while the last one is classification layer. The pink and green backgrounds represent feature spaces with different dimensionalities where the source and target domains respectively locate, whereas the yellow background means the shared feature space with the same dimensionality. The working progress of the network can be divided into the following stages: 1) The weight matrix $\boldsymbol{W}_s^{(m)}$, bias vector $\boldsymbol{b}_t^{(m)}$ and classifier parameter $\boldsymbol{W}^{(n)}$ of network N_1, are learned using source-domain data by means of supervised method, where $m \in \{1,\ 2,\ \cdots,\ n-1\}$. The weight matrix $\boldsymbol{W}_t^{(m)}$ and bias vector $\boldsymbol{b}_t^{(m)}$ of network N_2 are learned using target-domain data by means of unsupervised method. Thus, the initial pseudo labels $\boldsymbol{Y}_u$ of the unlabeled data in target domain are obtained; 2) According to features in the n-th layer $\xi_s^{(n)}$ and $\xi_t^{(n)}$ for source and target domains, the marginal distribution matching term $\Gamma(\boldsymbol{p}_s, \boldsymbol{p}_t)$ between domains is obtained by MMD. The manifold alignment geometric term $G(\xi_s^{(n)}, \xi_t^{(n)})$ is obtained based on the intrinsic relationships among samples in each domain. According to the true labels in source domain $\boldsymbol{Y}_s$, true and pseudo labels in target domain $\boldsymbol{Y}_l$ and $\boldsymbol{Y}_u$, the manifold alignment similarity term $E(\xi_s^{(n)}, \xi_t^{(n)})$ and the dissimilarity term $D(\xi_s^{(n)}, \xi_t^{(n)})$ based on labels are obtained; 3) According to classifier output $\boldsymbol{p}_s$ and $\boldsymbol{p}_t$, the conditional distribution matching term $\Gamma(\boldsymbol{p}_s, \boldsymbol{p}_t)$ between domains is obtained by MMD as well. Then the loss function $L(\boldsymbol{W}^{(n)}, \xi_s^{(n)}, \xi_l^{(n)})$ of the softmax classifier is obtained based on $\boldsymbol{p}_s$, $\boldsymbol{Y}_s$, $\boldsymbol{p}_l$ and $\boldsymbol{Y}_l$; 4) According to overall objective function J, $\boldsymbol{W}_s^{(m)}$, $\boldsymbol{W}_t^{(m)}$, $\boldsymbol{b}_s^{(m)}$, $\boldsymbol{b}_t^{(m)}$ and $\boldsymbol{W}^{(n)}$ are iteratively updated by gradient descent method till convergence.

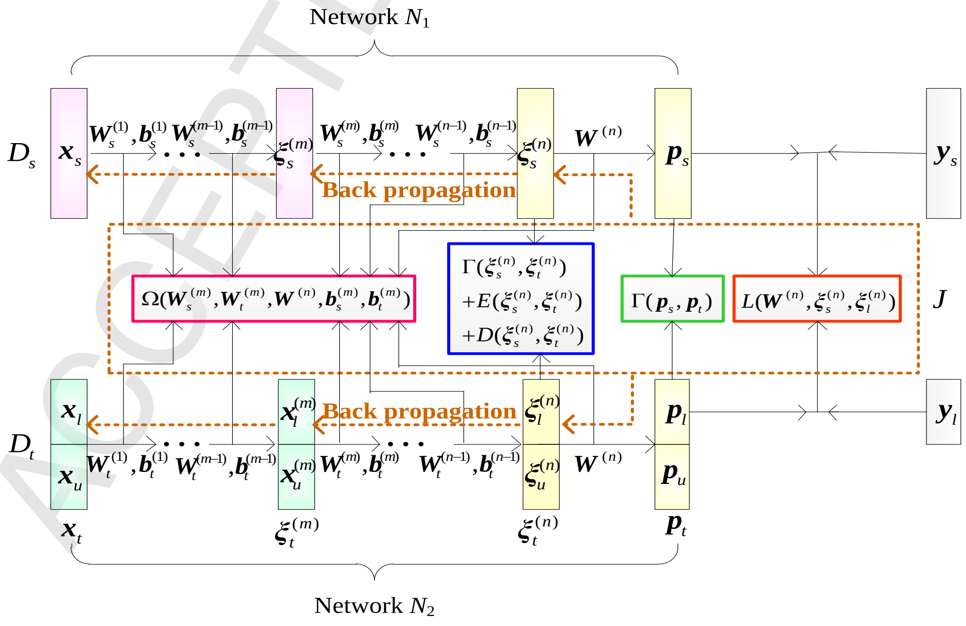

Network N_1
D_s
$\boldsymbol{x}_s$
$\boldsymbol{W}_s^{(1)}, \boldsymbol{b}_s^{(1)}$
$\boldsymbol{W}_s^{(m-1)}, \boldsymbol{b}_s^{(m-1)}$
$\xi_s^{(m)}$
$\boldsymbol{W}_s^{(m)}, \boldsymbol{b}_s^{(m)}$
$\boldsymbol{W}_s^{(n-1)}, \boldsymbol{b}_s^{(n-1)}$
$\xi_s^{(n)}$
$\boldsymbol{W}^{(n)}$
$\boldsymbol{p}_s$
$\boldsymbol{y}_s$
Back propagation
$\Omega(\boldsymbol{W}_s^{(m)}, \boldsymbol{W}_t^{(m)}, \boldsymbol{W}^{(n)}, \boldsymbol{b}_s^{(m)}, \boldsymbol{b}_t^{(m)})$
$\Gamma(\xi_s^{(n)}, \xi_t^{(n)})$ $+E(\xi_s^{(n)}, \xi_t^{(n)})$ $+D(\xi_s^{(n)}, \xi_t^{(n)})$
$\Gamma(\boldsymbol{p}_s, \boldsymbol{p}_t)$
$L(\boldsymbol{W}^{(n)}, \xi_s^{(n)}, \xi_l^{(n)})$
J
D_t
$\boldsymbol{x}_l$
$\boldsymbol{x}_l^{(m)}$
Back propagation
$\xi_l^{(n)}$
$\boldsymbol{p}_l$
$\boldsymbol{y}_l$
$\boldsymbol{x}_u$
$\boldsymbol{W}_t^{(1)}, \boldsymbol{b}_t^{(1)}$
$\boldsymbol{W}_t^{(m-1)}, \boldsymbol{b}_t^{(m-1)}$
$\boldsymbol{x}_u^{(m)}$
$\boldsymbol{W}_t^{(m)}, \boldsymbol{b}_t^{(m)}$
$\boldsymbol{W}_t^{(n-1)}, \boldsymbol{b}_t^{(n-1)}$
$\xi_u^{(n)}$
$\boldsymbol{W}^{(n)}$
$\boldsymbol{p}_u$
$\boldsymbol{x}_t$
$\xi_t^{(m)}$
$\xi_t^{(n)}$
$\boldsymbol{p}_t$
Network N_2

Fig. 2 The topology of HDANA network

2.2 The working principle of HDANA

The HDANA network is stacked by multiple AEs. The AE is trained layer-wise by using unsupervised methods. For the *m*-th layer, $\boldsymbol{\xi}_{ri}^{(m)}$ represents the feature of the *i*-th input sample in source and target domains, where $r \in \{s, t\}$ and $\boldsymbol{\xi}_{ri}^{(1)} = \boldsymbol{x}_{ri}$. The coding and decoding processes are as follows

$$\boldsymbol{\xi}_{ri}^{(m+1)} = f(\boldsymbol{W}_r^{(m)}\boldsymbol{\xi}_{ri}^{(m)} + \boldsymbol{b}_r^{(m)}) \tag{1}$$

$$\hat{\boldsymbol{\xi}}_{ri}^{(m)} = g(\boldsymbol{W}'^{(m)}_r\boldsymbol{\xi}_{ri}^{(m+1)} + \boldsymbol{b}'^{(m)}_r) \tag{2}$$

