To appear in the *Journal of Experimental & Theoretical Artificial Intelligence*
Vol. 00, No. 00, Month 20XX, 1–23

Domain transfer convolutional attribute embedding

Fang Su[a*] , Jing-Yan Wang[b]

[a] *School of Economics and Management, Shaanxi University of Science & Technology, Xi'an, ShaanXi Province, P.R.C, 710021*

[b] *New York University Abu Dhabi, Abu Dhabi, United Arab Emirates*

(*v5.0 released July 2015*)

In this paper, we study the problem of transfer learning with the attribute data. In the transfer learning problem, we want to leverage the data of the auxiliary and the target domains to build an effective model for the classification problem in the target domain. Meanwhile, the attributes are naturally stable cross different domains. This strongly motives us to learn effective domain transfer attribute representations. To this end, we proposed to embed the attributes of the data to a common space by using the powerful convolutional neural network (CNN) model. The convolutional representations of the data points are mapped to the corresponding attributes so that they can be effective embedding of the attributes. We also represent the data of different domains by a domain-independent CNN, ant a domain-specific CNN, and combine their outputs with the attribute embedding to build the classification model. An joint learning framework is constructed to minimize the classification errors, the attribute mapping error, the mismatching of the domain-independent representations cross different domains, and to encourage the the neighborhood smoothness of representations in the target domain. The minimization problem is solved by an iterative algorithm based on gradient descent. Experiments over benchmark data sets of person re-identification, bankruptcy prediction, and spam email detection, show the effectiveness of the proposed method.

Keywords: Transfer Learning; Attribute Embedding; Convolutional Neural Network; Bankruptcy Prediction

1. Introduction

1.1. *Backgrounds*

In the machine learning problems, domain transfer learning has recently attracted much attention (Lopez-Sanchez, Arrieta and Corchado, 2018; Wang, Song, Marquez-Lago, Leier, Li, Lithgow, Webb and Shen, 2017; Yang and Zhang, 2017; Zhang, Yang and Zhang, 2017a). Transfer learning refers to the learning problem of a predictive model for a target domain, by leveraging the data from both the target domain and one or more auxiliary domains. The target domain is in lack of class labels, which makes the learning in the target domain difficult. The auxiliary domains have the same input space and the label space, however, the data distribution of the auxiliary domains are significantly different from the target domain, thus the auxiliary domain data cannot be directly used to learn the model in the target domain. To solve this problem, domain-transfer learning is proposed to transfer the data representation and/or the models of the auxiliary domains to fit the data of the target domain, so that the classification performance of the

*Corresponding author.

target domain can be improved. For example, in the problem of person re-identification, to identify one person captured by one camera, we learn a classifier to predict the ID of the image of a person. Usually, there are more than one cameras, and we can use the data of different cameras to help the learning for one target camera. Because of the angle of different cameras are different, the data of the multiple cameras cannot be directly combined. Thus the transfer learning technology is needed to leverage the gaps between the cameras (An, Chen and Yang, 2017; Hassen, Loukil, Ouni and Jallouli, 2018; Ibn Khedher, El-Yacoubi and Dorizzi, 2017; Zhao, Wang, Wong, Zheng, Yang and Miao, 2017).

One shortage of traditional transfer learning methods is the that the attributes of the data are not used by the classification model. But the attributes of the data actually has the nature of stability across the domains. For example, in the problem of person re-identification, because of the change of the angles of the cameras, the appearances of the same person in different cameras may change, the attributes usually keep stable, such as the attribute of long hair, wearing short pans, and/or carrying a bag. Thus using the attributes of the data is critical for the transfer learning (Kulkarni, Sharma, Zepeda and Chevallier, 2014; Peng, Tian, Xiang, Wang, Pontil and Huang, 2017; Suzuki, Sato, Oyama and Kurihara, 2014a,b). In this paper, we study the problem of effective use of both input data and attribute for the domain-transfer learning problem, and propose a novel method of attribute embedding based on the popular convolutional neural network (CNN) (Fujino, Hatanaka, Mori and Matsumoto, 2018; Jing, Zhao, Li and Xu, 2017; Puri, Tewari, Katyal and Garg, 2018; Roa-Barco, Serradilla-Casado, Velasco-Vzquez, Lpez-Zorrilla, Graa, Chyzhyk and Price, 2018; Todoroki, Han, Iwamoto, Lin, Hu and Chen, 2018; Waijanya and Promrit, 2018) to solve this problem. Further, we develop a novel model using the attribute embedding as the input for the learning of the target domain classification model.

1.2. *Relevant works*

Our work is an effective representation method of attributes for the problem of transfer learning problem. However, there only two existing works in this direction, and we introduce them as follows.

- Peng et al. (Peng et al., 2017) proposed to represent the attribute vectors of each data point by using an attribute dictionary. Each data point is reconstructed by the elements in the dictionary, and the reconstruction coefficients are used as the new representation of the attributes. The attribute vector of a data point is mapped to the new representation vector by a linear transformation matrix so that the new representation vector is linked to the attribute vector. To leverage the auxiliary and the target domains, the same attribute representation method is applied to both auxiliary and target domains. The learning process is regularized by the class-intra similarity in the auxiliary domains, and by the neighborhood in the target domain.
- Su et al. (Su, Yang, Zhang, Tian, Davis and Gao, 2017) proposed a low-rank attribute embedding method for the problem of person re-identification of multiple cameras. The proposed method tries to solve the problem of multiple cameras based person re-identification as a multi-task learning problem. The proposed method uses both the low-level features with mid-level attributes as the input of the identification model. The embedding of attributes maps the attributes to a continuous space to explore the correlative relationship between each pair of attributes and also recovers the missing attributes.

Both these two methods of attribute representation are based on the linear transformation. However, a simple linear function may be insufficient to represent the attributes effectively. In the domain transfer area, a group of methods (Ding and Fan, 2013, 2015; Ding, Fan, Zhang, Ge and Chou, 2012; Zhang, Ding and Fan, 2017b) embedded a physical structure of high dimensional data into another domain with low dimensionality using a non-linear mapping, which is trained by balancing the effect of data and a heuristic physical prior. These methods inspired us to embed the attributes of the data to a common space.

1.3. *Our contribution*

In this paper, we propose a novel attribute embedding method for attributes for the problem of domain transfer learning. The embedding of attributes is based on CNN model. The convolutional output of the input data is further mapped to the attribute vector. In this way, the attribute embedding vector not only represents the attributes of a data point but also contains the pattern of the input data constructed by the CNN model, which has been proven to be a powerful representation model. To construct the classification model for each domain, we also learn a domain-independent convolutional representation and a domain-specific convolutional representation. The domain-independent convolutional representation maps the data of different domains to a shared data space to capture the patterns shared over all the domain. The domain-specific convolutional representation is used to represent the patterns specifically contained by each domain. The classification model of each domain is based on the three types of convolutional representations, i.e., attribute embedding, domain-independent and domain-specific representations. To learn the parameters of the models, we propose to minimize the mapping errors of the attributes, the classification errors across different domains, the mismatching of different domains in the domain-independent representation space, and the dissimilarity between the neighboring data points in the target domain. The joint minimization problem is solved by an alternate optimization strategy and the gradient descent algorithm.

1.4. *Paper organization*

This paper is organized as follows. In section 2, we introduce the proposed model and the learning method of the parameters of the model. In section 3, we test the proposed method over some benchmark datasets, compare it to the state-of-the-art methods. In section 4, we give the conclusion of this paper.

2. Method

In this section, we will introduce the proposed method of attribute embedding and cross-domain learning. The proposed model and the corresponding learning problem is firstly introduced, and then the optimization method is developed to solve the learning problem. Finally, we give an iterative algorithm based on the optimization results.

2.1. *Problem embedding*

In the problem setting of cross-domain learning, we assume we have T domains. The first $T-1$ domain are the auxiliary domains, while the T-th domain is the target domain.

