

## Research Article

# Domain Adaption Based on ELM Autoencoder

Wan-Yu Deng, Yu-Tao Qu, and Qian Zhang

*School of Computer, Xi'an University of Posts & Telecommunications, Xi'an 710121, China*

Correspondence should be addressed to Wan-Yu Deng; [dengwanyu@126.com](mailto:dengwanyu@126.com)

Received 21 June 2016; Revised 26 November 2016; Accepted 18 May 2017; Published 19 June 2017

Academic Editor: Jason Gu

Copyright © 2017 Wan-Yu Deng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose a new ELM Autoencoder (ELM-AE) based domain adaption algorithm which describes the subspaces of source and target domain by ELM-AE and then carries out subspace alignment to project different domains into a common new space. By leveraging nonlinear approximation ability and efficient one-pass learning ability of ELM-AE, the proposed domain adaption algorithm can efficiently seek a better cross-domain feature representation than linear feature representation approaches such as PCA to improve domain adaption performance. The widely experimental results on Office/Caltech-256 datasets show that the proposed algorithm can achieve better classification accuracy than PCA subspace alignment algorithm and other state-of-the-art domain adaption algorithms in most cases.

## 1. Introduction

With the rapid development of Internet and social networks, a huge amount of data (e.g., web data and social data [1]) is being generated rapidly at every moment [2, 3]. With the explosive growth of data, its processing becomes more and more essential. Among data processing, the feature extraction is one of the most important technologies to deal with data. Feature extraction is used to represent the sample data and extract the most useful characteristics. The performance of a machine learning algorithm depends on whether the extracted feature can well represent the data. When the data dimension is too large, there will be a lot of problems. For example, the computational efficiency will be decreased and may cause overfitting problems. Principle Component Analysis (PCA) [4–6] is one of state-of-the-art methods of feature extraction. PCA is used to reduce the dimensions of data under such circumstance and tries to keep the useful information as more as possible. In particular, it uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. This transformation is defined in such a way that the first principal component has the largest possible variance, and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the

preceding components. However, PCA bears its inherent limitations when it is employed to represent the data: (1) it requires that the data is linear and (2) the features of PCA decomposition are orthogonal. In real applications, many datasets do not conform to this property. Based on this, we introduce a new method, ELM Autoencoder (ELM-AE), to extract the feature space of the data and learn the projecting function of subspace. The feature space that one extracts is not based on orthogonal transformation and the space of the data can be linear and nonlinear cases.

Domain adaptation (DA) is mainly to train a robust classifier, which can recognize the fact that the data is from the different distributions. It is widely used in computer vision and pattern recognition. The usual DA methods are divided into two classes [7]: (1) semisupervised DA algorithm [8], which will have a small amount of label data from the target domain, and (2) unsupervised DA algorithm, in which there is no label data from the target domain. Here we mainly study the unsupervised DA algorithm and proposed a DA algorithm based on ELM-AE.

Recently, DA has been applied to many fields, embracing speech and language processing [9–11], computer vision [12–14], statistics, and machine learning [12–14]. A robust classifier can be trained in DA to deal with the multidomain mixed classification tasks. This algorithm is particularly suitable to handle the unsupervised data with no class labels. Typical

implementation of DA method is learning a new space in which the differences of feature representation between source and target domains can be minimized.

DA method based PCA has been extensively researched [4–6]; we can find a common subspace through PCA in which the diversities between the two different distributions data of the source and target domains are minimized. In [10], Blitzer et al. proposed a method to learn a new feature space through feature relationship between different domains. Source data representation can be obtained by linear transformation of target data according to Jhuo et al. mentioned in [15]. In [16], Gong et al. proposed a geodesic flow kernel (GFK), which mainly counts the changes of both source data and the target data in the geometry and statistics. Fernando et al. proposed a DA method based PCA in [7]. They obtained the feature space of the source and target data by applying PCA, respectively. Then, the representative feature of source data was projected into the feature space of target data and the representative feature of the target data was projected into the feature space of source data. Fernando et al. also proposed three methods in [7]: DA-SA1, where subspace  $\mathbf{W}_S$  built from source domain uses PCA, DA-SA2, where they use PCA to project the subspace of the target domain which is denoted by  $\mathbf{W}_T$ , and NA, where they only use the original input space without learning a subspace.

The rest of the paper is organized as follows. We present the related work in Section 2. Section 3 is devoted to the presentation of the ELM-AE method and the consistency theorem on the similarity measure deduced from the learned projecting function. In Section 4, the subspace alignment algorithm based on ELM-AE is introduced. We carry out our experiments on various datasets in Section 5. In Section 6, we get the conclusions.

## 2. Related Work

In this section, we will show a kind of feature extraction method (PCA) which has been used in subspace alignment domain adaption algorithm. In some applications involving many related features, a great number of features will not only increase the complexity of the problem but also make it difficult to give a reasonable analysis of the problem. In general, although each feature provides some information, their importance is different. In many cases, there is certain correlation among the feature; thus, the information provided by these feature, to some extent, will coincide. Therefore, it is expected to represent these features by a small amount of the new and unrelated feature to reflect the vast majority of information provided by the original feature and then achieve the better solution of the problems through new feature.