The weight matrices $\boldsymbol{W}_r^{(m)}$ and $\boldsymbol{W}'^{(m)}_r$, the bias vectors $\boldsymbol{b}_r^{(m)}$ and $\boldsymbol{b}'^{(m)}_r$ are learned by minimizing the reconstruction error as

$$\min_{\boldsymbol{W}_r^{(m)}, \boldsymbol{b}_r^{(m)}, \boldsymbol{W}'^{(m)}_r, \boldsymbol{b}'^{(m)}_r} \sum_{i=1}^{n_r} \left\| \hat{\boldsymbol{\xi}}_{ri}^{(m)} - \boldsymbol{\xi}_{ri}^{(m)} \right\|_2^2 \tag{3}$$

where $\boldsymbol{\xi}_{ri}^{(m)} \in \mathrm{R}^{d_r^{(m)}}$ and $\boldsymbol{\xi}_{ri}^{(m+1)} \in \mathrm{R}^{d_r^{(m+1)}}$ represent that feature dimensionalities of the *m*-th and *m*+1-th layers are respectively $d_r^{(m)}$ and $d_r^{(m+1)}$, $\hat{\boldsymbol{\xi}}_{ri}^{(m)} \in \mathrm{R}^{d_r^{(m)}}$ denotes the reconstruction of $\boldsymbol{\xi}_{ri}^{(m)}$, $\boldsymbol{W}_r^{(m)} \in \mathrm{R}^{d_r^{(m+1)} \times d_r^{(m)}}$ and $\boldsymbol{b}_r^{(m)} \in \mathrm{R}^{d_r^{(m+1)} \times 1}$ respectively mean the weight matrix and bias vector in coding process, whereas $\boldsymbol{W}'^{(m)}_r \in \mathrm{R}^{d_r^{(m)} \times d_r^{(m+1)}}$ and $\boldsymbol{b}'^{(m)}_r \in \mathrm{R}^{d_r^{(m)} \times 1}$ respectively mean the weight matrix and bias vector in decoding process. $f(\cdot)$ and $g(\cdot)$ are both sigmoid activation functions. Multi-layer AEs are stacked and the feature of the *n*-th layer of both domains $\boldsymbol{\xi}_{ri}^{(n)}$ is obtained. It is noted that the feature dimensionalities of domains are identical, i.e., $d_s^{(n)} = d_t^{(n)}$. Labels of the *n*+1-th layer are predicted by softmax classifier, and the possibility of samples belonging to each class is estimated as follows.

$$\boldsymbol{p}_{ri} = \begin{bmatrix} p(y_{ri} = 1 \mid \boldsymbol{\xi}_{ri}^{(n)}, \boldsymbol{W}^{(n)}) \\ p(y_{ri} = 2 \mid \boldsymbol{\xi}_{ri}^{(n)}, \boldsymbol{W}^{(n)}) \\ \vdots \\ p(y_{ri} = c \mid \boldsymbol{\xi}_{ri}^{(n)}, \boldsymbol{W}^{(n)}) \end{bmatrix} = \frac{1}{\sum_{k=1}^{c} e^{\boldsymbol{W}_k^{(n)}\boldsymbol{\xi}_{ri}^{(n)}}} \begin{bmatrix} e^{\boldsymbol{W}_1^{(n)}\boldsymbol{\xi}_{ri}^{(n)}} \\ e^{\boldsymbol{W}_2^{(n)}\boldsymbol{\xi}_{ri}^{(n)}} \\ \vdots \\ e^{\boldsymbol{W}_c^{(n)}\boldsymbol{\xi}_{ri}^{(n)}} \end{bmatrix} \tag{4}$$

Based on true labels $\boldsymbol{Y}_s$ and $\boldsymbol{Y}_t$ of samples in the source and target domains, the loss function of softmax classifier is obtained as

$$L(\boldsymbol{W}^{(n)}, \boldsymbol{\xi}_s^{(n)}, \boldsymbol{\xi}_t^{(n)}) = -[\frac{1}{n_s}\sum_{i=1}^{n_s}\sum_{k=1}^{c} 1\{y_{si} = k\} \log \frac{e^{\boldsymbol{W}_k^{(n)}\boldsymbol{\xi}_{si}^{(n)}}}{\sum_{k=1}^{c} e^{\boldsymbol{W}_k^{(n)}\boldsymbol{\xi}_{si}^{(n)}}} + \frac{1}{n_l}\sum_{j=1}^{n_l}\sum_{k=1}^{c} 1\{y_{lj} = k\} \log \frac{e^{\boldsymbol{W}_k^{(n)}\boldsymbol{\xi}_{lj}^{(n)}}}{\sum_{k=1}^{c} e^{\boldsymbol{W}_k^{(n)}\boldsymbol{\xi}_{lj}^{(n)}}}] \tag{5}$$

where $\boldsymbol{W}_k^{(n)}$ represents the *k*-th row of $\boldsymbol{W}^{(n)}$, $k \in \{1,\ 2,\ \cdots,\ C\}$ represents a label of the *k*-th class.

In the *n*-th feature layer, the distribution similarity is measured by MMD to match marginal distribution between the source and target domains. Therefore, the marginal distribution difference to be minimized is

$$\Gamma(\boldsymbol{\xi}_s^{(n)}, \boldsymbol{\xi}_t^{(n)}) = \left\| \frac{1}{n_s}\sum_{i=1}^{n_s} \boldsymbol{\xi}_{si}^{(n)} - \frac{1}{n_t}\sum_{j=1}^{n_t} \boldsymbol{\xi}_{tj}^{(n)} \right\|_2^2 \tag{6}$$

The traditional manifold learning methods only deal with dimensionality reduction from a single manifold. Therefore, the manifold alignment method is introduced, which can make different manifolds mapping to the same space and keep geometric structure unchanged by case matching. According to [22], the manifold alignment method based on label information is applied to the heterogeneous domain adaptation problem. The geometric term is used to preserve its own manifold within each domain, which is defined as

$$G(\xi_s^{(n)},\xi_t^{(n)}) = \frac{1}{2}\sum_{r\in\{s,t\}}\sum_{i,j=1}^{n_r} W_g^r(i,j)\left\|\xi_{ri}^{(n)}-\xi_{rj}^{(n)}\right\|_2^2 = \frac{1}{2}\sum_{r\in\{s,t\}} \mathbf{1}_{n_r}^{\mathrm{T}}(\boldsymbol{W}_g^r \odot \boldsymbol{H}_g^r)\mathbf{1}_{n_r} \quad (7)$$

where $\boldsymbol{H}_g^r(i,j)=\left\|\xi_{ri}^{(n)}-\xi_{rj}^{(n)}\right\|_2^2$, $\mathbf{1}_{n_r}\in\mathbf{R}^{n_r}$ is the vector with all elements equal 1. To avoid neighborhood parameter selection, the graph construction method in [23] is adopted. Elements in local similarity matrix $\boldsymbol{W}_g^r\in\mathbf{R}^{n_r\times n_r}$ are defined as

$$W_g^r(i,j)=\begin{cases}\exp(-d(\boldsymbol{x}_{ri},\boldsymbol{x}_{rj})), & \text{if } \exp(-d(\boldsymbol{x}_{ri},\boldsymbol{x}_{rj}))>1/n_r\sum_{a=1}^{n_r}\exp(-d(\boldsymbol{x}_{ri},\boldsymbol{x}_{ra}))\\ 0, & \text{otherwise}\end{cases} \quad (8)$$

where $d(\boldsymbol{x}_{ri},\boldsymbol{x}_{rj})=\left\|\boldsymbol{x}_{ri}-\boldsymbol{x}_{rj}\right\|^2 \Big/ \sum_{a=1}^{n_r}\left\|\boldsymbol{x}_{ri}-\boldsymbol{x}_{ra}\right\|^2$, $\exp(-d(\boldsymbol{x}_{ri},\boldsymbol{x}_{rj}))$ represents the similarity degree between $\boldsymbol{x}_{ri}$ and $\boldsymbol{x}_{rj}$, $1/n_r\sum_{a=1}^{n_r}\exp(-d(\boldsymbol{x}_{ri},\boldsymbol{x}_{ra}))$ is the mean similarity degree between $\boldsymbol{x}_{ri}$ and all other samples.