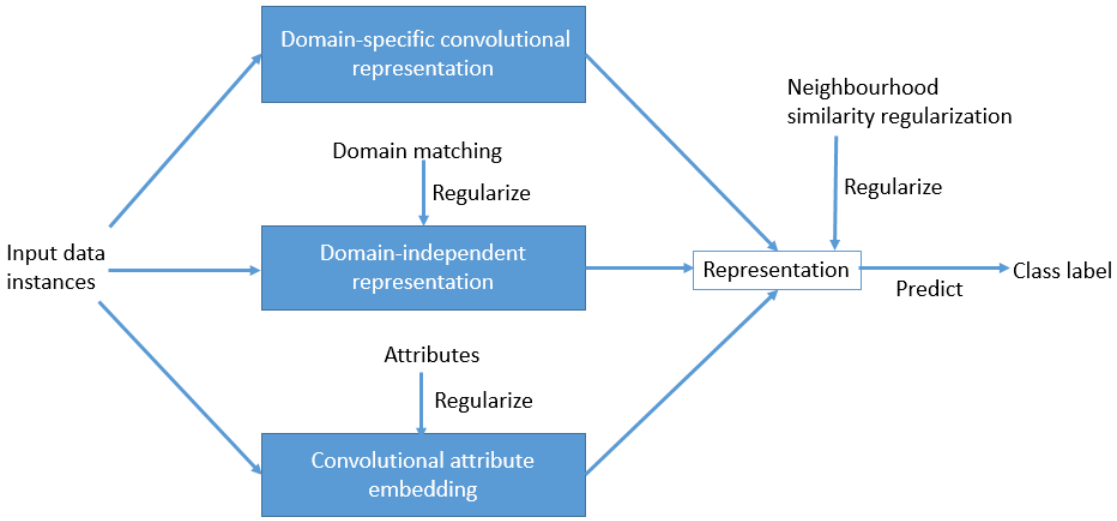

Figure 1. Learning framework of the proposed attribute embedding and classification model.

The problem is to learn an effective model for the classification of target domain. The input data for the training is given as follows.

- The input data sets of the T domains are denoted as $\mathcal{X}^t|_{t=1}^T$, where $\mathcal{X}^t = \{X_i^t\}$ is the data set of the t-th domain, and $X_i^t = \left[\mathbf{x}_{i1}^t, \cdots, \mathbf{x}_{i|X_i^t|}^t\right] \in R^{d \times |X_i^t|}$ is the input matrix of the i-th data point of the t-th domain, and each column of the matrix is a feature vector of a instance.
- Moreover, for each data point X_i^t, an attribute vector is attached, $\mathbf{a}_i^t = \left[a_{i1}^t, \cdots, a_{i|a|}^t\right] \in \{1,0\}^{|a|}$, where $a_{ij}^t = 1$ if the i-th data point of the t-th domain has the j-th attribute, and 0 otherwise.
- Meanwhile, all the data points of the auxiliary domains, and a small part of the data points of the target domain has the class label vector. For a data point, X_i^t, a class label vector $\mathbf{y}_i^t = \left[\mathbf{y}_{i1}^t, \cdots, \mathbf{y}_{i|y|}^t\right] \in \{1,0\}^{|y|}$, where $\mathbf{y}_{ij}^t = 1$ if X_i^t belongs to the j-th class, and 0 otherwise.

Our learning framework is shown in Figure 1. As shown in the figure, our model is composed of three convolutional representation sub-models, namely the convolutional attribute embedding model, the domain-independent representation model, and the domain-specific convolutional representation model. The convolutional attribute embedding model is used to leverage the input instances and the attributes, thus its learning is regularized by the attributes of the given data points. The domain-independent representation is a model for the data of all domains, and its main function is to map the data of different domains to a common space, thus it is regularized by the domain-matching term. The domain-specific convolutional representation is designed for different domains to handle the discrepancy of data of different domains. The representation of an input data point is the combination of the outputs of the three models, and it is used to predict the class label, and meanwhile, it is also regularized the neighborhood in the target domain. Accordingly, to construct the classification model, we consider the following problems.

Convolutional attribute embedding We propose to embed the attributes of each input data point to a vector, and use the convolutional representation of the input data

as the embedding vector. The embedding vectors will be further used as input of the classification model. Given the the input matrix of a data point, $X = [\mathbf{x}_1, \cdots, \mathbf{x}_{|X|}]$, to obtain its convolutional representation, we have a four-step process:

(1) We first use a sliding window of α instances, and concatenate the instances within the window to a new vector $\mathbf{z} \in R^{\alpha d}$. With the sliding window moving from the beginning to the end with a step of 1 instance, we obtain the new input data matrix,

$$Z = [\mathbf{z}_1, \cdots, \mathbf{z}_{|X|+\alpha-1}], \tag{1}$$

where $\mathbf{z}_i = \begin{bmatrix} \mathbf{x}_i \\ \vdots \\ \mathbf{x}_{i+\alpha-1} \end{bmatrix} \in R^{\alpha d}$ is the output of the i-th step of sliding. It is the concatenation of vectors of the i-th instance to the $i+\alpha-1$-th instance.

(2) Then a filter matrix $W_a \in R^{(\alpha d) \times m}$ is applied to the new input matrix, where each column of the filter matrix is a filter vector. The filtering result is the multiplication between $W_a^\top$ and Z, $W_a^\top Z$.

(3) Filtering is followed by an activation operation. In the activation operation, each element of the input matrix is transformed by a non-linear activation function, which is defined as the Rectified Linear Units (ReLU) $g(x) = \max(0, x)$. The output of activation operation is denoted as $g(W_a^\top Z)$.

(4) The last step is max-pooling. Given the output of the activation operation, we select the maximum element from each row, the output is denoted as $\max\left(g(W_a^\top Z)\right)$, where $\max(X)$ is a row-wise maximization operator.

The overall output of the convolutional representation can be obtained by the chain function of the four steps, denoted as

$$\begin{aligned} f_a(X) &= \max\left(g(W_a^\top Z)\right) \\ &= \begin{bmatrix} \max(g(\mathbf{w}_{a1}^\top Z)) \\ \vdots \\ \max(g(\mathbf{w}_{am}^\top Z)) \end{bmatrix} \end{aligned} \tag{2}$$

Since this convolutional representation of X is used as its attribute vector embedding, we propose to link it to the attribute vector $\mathbf{a}$ by a linear mapping function,

$$f_a(X) \leftarrow \Theta^\top \mathbf{a}, \tag{3}$$

where $\Theta \in R^{|a| \times m}$ is the mapping matrix. The convolutional representation of X is generated from its original data instances, meanwhile it is a mapping of the attributes of the input data. In this way, the embedding of the attributes only has the property of the attribute properties themselves, but also relies on the effective convolutional representation of the input data itself. Thus the embedding leverage the convolutional representation and the attributes well.

To reduce the mapping errors, we proposed to minimize the Frobenius norm distance between the convolutional representations and the mapping results for all the data points

of all domains,

$$\min_{\Theta, W_a} \sum_{t=1}^{T} \left(\sum_{i=1}^{n_t} \left\| f_a(X_i^t) - \Theta \mathbf{a}_i^t \right\|_F^2 \right), \tag{4}$$

where Θ is the mapping matrix of the linear mapping function for attribute vectors. By minimizing this objective, we obtain an effective embedding of the attributes of the input data. Please note that the embedding is for the attribute vector, instead for each of the attribute element.

Domain-independent representation To predict the class labels for the data points in multiple domains, we proposed the data of each domain to represent the data into a base convolutional representation, and a domain-specific convolutional representing. The base convolutional representation function is shared across all the domains. It tries to extract features relevant to the class labels, but independent of the specific domain. The base convolutional recreation function is also based on the sliding window, filtering, activation, and max-pooling. The base convolutional representation of X is defined as,

$$f_0(X) = \max\left(g(W_0^\top Z) \right), \tag{5}$$

where Z is the output of sliding window, and W_0 is the filter matrix of the base convolutional representation function.

Since the base convolutional representation is domain-independent, we hope the representations of data points from different domains can be similarity to each other. To this end, we impose that the distribution of the base representations of different domains is of the same. We use the mean vector of the representations of each domain as the presentation of the distribution of the domain. For the t-th domain, the mean vector is given as $\frac{1}{n_t} \sum_{i=1}^{n_t} f_0(X_i^t)$. To reduce the mismatch among the domains, we proposed to minimize the Frobenius norm distances between the mean vectors of each pair of domains.

$$\min_{W_0} \sum_{t,t'=1, t<t'}^{T} \left\| \frac{1}{n_t} \sum_{i=1}^{n_t} f_0(X_i^t) - \frac{1}{n_{t'}} \sum_{i=1}^{n_{t'}} f_0(X_i^{t'}) \right\|_F^2. \tag{6}$$

By minimizing this problem, we hope the base convolutional representation function can map data points of different domain to a common shared data space.

Cross-domain class label estimation To predict the class labels for the data points of different domains, we also consider the representation of the data points according to the domains. This is the domain-specific representations. The representation is also based on convolutional network function, and the function of the t-th domain of a data point X is given as,

$$f_t(X) = \max\left(g(W_t^\top Z) \right), \tag{7}$$

where W_t is the filter matrix of the t-th domain specific convolutional representations.