PCA was invented in 1901 by Karl Pearson in [17] as an analogue of the principal axis theorem in mechanics. It can be mathematically defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by some projection of the data comes to lie on the first coordinate, the second greatest variance on the second coordinate, and so on. In particular, it mainly includes the following steps [17].

(1) To normalize the raw data matrix  $\mathbf{X}_{n \times D}$ , which is equivalent to the panning and expansion of the original feature,

$$\mathbf{X}_{n \times D} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1D} \\ x_{21} & x_{22} & \cdots & x_{2D} \\ \cdots & \cdots & \cdots & \cdots \\ x_{n1} & x_{n2} & \cdots & x_{nD} \end{bmatrix}, \quad (1)$$

where  $x_{ij} = (x_{ij} - \bar{x}_j)/s_j$ , ( $i = 1, 2, \dots, N$ ;  $j = 1, 2, \dots, D$ );  $\bar{x}_j$  represents the average of the  $j$ th column and  $s_j = \sqrt{\text{cov}(j, j)}$ .

(2) Solving covariance matrix,

$$\text{cov}(i, j) = \frac{\sum_{k=1}^N (x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j)}{\sqrt{\sum_{k=1}^N (x_{ki} - \bar{x}_i)^2 (x_{kj} - \bar{x}_j)^2}}, \quad (2)$$

where  $\text{cov}(i, j)$  represents the correlation coefficient between the  $i$ th column and  $j$ th column.

The covariance matrix can be obtained by the following equation:

$$\mathbf{Z} = \begin{bmatrix} s_1^2 & \text{cov}(1, 2) & \cdots & \text{cov}(1, D) \\ \text{cov}(2, 1) & s_2^2 & \cdots & \text{cov}(2, D) \\ \cdots & \cdots & \cdots & \cdots \\ \text{cov}(D, 1) & \text{cov}(D, 2) & \cdots & s_D^2 \end{bmatrix}. \quad (3)$$

(3) Calculate the eigenvalues and eigenvectors of the covariance matrix and decompose the covariance matrix by its feature in [18]:

$$\mathbf{Z} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T, \quad (4)$$

where  $\mathbf{\Lambda}$  is a diagonal matrix composed by eigenvalue of the covariance matrix  $\mathbf{Z}$ ,  $\mathbf{U}$  is the orthogonal matrix composed by eigenvectors of  $\mathbf{Z}$  in column, and it is the principal coordinate of the new variables. Characteristic value represents the size of the new variable variance, and the eigenvalues we obtained will gradually decrease.

(4) When the eigenvalues are small enough, we think it has little to deal with our source data. Thus, we will choose the first  $m$  larger eigenvalues and eigenvectors to constitute our projecting space being  $\mathbf{U}_{D \times m}$ .

(5) By using (5), get the new data representation. Every line in matrix  $\mathbf{F}$  is equivalent to the projection of all the lines of the original matrix in the principal component axis. These new vectors of projection can be used to express our source data.

$$\mathbf{F}_{n \times m} = \mathbf{X}_{n \times D} \mathbf{U}_{D \times m}. \quad (5)$$

PCA method can find the most important variable combination of the original data. By showing the greatest variance, it can effectively and intuitively reflect the relationship between samples. Moreover, it can approximately express the original data by the largest principal component projecting. However, PCA method has its limitations: (1) it requires that the principal component must be a linear combination of the original data and (2) it requires that each principal component must be uncorrelated. These will lead to PCA being not able to solve some practical problems well when it encounters them.

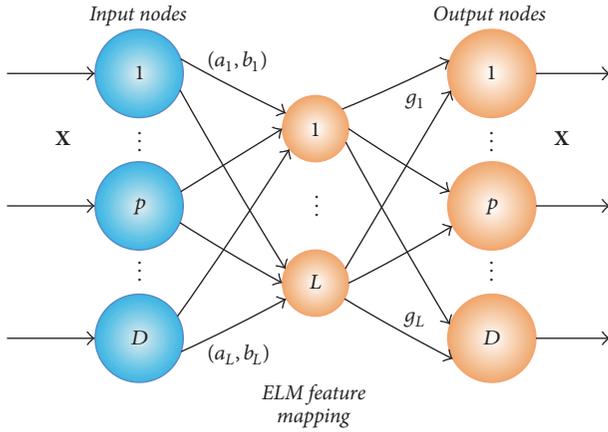


FIGURE 1: ELM-AE has the same network structure as ELM except that its target output is the same as input  $X$ .

### 3. ELM-AE

In this section, we will introduce one new feature representation algorithm, ELM-AE, which is based on one very fast and effective neural network named Extreme Learning Machine (ELM) [19, 20]. Just like the traditional ELM [21–23], ELM-AE contains three layers: input layer, hidden layer, and output layer. The difference is that the target output is the same as the input in ELM-AE. Figure 1 shows ELM-AE's network structure for compressed, sparse, and equal dimension representation.

