Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 4365372, 13 pages
http://dx.doi.org/10.1155/2016/4365372

Research Article

Instance-Wise Denoising Autoencoder for High Dimensional Data

Lin Chen and Wan-Yu Deng

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

Correspondence should be addressed to Wan-Yu Deng; dengwanyu@126.com

Received 9 April 2016; Revised 31 August 2016; Accepted 15 September 2016

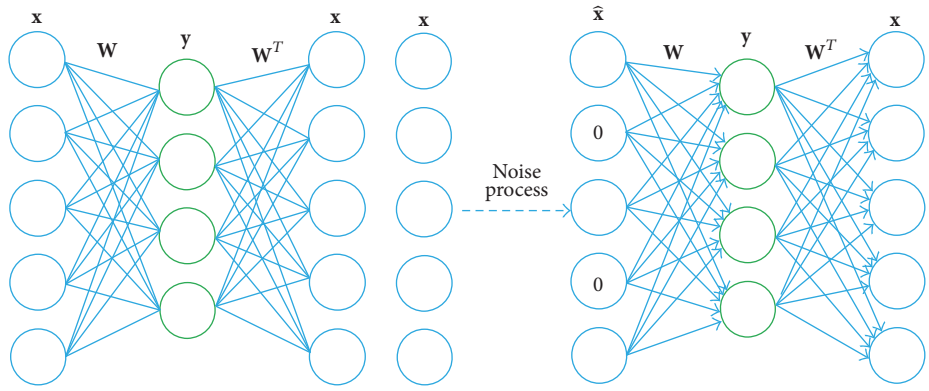
Academic Editor: Erik Cuevas

Denoising Autoencoder (DAE) is one of the most popular fashions that has reported significant success in recent neural network research. To be specific, DAE randomly corrupts some features of the data to zero as to utilize the cooccurrence information while avoiding overfitting. However, existing DAE approaches do not fare well on sparse and high dimensional data. In this paper, we present a Denoising Autoencoder labeled here as Instance-Wise Denoising Autoencoder (IDA), which is designed to work with high dimensional and sparse data by utilizing the instance-wise cooccurrence relation instead of the feature-wise one. IDA works ahead based on the following corruption rule: if an instance vector of nonzero feature is selected, it is forced to become a zero vector. To avoid serious information loss in the event that too many instances are discarded, an ensemble of multiple independent autoencoders built on different corrupted versions of the data is considered. Extensive experimental results on high dimensional and sparse text data show the superiority of IDA in efficiency and effectiveness. IDA is also experimented on the heterogenous transfer learning setting and cross-modal retrieval to study its generality on heterogeneous feature representation.

1. Introduction

Denoising Autoencoder (DAE) [1–5] is an extension of the classical autoencoder [6, 7], where feature denoising is key for the autoencoder to generate better features. In contrast to the classic autoencoder, the input vector in DAE is first corrupted by randomly setting some of features to zero. Then attempts are made to reconstruct the uncorrupted input from the corrupted version. Operating based on the principle of predicting the uncorrupted values from the corrupted input, DAE has been shown to generalize well even with noised input. However, DAE and its variants do not fare well on high dimensional and sparse data since many features are already zeros in nature, and any further reset of the feature vector has no more effects on the original data. Moreover, high dimensional data also lead to uneven distribution of uncorrupted features. To address the above challenges, in this paper, we propose a denoising scheme that is designed for high dimensional and sparse data, which is labeled here as the Instance-Wise Denoising Autoencoder (IDA). To be more specific, if one nonzero feature of the instance is chosen, then this instance will be removed totally. That means that many instances will be removed from the data. Obviously, this will lead to serious information loss. Therefore a recovery strategy is further adopted where multiple independent autoencoders are constructed based on different versions of corrupted inputs and then combined to obtain the final solution. In IDA, instances are directly dropped and thus can reduce the training data size significantly. Obviously, this will be considerably useful to large scale data analytics. Additionally, autoencoders in the model are independent of data retrieval level to command execution level, and this leads IDA natural to be parallelized and carried out on a single multicores CPU computer or a distributed computing platform. In the paper, we verify the performance on classic high dimensional and sparse text data. The experimental results show that the proposed autoencoder is very fast and effective.

Furthermore, we study IDA's application on heterogenous feature representation and propose the Heterogenous-SIDA based on the heterogenous feature fusion framework [8]. Experiments on transfer learning and cross-modal retrieval show that IDA can obtain better performance than mSDA



(a) Classical autoencoder (b) Denoise autoencoder

Figure 1: Structure of classical autoencoder and denoise autoencoder.

which is embedded autoencoder of the fusion framework [8].

The core contributions of the current paper are as follows: (i) an Instance-Wise Denoising Autoencoder (IDA) method is proposed improving generalization performance and efficiency. (ii) A procedure for building a fast deep learning structure rapidly via stacking IDA for large scale high dimensional problems is proposed. (iii) The deep learning approach is further introduced to two heterogenous feature learning tasks including cross-language classification and cross-modal retrieval.

2. Review on Denoising Autoencoder

In a classic autoencoder [6], the aim is to learn a distributed representation that captures the coordinates along the core factors of variation in the data. As shown in Figure 1(a), an autoencoder takes in input $\mathbf{x}$ and maps it to a hidden space $\mathbf{y} = f(\mathbf{W}\mathbf{x})$ through a deterministic mapping with weights $\mathbf{W}$ in the step called *encoder*. Then, in the *decoder* step, the latent space $\mathbf{y}$ or code is mapped back into a reconstructed feature space $\mathbf{z} = g(\mathbf{W}^T\mathbf{y})$ that bears the same shape as $\mathbf{x}$ through a similar transformation g. The parameter of this model $\mathbf{W}$ ($\mathbf{W}^T$ denotes the transposition of $\mathbf{W}$) is optimized such that the average of reconstruction error is minimized. The considered reconstruction error can be the cross-entropy loss [9] or the squared error loss as follows:

$$\min_{\mathbf{W}} \quad \left\| g\left(\mathbf{W}^T\mathbf{y}\right) - \mathbf{x} \right\|_F^2. \tag{1}$$

However, the basic autoencoder alone is not sufficient to be the basis of a deep architecture because it has a tendency of overfitting. In other words, the reconstruction criterion alone is unable to guarantee the extraction of useful features as it can lead to the obvious solution of *simply copying the input* or similarly uninteresting ones that maximize the mutual information in a trivial manner. Denoising Autoencoder (DAE) [1] is an extension of the classical autoencoder introduced specifically to address this phenomenon. As shown in Figure 1(b), DAE is trained to reconstruct a "clean" or "repaired" version of the corrupted input. This is achieved by first corrupting the original input $\mathbf{x}$ to arrive at $\hat{\mathbf{x}}$ by means of a stochastic corruption process consisting in randomly setting some of the values in the input vector to zero [1]. Corrupted input $\hat{\mathbf{x}}$ is then mapped, as with the basic autoencoder, to a hidden representation $\mathbf{y} = f(\hat{\mathbf{x}}\mathbf{W})$ from which we reconstruct a $\mathbf{z} = g(\mathbf{y}\mathbf{W}^T)$. Parameter $\mathbf{W}$ is trained to minimize the average reconstruction error over a training set, that is, to have $\mathbf{z}$ as close as possible to the uncorrupted input $\mathbf{x}$. There is a crucial limitation of DAE, which is high computational cost due to the expensive nonlinear optimization process. To this end, Chen et al. [4] proposed Marginalized Denoising Autoencoders (mDAE) which replace the encoder and decoder with one linear transformation matrix. mDAE provides a closed-form solution for the parameters and thus eliminates the use of other optimization algorithms, for example, stochastic gradient descent and backpropagation. Liang and Liu [10] combined stacked Denoising Autoencoder with dropout technology together and reduced time complexity during fine-tuning phase. Moreover, when the input is heavily corrupted during training the network tends to learn coarse-grained features, whereas when the input is only slightly corrupted, the network tends to learn fine-grained features. To address this problem, Geras and Sutton [3] proposed scheduled Denoising Autoencoders that learn features at multiple different levels of scale which starts with a high level of noise that lowers as training progresses. To reduce the effect of outliers, Jiang et al. [5] proposed a robust $\ell_{2,1}$-norm to measure reconstruction error to learn a more robust model. To improve denoising performance, Cho [11] proposed a simple sparsification method of the latent representation found by the encoder. Wang et al. [12] proposed a probabilistic formulation for stacked denoise autoencoder (SDAE) and then extend it to a relational SDAE (RSDAE) model which jointly performs deep representation learning and relational learning in a principled way under a probabilistic framework. These DAE algorithms address many shortcomings of traditional autoencoders such as their inability in principle to learn useful overcomplete representations and have been shown to generalize well even with noised input. However, DAE and its variants do not fare well on high dimensional and sparse data since many features