To estimate the class label from a data point of the t-th domain, we combine both the domain-independent and domain-specific convolutional representations of the input

data, $f_0(X)$ and $f_t(X)$, and also the attribute embedding of the data, $f_a(X)$. They are concatenated to a longer vector,

$$f(X) = \begin{bmatrix} f_0(X) \\ f_t(X) \\ f_a(X) \end{bmatrix} \in R^{m_0+m_t+m_a}, \tag{8}$$

and the longer vector is transformed to a $|y|$-dimensional vector of scores of classification by a matrix $U = \begin{bmatrix} U_0 \\ U_t \\ U_a \end{bmatrix} \in R^{(m_0+m_t+m_a)\times|y|}$ in a classification function,

$$h_t(X) = U^\top f(X) = U_0^\top f_0(X) + U_t^\top f_t(X) + U_a^\top f_a(X), t = 1, \cdots, T, \tag{9}$$

where U_0, U_t, and U_a are the transformation matrices for the domain-independent representation, domain-specific representation, and the attribute embedding. The classification function is used to predict the class labels, thus we propose to reduce the prediction errors measured by the Frobenius norm distance between the class label vectors and the outputs of $h_t(X)$ for the data pints with available label vectors,

$$\min_{U_t, W_t|_{t=0}^T, U_a, W_a} \left\{ \sum_{t=1}^{T-1} \left(\sum_{i=1}^{n_t} \|\mathbf{y}_i^t - h_t(X_i^t)\|_F^2 \right) + \sum_{i=1}^{l_T} \|\mathbf{y}_i^T - h_t(X_i^T)\|_F^2 \right\}. \tag{10}$$

Please in the objective function of this problem, for the auxiliary domains, all the data points are labeled, but for the target domain, only the first l_T data points are labeled. Thus for the target domain, we only consider the first l_T data points.

Neighbourhood similarity regularization For the unlabeled data points in the target domain, we also regularize them by imposing their representations to be constant with the labeled data points in the neighborhood, so that the supervision information can also be propagated to them. To this end, we hope for any neighboring two data points in the target domain, their overall representation vectors are close to each other. We propose to minimize the Frobenius norm distance between the representations of neighboring data points in the target domain,

$$\min_{U_T, W_T, U_a, W_a} \sum_{i,i'=1}^{n_T} M_{ii'} \left\| f(X_i^T) - f(X_{i'}^T) \right\|_F^2, \tag{11}$$

where $M_{ii'} = 1$ if X_i and $X_{i'}$ are neighbor to each other, and 0 otherwise. In this way, if a data point is not labeled, but its representation is also regularized by the other representations of the target domain, especially the representations of the labeled data points. Thus the learning of the unlabeled data points is also benefiting from the labels.

2.2. *Problem optimization*

The learning framework is constructed by combining the learning problems mentioned above,

$$\min_{U_t,W_t|_{t=0}^{T},U_a,W_a,\Theta}\Bigg\{o(U_t,W_t|_{t=0}^{T},U_a,W_a,\Theta)=\sum_{t=1}^{T-1}\left(\sum_{i=1}^{n_t}\|\mathbf{y}_i^t-h_t(X_i^t)\|_F^2\right)+\sum_{i=1}^{l_T}\|\mathbf{y}_i^T-h_t(X_i^T)\|_F^2+C_1\sum_{t=1}^{T}\left(\sum_{i=1}^{n_t}\left\|f_a(X_i^t)-\Theta\mathbf{a}_i^t\right\|_F^2\right)+C_2\sum_{t,t'=1,t<t'}^{T}\left\|\frac{1}{n_t}\sum_{i=1}^{n_t}f_0(X_i^t)-\frac{1}{n_{t'}}\sum_{i=1}^{n_{t'}}f_0(X_i^{t'})\right\|_F^2+C_3\sum_{i,i'=1}^{n_T}M_{ii'}\left\|f(X_i^T)-f(X_{i'}^T)\right\|_F^2\Bigg\},\tag{12}$$

where o is the objective function, and $C_k, k=1,\cdots,3$ are the tradeoff weights of different regularization terms. This joint learning framework can learn the effective representations of the input data of different domains. The three types of representations are all based on convolutional networks. The attribute embedding are regularized the attribute vectors. The domain-independent representations are regularize by both the labels and the mismatching of the distributions of different domains. The domain-specific representations are only regularized by the labels of the corresponding domains. For the target domain, all the representations are regularized by the neighborhood structure.

To solve this problem, we proposed to use the alternate optimization method. When one parameter is being optimized, others are fixed. In an iterative algorithm, the parameters are updated to optimize the problem alternately. In the following sections, we will discuss how to update the parameters respectively.

2.2.1. Optimization of filters of $f_a(X)$

When the filters of W_a is optimized, we substitute (2), (8) and (9) to (12), and remove all the terms which are irrelevant to W_a, the problem is reduced to

$$\min_{W_a}\Bigg\{o_1(W_a)=\sum_{t=1}^{T-1}\left(\sum_{i=1}^{n_t}\|U_a^\top f_a(X_i^t)-\left(\mathbf{y}_i^t-U_0^\top f_0(X_i^t)+U_t^\top f_t(X_i^t)\right)\|_F^2\right)+\sum_{i=1}^{l_T}\|U_a^\top f_a(X_i^T)-\left(\mathbf{y}_i^T-U_0^\top f_0(X_i^T)+U_T^\top f_T(X_i^T)\right)\|_F^2+C_1\sum_{t=1}^{T}\left(\sum_{i=1}^{n_t}\left\|f_a(X_i^t)-U\mathbf{a}_i^t\right\|_F^2\right)C_3\sum_{i,i'=1}^{n_T}M_{ii'}\left(\left\|f_a(X_i^T)-f_a(X_{i'}^T)\right\|_F^2\right)\Bigg\}.\tag{13}$$

To update the filters of attribute embedding function, we use the coordinate gradient descent algorithm. In this algorithm, the filters are updated sequentially. When one filter is updated, others are fixed. When the k-th filter $\mathbf{w}_{ak}$ (the k-th column of W_a) is considered, we update it according to the direction of the gradient of $o_1(W_a)$ regarding $\mathbf{w}_{ak}$. The gradient function is given as following according to chain rule,

$$\begin{aligned}\frac{\partial o_1}{\partial \mathbf{w}_{ak}} &= 2\sum_{t=1}^{T-1}\sum_{i=1}^{n_t}\left[U_a\left(U_a^\top f_a(X_i^t) - \left(\mathbf{y}_i^t - U_0^\top f_0(X_i^t) + U_t^\top f_t(X_i^t)\right)\right)\right]_k \frac{\partial f_a(X_i^t)_k}{\partial \mathbf{w}_{ak}} \\ &+ 2\sum_{i=1}^{l_T} U_a\left[U_a^\top f_a(X_i^T) - \left(\mathbf{y}_i^T - U_0^\top f_0(X_i^T) + U_T^\top f_T(X_i^T)\right)\right]_k \frac{\partial f_a(X_i^T)_k}{\partial \mathbf{w}_{ak}} \\ &+ 2C_1\sum_{t=1}^{T}\sum_{i=1}^{n_t}\left[f_a(X_i^t) - U\mathbf{a}_i^t\right]_k \frac{\partial f_a(X_i^t)_k}{\partial \mathbf{w}_{ak}} \\ &+ 2C_3\sum_{i,i'=1}^{n_T} M_{ii'}\left[f_a(X_i^T) - f_a(X_{i'}^T)\right]_k \frac{\partial f_a(X_i^T)_k}{\partial \mathbf{w}_{ak}},\end{aligned} \tag{14}$$

where $[\mathbf{f}]_k$ is the k-th element of the vector of $\mathbf{f}$,

$$\begin{aligned}&\frac{\partial f_a(X)_k}{\partial \mathbf{w}_{ak}} = \nabla g\left(\mathbf{w}_{ak}^\top \mathbf{z}_{j^*}\right)\mathbf{z}_{j^*}, \nabla g(x) = \begin{cases}1, & if\ x > 0\\ 0, & otherwise,\end{cases} \ and \\ &j^* = \arg\max_j(g(\mathbf{w}_{ak}^\top \mathbf{z}_i))),\end{aligned} \tag{15}$$

The updating rule of $\mathbf{w}_{ak}$ is as follows,

$$\mathbf{w}_{ak} \leftarrow \mathbf{w}_{ak} - \tau\frac{\partial o_1}{\partial \mathbf{w}_{ak}}, \tag{16}$$

where τ is the descent step.