Suppose that there are  $L$  hidden neuron in hidden layer and  $D$  neurons (data dimension) in input layer and output layer; according to its source data dimension  $D$  and the neurons number  $L$  in hidden layer, ELM-AE can be divided into three kinds of structures [24, 25]:

- (1) Compressed ( $D > L$ ): representative feature is projected into a low-dimensional feature space from a high-dimensional input data space
- (2) Sparse ( $D < L$ ): representative feature is projected into a high-dimensional feature space from a low-dimensional input data space
- (3) Equal ( $D = L$ ): representative feature is generated from projecting in which the dimension of the input data space is equal to the feature space

According to ELM theory [21, 26, 27], the hidden neurons for ELM can be randomly generated. In general, we usually choose orthogonal randomly generated hidden parameters to improve ELM-AE's generalization performance. In ELM-AE, the orthogonal random weights and biases of the hidden nodes project the input data into a different or equal dimension space, as shown by the Johnson-Lindenstrauss lemma [28], and are calculated as follows:

$$\begin{aligned} \mathbf{h} &= g(\mathbf{a} \cdot \mathbf{x} + \mathbf{b}), \\ \mathbf{a}^T \mathbf{a} &= \mathbf{I}, \\ \mathbf{b}^T \mathbf{b} &= \mathbf{I}, \end{aligned} \quad (6)$$

where  $\mathbf{a} = [a_1, \dots, a_L]$  are the orthogonal random weights and  $\mathbf{b} = [b_1, \dots, b_L]$  are the orthogonal random biases between the input and hidden nodes;  $g(x)$  is activation function of ELM-AE.

ELM-AE's output weight  $\beta$  is our transformation matrix; we can project the input data into the subspace by  $\beta$ . For sparse and compressed ELM-AE representations, we calculate output weights  $\beta$  as follows:

$$\beta = \left( \frac{\mathbf{I}}{C} + \mathbf{H}^T \mathbf{H} \right)^\dagger \mathbf{H}^T \mathbf{X}, \quad (7)$$

where  $\mathbf{H} = [\mathbf{h}_1, \dots, \mathbf{h}_N]$  are ELM-AE's hidden layer outputs and  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]$  are not only input data but also output data. For equal dimension ELM-AE representations, we calculate output weights  $\beta$  as follows:

$$\begin{aligned} \beta &= \mathbf{H}^{-1} \mathbf{T}, \\ \beta^T \beta &= \mathbf{I}. \end{aligned} \quad (8)$$

### 4. Subspace Alignment Based on ELM-AE

In this section, we mainly introduce the subspace alignment algorithm based on ELM-AE and illustrate the principle of the algorithm in detail.

**4.1. Subspace Alignment.** In classification or regression, we generally deal with the labeled data and those unlabeled data that come from the same distribution. However, in practical applications, this assumption will be challenged, since data we need to process are from different domains. Due to the change of the data type, it will not achieve a good result when it uses the model trained on the former data to classify the later test data. In order to obtain robust machine learning model, it is necessary to take into account the shift between these two domains (we refer to these different but related marginal distributions as domains). The purpose of DA is to make full use of the information generated during the transformation between the source domain and the target domain and then automatically adapt it.

In this paper, we enable each domain to be projected into its subspace. Then we enable the subspace of the source domains to be closed to the target domain for the purpose of the reduction of the difference between the two structures. It needs a transformation function to enable the subspace of the source domain to be projected into the subspace of the target domain. Based on this idea, subspace alignment algorithm is described as follows in detail.

**4.2. Subspace Generation.** To learn the shift between these two domains, it is needed to process the raw data using ELM-AE to represent the original data and make full use of the information of these two domains. Firstly, we transform every source and target sample in the form of a  $D$ -dimensional vector. Then, use ELM-AE to calculate the output weights of the  $L$  hidden layer nodes, which are the subspaces between the source domains and the target domains, respectively, denoted by  $\mathbf{W}_S \in R^{D \times L}$  and  $\mathbf{W}_T \in R^{D \times L}$ .

**Input:** Source data  $\mathbf{X}_S$ , Target data  $\mathbf{X}_T$ , Source labels  $\mathbf{T}_S$ , Target labels  $\mathbf{T}_T$ , the number of hidden neurons  $L$ , Activation function  $g(x)$ .  
**Output:** Predicted target labels  $\mathbf{T}_T$ ;  
 $\mathbf{W}_S \leftarrow \text{ELM-AE}(\mathbf{X}_S, g(x), L)$ ;  
 $\mathbf{W}_T \leftarrow \text{ELM-AE}(\mathbf{X}_T, g(x), L)$ ;  
 $\widetilde{\mathbf{W}}_S = \mathbf{W}_S \mathbf{W}_S^T \mathbf{W}_T$ ;  
 $\mathbf{X}_S = \mathbf{X}_S \widetilde{\mathbf{W}}_S$ ;  
 $\mathbf{X}_T = \mathbf{X}_T \mathbf{W}_T$ ;  
 $\mathbf{T}_T = \text{Classifier}(\mathbf{X}_S, \mathbf{X}_T, \mathbf{T}_S)$ ;

ALGORITHM 1: (SA-ELM-DA): subspace alignment based on ELM-AE.