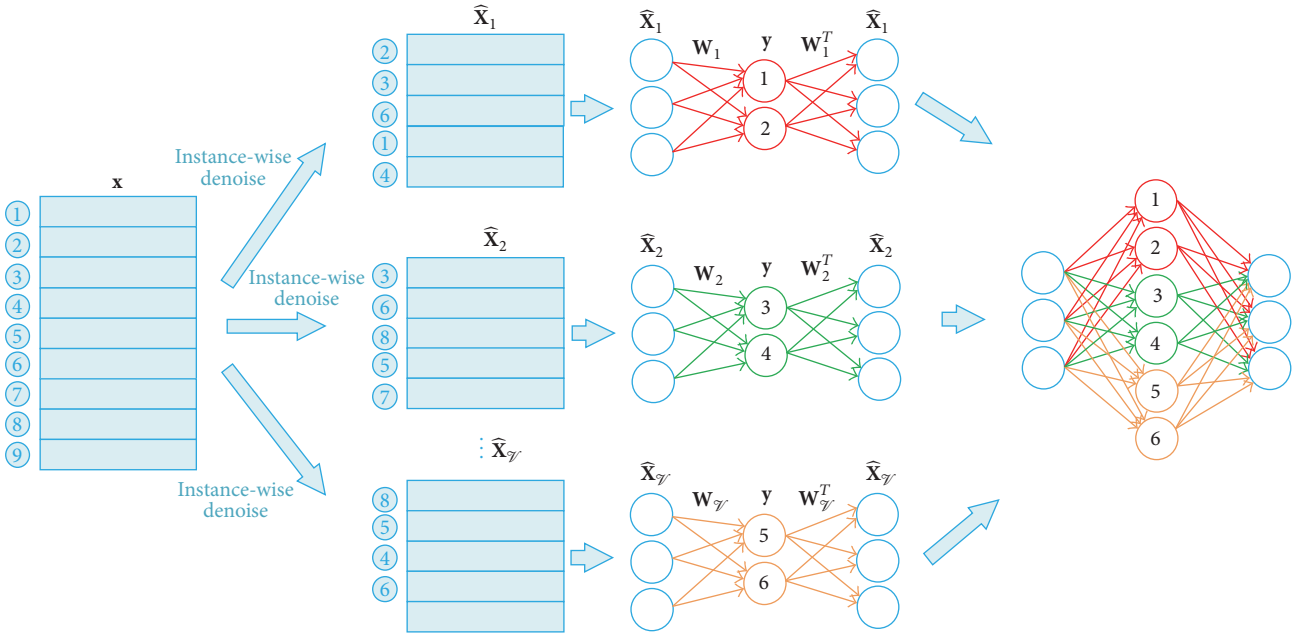


Figure 2: IDA.

are already zero in nature; any further reset of the feature vector has no more effects on the original data. Moreover, high dimensional data also lead to uneven distribution of uncorrupted features. To address the above challenges, in this paper, we propose a denoising scheme that is designed for high dimensional sparse data, which is labeled here as the Instance-Wise Denoising Autoencoder (IDA).

3. Methodology

3.1. Instance-Wise Denoising Autoencoder (IDA). In this section we introduce a novel denoising method of autoencoder, which preserves its strong feature learning capabilities and alleviates the concerns mentioned.

Given N original instances $\mathbf{X} = \{\mathbf{x}_i \mid \mathbf{x}_i \in \mathbf{R}^m\}_{i=1}^N$, and corrupt them by the modified strategy—if one nonzero feature of the instance is selected with a given probability p, then this instance will be reset to zero totally. To be further specific, we generate m-bits boolean vector $\mathbf{m} = [0, 1, 0, \ldots, 1]_{m\times 1}$ with probability p of nonzero occurrence where each element corresponds to a feature. If the indices $nonzero(\mathbf{m})$ of nonzero element of $\mathbf{m}$ and $nonzero(\mathbf{x})$ of an instance $\mathbf{x}$ have overlap (i.e., $nonzero(\mathbf{m}) \cap nonzero(\mathbf{x}) \neq \Phi$), all features of the instance will be reset as $\mathbf{0}$; otherwise, the instance will be retained. It can be written as follows:

$$\widehat{\mathbf{x}} = \begin{cases} \mathbf{x}, & nonzero\,(\mathbf{m}) \cap nonzero\,(\mathbf{x}) \neq \Phi \\ \mathbf{0}, & \text{otherwise.} \end{cases} \tag{2}$$

After denoising, the resultant input is denoted as $\widehat{\mathbf{X}}$. We reconstruct the inputs through minimizing the following reconstruction loss:

$$\min_{\mathbf{W}} \quad \left\| \widehat{\mathbf{X}} - g\left(f\left(\widehat{\mathbf{X}}\mathbf{W}\right)\mathbf{W}^T\right)\right\|_p^\sigma, \tag{3}$$

where σ and p denote different norm with corresponding $\sigma > 0$, $p = 1/2, 1, 2, \ldots, +\infty$. According to different parameters σ, p, f, and g, different coding can be obtained. The optimization methods and computation cost are also different.

(i) g and f Are Linear, $p = 2$, and $\sigma = 1$. The loss function can be rewritten as $\min_{\mathbf{W}} \|\widehat{\mathbf{X}} - \widehat{\mathbf{X}}\mathbf{W}\mathbf{W}^T\|_2^1$. The reconstruction error norm is Frobenius norm [13] and then the solution will be obtained in a closed-form directly.

(ii) f Is Nonlinear and g Is Linear, $p = 2$, and $\sigma = 1$. The loss function can be rewritten as $\min_{\mathbf{W}} \| \widehat{\mathbf{X}} - f(\widehat{\mathbf{X}}\mathbf{W})\mathbf{W}^T\|_2^1$; the solution can be obtained by Extreme Learning Machine Based Autoencoder (ELM-AE) [14] where $\mathbf{W}$ is first randomly assigned and then replaced by optimized result.