2.2.2. Optimization of filters of $f_0(X)$

To optimize the filters of the domain-independent convolutional representation function, $f_0(X)$, we also fix other parameters and remove the irrelevant terms. The following problem is obtained,

$$\begin{aligned}\min_{W_0}\Bigg\{ o_2(W_0) = &\sum_{t=1}^{T-1}\sum_{i=1}^{n_t} \|U_0^\top f_0(X_i^t) - \left(\mathbf{y}_i^t - U_a^\top f_a(X_i^t) + U_t^\top f_t(X_i^t)\right)\|_F^2 \\ &+ \sum_{i=1}^{l_T} \|U_0^\top f_0(X_i^T) - \left(\mathbf{y}_i^T - U_a^\top f_a(X_i^T) + U_T^\top f_T(X_i^T)\right)\|_F^2 \\ &+ C_2 \sum_{t,t'=1,t<t'}^{T} \left\| \frac{1}{n_t}\sum_{i=1}^{n_t} f_0(X_i^t) - \frac{1}{n_{t'}}\sum_{i=1}^{n_{t'}} f_0(X_i^{t'}) \right\|_F^2 \\ &+ C_3 \sum_{i,i'=1}^{n_T} M_{ii'} \left\| f_0(X_i^T) - f_0(X_{i'}^T) \right\|_F^2 \Bigg\}.\end{aligned} \tag{17}$$

Similarly to the optimization of filters of $f_a(X)$, we also use the coordinate gradient descent algorithm to update the filters of $f_0(X)$. When a filter $\mathbf{w}_{0k}$ is considered, we calculate the sub-gradient function of $o_2(W_0)$ with regard to $\mathbf{w}_{0k}$ as follows,

$$\begin{aligned}\frac{\partial o_2}{\partial \mathbf{w}_{0k}} = &2\sum_{t=1}^{T-1}\sum_{i=1}^{n_t} \left[U_0\left(U_0^\top f_0(X_i^t) - \left(\mathbf{y}_i^t - U_a^\top f_a(X_i^t) + U_t^\top f_t(X_i^t)\right)\right)\right]_k \frac{\partial f_0(X_i^t)}{\partial \mathbf{w}_{0k}} \\ &+2\sum_{i=1}^{l_T} \left[U_0\left(U_0^\top f_0(X_i^T) - \left(\mathbf{y}_i^T - U_a^\top f_a(X_i^T) + U_T^\top f_T(X_i^T)\right)\right)\right]_k \frac{\partial f_0(X_i^T)}{\partial \mathbf{w}_{0k}} \\ &+ 2C_2 \sum_{t,t'=1,t<t'}^{T} \left[\frac{1}{n_t}\sum_{i=1}^{n_t} f_0(X_i^t) - \frac{1}{n_{t'}}\sum_{i=1}^{n_{t'}} f_0(X_i^{t'})\right]_k \\ &\left(\frac{1}{n_t}\sum_{i=1}^{n_t}\frac{\partial f_0(X_i^t)}{\partial \mathbf{w}_{0k}} - \frac{1}{n_{t'}}\sum_{i=1}^{n_{t'}}\frac{\partial f_0(X_i^{t'})}{\partial \mathbf{w}_{0k}}\right) \\ &+ 2C_3 \sum_{i,i'=1}^{n_T} M_{ii'} \left[f_0(X_i^T) - f_0(X_{i'}^T)\right]_k \left(\frac{\partial f_0(X_i^T)}{\partial \mathbf{w}_{0k}} - \frac{\partial f_0(X_{i'}^T)}{\partial \mathbf{w}_{0k}}\right)\end{aligned}, \tag{18}$$

where $\frac{\partial f_0(X)}{\partial \mathbf{w}_{0k}}$ is defined as the same as (15), and the update rule of $\mathbf{w}_{0k}$ is

$$\mathbf{w}_{0k} \leftarrow \mathbf{w}_{0k} - \tau \frac{\partial o_2}{\partial \mathbf{w}_{0k}}. \tag{19}$$

2.2.3. Optimization of filters of $f_t(X), t = 1, \cdots, T-1$

To update the filters of a domain-specific convolutional representation function of an auxiliary domain, $f_t(X), t = 1, \cdots, T-1$, we have the following optimization problem by fixing other parameters and removing the irrelevant terms,

$$\min_{W_t}\left\{ o_3(W_t) = \sum_{i=1}^{n_t} \| \left(U_t^\top f_t(X_i^t) - \left(\mathbf{y}_i^t - U_a^\top f_a(X_i^t) + U_0^\top f_0(X_i^t)\right)\right)\|_F^2 \right\}. \tag{20}$$

To optimize the filters, we also use the coordinate gradient descent algorithm. The gradient descent function of o_3 with regard to a filter of W_t, $\mathbf{w}_{tk}$ is given as follows,

$$\frac{\partial o_3}{\partial \mathbf{w}_{tk}} = 2\sum_{i=1}^{n_t}\left[U_t\left(U_t^\top f_t(X_i^t) - \left(\mathbf{y}_i^t - U_a^\top f_a(X_i^t) + U_0^\top f_0(X_i^t)\right)\right)\right]_k \frac{\partial f_t(X_i^t)}{\partial \mathbf{w}_{tk}}, \tag{21}$$

where $\frac{\partial f_t(X)}{\partial \mathbf{w}_{tk}}$ is defined as same as (15). Accordingly, $\mathbf{w}_{tk}$ is updated as

$$\mathbf{w}_{tk} \leftarrow \mathbf{w}_{tk} - \tau\frac{\partial o_3}{\partial \mathbf{w}_{tk}}. \tag{22}$$

2.2.4. Optimization of filters of $f_T(X)$

To update the filters of the target domain-specific convolutional representation function, $f_T(X)$, we only consider the terms of the objective function which are relevant to $f_T(X)$, and fix other parameters. The following optimization problem is obtained,

$$\begin{aligned}\min_{W_T}\Bigg\{ o_4(W_T) = &\sum_{i=1}^{l_T}\|U_T^\top f_T(X_i^T) - \left(\mathbf{y}_i^T - U_a^\top f_a(X_i^T) + U_0^\top f_0(X_i^T)\right)\|_F^2 \\ &+C_3\sum_{i,i'=1}^{n_T} M_{ii'}\left\|f_T(X_i^T) - f_T(X_{i'}^T)\right\|_F^2\Bigg\},\end{aligned} \tag{23}$$

and its gradient function with regard to the k-th filter is

$$\begin{aligned}\frac{\partial o_4}{\partial \mathbf{w}_{Tk}} = &2\sum_{i=1}^{l_T}\left[U_T\left(U_T^\top f_T(X_i^T) - \left(\mathbf{y}_i^T - U_a^\top f_a(X_i^T) + U_0^\top f_0(X_i^T)\right)\right)\right]_k \frac{\partial f_T(X_i^T)}{\partial \mathbf{w}_{Tk}} \\ &+ 2C_3\sum_{i,i'=1}^{n_T} M_{ii'}\left[f_T(X_i^T) - f_T(X_{i'}^T)\right]_k\left(\frac{\partial f_T(X_i^T)}{\partial \mathbf{w}_{Tk}} - \frac{\partial f_T(X_{i'}^T)}{\partial \mathbf{w}_{Tk}}\right).\end{aligned} \tag{24}$$

Accordingly, the update rule of $\mathbf{w}_{Tk}$ is

$$\mathbf{w}_{Tk} \leftarrow \mathbf{w}_{Tk} - \tau\frac{\partial o_4}{\partial \mathbf{w}_{Tk}}. \tag{25}$$

2.2.5. Optimization of U_a, $U_t, t = 0, \cdots, T$

To optimize the transformation matrices U_a, U_0, and $U_t, t = 1, \cdots, T$, we only consider the following optimization problem,

$$\min_{U_a,U_0,U_1,\cdots,U_T} \Bigg\{ o_5(U_a,U_0,U_1,\cdots,U_T) = \sum_{t=1}^{T-1}\sum_{i=1}^{n_t} \|\mathbf{y}_i^t - \left(U_0^\top f_0(X_i^t) + U_t^\top f_t(X_i^t) + U_a^\top f_a(X_i^t)\right)\|_F^2 + \sum_{i=1}^{l_T} \|\mathbf{y}_i^T - \left(U_0^\top f_0(X_i^T) + U_t^\top f_t(X_i^T) + U_a^\top f_a(X_i^T)\right)\|_F^2 \Bigg\}. \tag{26}$$