**4.3. The Subspace Alignment Based on the ELM-AE.** In this section, we will introduce how to learn the transformation function between two subspaces and use it to achieve the subspace alignment. We suggest to project each source ( $\mathbf{x}_S$ ) and target ( $\mathbf{x}_T$ ) sample (where  $\mathbf{x}_S \in R^{1 \times d}$ ,  $\mathbf{x}_T \in R^{1 \times d}$ ) into its respective subspace  $\widetilde{\mathbf{x}}_S$  and  $\widetilde{\mathbf{x}}_T$  by the operations  $\mathbf{x}_S \mathbf{W}_S$  and  $\mathbf{x}_T \mathbf{W}_T$ , respectively. Then, we learn a linear transformation function that aligns the source subspace coordinate system to the target one. This step allows us to directly compare source and target samples in their respective subspaces without unnecessary data projections. To achieve this task, we align basis vectors by using a transformation matrix  $\mathbf{M}$  from  $\mathbf{W}_S$  to  $\mathbf{W}_T$ .  $\mathbf{M}$  is learned by minimizing the following Bregman matrix divergence [7]:

$$\mathbf{F}(\mathbf{M}) = \|\mathbf{W}_S \mathbf{M} - \mathbf{W}_T\|_F^2, \quad (9)$$

$$\mathbf{M}^* = \arg \min_M (\mathbf{F}(\mathbf{M})), \quad (10)$$

where  $\|\cdot\|_F^2$  is the Frobenius norm.

Because the orthogonal operation is invariant for Frobenius norm, (9) can be written as follows:

$$\mathbf{F}(\mathbf{M}) = \|\mathbf{W}_S^T \mathbf{W}_S \mathbf{M} - \mathbf{W}_S^T \mathbf{W}_T\|_F^2 = \|\mathbf{M} - \mathbf{W}_S^T \mathbf{W}_T\|_F^2. \quad (11)$$

In this way, we can calculate the optimal transformation matrix:

$$\mathbf{M}^* = \mathbf{W}_S^T \mathbf{W}_T. \quad (12)$$

By changing matrix  $\mathbf{M}$ , we can get the new coordinate system:

$$\widetilde{\mathbf{W}}_S = \mathbf{W}_S \mathbf{W}_S^T \mathbf{W}_T, \quad (13)$$

where  $\widetilde{\mathbf{W}}_S$  is the target aligned source coordinate system. Subspace alignment algorithm based on ELM-AE (SA-ELM-DA in short) can be summarized in Algorithm 1.

As for the ELM-AE activation function, it can be ‘‘sigmoid,’’ ‘‘RBF,’’ and ‘‘Sin’’ functions and so on.  $L$  represents the number of neurons in the hidden layer.  $C$  is a regularized parameter to improve the generalization performance of the ELM-AE.  $\mathbf{W}_S$  and  $\mathbf{W}_T$  are not only the output weight of the ELM-AE but also the projecting matrix of the subspace.

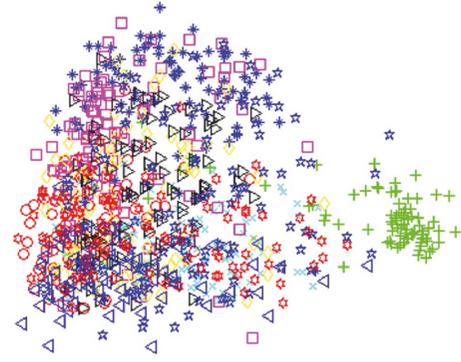


FIGURE 2: Visualization of domain A.

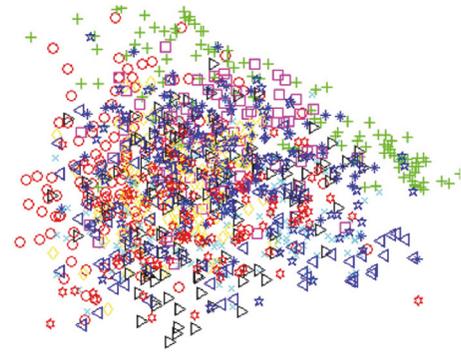


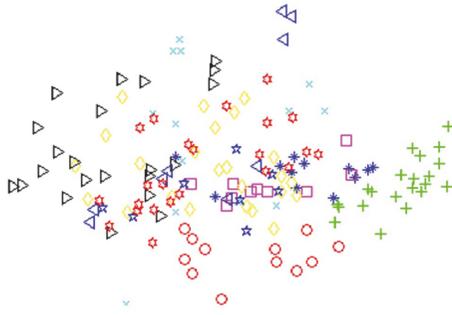
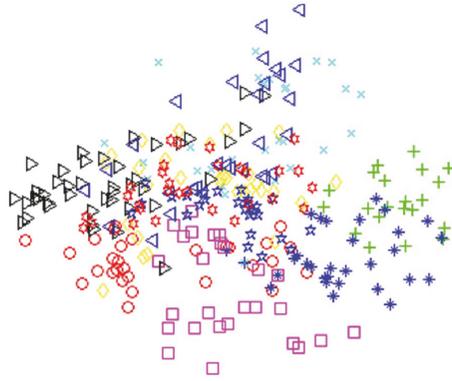
FIGURE 3: Visualization of domain C.