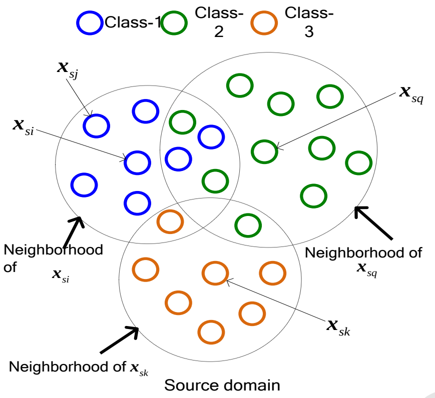


Fig. 3 Diagram of geometric term constructing

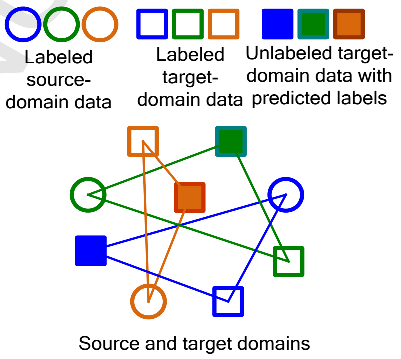

Labeled source-domain data
Labeled target-domain data
Unlabeled target-domain data with predicted labels
Source and target domains

Fig. 4 Diagram of similarity and dissimilarity terms constructing

Taking the source-domain data as an example, Fig. 3 intuitively shows the construction of the geometric term, where three different colors are used to denote three classes. If the similarity degree between $\boldsymbol{x}_{si}$ and $\boldsymbol{x}_{sj}$ is larger than the mean similarity degree between $\boldsymbol{x}_{ri}$ and all other samples, $\boldsymbol{x}_{sj}$ is the neighborhood of $\boldsymbol{x}_{si}$ and the connecting weight is $W_g^s(i,j)$. The similarity neighborhood of $\boldsymbol{x}_{si}$ is defined as a hypersphere with $\boldsymbol{x}_{si}$ as the center and the mean similarity degree as the radius. The benefit of introducing geometric term is to make manifold structure unchanged, while the defect is that the label information is ignored. Thus, the hypersphere may contain other classes of samples. Taking the neighborhood of $\boldsymbol{x}_{si}$ as an example, except Class-1 samples, it contains two Class-2 and one Class-3 samples. Aiming at the problem, the similarity and dissimilarity terms are introduced to further improve classification performance.

Fig. 4 intuitively shows the construction of the similarity and dissimilarity terms, in which circle and square represent the source-domain and target-domain data respectively and each color denotes one class. A solid line is used to connect two samples with the same class label. The similarity term is defined as

$$E(\xi_s^{(n)},\xi_t^{(n)}) = \frac{1}{2}\sum_{i,j=1}^{n_s+n_t} W_e(i,j)\left\|\xi_i^{(n)}-\xi_j^{(n)}\right\|_2^2 = \frac{1}{2}\mathbf{1}_{n_s+n_t}^{\mathrm{T}}(\boldsymbol{W}_e\odot\boldsymbol{H})\mathbf{1}_{n_s+n_t} \quad (9)$$

where $\xi_r^{(n)}=\{\xi_{ri}^{(n)}\}|_{i=1}^{n_r}$ with $\xi_r^{(n)}\in\mathrm{R}^{d_r^{(n)}\times n_r}$, $\xi^{(n)}=[\xi_s^{(n)},\xi_t^{(n)}]$ with $\xi^{(n)}\in\mathbf{R}^{d_s^{(n)}\times(n_s+n_t)}$,

$\boldsymbol{H}(i,j)=\left\|\boldsymbol{\xi}_i^{(n)}-\boldsymbol{\xi}_j^{(n)}\right\|_2^2$, $1_{n_s+n_t}\in \mathrm{R}^{n_s+n_t}$ is a vector that all elements equal 1.

For the similarity term, when samples $\boldsymbol{\xi}_i^{(n)}$ and $\boldsymbol{\xi}_j^{(n)}$ have the same label, $W_e(i,j)=1$, otherwise $W_e(i,j)=0$. The role of similarity term is to ensure that the samples of all domains with the same class label have similar feature representation during the process of parameter optimization. What we want to achieve is to pull all samples have the same class label close to each other, whatever from the source or target domain. Note that predicted pseudo labels are used regarding unlabeled samples in target domain.

Opposite to the similarity term, the dissimilarity term is defined as

$$\begin{aligned} D(\boldsymbol{\xi}_s^{(n)},\boldsymbol{\xi}_t^{(n)}) &= \frac{1}{2}\sum_{i,j=1}^{n_s+n_t} W_d(i,j)\left\|\boldsymbol{\xi}_i^{(n)}-\boldsymbol{\xi}_j^{(n)}\right\|_2^2 \\ &= \frac{1}{2}\mathbf{1}_{n_s+n_t}^{\mathrm{T}}(\boldsymbol{W}_d \odot \boldsymbol{H})\mathbf{1}_{n_s+n_t} \end{aligned} \tag{10}$$

where $W_d(i,j)=1$ when labels of samples $\boldsymbol{\xi}_i^{(n)}$ and $\boldsymbol{\xi}_j^{(n)}$ differ, otherwise $W_d(i,j)=0$. The role of dissimilarity term is to ensure that the samples of all domains with different class labels have different feature representation during the process of parameter optimization. What we want to achieve is to pull all samples have different same class labels apart from each other, whatever from the source or target domain.

In the classification layer, the conditional distribution of both domains $P(y_s\,|\,\boldsymbol{\xi}_s^{(n)})$ and $P(y_t\,|\,\boldsymbol{\xi}_t^{(n)})$ is matched by likewise MMD metric. Therefore, the conditional distribution difference is written as

$$\Gamma(\boldsymbol{p}_s,\boldsymbol{p}_t)=\left\|\frac{1}{n_s}\sum_{i=1}^{n_s}\boldsymbol{p}_{si}-\frac{1}{n_t}\sum_{j=1}^{n_t}\boldsymbol{p}_{tj}\right\|_2^2 \tag{11}$$

Further to avoid over-fitting, the weight decay term is introduced to reduce the update speed of weight, which is

$$\Omega(\boldsymbol{W}_s^{(m)},\boldsymbol{W}_t^{(m)},\boldsymbol{W}^{(n)},\boldsymbol{b}_s^{(m)},\boldsymbol{b}_t^{(m)})=\sum_{r\in\{s,t\}}\sum_{m=1}^{n-1}(\left\|\boldsymbol{W}_r^{(m)}\right\|_2^2+\left\|\boldsymbol{b}_r^{(m)}\right\|_2^2)+\left\|\boldsymbol{W}^{(n)}\right\|_2^2 \tag{12}$$

Combining (5), (6), (7), (9), (10), (11) and (12), the overall objective function is

$$\begin{aligned} J = {} & L(\boldsymbol{W}^{(n)},\boldsymbol{\xi}_s^{(n)},\boldsymbol{\xi}_t^{(n)})+\alpha\Gamma(\boldsymbol{\xi}_s^{(n)},\boldsymbol{\xi}_t^{(n)})+\beta\Gamma(\boldsymbol{p}_s,\boldsymbol{p}_t)+\gamma[G(\boldsymbol{\xi}_s^{(n)},\boldsymbol{\xi}_t^{(n)})+E(\boldsymbol{\xi}_s^{(n)},\boldsymbol{\xi}_t^{(n)})-D(\boldsymbol{\xi}_s^{(n)},\boldsymbol{\xi}_t^{(n)})] \\ & +\lambda\Omega(\boldsymbol{W}_s^{(m)},\boldsymbol{W}_t^{(m)},\boldsymbol{W}^{(n)},\boldsymbol{b}_s^{(m)},\boldsymbol{b}_t^{(m)}) \end{aligned} \tag{13}$$

where α, β, γ, λ are all coefficients to balance effect of each term to the objective function.