- **Optimization of** U_a To optimize U_a, we rewrite the objective as follows,

$$\begin{aligned} &o_5(U_a,U_0,U_1,\cdots,U_T) = \left\|\Omega - U_a^\top F_a\right\|_F^2, \\ &where\ F_a = [\underbrace{f_a(X_1^1),\cdots,f_a(X_{n_1}^1)}_{n_1},\cdots,\underbrace{f_a(X_1^T),\cdots,f_a(X_{l_T}^T)}_{l_T}], \\ &\Omega = [\underbrace{\boldsymbol{\omega}_1^1,\cdots,\boldsymbol{\omega}_{n_1}^1}_{n_1},\cdots,\underbrace{\boldsymbol{\omega}_1^T,\cdots,\boldsymbol{\omega}_{l_T}^T}_{l_T}], \\ &and\ \boldsymbol{\omega}_i^t = \mathbf{y}_i^t - \left(U_0^\top f_0(X_i^t) + U_t^\top f_t(X_i^t)\right). \end{aligned} \tag{27}$$

We set the derivative of the object regarding to the U_a to zero to obtain the optimal solution of F_a,

$$\begin{aligned} &\frac{\partial o_5}{\partial U_a} = -2F_a\Omega^\top + 2F_a{F_a}^\top U_a = 0 \\ &\Rightarrow U_a = \left(F_a F_a^\top\right)^{-1} F_a\Omega^\top. \end{aligned} \tag{28}$$

- **Optimization of** U_0 To optimize U_0, we rewrite the objective as follows,

$$\begin{aligned} &o_5(U_a,U_0,U_1,\cdots,U_T) = \left\|\Upsilon - U_0^\top F_0\right\|_F^2 \\ &where\ F_0 = [\underbrace{f_0(X_1^1),\cdots,f_0(X_{n_1}^1)}_{n_1},\cdots,\underbrace{f_0(X_1^T),\cdots,f_0(X_{l_T}^T)}_{l_T}], \\ &\Upsilon = [\underbrace{\boldsymbol{\upsilon}_1^1,\cdots,\boldsymbol{\upsilon}_{n_1}^1}_{n_1},\cdots,\underbrace{\boldsymbol{\upsilon}_1^T,\cdots,\boldsymbol{\upsilon}_{l_T}^T}_{l_T}], \\ &and\ \boldsymbol{\upsilon}_i^t = \mathbf{y}_i^t - \left(U_a^\top f_a(X_i^t) + U_t^\top f_t(X_i^t)\right). \end{aligned} \tag{29}$$

We set the derivative regarding U_0 to zero and obtain the solution of U_0,

$$\begin{aligned}&\frac{\partial o_5}{\partial U_0} = -2F_0\Upsilon^\top + 2F_0{F_0}^\top U_0 = 0\\&\Rightarrow U_0 = \left(F_0F_0^\top\right)^{-1}F_0\Upsilon^\top.\end{aligned} \tag{30}$$

- **Optimization of $U_t|_{t=1}^T$** To optimize U_t, we rewrite the objective as follows,

$$\begin{aligned}&o_5(U_a, U_0, U_1, \cdots, U_T)\\&= \sum_{i=1}^{n_t} \|\mathbf{y}_i^t - \left(U_0^\top f_0(X_i^t) + U_t^\top f_t(X_i^t) + U_a^\top f_a(X_i^t)\right)\|_F^2 + R_t\\&= \left\|\Pi_t - U_t^\top F_t\right\|_F^2 + R_t\\&where\ F_t = \left[f_t(X_1^t), \cdots, f_t(X_{n_t}^t)\right],\\&\Pi_t = \left[\boldsymbol{\pi}_1^t, \cdots, \boldsymbol{\pi}_{n_t}^t\right],\\&and\ \boldsymbol{\pi}_i^t = \mathbf{y}_i^t - \left(U_0^\top f_0(X_i^t) + U_a^\top f_a(X_i^t)\right).\end{aligned} \tag{31}$$

where R_t is combination of the terms which are irrelevant to U_t. By setting the derivative of the objective regarding U_t to zero, we obtain the solution of U_t,

$$\begin{aligned}&\frac{\partial o_5}{\partial U_t} = -2F_t\Upsilon^\top + 2F_t{F_0}^\top U_t = 0\\&\Rightarrow U_t = \left(F_tF_t^\top\right)^{-1}F_t{\Pi_t}^\top.\end{aligned} \tag{32}$$

2.2.6. Optimization of Θ

To optimize the attribute mapping matrix, Θ, we fix the other parameters and consider the following sub-optimization problem.

$$\begin{aligned}&\min_{\Theta}\left\{o_6(\Theta) = C_1\sum_{t=1}^{T}\left(\sum_{i=1}^{n_t}\left\|f_a(X_i^t) - \Theta^\top\mathbf{a}_i^t\right\|_F^2\right)\right.\\&\left.= C_1\left\|F_a - \Theta^\top A\right\|_F^2\right\},\\&where\ A = [\underbrace{\mathbf{a}_1^1, \cdots, \mathbf{a}_{n_1}^1}_{n_1}, \cdots, \underbrace{\mathbf{a}_1^T, \cdots, \mathbf{a}_{l_T}^T}_{l_T}].\end{aligned} \tag{33}$$

By setting the derivative of o_6 regarding to Θ to zero, we obtain the optimal solution of Θ as follows,

$$\begin{aligned} &\frac{\partial o_6}{\partial \Theta} = -2C_1 A F_a^\top + 2C_1 A A^\top \Theta = 0 \\ &\Rightarrow \Theta = \left(A A^\top\right)^{-1} A F_a^\top. \end{aligned} \tag{34}$$

2.3. *Details of algorithm implementation*

In this section, we describe the details of the iterative algorithm for learning the parameters of the convolutional attribute embedding model, and the details of the implementation. The detailed description of the iterative algorithm is given in Algorithm 1. In this algorithm, we update the filters of three convolutional layers, the transformation matrices, and the attribute transformation matrix, alternately. At the very beginning of the algorithm, we initialize the parameters and an objective function value to zeros. The updating processes are iterated until a maximum iteration number or the amount of decreasing of the objective value is smaller than a threshold. The flowchart of the iterative algorithm is given in Figure 2. The algorithm is implemented by Python programming language with Tensorflow supporting.

Algorithm 1 Iterative algorithm of MRSO.

Input: Training set of data points $\{(X_i^t, \mathbf{a}_i^t, \mathbf{y}_i^t)|_{i=1}^{n_t}\}|_{t=1}^T$;
Input: Tradeoff parameters C_1, C_2, and C_3;
Input: Maximum number of iterations, η;
Input: Objective value threshold, ε.
Initialize iteration indicator $\iota = 1$.
Initialize model parameters and objective value $o^0 = 0$.
while $\iota \leq \eta$ or objective value $|o^\iota - o^{\iota-1}| \leq \varepsilon$ **do**
  Update the filters of the convolutional attribute embedding model, f_a, according to (16).
  Update the filters of the domain-independent convolutional representation model, f_0, according to (19).
  for $i = t, \cdots, T-1$ **do**
    Update the filters of the domain-specific convolutional representation model of the t-th source domain, f_t, according to (22).
  end for
  Update the filters of the domain-specific convolutional representation model of the target domain, f_T, according to (25).
  Update the transformation matrix of attribute embedding model, U_a, according to (28).
  Update the transformation matrix of the domain-independent model, U_0, according to (30).
  Update the transformation matrix of the domain-specific model, $U_t|_{t=1}^T$, according to (32).
  Update the attribute mapping matrix, Θ, according to (34).
  Update the objective value o^ι according to (12).
  $\iota = \iota + 1$.
end while
Output: $\mathbf{w}^T$ and $z_1^T, \cdots, z_n^T$.