The advantage of the proposed algorithm is twofold: (1) by generating the subspace, we can train a robust classifier, which has a high degree of accuracy when classifying the different distributions of the data; (2) compared to PCA, our algorithm can also have good performance in dealing with the nonlinear data.

## 5. Experimental Study

In this experiment, we will carry out the experiments on a variety of data to evaluate our method and compare with some other state-of-the-art DA algorithms.

**5.1. DA Datasets.** We compare all DA algorithms using Office [29] dataset and Caltech-256 [30] dataset which contain four domains. The Office dataset consists of images from webcam (denoted by  $W$ ), DSLR images (denoted by  $D$ ), and Amazon images (denoted by  $A$ ). The Caltech-256 images are denoted by  $C$ .  $A$  contains 958 images,  $C$  contains 1123 images,  $D$  contains 157 images, and  $W$  is a collection of 295 images. Following the same setup as in [30], we use each source of images as a domain; consequently we get four domains ( $A$ ,  $C$ ,  $D$ , and  $W$ ) and 12 DA problems. DA problem can be described as the notation  $S \rightarrow T$ , in which  $S$  is the source domain and  $T$  is the target domain. We use random selection method to obtain the two dimensions of the original datasets, and one example is visualized in Figures 2–5.

FIGURE 4: Visualization of domain  $D$ .FIGURE 5: Visualization of domain  $W$ .

From Figures 2–5, we can find that the points from different categories mix together. But the visualizations of 2-dimensional representation cannot conclude that the original datasets are nonlinear. To better show the nonlinear ability of the proposed algorithm, we use linear activation function and nonlinear activation function (sigmoid), respectively, in the experiments. From Tables 1 and 2, we can find that when we use the linear activation function, the average classification accuracy of SA-ELM-DA is significantly reduced not only using SURF feature but also using CNN feature; it further illustrates that our datasets are nonlinear and also illustrates the nonlinear ability of the proposed algorithm.

**5.2. Experimental Setup.** We use SURF feature and CNN feature to cooperate with our algorithm SA-ELM-DA, respectively, and denote them as SA-ELM-DA(SURF) and SA-ELM-DA(CNN). SURF is a feature extraction algorithm based on scale invariant: first, using Hessian of each pixel to build scale space and then using nonmaximum suppression to select initial feature points. The 3-dimensional linear interpolation method is used to get the feature points of the subpixel level, and the points whose values are less than a certain threshold are also removed. The main direction of the feature points is obtained by counting harr wavelet feature in the field of feature points. Finally, construct surf feature point description operator and get an 800-dimensional representation. Convolution neural network (CNN) [31] is an algorithm

commonly used in depth learning; we use imagenet-vgg-f model of CNN in MatConvNet (a convolution network toolkit in MATLAB developed by Andrea Vedaldi) to extract the output of the fully connected layer on layer 20 and get a 4096-dimensional representation of images.

We compare our ELM-AE subspace DA approach with PCA subspace DA algorithm and other three baselines as follows.

**DA-SA1 [7].** Subspace  $W_S$  built from source domain uses PCA;  $W_S$  is used to project source data and target data to complete the domain adaptation and calculate average accuracy of domain adaptation by cross-validation.

**DA-SA2 [7].** PCA is used to project the subspace of the target domain which is denoted by  $W_T$ , followed by using  $W_T$  to project source data and target data and calculate the average accuracy of domain adaption by cross-validation.

**NA [7].** No projection is generated, the classification of multidomain data in original space is completed without learning a subspace in domain adaption. Calculate the average classification accuracy by cross-validation in domain adaption.

**GFK [16].** By integrating an infinite number of subspaces that characterize changes in geometric and statistical properties from the source to the target domain, it can automatically infer important algorithmic parameters without requiring extensive cross-validation.

**SA-ELM-DA.** Learn the feature representation by ELM-AE from source and target data and generate respective subspace, complete subspace alignment by transformation matrix to achieve domain adaption, and calculate the average classification accuracy.

In the experiments, we compare the performance of subspace generation for each DA algorithm using a 1-Nearest-Neighbor (NN) classifier and a SVM classifier. The number of hidden nodes in experiments is determined by cross-validation.

**5.3. Experimental Results.** For domain adaptation based on ELM-AE performance with Office/Caltech-256 datasets, we use Office [29]/Caltech-256 [30] datasets which consist of four domains ( $A$ ,  $C$ ,  $D$ , and  $W$ ) to compare with different DA algorithms in our experiments. The results of unsupervised DA problems using a NN classifier are shown in Table 1. In 7 out of the 12 DA problems, our algorithm has an advantage over the others; it improves the average accuracy significantly in domain adaptation. The results acquired from a SVM [32] classifier in the unsupervised DA case are shown in Table 2. Our method has better performance compared to the others in terms of 10 DA problems. Other results are closed to the optimal performance on average accuracy. SA-ELM-DA(SURF, linear) and SA-ELM-DA(CNN, linear) are only used to prove that the datasets are nonlinear in Tables 1 and 2; we will no longer analyze the results of SA-ELM-DA(SURF, linear) and SA-ELM-DA(CNN, linear) in here.