2.3 HDANA parameter optimization

To minimize the objective function, the gradient descent method via back propagation is used to learn model parameters. It is necessary to calculate the partial derivative of the objective function with respect to the weight and bias of each layer. The partial derivative of the loss function term $L(\boldsymbol{W}^{(n)},\boldsymbol{\xi}_s^{(n)},\boldsymbol{\xi}_t^{(n)})$ of the softmax classifier, of the marginal distribution matching term $\Gamma(\boldsymbol{\xi}_s^{(n)},\boldsymbol{\xi}_t^{(n)})$, of the conditional distribution matching term $\Gamma(\boldsymbol{p}_s,\boldsymbol{p}_t)$ and of the weight decay term $\Omega(\boldsymbol{W}_s^{(m)},\boldsymbol{W}_t^{(m)},\boldsymbol{W}^{(n)},\boldsymbol{b}_s^{(m)},\boldsymbol{b}_t^{(m)})$ are easy to obtain. Therefore in this paper, only the manifold alignment terms $G(\boldsymbol{\xi}_s^{(n)},\boldsymbol{\xi}_t^{(n)})$, $E(\boldsymbol{\xi}_s^{(n)},\boldsymbol{\xi}_t^{(n)})$ and $D(\boldsymbol{\xi}_s^{(n)},\boldsymbol{\xi}_t^{(n)})$ regarding partial derivative of the parameter model is analyzed, which are

$$\frac{\partial G(\boldsymbol{\xi}_s^{(n)},\boldsymbol{\xi}_t^{(n)})}{\partial \boldsymbol{\xi}_{ri}^{(n)}}=(\boldsymbol{\xi}_{ri}^{(n)}\mathbf{1}_{n_r}^{\mathrm{T}}-\boldsymbol{\xi}_r^{(n)})(\boldsymbol{W}_g^{r}+(\boldsymbol{W}_g^{r})^{\mathrm{T}})_{(:,i)} \tag{14}$$

$$G\delta_{ri}^{(n)}=\frac{\partial G(\boldsymbol{\xi}_s^{(n)},\boldsymbol{\xi}_t^{(n)})}{\partial \boldsymbol{\xi}_{ri}^{(n)}}\odot\boldsymbol{\xi}_{ri}^{(n)}\odot(\mathbf{1}_{d_r^{(n)}}-\boldsymbol{\xi}_{ri}^{(n)}) \tag{15}$$

$$G\delta_{ri}^{(m)}=(W_r^{(m)})^{\mathrm{T}}G\delta_{ri}^{(m+1)}\odot\boldsymbol{\xi}_{ri}^{(m)}\odot(\mathbf{1}_{d_r^{(m)}}-\boldsymbol{\xi}_{ri}^{(m)}) \tag{16}$$

$$\frac{\partial G(\boldsymbol{\xi}_s^{(n)},\boldsymbol{\xi}_t^{(n)})}{\partial \boldsymbol{W}_r^{(m)}}=\frac{1}{n_r}G\boldsymbol{\delta}_r^{(m+1)}(\boldsymbol{\xi}_r^{(m)})^{\mathrm{T}} \tag{17}$$

$$\frac{\partial G(\boldsymbol{\xi}_s^{(n)},\boldsymbol{\xi}_t^{(n)})}{\partial \boldsymbol{b}_r^{(m)}}=\frac{1}{n_r}\sum_{j=1}^{n_r}G\boldsymbol{\delta}_r^{(m+1)}(:,j) \qquad 18)$$

where $G\delta_{ri}^{(m)}\in \mathrm{R}^{d_r^{(m)}}$ is intermediate variable. Formation of the partial derivative of the similarity term $E(\boldsymbol{\xi}_s^{(n)},\boldsymbol{\xi}_t^{(n)})$ and that of the dissimilarity term $D(\boldsymbol{\xi}_s^{(n)},\boldsymbol{\xi}_t^{(n)})$ are similar to $G(\boldsymbol{\xi}_s^{(n)},\boldsymbol{\xi}_t^{(n)})$. Note that $\boldsymbol{\xi}_s^{(n)}=\boldsymbol{\xi}^{(n)}(:,1:n_s)$ and $\boldsymbol{\xi}_t^{(n)}=\boldsymbol{\xi}^{(n)}(:,n_s+1:n_s+n_t)$ should be considered separately. After partial derivatives with regard to weight and bias for each term in objective function *J*, the network parameters are updated via the following equations as

$$\begin{aligned}\boldsymbol{W}_r^{(m)}&=\boldsymbol{W}_r^{(m)}-\mu\frac{\partial J}{\partial \boldsymbol{W}_r^{(m)}}\\ \boldsymbol{b}_r^{(m)}&=\boldsymbol{b}_r^{(m)}-\mu\frac{\partial J}{\partial \boldsymbol{b}_r^{(m)}}\\ \boldsymbol{W}^{(n)}&=\boldsymbol{W}^{(n)}-\mu\frac{\partial J}{\partial \boldsymbol{W}^{(n)}}\end{aligned} \tag{19}$$

where μ represents learning rate.

2.4 Algorithm step

With mentioned above, the HDANA algorithm is as follows.

Input: source-domain data $D_s=\{\boldsymbol{x}_{si},y_{si}\}|_{i=1}^{n_s}$, labeled target-domain data $D_l=\{\boldsymbol{x}_{lj},y_{lj}\}|_{j=1}^{n_l}$, unlabeled target-domain data $D_u=\{\boldsymbol{x}_{uj}\}|_{j=1}^{n_u}$, number of network layers *n*+1, number of nodes in classification layer *c*, learning rate of parameters μ, balance coefficients α, β, γ, λ and maximum iteration times *T*.

Output: The predicted labels of target-domain data.

Step 1: Learn parameters $\boldsymbol{W}_s^{(m)}$ and $\boldsymbol{b}_s^{(m)}$ of N_1 and classifier parameter $\boldsymbol{W}^{(n)}$ by supervised method. Learn parameters $\boldsymbol{W}_t^{(m)}$ and $\boldsymbol{b}_t^{(m)}$ of N_2 by unsupervised method. Get initial pseudo labels of unlabeled data $\{y_{uj}\}|_{j=1}^{n_u}$.

Step 2: Get features of both domains in the *n*-th layer $\boldsymbol{\xi}_s^{(n)}$ and $\boldsymbol{\xi}_t^{(n)}$, as well as marginal distribution matching term $\Gamma(\boldsymbol{\xi}_s^{(n)},\boldsymbol{\xi}_t^{(n)})$ according to (6).

Step 3: Get the manifold alignment geometrical term $G(\boldsymbol{\xi}_s^{(n)},\boldsymbol{\xi}_t^{(n)})$, similarity term $E(\boldsymbol{\xi}_s^{(n)},\boldsymbol{\xi}_t^{(n)})$ and dissimilarity term $D(\boldsymbol{\xi}_s^{(n)},\boldsymbol{\xi}_t^{(n)})$ according to (7), (9) and (10).

Step 4: Get conditional distribution difference term $\Gamma(\boldsymbol{p}_s,\boldsymbol{p}_t)$ according to (11).

Step 5: Based on the true labels of both domains, get the loss function term $L(\boldsymbol{W}^{(n)},\boldsymbol{\xi}_s^{(n)},\boldsymbol{\xi}_t^{(n)})$ of the softmax classifier.

Step 6: Get the weight decay term $\Omega(\boldsymbol{W}_s^{(m)},\boldsymbol{W}_t^{(m)},\boldsymbol{W}^{(n)},\boldsymbol{b}_s^{(m)},\boldsymbol{b}_t^{(m)})$ according to (12).

Step 7: Get the overall objective function *J* according to (13).