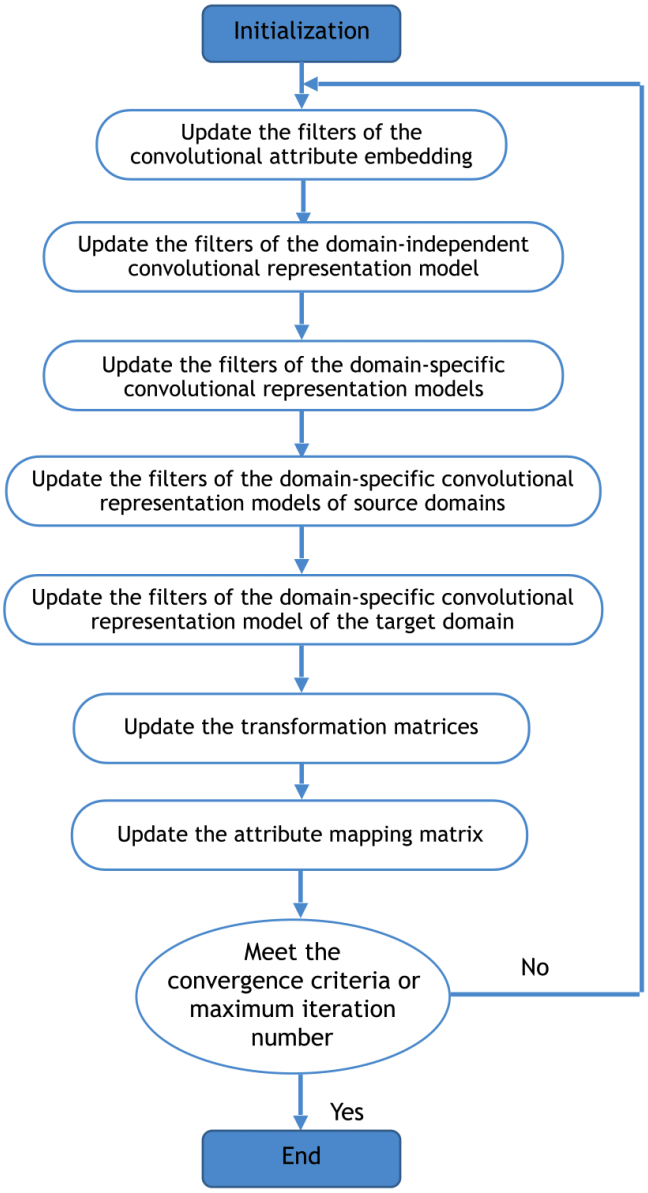

Figure 2. Flowchart of the iterative algorithm of MRSO.

3. Experiment

In this section, we evaluate the proposed method over several domain-transfer problems.

3.1. *Data sets*

In the experiments, we use the following six data sets.

- **CUHK03 data set** This data set was developed for the problem of person re-identification problems (Li, Zhao, Xiao and Wang, 2014). It contains 13,164 images of 1,360 persons. For each image, we annotate it by 108 attributes, including gender (male/female), wearing long hair, etc. The images are captured by six different

cameras. The problem of person re-identification is to train a classifier over the images of some cameras, and then use the classifier to identify an image captured from other cameras. We treat each camera as a domain, and we use each domain as a target domain in turn. The dataset can be downloaded from `http://www.ee.cuhk.edu.hk/~xgwang/CUHK_identification.html`. The attributes that we annotate in CUHK03 are listed in Table 1.

- **Market-1501 data set** This data set is for the problem of person re-identification Zheng, Shen, Tian, Wang, Wang and Tian (2015). It contains 32,668 images of detected persons. The number of classes (identities) is 1,501. The number of cameras is six, and the number of cameras of images for each person varies from 2 to 6. The attribute set for this set is the same as the CUHK03 data set.
- **iLIDS-VID data set** This data set is for the person re-identification problem Wang, Gong, Zhu and Wang (2014). It contains images of 300 individuals, and for each individual, the image from two cameras are collected, thus there are 600 image sets in total. The number of images in each image set varies from 23 to 192.
- **SAIVT-SoftBio data set** This data set of multiple camera person re-identification has images of 8 cameras Bialkowski, Denman, Sridharan, Fookes and Lucey (2012). It contains images of 152 different individuals in total. However, not all the individuals are detected in all the cameras, we only consider the persons who are detected in the 3-rd, 5-th, and 8-th cameras.
- **Bankrupt prediction set** This data contains the stock price wave data of 3 years of 374 companies of three different countries, China, USA, and UK. We collected this data for the problem of prediction of company bankrupt. Each company is also labeled by a list of business type attributes. To represent the price change wave of a company, we use a sliding window to split the wave into short-term frames and treat each frame as an instance. In this way, each company is treated as a data points, presented by a set of short-term frames, and a list of binary attributes of business types. Moreover, each country is treated as domain, thus we have three domains in our setting. The prediction problem of this data set is to predict if a given company will be in bankrupt within the future 3 years. Again, we treat each country as a target domain in turn and use the other two countries as auxiliary domains.
- **Spam email data set** This data set is for the spam email detection competition of the ECML/PKDD Discovery Challenge 2006 (Bickel, 2006). It contains texts of emails of 15 email users, and for each user, there are 400 emails. Among the 400 emails of each user, half of them are spam emails, while the remaining half are non-spam emails. Each email text is composed of a set of words. To present each email, we use the word embedding technology to obtain an embedding vector for each word of the email text, and thus each email is transformed to a set of embedding vectors, which is treated as instances in our model. Moreover, we also apply a topic classifier and a sentiment classifier to each email text to extract attributes of the text and use the extracted attributes as additional information. Each user is treated as a domain, and we also use each user as a target domain in turn.

The motive to use the data sets of person re-identification, bankruptcy prediction, and spam email detection is to show that the proposed method can be generalized to different types of applications, including computer vision, natural language processing, and economics.

Table 1. Attributes used to annotate CUHK03.

upperBodyRed	lowerBodyBrown	personalLess30
hairBrown	lowerBodyLogo	lowerBodyTrousers
footwearShoes	carryingNothing	upperBodyBlue
upperBodyBrown	upperBodyLogo	hairWhite
hairRed	footwearPurple	personalLarger60
hairGrey	upperBodyWhite	lowerBodyHotPants
carryingFolder	lowerBodyThinStripes	hairPurple
upperBodyThinStripes	lowerBodyShorts	accessoryHeadphone
footwearLeatherShoes	upperBodyPurple	footwearYellow
upperBodyGrey	lowerBodyOrange	accessorySunglasses
upperBodyLongSleeve	upperBodyOther	accessoryFaceMask
accessoryMuffler	upperBodyNoSleeve	footwearBlue
lowerBodyJeans	upperBodyOrange	upperBodyJacket
hairGreen	footwearPink	lowerBodyShortSkirt
personalLess45	upperBodyFormal	carryingUmbrella
footwearGreen	lowerBodyYellow	carryingBabyBuggy
footwearRed	lowerBodyLongSkirt	hairYellow
footwearBlack	lowerBodyWhite	lowerBodyGreen
upperBodyYellow	footwearSandals	hairLong
accessoryNothing	upperBodyThickStripes	upperBodyPlaid
carryingPlasticBags	upperBodyShortSleeve	hairShort
upperBodyCasual	accessoryKerchief	carryingSuitcase
footwearSneakers	footwearGrey	upperBodyVNeck
accessoryHat	hairOrange	personalLess60
accessoryHairBand	lowerBodySuits	upperBodySuit
upperBodyBlack	footwearWhite	lowerBodyGrey
carryingLuggageCase	lowerBodyCasual	upperBodyGreen
carryingOther	lowerBodyPurple	footwearOrange
upperBodyTshirt	lowerBodyRed	lowerBodyPlaid
lowerBodyFormal	lowerBodyBlack	personalLess15
upperBodyPink	lowerBodyPink	personalMale
hairBlack	carryingBackpack	footwearStocking
footwearBrown	hairBald	personalFemale
footwearBoots	accessoryShawl	lowerBodyBlue
carryingShoppingTro	lowerBodyCapri	carryingMessengerBag

3.2. *Experimental setting*

In our experiments, given a data set of several domains, we treat each domain as a target domain in turn, while treating the other domains as the auxiliary domains to help train the model. The data points in a target domain are further split into a training set and a

test set with equal sizes randomly. Meanwhile, for the training set of the target domain, we further split it into equal-sized subsets. One subset is used as a labeled set, and the other set is used as an unlabeled set. We train the model over the data points of the auxiliary domains and the training set of the target domain and then test it over the test set of the target domain. The classification rate over the test set is used as the performance measure. The average classification rate over different target domains is reported and compared. The average classification rate is computed as follows,

$$\begin{aligned} &average\ classification\ rate \\ &= \frac{1}{\#target\ domains} \sum_{t} \frac{\#correctely\ classified\ data\ points\ of\ the\ t-th\ domain}{\#total\ test\ data\ points\ of\ the\ t-th\ domain} \end{aligned} \tag{35}$$

3.3. *Results*

In the experiments, we first compare the proposed domain transfer convolutional attribute embedding (DTCAE) algorithm to some state-of-the-art domain-transfer attribute representation methods, and then study the properties of the proposed algorithm experimentally.