TABLE 1

(a) Recognition accuracy with DA using a NN classifier (Office dataset + Caltech-256)

Algorithm	$C \rightarrow A$	$D \rightarrow A$	$W \rightarrow A$	$A \rightarrow C$	$D \rightarrow C$	$W \rightarrow C$
NA [7]	21.53%	28.71%	23.64%	22.55%	26.26%	19.21%
DA-SA1 [7]	38.00%	29.80%	35.50%	30.90%	29.60%	31.30%
DA-SA2 [7]	40.50%	33.00%	38.00%	33.30%	31.20%	31.90%
PCA [7]	39.00%	38.00%	37.40%	35.30%	32.40%	32.30%
GFK [16]	36.90%	32.52%	31.10%	35.61%	29.86%	27.26%
SA-ELM-DA(SURF, sigmoid)	39.72%	36.89%	34.42%	<b>35.74%</b>	<b>33.24%</b>	<b>33.61%</b>
SA-ELM-DA(SURF, linear)	25.62%	30.02%	26.31%	24.00%	26.02%	19.00%
SA-ELM-DA(CNN, linear)	20.15%	15.52%	20.92%	17.43%	11.75%	13.47%
SA-ELM-DA(CNN, sigmoid)	<b>58.36%</b>	<b>43.79%</b>	<b>71.83%</b>	<b>49.74%</b>	<b>34.54%</b>	<b>41.63%</b>

(b) Recognition accuracy with DA using a NN classifier (Office dataset + Caltech-256)

Algorithm	$A \rightarrow D$	$C \rightarrow D$	$W \rightarrow D$	$A \rightarrow W$	$C \rightarrow W$	$D \rightarrow W$
NA [7]	21.34%	21.08%	54.01%	25.24%	20.32%	62.44%
DA-SA1 [7]	34.60%	37.40%	71.80%	35.10%	33.50%	74.00%
DA-SA2 [7]	34.70%	36.40%	72.90%	36.80%	34.40%	78.40%
PCA [7]	37.60%	39.60%	80.30%	38.60%	36.80%	83.60%
GFK [16]	35.24%	35.21%	70.63%	34.46%	33.71%	74.92%
SA-ELM-DA(SURF, sigmoid)	36.86%	<b>39.92%</b>	<b>80.83%</b>	36.76%	<b>36.92%</b>	<b>84.41%</b>
SA-ELM-DA(SURF, linear)	30.00%	32.60%	83.38%	30.00%	26.00%	79.47%
SA-ELM-DA(CNN, linear)	18.74%	13.54%	15.76%	17.16%	12.18%	13.64%
SA-ELM-DA(CNN, sigmoid)	<b>46.94%</b>	35.86%	53.82%	<b>52.61%</b>	<b>46.69%</b>	54.69%

In Table 2, we use SA-ELM-DA(CNN) to complete the domain adaptation; the average accuracy has been improved significantly. It is obviously shown that our algorithm works not only for NN-like local classifiers but also with more global SVM classifiers superior to other DA algorithms. In Table 1, SA-ELM-DA(SURF) has an increase of 15.20% compared to NA [7] on average accuracy; compared to DA-SA1 [7] and DA-SA2 [7], our algorithm improves the average accuracy by 2.30% and 4.00%, respectively. GFK [16] algorithm has been shown to have good performance, compared to NA [7]; it increased by 11.6% on average accuracy, but our algorithm has an increase of 4.20% compared to GFK [16] on average accuracy. In Table 2, our algorithm has an average accuracy of 4.70% over DA-SA1 [7] and improves the average accuracy by 4.40% over DA-SA2 [7]. Compared to GFK [16], our algorithm improves by 3.40% on average accuracy. PCA [7] has shown a best result, but our algorithm has an increase of 1.10% compared to PCA [7] on average accuracy. Our algorithm has a better performance in the task of multidomain data. DA algorithm based on PCA [7] has been proven to have good performance in dealing with unsupervised data. Compared with the aforementioned DA algorithm, SA-ELM-DA performs better in handling the unsupervised data model. Our algorithm has been greatly improved which was compared to the NA [7] with no field conversion algorithm on average accuracy. Therefore, SA-ELM-DA is more suitable to deal with data from different fields. Compared to GFK [16], the performance of algorithm has much room for improvement. As we can conclude from Tables 1 and 2, SA-ELM-DA(CNN)

has the best performance on average accuracy; when we use CNN feature to complete the domain adaptation, the average accuracy has been improved significantly except for two domains about  $W$  and  $D$ . There are a few samples in  $W$  and  $D$ ; they may lead to extracting inaccurate feature when using CNN, which results in a significant decline in performance of domain adaption between  $W$  and  $D$ . SA-ELM-DA has a significant improvement on average accuracy not only using SURF feature but also using CNN feature. Because the datasets are nonlinear, the average accuracy has a significant improvement when using nonlinear activation function (sigmoid) to project data. The algorithm we proposed can handle nonlinear data compared to DA based on PCA [7]. It is firmly believed that our method will be widely applied in various types of data processing such as image processing, machine learning, and computer vision in the future. For example, in medical image processing, there are differences in the performance of the same diseases because of individual differences. Through domain adaption, we can learn a common model to narrow the differences of the same type of diseases and improve the diagnosis rate of diseases.