As shown in Figure 2, to recover the information loss led by instances corruption, multiple independent encoders combination is adjusted. In particular, $\mathscr{V} > 1$ versions of denoising inputs $\{\widehat{\mathbf{X}}_i\}_{i=1}^{\mathscr{V}}$ and corresponding independent autoencoder are constructed as the following:

$$\begin{aligned} \min_{\mathbf{W}} \quad & \left\| \widehat{\mathbf{X}}_i - g\left(f\left(\widehat{\mathbf{X}}_i\mathbf{W}_i\right)\mathbf{W}_i^T\right)\right\|_p^\sigma \\ \text{subject to:} \quad & i = 1, 2, \ldots, \mathscr{V}, \end{aligned} \tag{4}$$

where $\widehat{\mathbf{X}}_i$ and $\mathbf{W}_i$ denote the different version of corrupted input and its corresponding encoder. Because all autoencoders are independent of the process from data retrieval to operation execution, they are suitable to be parallelized and carried out on a multicore CPU computer or a distributed computing platform.

Input: Training data $\mathbf{X} = \{\mathbf{x}_i \mid \mathbf{x}_i \in \mathbf{R}^m\}_{i=1}^n$, p zeros probability in mask vectors
While $\|\mathbf{X} - f(g(\mathbf{X}\mathbf{W})\mathbf{W}^T)\|_p^\sigma > \theta$ (*error threshold*) **do**
  for $i = 1$ **to** $\mathscr{V}$ **do**
    Generate random mask boolean vector according to the density p:
      $\mathbf{m}_k = [m_1, m_2, \ldots, m_m]$ where $\sum(nonzero(\mathbf{m}_k))/|\mathbf{m}_k| \le p$
    Perform denoising with mask:

$$\widehat{\mathbf{x}}_k = \begin{cases} \mathbf{x}_k, & nonzero\left(\mathbf{m}_k\right) \cap nonzero\left(\mathbf{x}_k\right) \neq \Phi \\ \mathbf{0}, & \text{otherwise,} \end{cases} \qquad k = 1, \ldots, N$$

    Get corrupted inputs $\widehat{\mathbf{X}}_i = \{\widehat{\mathbf{x}}_k \mid \mathbf{x}_k \neq \mathbf{0}\}_{k=1}^{\widehat{N}}$
    Compute autoencoder according to different $f, g, \sigma > 0$, $p = 1/2, 1, 2, \ldots, +\infty$:

$$\min_{\mathbf{W}_i} \left\| \widehat{\mathbf{X}}_i - f\left(g\left(\widehat{\mathbf{X}}_i \mathbf{W}_i\right)\mathbf{W}_i^T\right)\right\|_p^\sigma$$

  end
  Obtain the final solution by combining $\mathscr{V}$ autoencoders together: $\mathbf{W} = [\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_{\mathscr{V}}]$
end
Return: $\mathbf{W}$

Algorithm 1: Instance-Wise Denoising Autoencoder (IDA).

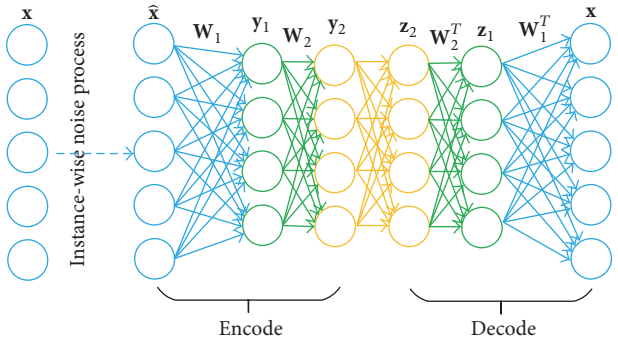


Figure 3: SIDA.

After obtaining the code of every autoencoder, we can reach the final solution by combining them together; that is,

$$\mathbf{W} = \left[\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_{\mathscr{V}}\right]. \tag{5}$$

In comparison to feature-wise denoising scheme where some features are corrupted, Instance-Wise Denoising scheme has the following benefits: (i) tackling the challenging of high dimensional sparse data, (ii) reducing the data instance size used explicitly. For example, for a problem with 1 million data instances, if only 1% of the instances are retained in the corrupted inputs, the computational cost will be reduced to only 0.01% of the original (here we refer to the widely used $\mathcal{O}(N^2M)$ [13]), and (iii) being easy to be implemented in parallel paradigm.

3.2. Stacked Instance-Wise Denoising Autoencoder (SIDA). IDA can be stacked to build deep network which has more than one hidden layer. Generative deep model built by stacking multilayer autoencoders can obtain a more useful representation to express the multilevel structure of image features or other data. Figure 3 shows a typical instance of SIDA structure, which includes two encoding layers and two decoding layers. Supposing there are L hidden layers in the encoding part, we have the activation function of the kth encoding layer:

$$y^{(k+1)} = f\left(\mathbf{W}^{(k+1)}\mathbf{y}^{(k)}\right), \quad k = 0, \ldots, L-1, \tag{6}$$

where the input $\mathbf{y}^{(0)}$ is the original data $\mathbf{x}$. The output $\mathbf{y}^{(L)}$ of the last encoding layer is the high level features extracted by the SIDA network. In the decoding steps, the output of the first decoding layer is regarded as the input of the second decoding layer. The decoding function of the kth decoding layer is

$$\mathbf{z}^{(k+1)} = g\left(\mathbf{W}^{(L-k)T}\mathbf{z}^{(k)}\right), \quad k = 0, \ldots, L-1, \tag{7}$$

where the input $\mathbf{z}^{(0)}$ of the first decoding layer is the output $\mathbf{y}^{(L)}$ of the last encoding layer. The output $\mathbf{z}^{(L)}$ of the last decoding layer is the reconstruction of the original data. The training process of SIDA is provided as follows.

Step 1. Train the first IDA, which includes the first encoding layer and the last decoding layer. Obtain the network weight $\mathbf{W}^{(1)}$ and the output $\mathbf{y}^{(1)}$ of the first encoding layer.

Step 2. Use $\mathbf{y}^{(k)}$ as the input data of the $(k+1)$th encoding layer. Train the $(k+1)$th IDA and obtain $\mathbf{W}^{(k+1)}$ and $\mathbf{y}^{(k+1)}$, where $k = 1, \ldots, L-1$ and L is the number of hidden layers in the network.

It can be seen that each IDA is trained independently, and therefore the training of SIDA is called layer-wise training (Algorithm 1).

3.3. SIDA for Transfer Learning. In the previous sections, we have shown the superiority of our algorithm in computational complexity for single source data. Nevertheless, with the multimedia data becoming current mainstream of information dissemination in the network, heterogenous data mining is

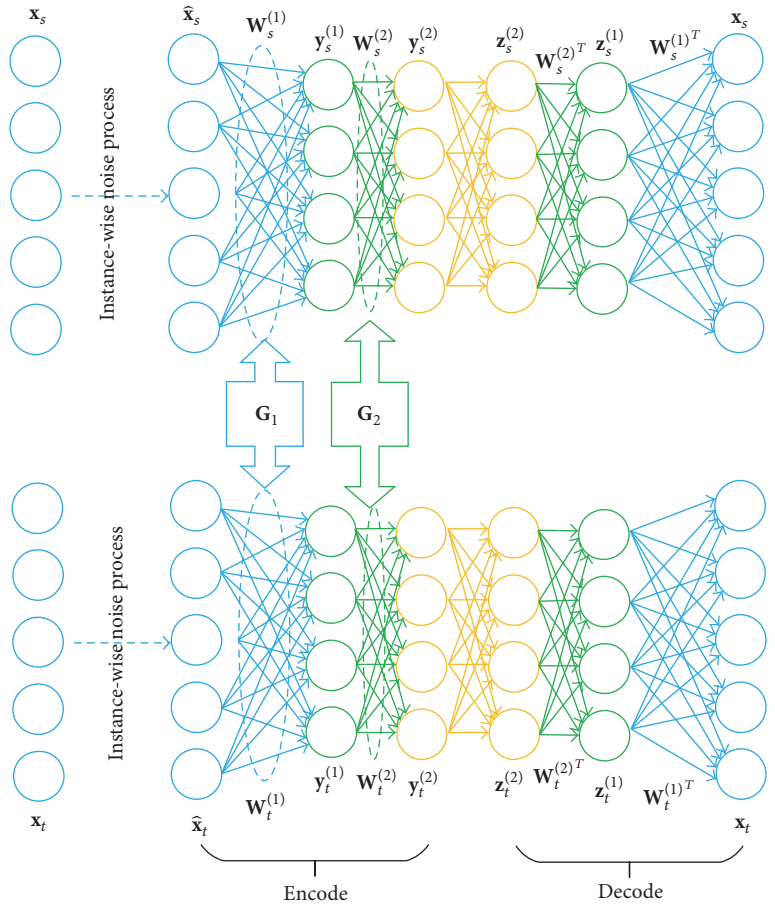


FIGURE 4: Heterogenous-SIDA (for transfer learning and cross-modal retrieval).