3.3.1. Comparison to state-of-the-art

Attribute embedding for domain-transfer learning problem is a new topic and there are only two existing methods. In the experiment, we compare the proposed algorithm against the two existing methods, which are the Joint Semantic and Latent Attribute Modelling (JSLAM) method proposed by Peng et al. (Peng et al., 2017), and the Multi-Task Learning with Low Rank Attribute Embedding (MTL-LORAE) method proposed by Su et al. (Su et al., 2017). The comparison results over the three benchmark data sets are shown in Figure 3. According to the reported average accuracies over the benchmark datasets, our algorithm DTCAE achieves the best performance over all the three datasets. For example, over the CUHK03 data set, the DTCAE is the only compared method which has an average accuracy higher than 0.800. Meanwhile, over the spam email dataset, only DTCAE obtains an average accuracy higher than 0.900. The reasons for our improvement over the compare methods are described in detail as follows.

- One reason for the improvement achieved by our model over the compared methods, JSLAM and MTL-LORAE, are the usage of convolutional attribute embedding layer. Both JSLAM and MTL-LORAE use simple linear functions to embed the attributes. These models heavily rely on the quality of the features extracted from the original data to represent the attributes. However, features are hand-crafted which is not specifically designed for the targeted attributes. However, our model is based on convolutional layers which use a group of filters to automatically extract features for the attributes recognition. The filters are adjusted to fit the attributes during the learning process. Compared to the linear model with hand-crafted features which ignores the attributes, the convolutional attribute embedding model can learn both the features and the attribute estimation function jointly. This makes the model more accurate for the attribute embedding.
- Another reason for the improvement is the usage of domain-independent and

domain-specific convolutional representation layers to extract the features shared by different domains and the features specific for each domain. However, JSLAM ignores the difference between features of different domains and uses a shared dictionary to represent hand-crafted features extracted from the original features. Compared to JSLAM, our model has the advantage of automatic feature learning, and effective domain feature extraction and sharing. This makes our model more suitable for the domain-transfer learning problem.

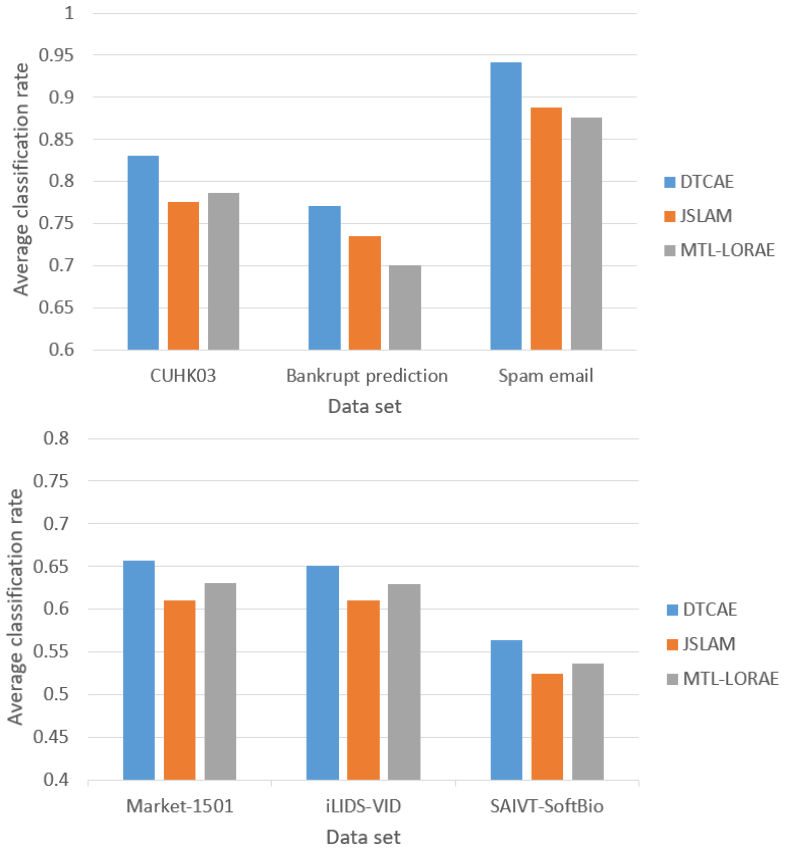

Figure 3. Comparison results over the benchmark data sets.

3.3.2. Sensitivity to tradeoff parameters

In the objective function of our model, there are three tradeoff parameters, C_1, C_2, and C_3. These parameters weight the importance of attribute embedding, domain-independent representation, and neighborhood similarity regularization. To verify the effect of these terms, we also study the performance of the proposed algorithm against different values of the tradeoff parameters. The sensitivity curves of to the tradeoff parameters are reported in Figure 4. As shown in the figure, for the C_1, our algorithm DTCAE is sensitive to the change of the value of C_1. When C_1 is increasing from 0.1 to 50, the average classification rates over all the three datasets increase significantly. This indicates the importance of the attributes embedding for the domain-transfer learning. However, it seems the proposed algorithm DTCAE is stable to the change of C_2. But a larger C_2 still achieves slightly better average classification rates. Finally, regarding C_3, we cannot observe a significant change when then values are varying. It seems a median

value of C_3 can give the best results.

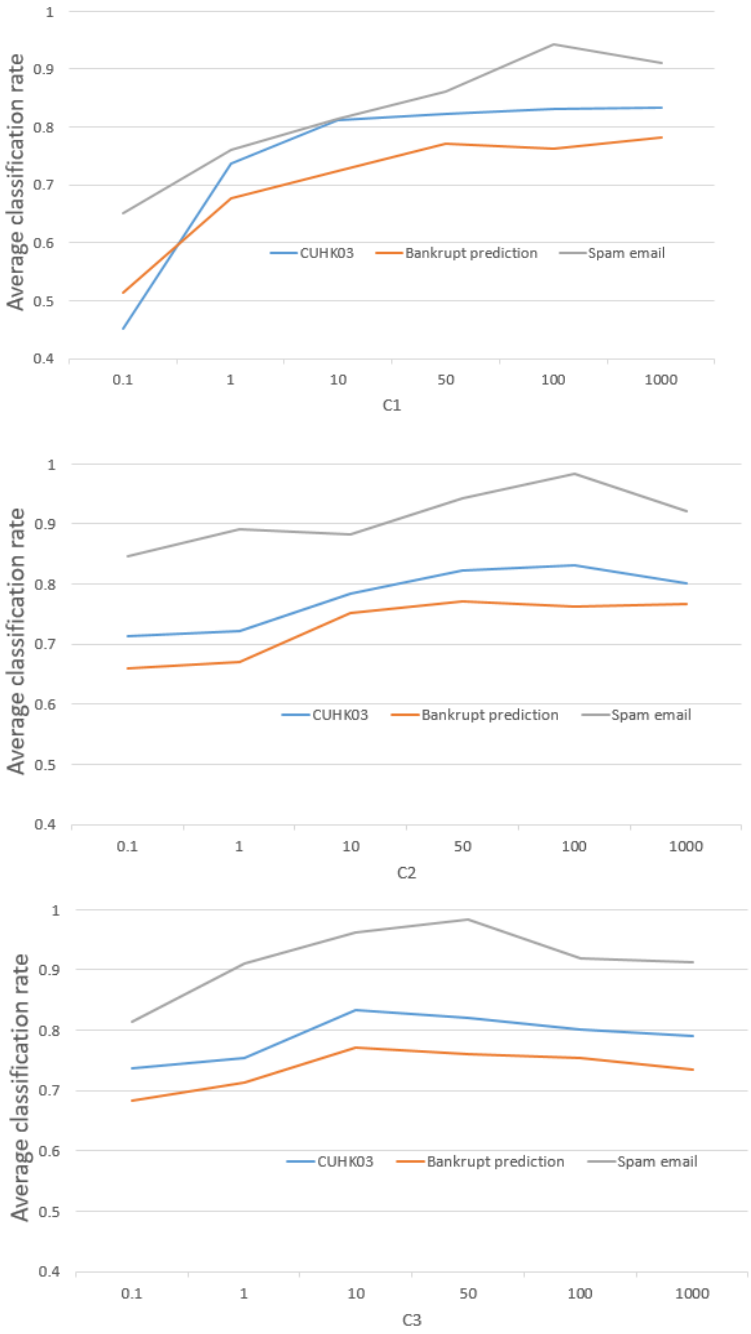

Figure 4. Sensitivity curve of tradeoff parameters.