Step 8: Optimize the two network weights $\boldsymbol{W}_s^{(m)}$ and $\boldsymbol{W}_t^{(m)}$, biases $\boldsymbol{b}_s^{(m)}$ and $\boldsymbol{b}_t^{(m)}$, and classifier parameter $\boldsymbol{W}^{(n)}$ according to (19). Update the pseudo labels $\{y_{uj}\}|_{j=1}^{n_u}$ of unlabeled samples in target domain.

Step 9: Repeat Steps 2-8, till the maximum iteration *T* is reached.

Step 10: Get the output of softmax classifier according to (4). The label corresponding to the maximum probability is the predicted label of the unlabeled target-domain sample.

3 Experimental Results

3.1 The datasets and settings

To verify the effectiveness of the proposed HDANA method, 16 groups of domain adaptation tasks are constructed on cross-domain object recognition dataset Office+Caltech-256 and cross-lingual text categorization dataset Multilingual Reuters Collection [20].

The Office+Caltech-256 dataset is illustrated in Table 1. It contains 3 subsets: Amazon (images from online merchants), Webcam (low-resolution images by a web camera), DSLR (high-resolution images by a digital SLR camera). Each subset has 31 classes. Two types of features are used: the 800-dimension SURF feature [24] and 4096-dimension DeCAF feature [25].

Table 1 Illustration of dataset Office+Caltech-256

Dataset	Type	No. samples	SURF dimension	DeCAF dimension	No. classes
Amazon	Object	958	800	4096	10
Webcam	Object	295	800	4096	10
DSLR	Object	157	800	4096	10
Caltech-256	Object	1123	800	4096	10

The Multilingual Reuters Collection dataset is illustrated in Table 2. It contains a total of 11000 articles from 6 classes, in which the following 5 languages, English, French, Italian, German and Spanish, are respectively used. According to [18], the final dimensions of these languages are 1131, 1230, 1041, 1417 and 807 after PCA reduction.

Table 2 Illustration of dataset Multilingual Reuters Collection

Dataset	Type	No. samples	Feature dimension	No. classes
English	Text	18758	1131	6
French	Text	26648	1230	6
Italian	Text	24039	1041	6
German	Text	29953	1417	6
Spanish	Text	11547	807	6

3.2 The comparative experiments

To evaluate the classification performance of HDANA, the support vector machine (SVM_t) and autoencoder (AE_t) are used as comparative benchmarks, and labeled target-domain data are directly used to train SVM and AE to predict unlabeled data. As both SSKMDA [11] and SCP [20] methods need additional unlabeled source-domain data, neither is considered in this paper. Instead, the following associated HDA methods as DAMA [16], MMDT [9], SHFA [18], G-JDA [21] and TNT [15] are taken and compared. Among them, experiments by SVM_t, DAMA, MMDT and G-JDA are coming from [21]. A 3-layer autoencoder is used and parameters are optimized in AE_t. For SHFA and TNT, we use the source code provided by authors and all the parameters are carefully tuned.

At first, we perform the object recognition experiments across feature spaces. Since the size of DSLR (D) subset is much small, only the three Amazon, Webcam and Caltech-256 subsets are selected. According to the settings in [21], 20 and 3 images per class are respectively picked up as labeled samples in both domains. The remaining target-domain samples are unlabeled ones to be identified. The average classification results (classification accuracy with standard deviation) of 20 random trials are shown in Table 3. It is shown that: 1) SVM_t and AE_t are the benchmark methods for non-domain adaptation, and SVM_t cannot achieve satisfactory performance whereas AE_t is very competitive, indicating advantages of the deep network; 2) The proposed HDANA is obviously advantageous over others, and the classification accuracies on all tasks are much higher than that of AE_t. Therefore, HDANA can better perform the domain adaptation process of cross-feature and thus achieve the highest classification accuracy of unlabeled target-domain samples.

Table 3 Comparison of average classification results on cross-feature tasks (%)

D_s→D_t	SVM_t	AE_t	DAMA[16]	MMDT[9]	SHFA[18]	G-JDA[21]	TNT[15]	HDANA
				SURF to $DeCAF_6$				
W→W	84.7±1.0	87.4±0.3	87.3±0.9	88.2±0.6	89.2±0.8	89.4±0.9	92.6±0.4	**93.6±0.5**
C→C	70.3±1.3	73.0±0.2	72.8±0.8	76.4±0.7	74.3±0.9	86.7±0.5	86.8±0.9	**89.8±0.3**
A→A	82.3±0.9	85.7±0.2	86.7±0.5	86.5±0.5	85.7±0.3	92.3±0.2	92.1±0.8	**95.5±0.5**
				$DeCAF_6$ to SURF				
W→W	52.3±1.2	57.8±0.1	57.0±0.9	57.7±0.8	60.8±0.7	63.8±0.9	62.3±0.9	**67.9±0.4**
C→C	27.6±0.6	30.1±0.1	28.8±0.6	30.8±0.7	28.7±0.9	33.7±0.8	35.3±0.9	**36.5±0.3**
A→A	38.9±0.7	41.9±0.2	40.6±0.6	45.0±0.7	47.2±0.8	50.3±0.7	50.8±1.0	**54.2±0.6**

Then we consider a more challenging object recognition task, in which the cross-dataset and cross-feature are simultaneously achieved. A, W and C are selected as source domains and D as target domain. In analogy the average classification results of 20 trials are listed in Table 4. It is shown that by comparison, HDANA still achieves the best. The classification accuracy is much higher than that of TNT with shallow structure. The consistency of label and structure information can be kept by introducing manifold alignment term. Hence excellent results can be available as well even for domain adaptation problems that both cross-dataset and cross-feature are simultaneously involved.

Table 4 Comparison of average classification results on cross-feature and cross-dataset tasks (%)

D_s→D_t	SVM_t	AE_t	DAMA[16]	MMDT[9]	SHFA[18]	G-JDA[21]	TNT[15]	HDANA
				SURF to $DeCAF_6$				
W→D	89.7±0.8	89.4±0.1	89.4±0.6	90.8±0.6	92.0±1.0	95.0±0.4	95.5±1.0	**96.9±0.3**
C→D		89.5±0.1	89.8±0.6	91.2±0.6	91.9±0.8	92.8±0.8	92.7±1.1	**95.3±0.6**
A→D		89.4±0.1	90.5±0.5	90.5±0.6	91.9±1.1	94.3±0.7	90.6±0.9	**96.1±0.5**
				$DeCAF_6$ to SURF				
W→D	51.3±0.8	53.3±0.9	56.2±0.9	52.3±0.9	55.8±1.0	55.5±0.8	58.3±0.8	**61.4±0.6**
C→D		53.0±0.2	52.8±0.6	55.2±0.8	55.7±1.1	57.2±1.0	58.0±0.8	**60.6±0.3**
A→D		52.4±0.2	51.7±0.9	53.9±0.6	56.3±0.9	56.9±0.7	59.1±0.7	**60.6±0.4**

For cross-lingual text categorization task, we select English (E), French (F), Italian (I) and German (G) as the source domain and randomly pick up 100 samples as labeled data from each class, and pick Spanish (S) as target domain. 10 or 20 samples per class are randomly selected from the target domain as labeled data, and then 500 samples per class in the remaining subset are selected as unlabeled data. The average classification results of 20 random trials are listed in Table 5. By comparison, the proposed HDANA achieves the highest values among all the classification tasks and thus is significantly superior to DAMA, MMDT, SHFA, G-JDA, and TNT. HDANA is also applicable to cross-lingual text categorization task.