more important than ever. In the this section, we are going to discuss how to integrate SIDA into the heterogeneous feature learning framework [15] and then apply it to two classical heterogenous data mining tasks consisting in heterogenous transfer learning and cross-modal retrieval. We term this SIDA for heterogenous data mining as Heterogenous-SIDA.

Transfer learning has demonstrated its success in different applications. Our study focuses on heterogenous transfer learning which aims to learn a feature mapping across heterogeneous feature spaces based on some cross-domain correspondences. In this field, Shi et al. [16] proposed a spectral transformation based heterogenous transfer learning method which employs spectral transformation to map cross-domain data into a common feature space through linear projection. Duan et al. [17] used two different projection matrices to transform the data from two domains into a common subspace and then use two new feature mapping functions to augment the transformed data with their original features and zeros. Kulis et al. [18] proposed learning an asymmetric nonlinear kernel transformation that maps points from one domain to another. Zhou et al. [8] proposed a multiclass heterogeneous transfer learning algorithm that reconstructs a sparse feature transformation matrix to map the weight vector of classifiers learned from the source domain to the target. Glorot et al. [19] trained a stacked denoise autoencoder (SDAE) to reconstruct the input (ignoring the labels) on the union of the source and target data, and then a classifier is trained on the resulting feature representation. Chen et al. [20] proposed a marginalized Stacked Denoising Autoencoder (mSDA) for domain adaptation where the closed-form solution is achieved for SDAE. Zhou et al. [15] further applied mSDA to learn the deep learning structure as well as the feature mappings between cross-domain heterogeneous features to reduce the bias issue caused by the cross-domain correspondences.

The Heterogenous-SIDA model can be trained based on the multilayer heterogenous data fusion framework [15] as in Figure 4. In particular, given a set of data pairs from two different domains $\mathbf{D} = \{(\mathbf{x}_{s_i} \in \mathbf{R}^{m_2}, \mathbf{x}_{t_i} \in \mathbf{R}^{m_1})\}_{i=1}^{n}$, the objective is to learn the weight matrices $\{\mathbf{W}_s^{(k)}, \mathbf{W}_t^{(k)}\}_{k=1}^{K}$ that project the source and target data to the kth hidden layer, $\mathbf{H}_s^{(k)} = f(\mathbf{W}_s^{(k)}\mathbf{X}_s)$ and $\mathbf{H}_t^{(k)} = f(\mathbf{W}_t^{(k)}\mathbf{X}_t)$, respectively, and also two feature mappings $\{\mathbf{G}_s^{(k)}, \mathbf{G}_t^{(k)}\}_{k=1}^{K}$ that map the data to a common space such that the disparity between source and target domain data is minimized:

$$\left\|\mathbf{H}_s^{(k)}\mathbf{G}_s^{(k)} - \mathbf{H}_t^{(k)}\mathbf{G}_t^{(k)}\right\|_F^2 + \lambda_s \left\|\mathbf{G}_s^{(k)}\right\|_F^2 + \lambda_t \left\|\mathbf{G}_t^{(k)}\right\|_F^2, \quad (8)$$

where $\lambda_s > 0$ and $\lambda_t > 0$ are regularization terms to avoid overfitting.

$\{\mathbf{G}_s^{(k)}\}_{k=1}^K$ and $\{\mathbf{G}_t^{(k)}\}_{k=1}^K$ can be computed by alternative optimized algorithm. However, here for simplification, we still let one weight matrix be unit matrix $\mathbf{I}$ and just learn one feature mapping $\{\mathbf{G}^{(k)}\}_{k=1}^K$:

$$\left\|\mathbf{H}_s^{(k)} - \mathbf{H}_t^{(k)}\mathbf{G}^{(k)}\right\|_F^2 + \lambda\left\|\mathbf{G}^{(k)}\right\|_F^2, \tag{9}$$

where $\lambda > 0$ is a regularization term.

The closed-form solution can then be obtained by

$$\mathbf{G}^{(k)} = \mathbf{H}_t^{(k)^T}\left(\mathbf{H}_t^{(k)}\mathbf{H}_t^{(k)^T} + \lambda\mathbf{I}\right)^{-1}\mathbf{H}_s^{(k)}. \tag{10}$$

Sometimes, the correlation between two domains may be nonlinear, so we extend the linear mapping to the nonlinear one by kernel method. The dual form can be written as the following:

$$\left\|\mathbf{H}_s^{(k)} - \mathscr{K}\left(\mathbf{H}_t^{(k)}, \mathbf{H}_t^{(k)}\right)\mathbf{G}^{(k)}\right\|_F^2 + \lambda\left\|\mathbf{G}^{(k)}\right\|_F^2, \tag{11}$$

where $\mathscr{K}$ is kernel function such as RBF kernel.

The feature mapping $\mathbf{G}^{(k)}$ can be obtained by

$$\mathbf{G}^{(k)} = \left(\mathscr{K}\left(\mathbf{H}_t^{(k)}, \mathbf{H}_t^{(k)}\right) + \lambda\mathbf{I}\right)^{-1}\mathbf{H}_s^{(k)}. \tag{12}$$

The representation of $\mathbf{H}_t^{(k)}$ can be written as

$$\mathbf{H}_t^{(k)} = \mathscr{K}\left(\mathbf{H}_t^{(k)}, \mathbf{H}_t^{(k)}\right)\left(\mathscr{K}\left(\mathbf{H}_t^{(k)}, \mathbf{H}_t^{(k)}\right) + \lambda\mathbf{I}\right)^{-1}\mathbf{H}_s^{(k)}. \tag{13}$$

After learning the multilevel features and mappings, for each source domain instance $\mathbf{x}_s$, by denoting $\mathbf{h}_s^{(k)}$ as the representation of the kth layer, one can define a new representation $\mathbf{z}_s$ by augmenting the original features with high level features of all the layers to arrive at $\mathbf{z}_s = [\mathbf{h}_s^{(1)^T}, \ldots, \mathbf{h}_s^{(k)^T}]$, where $\mathbf{h}_s^{(1)} = \mathbf{x}_s$.