3.3.3. Convergence analysis

Since the proposed algorithm DTCAE is an iterative algorithm. The variables are updated alternately. We are also interested in the convergence of the algorithm. Thus we plot the average classification rates with a different number of iterations. The curves over the three benchmark data sets are plotted in Figure 5. According to the curves of

Figure 5, when more iterations are used to update the variables of the model, the average classification rates increase stably. This is not surprising because a larger number of iterations reaches a smaller objective function. This verifies the effectiveness of the proposed model and its corresponding objective function. Moreover, we also observe that when the iteration number is larger than 100, the change of the performance is very small. This means that the algorithm converges and no more iteration is needed to improve the performance.

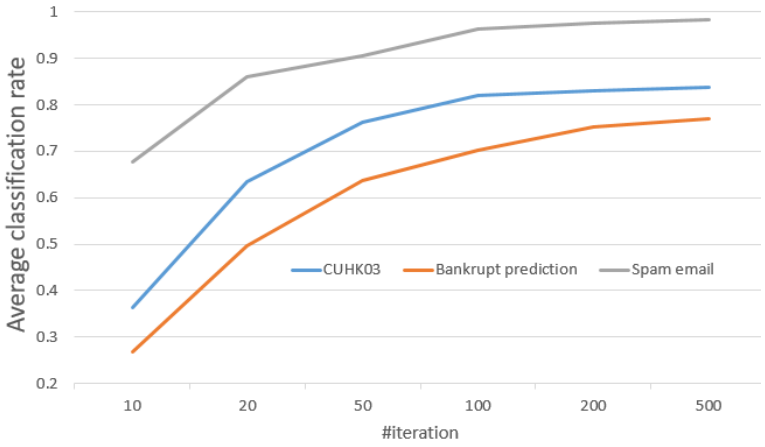

Figure 5. Convergence curves over the benchmark data sets.

4. Conclusion

In this paper, we propose a novel model for the problem of cross-domain learning problem with attribute data. The model is based on CNN model. We use a CNN model to map the input data to its attributes. Moreover, a domain-independent and domain-specific CNN model are also used to represent the data input itself. The attribute embedding, the domain-independent, and domain-specific representations are concatenated as the new representation of the data points, and we further a linear layer to map the new representation to the class labels. Moreover, we also impose the domain-independent representations of data points of different domains to be in a common distribution, and the neighboring data points of target domain to be similar to each other. We model the learning problem as a minimization problem and solve it by an iterative algorithm. The experiments on three benchmark data sets show its advantages.

Acknowledgments

This work was funded by National Natural Science Foundation of China under Grant 41401653, National Social Science Fund of China under Grant 17XJY018, MOE Project of Humanities and Social Sciences of China under Grant 16YJAZH051.

Statement of conflict of interest

The authors of this paper claim no conflict of interest for the work reported in this paper.

References

An, L., Chen, X., Yang, S., 2017. Multi-graph feature level fusion for person re-identification. Neurocomputing 259, 39–45.

Bialkowski, A., Denman, S., Sridharan, S., Fookes, C., Lucey, P., 2012. A database for person re-identification in multi-camera surveillance networks, in: Digital Image Computing Techniques and Applications (DICTA), 2012 International Conference on, IEEE. pp. 1–8.

Bickel, S., 2006. Ecml-pkdd discovery challenge 2006 overview, in: ECML-PKDD Discovery Challenge Workshop, pp. 1–9.

Ding, M., Fan, G., 2013. Multi-layer joint gait-pose manifold for human motion modeling, in: 2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG).

Ding, M., Fan, G., 2015. Multilayer joint gait-pose manifolds for human gait motion modeling. IEEE Transactions on Cybernetics 45, 2413–2424.

Ding, M., Fan, G., Zhang, X., Ge, S., Chou, L.S., 2012. Structure-guided manifold learning for video-based motion estimation, in: 2012 19th IEEE International Conference on Image Processing, pp. 1977–1980.

Fujino, S., Hatanaka, T., Mori, N., Matsumoto, K., 2018. The evolutionary deep learning based on deep convolutional neural network for the anime storyboard recognition. Advances in Intelligent Systems and Computing 620, 278–285.

Hassen, Y., Loukil, K., Ouni, T., Jallouli, M., 2018. Images selection and best descriptor combination for multi-shot person re-identification. Smart Innovation, Systems and Technologies 76, 11–20.

Ibn Khedher, M., El-Yacoubi, M., Dorizzi, B., 2017. Fusion of appearance and motion-based sparse representations for multi-shot person re-identification. Neurocomputing 248, 94–104.

Jing, L., Zhao, M., Li, P., Xu, X., 2017. A convolutional neural network based feature learning and fault diagnosis method for the condition monitoring of gearbox. Measurement: Journal of the International Measurement Confederation 111, 1–10.

Kulkarni, P., Sharma, G., Zepeda, J., Chevallier, L., 2014. Transfer learning via attributes for improved on-the-fly classification, in: 2014 IEEE Winter Conference on Applications of Computer Vision, WACV 2014, pp. 220–226.

Li, W., Zhao, R., Xiao, T., Wang, X., 2014. Deepreid: Deep filter pairing neural network for person re-identification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 152–159.

Lopez-Sanchez, D., Arrieta, A., Corchado, J., 2018. Deep neural networks and transfer learning applied to multimedia web mining. Advances in Intelligent Systems and Computing 620, 124–131.

Peng, P., Tian, Y., Xiang, T., Wang, Y., Pontil, M., Huang, T., 2017. Joint semantic and latent attribute modelling for cross-class transfer learning. IEEE Transactions on Pattern Analysis and Machine Intelligence .

Puri, U., Tewari, A., Katyal, S., Garg, B., 2018. Recognition of table images using k nearest neighbors and convolutional neural networks. Advances in Intelligent Systems and Computing 620, 326–333.

Roa-Barco, L., Serradilla-Casado, O., Velasco-Vzquez, M., Lpez-Zorrilla, A., Graa, M., Chyzhyk, D., Price, C., 2018. A 2d/3d convolutional neural network for brain white matter lesion detection in multimodal mri. Advances in Intelligent Systems and Computing 578, 377–385.

Su, C., Yang, F., Zhang, S., Tian, Q., Davis, L., Gao, W., 2017. Multi-task learning with low rank attribute embedding for multi-camera person re-identification. IEEE Transactions on Pattern

Analysis and Machine Intelligence .
Suzuki, M., Sato, H., Oyama, S., Kurihara, M., 2014a. Image classification by transfer learning based on the predictive ability of each attribute, in: Lecture Notes in Engineering and Computer Science, pp. 75–78.
Suzuki, M., Sato, H., Oyama, S., Kurihara, M., 2014b. Transfer learning based on the observation probability of each attribute, in: Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics, pp. 3627–3631.
Todoroki, Y., Han, X.H., Iwamoto, Y., Lin, L., Hu, H., Chen, Y.W., 2018. Detection of liver tumor candidates from ct images using deep convolutional neural networks. Smart Innovation, Systems and Technologies 71, 140–145.
Waijanya, S., Promrit, N., 2018. The poet identification using convolutional neural networks. Advances in Intelligent Systems and Computing 566, 179–187.
Wang, T., Gong, S., Zhu, X., Wang, S., 2014. Person re-identification by video ranking, in: European Conference on Computer Vision, Springer. pp. 688–703.
Wang, Y., Song, J., Marquez-Lago, T., Leier, A., Li, C., Lithgow, T., Webb, G., Shen, H.B., 2017. Knowledge-transfer learning for prediction of matrix metalloprotease substrate-cleavage sites. Scientific Reports 7, 5755.
Yang, L., Zhang, J., 2017. Automatic transfer learning for short text mining. Eurasip Journal on Wireless Communications and Networking 2017, 42.
Zhang, L., Yang, J., Zhang, D., 2017a. Domain class consistency based transfer learning for image classification across domains. Information Sciences 418-419, 242–257.
Zhang, X., Ding, M., Fan, G., 2017b. Video-based human walking estimation using joint gait and pose manifolds. IEEE Transactions on Circuits and Systems for Video Technology 27, 1540–1554.
Zhao, C., Wang, X., Wong, W., Zheng, W., Yang, J., Miao, D., 2017. Multiple metric learning based on bar-shape descriptor for person re-identification. Pattern Recognition 71, 218–234.
Zheng, L., Shen, L., Tian, L., Wang, S., Wang, J., Tian, Q., 2015. Scalable person re-identification: A benchmark, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 1116–1124.