## 6. Conclusion

We propose a new domain adaption algorithm based on the ELM-AE; meanwhile, we use different features to carry out our algorithm. In order to further enhance the accuracy of domain adaption, we further use the convolution neural network (CNN) [31] to extract the high-dimensional feature

TABLE 2

(a) Recognition accuracy with DA using a SVM Classifier (Office dataset + Caltech-256)

Algorithm	$C \rightarrow A$	$D \rightarrow A$	$W \rightarrow A$	$A \rightarrow C$	$D \rightarrow C$	$W \rightarrow C$
DA-SA1 [7]	44.30%	36.80%	32.90%	36.80%	29.60%	24.90%
DA-SA2 [7]	44.50%	38.60%	34.20%	37.30%	31.60%	28.40%
PCA [7]	46.10%	41.05%	39.30%	39.20%	35.00%	31.80%
GFK [16]	44.82%	37.91%	37.10%	38.37%	31.42%	29.14%
SA-ELM-DA(SURF, sigmoid)	<b>46.62%</b>	<b>41.15%</b>	39.02%	<b>40.88%</b>	<b>35.93%</b>	<b>33.17%</b>
SA-ELM-DA(SURF, linear)	27.82%	34.00%	36.04%	23.17%	32.03%	31.00%
SA-ELM-DA(CNN, linear)	61.62%	41.29%	52.73%	52.78%	35.87%	45.13%
SA-ELM-DA(CNN, sigmoid)	<b>80.80%</b>	<b>47.92%</b>	<b>68.89%</b>	<b>67.31%</b>	<b>36.79%</b>	<b>52.60%</b>

(b) Recognition accuracy with DA using a SVM Classifier (Office dataset + Caltech-256)

Algorithm	$A \rightarrow D$	$C \rightarrow D$	$W \rightarrow D$	$A \rightarrow W$	$C \rightarrow W$	$D \rightarrow W$
DA-SA1 [7]	36.10%	38.90%	73.60%	42.50%	34.60%	75.40%
DA-SA2 [7]	32.50%	35.30%	73.60%	37.30%	34.20%	80.50%
PCA [7]	38.80%	39.40%	77.90%	39.60%	38.90%	82.30%
GFK [16]	37.92%	36.14%	74.60%	39.81%	34.93%	79.10%
SA-ELM-DA(SURF, sigmoid)	<b>39.76%</b>	<b>41.46%</b>	<b>80.76%</b>	41.05%	<b>39.10%</b>	<b>83.88%</b>
SA-ELM-DA(SURF, linear)	36.04%	37.11%	82.32%	36.00%	33.24%	76.44%
SA-ELM-DA(CNN, linear)	46.34%	34.97%	43.28%	60.21%	57.34%	46.79%
SA-ELM-DA(CNN, sigmoid)	<b>49.43%</b>	<b>45.03%</b>	52.61%	<b>72.88%</b>	<b>64.34%</b>	55.34%

of images and complete the domain adaptation to improve the performance of our algorithm. The experimental results show that the SA-ELM-DA performs better than PCA and other state-of-the-art domain adaptation algorithms on average accuracy. Besides, in feature extracting, ELM-AE can also extract nonlinear feature well for those data with nonlinear relationship. Due to the good performance of our algorithm in dealing with the different distributions of data and its simple principle, we believe that our algorithm can have a wide range of applications in the future. We are going to extend our algorithm to cross-modal retrieval for large-scale data.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

## Acknowledgments

This work is supported by National Natural Science Foundation of China (Grant nos. 61572399, 61272120, and 61373116), Shaanxi New Star of Science & Technology (Grant no. 2013KJXX-29), New Star Team of Xian University of Posts & Telecommunications, and Provincial Key Disciplines Construction Fund of General Institutions of Higher Education in Shaanxi.

## References

- [1] E. Cambria, N. Howard, Y. Xia, and T.-S. Chua, "Computational Intelligence for Big Social Data Analysis [Guest Editorial]," *IEEE Computational Intelligence Magazine*, vol. 11, no. 3, pp. 8-9, 2016.
- [2] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 1521-1528, IEEE, Colorado Springs, RI, USA, June 2011.
- [3] G.-B. Huang, E. Cambria, K.-A. Toh, B. Widrow, and Z. Xu, "New trends of learning in computational intelligence [Guest Editorial]," *IEEE Computational Intelligence Magazine*, vol. 10, no. 2, pp. 16-17, 2015.
- [4] B. Chen, W. Lam, I. Tsang, and T.-L. Wong, "Extracting discriminative concepts for domain adaptation in text mining," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '09*, pp. 179-187, July 2009.
- [5] S. J. Pan, J. T. Kwok, and Q. Yang, "Transfer learning via dimensionality reduction," in *Proceedings of the 23rd AAAI Conference on Artificial Intelligence and the 20th Innovative Applications of Artificial Intelligence Conference, AAAI-08/IAAI-08, July 13, 2008-July 17, 2008*, pp. 677-682, Chicago, IL, USA, 2008.
- [6] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," in *Proceedings of the 21st International Joint Conference on Artificial Intelligence, IJCAI-09, July 11, 2009-July 17, 2009*, pp. 1187-1192, Pasadena, CA, USA, 2009.
- [7] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *Proceedings of the 2013 14th IEEE International Conference on Computer Vision, ICCV 2013*, pp. 2960-2967, aus, December 2013.
- [8] W. Li, L. Duan, D. Xu, and I. W. Tsang, "Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 6, pp. 1134-1148, 2014.