In heterogenous transfer learning, besides data pairs $\mathbf{D}_C = \{(\mathbf{x}_{s_i}^{(c)}, \mathbf{x}_{t_i}^{(c)})\}_{i=1}^{N_c}$ which come from the source and target domain, we always can collect a set of target domain unlabeled data $\mathbf{D}_T = \{\mathbf{x}_{t_i}\}_{i=1}^{N_t}$, and a set of source domain labeled data $\mathbf{D}_s = \{(\mathbf{x}_{s_i}, \mathbf{t}_{s_i})\}_{i=1}^{N_s}$. Based on Heterogenous-SIDA trained by $\mathbf{D}_C = \{(\mathbf{x}_{s_i}^{(c)}, \mathbf{x}_{t_i}^{(c)})\}_{i=1}^{N_c}$, we can obtain the augmented features of source data $\mathbf{Z}_s = [\mathbf{H}_s^{(1)^T}, \mathbf{H}_s^{(2)^T}, \ldots, \mathbf{H}_s^{(K)^T}]$; we then apply a standard classification (or regression/logic regression) algorithm on $(\mathbf{Z}_s, \mathbf{T}_s)$ to train a target predictor f_s.

For each target domain instance $\mathbf{x}_t$, the high level feature representation $\{\mathbf{h}_t^{(k)}\}_{k=1}^K$ is first generated, where $\mathbf{h}_t^{(1)} = \mathbf{x}_t$, then perform a feature mapping $\mathbf{z}_t = [\mathscr{K}(\mathbf{h}_t^{(1)}, \mathbf{H}_t^{(1)})^T\mathbf{G}^{(1)}, \ldots, \mathscr{K}(\mathbf{h}_t^{(K)}, \mathbf{H}_t^{(K)})^T\mathbf{G}^{(K)}]$, and finally make prediction by $f_s(\mathbf{z}_t)$.

3.4. SIDA for Cross-Modal Retrieval. Cross-modal retrieval is another important heterogenous feature representation application. Many cross-modal retrieval works focus on this issue through learning common space for two modality feature spaces. Rasiwasia et al. [21, 22] applied CCA to learn a common space between image and text cooccurrence data (image and text occurrence in one document). Semantic matching (SM) [21, 22] is to use Logistic regression in the image and text feature space to extract semantically similar feature to facilitate better matching. Bilinear model (BLM) [23] is a simple and efficient learning algorithm for bilinear models based on the familiar techniques of SVD and EM. LCFS [24] learns two projection matrices to map multimodal data into a common feature space, in which cross-modal data matching can be performed. GMLDA [25] adopts LDA under the multiview feature extraction framework. GMMFA [25] uses MFA for cross-modal retrieval under the multiview feature extraction framework.

In order to apply the proposed approach into cross-modal retrieval, instead of training a classifier, it needs to compute the similarity between cross-modal data in the common space. In particular, given a database $\mathscr{D} = \{D_1, \ldots, D_{|D|}\}$ of documents comprising image and text components, we consider the case where each document consists of a single image and its corresponding text; that is, $D_i = (\mathbf{x}_{s_i}, \mathbf{x}_{t_i})$, where image $\mathbf{x}_{s_i} \in \mathscr{R}^s$ and text $\mathbf{x}_{t_i} \in \mathscr{R}^t$ is represented as vectors in feature spaces $\mathscr{R}^s$ and $\mathscr{R}^t$, respectively. The images $\{\mathbf{x}_{s_i}\}_{i=1}^{|D|}$ and texts $\{\mathbf{x}_{t_i}\}_{i=1}^{|D|}$ are processed by Heterogeneous-SIDA and obtain multilayer weight matrix $\{\mathbf{W}_s^{(k)}\mathbf{W}_t^{(k)}\}_{k=1}^K$ and their corresponding $\{\mathbf{G}^{(k)}\}_{k=1}^K$. Given a query image $\mathbf{x}_s$ (text $\mathbf{x}_t$), its representation can be obtained by $\mathbf{z}_s \in [\mathbf{H}_s^{(1)^T}, \ldots, \mathbf{H}_s^{(K)^T}]$ ($\mathbf{z}_t \in [\mathbf{H}_t^{(1)^T}, \ldots, \mathbf{H}_t^{(K)^T}]$); cross-modal retrieval returns the text (image), represented by $\mathbf{z}_t \in [\mathbf{H}_t^{(1)^T}, \ldots, \mathbf{H}_t^{(K)^T}]$ ($\mathbf{z}_s \in [\mathbf{H}_s^{(1)^T}, \ldots, \mathbf{H}_s^{(K)^T}]$), that minimizes the distance between $\mathbf{z}_s$ and $\mathbf{z}_t$ in the common space by some distance measure such as the ℓ_1-distance [26], normalized correlation (NC) [27], and Kullback-Leibler divergence (KL) [28, 29].

Table 1: News20.bin and Rcv1.mul.

Dataset	# training/testing set	# dimensions
News20.bin	10,000/9,996	1,355,191
Rcv1.mul	15,564/518,571	47,236

4. Experimental Study

In this section, we present the experimental study of IDA on three popular machine learning tasks including text classification, cross-language sentiment classification, and image-text cross-modal retrieval to verify the performance of IDA from multiple aspects.

4.1. Results on High Dimensional Sparse Data. In order to compare the performance of IDA and SIDA (including serial and parallel implementation) on the high dimensional sparse data, here we select two popular datasets, `News20.bin` and `Rcv1.mul` as benchmarks. As detailed information of `News20.bin` and `Rcv1.mul` shown in Table 1, `News20.bin` contains 1,355,191 features and only about 0.0335% are nonzero values, and the dimensions of `Rcv1.mul` are 47,236 while only 0.14% are nonzero values. All the parameters are determined through cross-validation. We select the simple linear SVM as classifier. The experimental results including

Table 2: Training time (seconds) and testing accuracy (%) on News20.bin.

Methods	Testing accuracy (%)	Training time (s)
SDAE 1-layer 4500	95.25	67090
SDAE 2-layer 4500–1200	95.36	91703
SIDA 1-layer 4500	95.46	324.83
SIDA 2-layer 4500–1200	95.70	503.45
SIDA (parallel) 1-layer 4500	95.46	150.83 (2.6x)
SIDA (parallel) 2-layer 4500–1200	95.70	258.21 (1.94x)

Table 3: Training time (seconds) and testing accuracy (%) on Rcv1.mul.

Methods	Testing accuracy (%)	Training time (s)
SDAE 1-layer 4500	88.06	55300
SDAE 2-layer 4500–1200	88.23	69300
SIDA 1-layer 4500	88.14	272.54
SIDA 2-layer 500–1200	88.50	340.71
SIDA (parallel) 1-layer 4500	88.14	140.9 (1.94x)
SIDA (parallel) 2-layer 500–1200	88.50	173.1 (1.96x)

classification accuracy (%) and *training time* (in seconds) are shown in Tables 2 and 3.

We can find that, with nearly the same performance, SIDA is significantly faster than SDAE up to one hundred times. For example, on `News20.bin`, SIDA (2-layer 4500–1200) needs time of 503.45 seconds while SDAE needs about 18000 seconds, which is 100 times compared to SIDA.

When the autoencoders in SIDA are carried out in parallel, we can find that the speed is improved about nearly 2 times. For example, on `News20.bin`, SIDA obtains 2x speedup rate while on `Rcv.mult` it can be improved around 2.3 times. Our computer is a 4-core CPU, and no optimized strategy for parallel running is adopted. We just modify "for" to "parfor" in our Matlab implementation. SIDA can obtain more significant advantage than SDAE in more efficient distributed computing platform.

4.2. Cross-Modal Retrieval on Wikipedia Data. The Wikipedia dataset (http://www.svcl.ucsd.edu/projects/crossmodal/) which has 2866 image-text pairs is a challenging image-text dataset with large intraclass variations and small interclass discrepancies. The context of each text article describes people, places, or events, which are closely relevant to the content of the corresponding image document. There are 10 semantic categories in the Wikipedia dataset, including *art* & *architecture*, *geography* & *places*, *history*, *literature* & *theatre*, *biology*, *media*, *music*, *sports* & *recreation*, *royalty* & *nobility*, and *warfare* as shown in Table 4. Here we follow the data partitioning procedure adopted in [21, 22] where the original dataset is split into a training set of 2173 pairs and a testing set of 693 pairs. Then, we evaluate our proposed method against the following state-of-the-art cross-modal retrieval approaches.