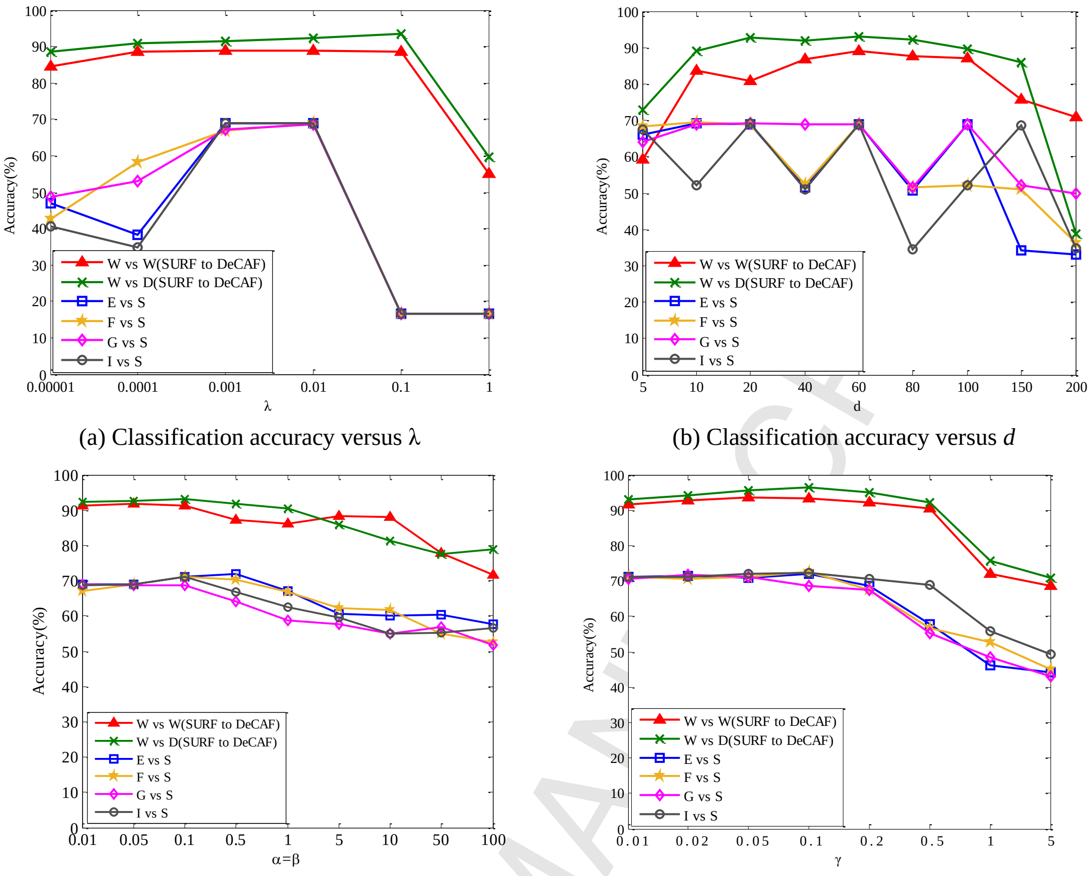
Table 5 Comparison of average classification results on cross-lingual text tasks (%)

D_s→D_t	SVM_t	AE_t	DAMA[16]	MMDT[9]	SHFA[18]	G-JDA[21]	TNT[15]	HDANA
Number of labeled samples in target domain per class=10								
E→S	67.3±0.6	61.8±0.8	66.1±0.8	68.9±0.6	68.6±0.7	69.4±0.8	70.0±1.0	**72.1±0.3**
F→S		62.1±0.3	61.5±1.0	69.3±0.6	69.1±0.6	70.5±0.7	70.9±0.8	**72.7±0.4**
G→S		61.7±0.3	63.4±0.8	68.7±0.5	69.2±0.7	69.6±1.0	69.9±0.9	**71.8±0.4**
I→S		62.3±0.7	65.3±0.8	69.5±0.6	70.4±0.6	70.1±1.0	71.0±0.9	**72.3±0.3**
Number of labeled samples in target domain per class=20								
E→S	74.5±0.4	70.8±0.5	73.1±0.4	75.5±0.4	75.4±0.3	76.0±0.7	76.1±0.7	**78.3±0.5**
F→S		71.2±0.4	72.2±0.5	75.3±0.5	75.2±0.6	76.8±0.8	77.5±0.8	**79.2±0.5**
G→S		71.4±0.3	69.3±0.7	75.7±0.5	75.5±0.5	76.8±0.7	77.1±1.1	**80.1±0.6**
I→S		71.3±0.4	73.0±0.4	76.3±0.4	76.1±0.6	76.6±0.7	76.8±0.9	**78.5±0.3**

3.3 Parameter analysis

The 5 hyper parameters including *d*, α , β , γ and λ in HDANA are taken for sensitivity analysis. Accuracy influenced by variation of all these parameters is shown in Fig. 5. As the feature dimension of both domain data is constant in the target recognition tasks, i.e., SURF feature equals constant 800 and DeCAF feature equals 4096, only one group of transfer task, i.e., W→W (SURF to DeCAF) and W→D (SURF to DeCAF), is selected from each class. In cross-lingual text categorization tasks, the dimensions of source-domain data are different. Therefore, experiments are respectively carried out on tasks on E→S, F→S, G→S and I→S. Considering there are 10 labeled samples of each class in the target domain, according to the settings in Section 3.1, the labeled samples should be randomly picked from both domains. Therefore, the average result is recorded by executing 10 times of experiments for each group. When one parameter is being measured, the rests remain constant. In addition for convenience, let the balance coefficient of marginal and conditional distribution identical, i.e., $\alpha = \beta$.

First, the weight decay regularization coefficient λ is adjusted. Fig. 5(a) shows that for tasks $W \to W$ and W→D, higher accuracy is obtained when λ belongs {0.001, 0.01, 0.1}. When $\lambda > 0.1$, the accuracy reduces significantly. For cross-lingual text categorization tasks E→S, F→S, G→S and I→S, it works better when λ={0.01, 0.1}. Then, the relationship between classification accuracy and number of nodes *d* in the shared feature layer is studied. Fig. 5(b) shows that for tasks W→W and W→D, higher accuracy is available at *d*=60, whereas for E→S, F→S, G→S and I→S, higher accuracy is available at *d*=10, 20. Further, the sensitivity experiment on marginal and conditional distribution balance coefficients α and β are carried out, as shown in Fig. 5(c). For each task, the accuracy due to parameter variation is small and higher value is achieved when α = β =0.1. Finally, the sensitivity curve of the manifold alignment term is drawn in Fig. 5(d). Classification accuracy achieves better when λ={0.01, 0.2}, while it is significantly reduced when $\gamma > 0.5$. Therefore, the optimal accuracy is available at $\gamma = 0.1$.



(a) Classification accuracy versus λ

(b) Classification accuracy versus *d*

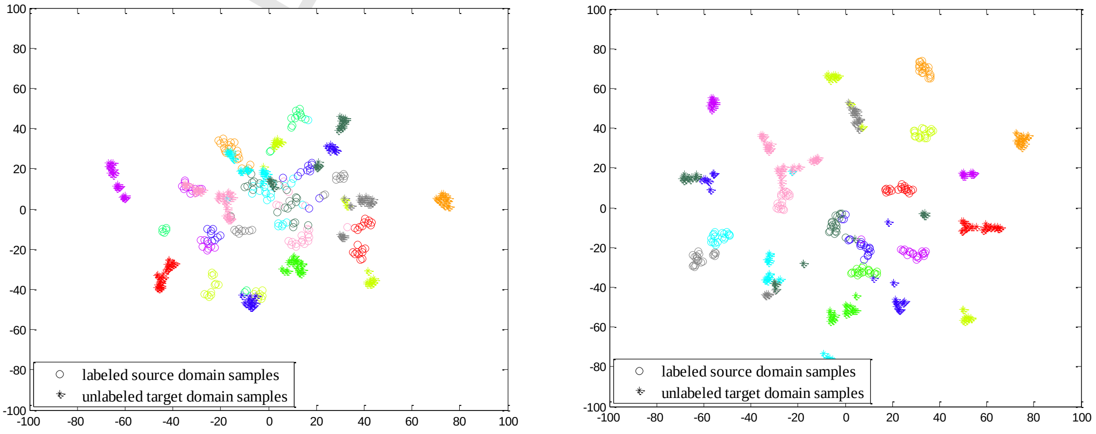
(c) Classification accuracy versus α and β