- [9] H. Daumé III, "Frustratingly easy domain adaptation," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL '07)*, pp. 256–263, June 2007.
- [10] J. Blitzer, R. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Proceedings of the 11th Conference on Empirical Methods in Natural Language Processing, EMNLP 2006, Held in Conjunction with COLING/ACL 2006, July 22, 2006–July 23, 2006*, pp. 120–128, Sydney, Australia, July 2006.
- [11] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer Speech and Language*, vol. 9, no. 2, pp. 171–185, 1995.
- [12] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Scholkopf, "Correcting sample selection bias by unlabeled data," in *Proceedings of the 20th Annual Conference on Neural Information Processing Systems, NIPS 2006*, pp. 601–608, BC, Canada, 2006.
- [13] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Proceedings of the 20th Annual Conference on Neural Information Processing Systems, NIPS 2006*, pp. 137–144, BC, Canada, 2006.
- [14] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.
- [15] I.-H. Jhuo, D. Liu, D. T. Lee, and S.-F. Chang, "Robust visual domain adaptation with low-rank reconstruction," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2012*, pp. 2168–2175, June 2012.
- [16] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 2066–2073, June 2012.
- [17] I. T. Jolliffe, *Principal Component Analysis*, Springer Series in Statistics, Springer, New York, NY, USA, 2nd edition, 2002.
- [18] L. Zwald and G. Blanchard, "On the convergence of eigenspaces in Kernel principal component analysis," in *Proceedings of the 2005 Annual Conference on Neural Information Processing Systems, NIPS 2005, December 5, 2005–December 8, 2005*, pp. 1649–1656, Vancouver, BC, Canada, 2005.
- [19] G. Huang, S. Song, and K. You, "Trends in extreme learning machines: a review," *Neural Networks*, vol. 61, pp. 32–48, 2015.
- [20] S. Wang, C. Deng, W. Lin, G. Huang, and B. Zhao, "NMF-Based Image Quality Assessment Using Extreme Learning Machine," *IEEE Transactions on Cybernetics*, vol. 47, no. 1, pp. 232–243, 2017.
- [21] G. Huang, L. Chen, and C. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Transactions on Neural Networks*, vol. 17, no. 4, pp. 879–892, 2006.
- [22] W. Mao, S. Zhao, X. Mu, and H. Wang, "Multi-dimensional extreme learning machine," *Neurocomputing A*, vol. 149, pp. 160–170, 2015.
- [23] W.-Y. Deng, Z. Bai, G.-B. Huang, and Q.-H. Zheng, "A Fast SVD-Hidden-nodes based Extreme Learning Machine for Large-Scale Data Analytics," *Neural Networks*, vol. 77, pp. 14–28, 2016.
- [24] L. L. C. Kasun, H. Zhou, G.-B. Huang, and C. M. Vong, "Representational learning with ELMs for big data," *IEEE Intelligent Systems*, vol. 28, no. 6, pp. 31–34, 2013.
- [25] C. Wang, J. Wang, and S. Gu, "Deep network based on stacked orthogonal convex incremental ELM autoencoders," *Mathematical Problems in Engineering*, vol. 2016, Article ID 1649486, 17 pages, 2016.
- [26] S. Ding, N. Zhang, X. Xu, L. Guo, and J. Zhang, "Deep extreme learning machine and its application in EEG classification," *Mathematical Problems in Engineering*, vol. 2015, Article ID 129021, 11 pages, 2015.
- [27] H. Zhou, G.-B. Huang, Z. Lin, H. Wang, and Y. C. Soh, "Stacked extreme learning machines," *IEEE Transactions on Cybernetics*, vol. 2, no. 2, pp. 1–13, 2014.
- [28] W. B. Johnson and J. Lindenstrauss, "Extensions of Lipschitz mappings into a Hilbert space," in *Proceedings of the Conf. Modern Analysis and Probability*, vol. 26, pp. 189–206, Amer. Math. Soc., Providence, RI.
- [29] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proceedings of the 11th European Conference on Computer Vision, ECCV 2010, September 5, 2010–September 11, 2010*, vol. 6314, pp. 213–226, Crete, Greece.
- [30] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *Proceedings of the 2011 IEEE International Conference on Computer Vision, ICCV 2011*, pp. 999–1006, esp, November 2011.
- [31] P. Qin, W. Xu, and J. Guo, "An empirical convolutional neural network approach for semantic relation classification," *Neurocomputing*, vol. 190, pp. 1–9, 2016.
- [32] W. Mao, J. Xu, C. Wang, and L. Dong, "A fast and robust model selection algorithm for multi-input multi-output support vector machine," *Neurocomputing*, vol. 130, pp. 10–19, 2014.



# Hindawi

Submit your manuscripts at  
<https://www.hindawi.com>