(i) Correlation Matching (CM) [21, 22]. This method applied CCA to learn a common space in which the possibility of whether two different modal data items represent the same semantic concept can be measured.

Table 4: Summary of the Wikipedia dataset.

Category	Training	Retrieval	Documents
Art & architecture	138	34	172
Biology	272	88	360
Geography & places	244	96	340
History	248	85	333
Literature & theatre	202	65	267
Media	178	58	236
Music	186	51	237
Royalty & nobility	144	41	185
Sport & recreation	214	71	285
Warfare	347	104	451

(ii) Semantic Matching (SM) [21, 22]. This method applied Logistic regression in the image and text feature space to extract semantically similar feature to facilitate better matching.

(iii) Semantic Correlation Matching (SCM) [21, 22]. This method applied Logistic regression in the space of CCA projected coefficients (a two-stage learning process).

(iv) Bilinear Model (BLM) [23]. This method is a suite of simple and efficient learning algorithms for bilinear models, based on the familiar techniques of SVD and EM.

(v) Learning Coupled Feature Spaces (LCFS) [24]. This method learns two projection matrices to map multimodal data into a common feature space in which cross-modal data matching can be performed.

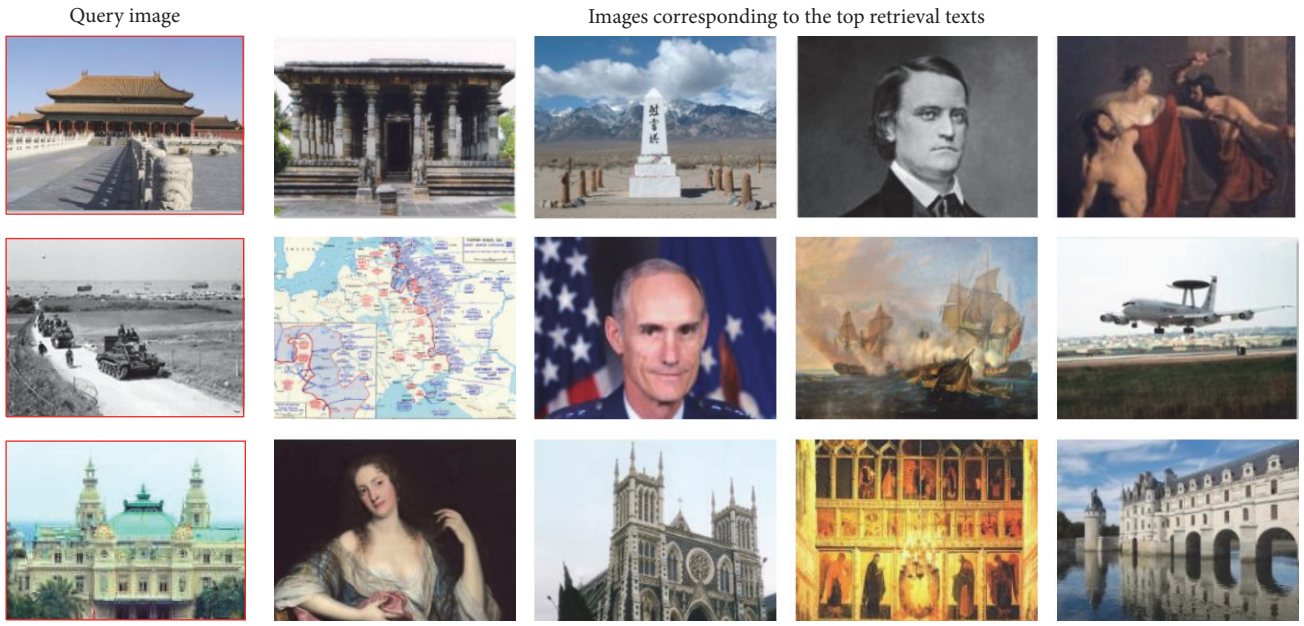


Figure 5: Image-to-text retrieval on Wikipedia. Query images are framed in (a). The four most relevant texts, represented by their ground-truth images, are shown in (b).

(vi) Generalized Multiview Linear Discriminant Analysis (GMLDA) [25]. This method applied LDA with the multiview feature extraction (MFA) framework.

(vii) Generalized Multiview Marginal Fisher Analysis (GMMFA) [25]. This method applied MFA with the multiview feature extraction (MFA) framework.

We here use mean average precision (MAP) to measure the retrieval performance [30]. Two tasks were considered: text retrieval based on an image query and image retrieval based on a query text. In the first case, each image is used as a query and produces ranking of all texts. In the second, the roles of images and text were reversed. The scores for text retrieval from an image query, image retrieval from a text query, and their average are presented in the Table 5. From the results obtained, the following conclusions can be made: (i) the proposed method is shown to be superior to the simple random retrieval which forms the baseline for comparison. (ii) The proposed method outperforms PCA, BLM, GMMFA, GMLDA, LCFS, CM, SM, and SCM [21, 22] on image retrieval given text query and vice versa.

Figure 5 shows several example image queries and the images corresponding to the top retrieved text by the Heterogenous-SIDA. Due to the limitation of pages, we only present the ground-truth images. The query images are framed in Figure 5(a), and the images associated with the four best text matches are shown on Figure 5(b). By comparing the category and text content, each of the top-4 retrieved texts contains one or more relevant words to the image query or they are belonging to the category of query image.

Figure 6 depicted two examples of the text queries and corresponding retrieval results using Heterogenous-SIDA.

Table 5: Retrieval performance (MAP scores).

Methods	Image query	Text query	Average
Random	0.118	0.118	0.118
PCA	0.112	0.173	0.143
BLM	0.256	0.202	0.229
GMMFA	0.275	0.214	0.245
GMLDA	0.275	0.210	0.243
LCFS	0.279	0.214	0.247
CM	0.249	0.196	0.223
SM	0.225	0.223	0.224
SCM	0.277	0.226	0.252
Heterogenous-SIDA	**0.296**	**0.232**	**0.264**

The text query is presented along with its corresponding ground-truth image. The top retrieved five images are shown below the text. By comparing the category and text content, we can find that Heterogenous-SIDA retrieves these images correctly since they are belonging to the category of query text ("history" at the top, "sports" at the bottom) or the corresponding text contains one or more relevant words to the text query.

Figure 7 shows the MAP scores achieved per category by the proposed method and state-of-the-art counterparts, SM, CM, and SCM [21, 22]. Note that, on most categories, the MAP of our method is competitive with those of CM, SM, and SCM.