(d) Classification accuracy versus γ

Fig. 5 The parameter sensitivity curves

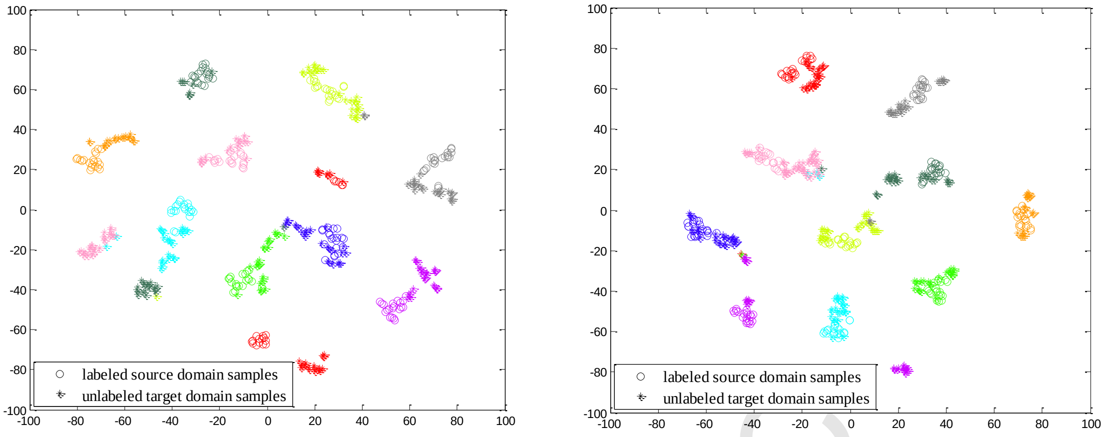
3.4 Feature visualization

To further prove that feature representation learned by HDANA has better discrimination ability, the t-SNE embedding of features is plotted on tasks W→W (SURF to DeCAF) and W→D (SURF to DeCAF). The t-SNE embedding is a high-dimensional data visualization method, which can locate the data points in a 2D or 3D space [26]. The t-SNE embedding of the original features, G-JDA features and HDANA features are drawn in Figs. 6-7. Each color represents one class and there are altogether 10. "◦" and "*" respectively mean labeled source and unlabeled target-domain samples.



(a) Original features

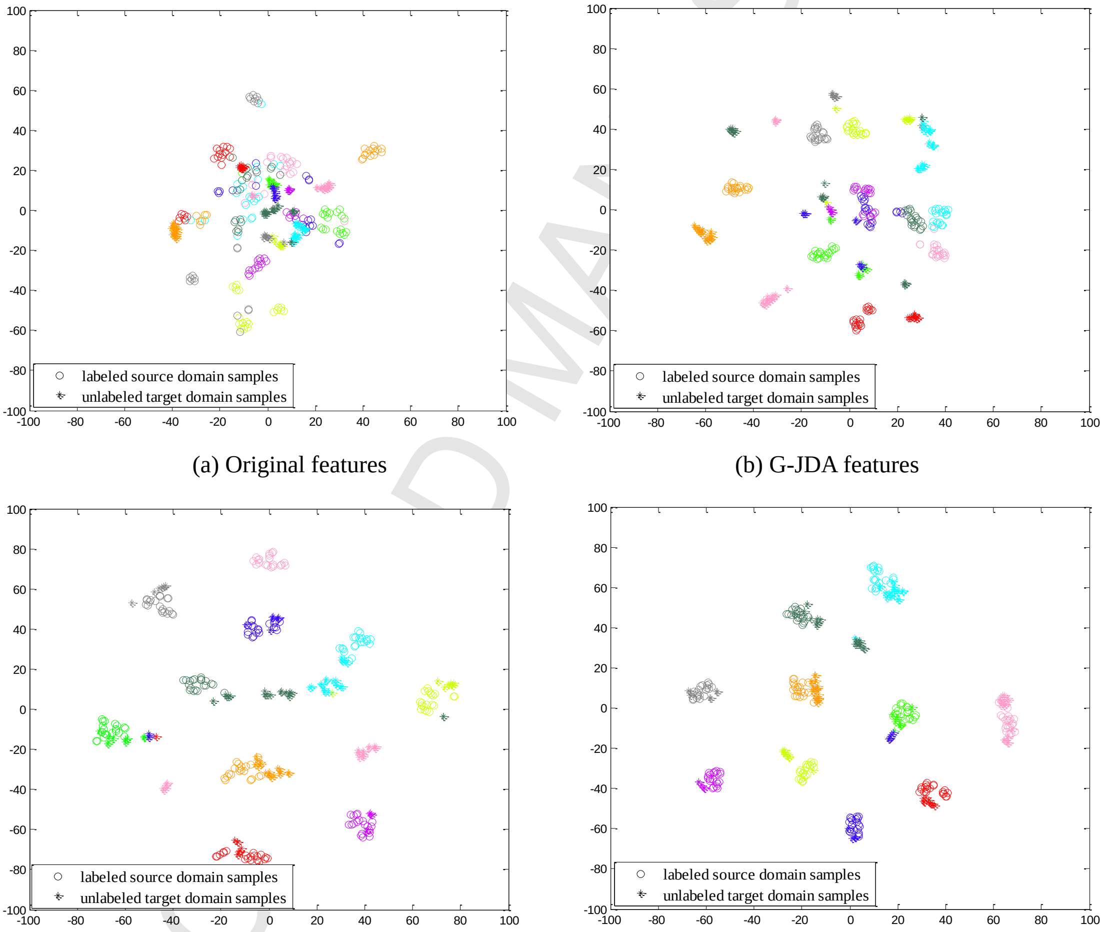
(b) G-JDA features



(c) TNT features

(d) HDANA features

Fig. 6 Feature visualization on task W→W (SURF to DeCAF)



(a) Original features

(b) G-JDA features

(c) TNT features

(d) HDANA features

Fig. 7 Feature visualization on task W→D (SURF to DeCAF)

It can be found from Figs. 6-7 that: 1) The original feature points from the source and target domains are mixed, indicating poor discrimination ability; 2) Compared with the original features, the G-JDA and TNT feature points are of compact inter-class and scattered intra-class; 3) The TNT features has a stronger discrimination ability than that of G-JDA features, which shows that the deep learning method can learn more effective feature representation than that of shallow learning method; 4) Comparatively HDANA not only has better such compactness and scatter, but also realizes alignment between the labeled source-domain samples and the unlabeled target-domain samples. Thus, it is proved that the features extracted by HDANA have better discrimination ability, thus better domain adaptation can be achieved and effective knowledge transfer can be realized. Therefore, the proposed HDANA has higher classification accuracy of target-domain samples.

4 Conclusions

Although existing shallow transfer learning methods can achieve promising results, they only utilize linear transformation or nonlinear transformation with kernel functions to bridge the gap between the source and target domains. Therefore, they are not effective enough when there is a large distribution difference and feature bias between two domains. To fill this gap in the literature, we proposed a semi-supervised HDA learning method, i.e., heterogeneous domain adaptation network based on autoencoder. Compared with existing approaches, HDANA has manifold advantages. 1) Under the framework of deep learning model, data in both domains is processed through two sets of deep networks and more abstractive shared features are obtained through multilayer nonlinear mapping; 2) The marginal and conditional distributions are simultaneously matched by empirical MMD metric on the shared feature layer to reduce distribution difference across domains; 3) The manifold alignment term based on labels is introduced. The geometric term based on graph is used to preserve the consistency of the geometric structure of domain data. The terms of similarity and dissimilarity are used to preserve the label consistency, to have similar feature representation for samples of the same class and to get different feature representations for samples of different classes. Hence the extracted features are more distinctive; 4) Utilizing label information of both domains, the loss term of softmax classifier is obtained for back propagation to further improve classification performance. Experiments demonstrate that the proposed HDANA method yields superior performance when compared with several state-of-the-art models in terms of classification accuracy. With the arrival of big data era and the rapid development of deep learning, a large number of deep models emerge. Compared with AEs, convolutional neural networks (CNNs) involve many more connections than weights and the architecture itself realizes a form of regularization. In addition, a CNN automatically provides some degree of translation invariance. Many researches have proved that CNNs outperform AEs. Therefore, the future work is how to effective apply CNNs to large-scale and multi-modal HDA problems including recognition tasks across text and image or across audio and video.