4.3. Transfer Learning Results. In this section, we present further studies on the performance of IDA for a transfer learning task: cross-language classification. In particular, the

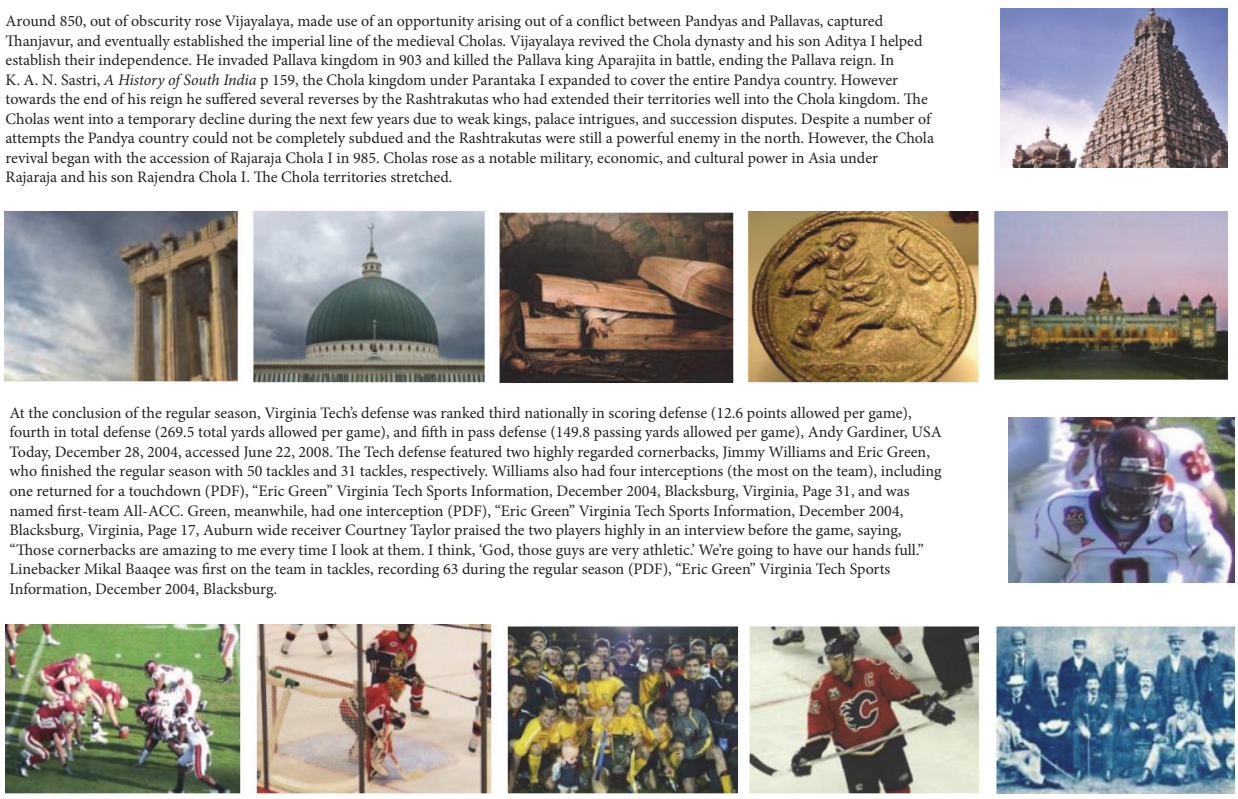


Figure 6: Two examples of text-based cross-modal retrieval using EIDA from Wikipedia. The query text and ground-truth image are shown on the top; retrieved images are presented at the bottom.

cross-language sentiment dataset [31] is considered here. This dataset comprises the Amazon product reviews on three product categories: *Books* (B), *DVDs* (D), and *music* (M). These reviews are written in four languages: *English* (EN), *German* (GE), *French* (FR), and *Japanese* (JP).

For each language, the reviews are split into training and testing set, including 2,000 reviews per categories. We use the *English* reviews in the training dataset as the source domain labeled data and *non-English* (each of the other 3 languages) reviews in a train file as target domain unlabeled data. Further, we use the Google translator on the *non-English* reviews in the testing dataset to construct the cross-domain (English versus non-English) unlabeled parallel data. The performances of all methods are then evaluated on the target domain unlabeled data.

Here we focus on cross-language cross-category learning between *English* and the other 3 languages (*German*, *French*, and *Japanese*). This is a more challenging task than only cross-language. For a comprehensive comparison, we constructed 18 cross-language cross-category sentiment classification tasks as follows:

(i) EN-B-FR-D and EN-B-FR-M.

(ii) EN-B-GE-D and EN-B-GE-M.

(iii) EN-B-JP-D and EN-B-JP-M.

(iv) EN-D-FR-B and EN-D-FR-M.

(v) EN-D-GE-B and EN-D-GE-M.

(vi) EN-D-JP-B and EN-D-JP-M.

(vii) EN-M-FR-B and EN-M-FR-D.

(viii) EN-M-GE-B and EN-M-GE-D.

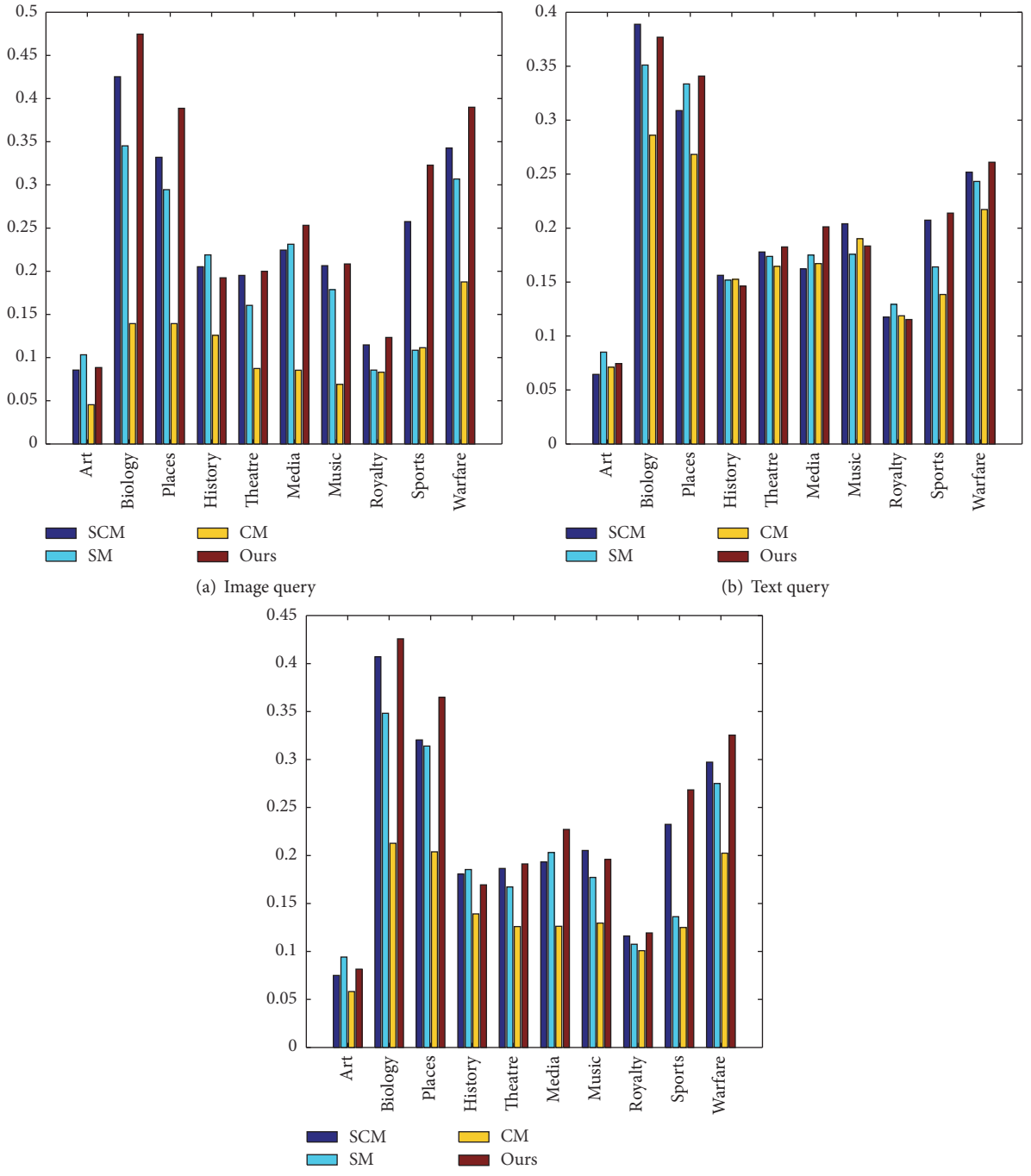
(ix) EN-B-JP-B and EN-B-JP-D.