Acknowledgements:

This work is supported by the National Natural Science Foundation of China (61273143, 61472424). It was also supported by National Funding from the FCT – *Fundação para a Ciência e a Tecnologia* through the UID/EEA/500008/2013 Project, by the Government of the Russian Federation, Grant 074-U01, and by Finep, with resources from Funttel, Grant No. 01.14.0231.00, under the *Centro de Referência em Radiocomunicações* - CRR project of the *Instituto Nacional de Telecomunicações* (Inatel), Brazil.

References:

[1]Bisio I, Delfino A, Lavagetto F, et al. Gender-driven emotion recognition through speech signals for ambient intelligence applications. IEEE Transactions on Emerging Topics in Computing, 2013, 1(2): 244-257.

[2]Bisio I, Lavagetto F, Marchese M, et al. Smartphone-based user activity recognition method for health remote monitoring applications. In: Proceedings of International Conference on Pervasive and Embedded Computing and Communication Systems, 2012: 200-205.

[3]Gong B, Shi Y, Sha F, et al. Geodesic flow kernel for unsupervised domain adaptation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2012: 2066-2073.

[4]Long M, Wang J, Ding G, et al. Transfer joint matching for unsupervised domain adaptation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2014: 1410-1417.

[5]Zhuang F, Cheng X, Luo P, et al. Supervised representation learning: transfer learning with deep autoencoders. In: Proceedings of International Conference on Artificial Intelligence, 2015: 4119-4125.

[6]Long M, Cao Y, Wang J, et al. Learning transferable features with deep adaptation networks. In: Proceedings of International Conference on Computer Science, 2015: 97-105.

[7]Long M, Wang J, Cao Y, et al. Deep learning of transferable representation for scalable domain adaptation. IEEE Transactions on Knowledge and Data Engineering, 2016, 28(8): 2027-2040.

[8]Kulis B, Saenko K, Darrell T. What you saw is not what you get: domain adaptation using asymmetric kernel transforms. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2011: 1785-1792.

[9]Hoffman J, Rodner E, Donahue J, et al. Efficient learning of domain-invariant image representations. arXiv preprint arXiv:1301.3224, 2013.

[10]Zhou J T, Tsang I W. Heterogeneous domain adaptation for multiple classes. In: Proceedings of International Conference on Artificial Intelligence and Statistics, 2014: 1095-1103.

[11]Xiao M, Guo Y. Feature space independent semi-supervised domain adaptation via kernel matching. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014, 37(1): 54-66.

[12] Tsai Y H H, Yeh Y R, Wang Y C F. Learning cross-domain landmarks for heterogeneous domain adaptation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2016: 5081-5090.

[13]Tsai Y H H, Yeh Y R, Wang Y C F. Heterogeneous domain adaptation with label and structure consistency. In: Proceedings of IEEE International Conference on Acoustics, 2016: 2842-2846.

[14]Shi X, Liu Q, Fan W, et al. Transfer learning on heterogenous feature spaces via spectral transformation. In: Proceedings of IEEE International Conference on Data Mining, 2010: 1049-1054.

[15]Chen W Y, Hsu T M H, Tsai Y H H, et al. Transfer neural trees for heterogeneous domain adaptation. In: Proceedings of European Conference on Computer Vision, 2016: 399-414.

[16]Wang C, Mahadevan S. Heterogeneous domain adaptation using manifold alignment. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence, 2011: 1541-1546.

[17]Duan L, Xu D, Tsang I. Learning with augmented features for heterogeneous domain adaptation. In: Proceedings of International Conference on Computer Science, 2012: 711-718.

[18]Li W, Duan LX, Xu D, et al. Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014, 36(6): 1134-1148.

[19]Wu XX, Wang H, Liu CW, et al. Cross-view action recognition over heterogeneous feature spaces. IEEE Transactions on Image Processing, 2015, 24(11): 4096-4108.

[20]Xiao M, Guo Y. Semi-supervised subspace co-projection for multi-class heterogeneous domain adaptation. In: Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 2015: 525-540.

[21]Hsieh Y T, Tao S Y, Tsai Y H H, et al. Recognizing heterogeneous cross-domain data via generalized joint distribution adaptation. In: Proceedings of IEEE International Conference on Multimedia and Expo, 2016: 1-6.

[22]Tuia D, Volpi M, Trolliet M, et al. Semisupervised manifold alignment of multimodal remote sensing images. IEEE Transactions on Geoscience and Remote Sensing, 2014, 52(12): 7708-7720.

[23]Yang B, Chen S. Sample-dependent graph construction with application to dimensionality reduction. Neurocomputing, 2010, 74(1-3): 301-314.

[24]Bay H, Tuytelaars T, Gool L V. SURF: speeded up robust features. Computer Vision and Image Understanding, 2006, 110(3): 404-417.

[25]Donahue J, Jia Y, Vinyals O, et al. DeCAF: A deep convolutional activation feature for generic visual recognition. Computer Science, 2013, 50(1): 815-830.

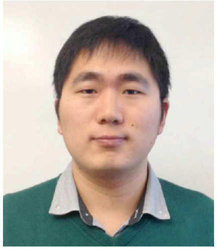
[26]Maaten L, Hinton G. Visualizing data using t-SNE. Journal of Machine Learning Research, 2008, 9: 2579-2605.

Biography of the authors

Xuesong Wang received the PhD degree from China University of Mining and Technology in 2002. She is currently a professor in the School of Information and Control Engineering, China University of Mining and Technology. Her main research interests include machine learning, bioinformatics, and artificial intelligence. Email: wangxuesongcumt@163.com

Yuting Ma received the bachelor degree from Harbin Engineering University in 2014. She is currently a master candidate in the School of Information and Control Engineering, China University of Mining and Technology. Her main research interest is transfer learning. Email: mayuting0703@yeah.net

Yuhu Cheng [M'15] received the PhD degree from the Institute of Automation, Chinese Academy of Sciences in 2005. He is currently a professor in the School of Information and Control Engineering, China University of Mining and Technology. His main research interests include machine learning, transfer learning, and intelligent system. Email: chengyuhu@163.com



Liang Zou received the B.Sc. degree in microelectronics from Anhui University, China, in 2010, and the M.Sc. degree in biomedical engineering from the University of Science and Technology of China, in 2013. He is currently pursuing the Ph.D. degree with the University of British Columbia. His current research interest includes biomedical signal processing and bioinformatics. Email: liangzou@ece.ubc.ca

Joel J.P.C. Rodrigues [S'01, M'06, SM'06] is a professor and senior researcher at the National Institute of Telecommunications (Inatel), Brazil and senior researcher at the Instituto de Telecomunicações, Portugal. He is the leader of NetGNA Research Group (http://netgna.it.ubi.pt), Past Chair of the IEEE ComSoc TCs on eHealth and Communications Software, and a Steering Committee member of the IEEE Life Sciences Technical Community. He is the Editor-in-Chief of three international journals, and a co-author of over 500 papers, three books, and two patents. He is the recipient of several Outstanding Leadership and Outstanding Service Awards by IEEE Communications Society and several best paper awards. Email: joeljr@ieee.org

Highlights of the paper

- A novel semi-supervised heterogeneous domain adaptation based on deep learning is proposed.
- The marginal and conditional distributions are simultaneously matched by MMD on the shared feature layer.
- The geometrical term based on graph is used to maintain consistency of the geometrical structure of inner domain data.
- The terms of similarity and dissimilar are used to maintain consistency of label information across domains.