For example, the task EN-B-FR-D uses all the *Books* reviews in *French* in the testing dataset and its *English* translations as the parallel dataset, the *DVDs* reviews in *French* as the target language testing dataset, and original *English Books* reviews as the source domain labeled data. We compare the proposed method with the following baselines.

(i) SVM-SC [15]. This method first trains a classifier on the source domain labeled data and then predicts the source domain parallel data. By using the correspondence, the predicted labels for source parallel data can be transferred into target parallel data. Next, it trains a model on the target parallel data with predicted labels to make predictions on the target domain test data.

(ii) CL-KCCA [32]. This method applied cross-lingual kernel canonical component analysis on the unlabeled parallel data to learn two projections for the source and target languages and then train a monolingual classifier with the projected source domain labeled data.

(iii) HeMap [16]. This method applied heterogeneous spectral mapping to learn mappings to project two domain data



(a) Image query

(b) Text query

(c) Average performance

Figure 7: MAP performance for each category.

items onto a common feature subspace. However, HeMap does not take the instance correspondence information into consideration.

(iv) mSDA-CCA [33]. This method adopts mSDA to learn a shared feature representation and conduct CCA on the correspondences between domains in the same layers.

(v) HHTL [15]. This method is our previous work where mSDA is applied to learn the deep learning structure as well as the feature mappings between cross-domain heterogeneous features to reduce the bias issue caused by the cross-domain correspondences.

The testing accuracy (%) is then summarized in Table 6. HTTL and Heterogenous-SIDA have the same framework and only difference is that the embedded autoencoder is different (in HTTL the embedded autoencoder is mSDA). Through comparing these two algorithms we can clearly verify the performance of SIDA and mSDA on high dimensional

Table 6: Transfer learning results on Amazon dataset: comparison results in terms of testing accuracy (%).

Task	SVM-SC	CL-KCCA	HeMap	Msda-CCA	HHTL (1 layer)	HHTL (3 layers)	Ours (1 layer)	Ours (3 layers)
EN-B-FR-D	73.25	50.00	49.45	72.96	72.65	76.75	76.2	**78.95**
EN-B-FR-M	62.40	47.20	50.70	64.29	63.30	67.65	70.1	**70.1**
EN-B-GE-D	72.55	77.50	66.12	78.23	70.95	75.10	75.25	**76.90**
EN-B-GE-M	58.95	50.00	48.35	62.52	59.60	69.55	74.3	**76.25**
EN-B-JP-D	69.50	69.46	49.55	71.95	69.45	72.56	70.65	**72.95**
EN-B-JP-M	53.15	55.23	52.05	57.17	53.56	62.39	70.45	**72.6**
EN-D-FR-B	71.35	47.10	48.35	72.32	71.25	79.27	81.3	**81.5**
EN-D-FR-M	71.65	47.10	49.90	68.87	71.70	75.43	77.75	**78.4**
EN-D-GE-B	76.40	54.87	50.80	75.32	74.00	79.35	79.5	**80.55**
EN-D-GE-M	69.05	54.87	50.55	75.69	74.05	78.55	77.4	**78.4**
EN-D-JP-B	70.00	61.16	50.12	72.60	65.55	68.12	72.05	**74.05**
EN-D-JP-M	59.20	66.89	49.80	57.75	66.02	70.58	73.7	**76.35**
EN-M-FR-B	76.20	73.54	50.55	74.23	73.80	75.84	78.75	**80.25**
EN-M-FR-D	73.65	47.10	50.87	72.76	74.15	77.04	75.85	**79.5**
EN-M-GE-B	74.50	81.50	49.45	76.82	75.45	78.30	77.6	**81.75**
EN-M-GE-D	74.60	80.20	50.20	72.28	74.45	81.42	78.5	**81.6**
EN-B-JP-B	67.85	63.75	48.85	68.83	65.35	71.65	71.45	**72.85**
EN-B-JP-D	69.35	60.46	48.80	71.06	68.55	74.25	73.6	**75.65**

Table 7: Transfer learning results on Amazon dataset: comparison results in terms of training time (s).

Task	SVM-SC	CL-KCCA	HeMap	Msda-CCA	HHTL (1 layer)	HHTL (3 layers)	Ours (1 layer)	Ours (3 layers)
EN-B-FR-D	73.25	50.00	49.45	72.96	10.62	14.23	**8.31**	10.54
EN-B-FR-M	62.40	47.20	50.70	64.29	10.22	15.65	**8.23**	10.11
EN-B-GE-D	72.55	77.50	66.12	78.23	10.74	14.34	**8.71**	11.51
EN-B-GE-M	58.95	50.00	48.35	62.52	11.64	13.67	**8.53**	12.24
EN-B-JP-D	69.50	69.46	49.55	71.95	10.74	14.32	**8.91**	11.84
EN-B-JP-M	53.15	55.23	52.05	57.17	10.75	15.74	**8.29**	13.14
EN-D-FR-B	71.35	47.10	48.35	72.32	11.23	14.44	**8.57**	12.57
EN-D-FR-M	71.65	47.10	49.90	68.87	11.36	16.46	**8.90**	11.51
EN-D-GE-B	76.40	54.87	50.80	75.32	10.85	14.86	**8.87**	11.94
EN-D-GE-M	69.05	54.87	50.55	75.69	10.23	14.94	**8.66**	12.64
EN-D-JP-B	70.00	61.16	50.12	72.60	10.68	13.43	**8.60**	12.44
EN-D-JP-M	59.20	66.89	49.80	57.75	11.47	16.27	**8.94**	11.67
EN-M-FR-B	76.20	73.54	50.55	74.23	10.82	14.38	**8.25**	11.98
EN-M-FR-D	73.65	47.10	50.87	72.76	11.43	14.46	**8.16**	12.56
EN-M-GE-B	74.50	81.50	49.45	76.82	10.97	13.64	**8.75**	12.09
EN-M-GE-D	74.60	80.20	50.20	72.28	10.85	14.43	**8.69**	11.17
EN-B-JP-B	67.85	63.75	48.85	68.83	10.68	16.85	**8.91**	11.38
EN-B-JP-D	69.35	60.46	48.80	71.06	10.97	14.93	**8.50**	11.23

and sparse data. The experimental results of our method and HTTL with the same layers show that, whether for 1 layer or 3 layers, our method can produce much better performance than HTTL. This shows that the proposed autoencoder method can learn useful higher-level features to alleviate the distribution bias with the same number of layers. Additionally, the training time of these algorithms is reported in Table 7. We can find that the proposed algorithm is faster than HHTL. For example, For 3 layers, our method is faster than HHTL(3) in most cases. Compared with the other 4 transfer learning methods including SVM-SC, CL-KCCA, HeMap, and mSDA-CCA, the proposed method is also very competitive efficient from the testing accuracy and training time. Due to the deep structure. Our method with multiple layers is not the fastest algorithm, but it showcased improved prediction accuracies over the other counterpart algorithms

significantly. This benefited from more appropriate high level features of SIDA and better cross-domain knowledge transfer in each layer.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work is supported by National Science Foundation of China 61572399, 61373116, and 61272120; Shaanxi New Star of Science & Technology 2013KJXX-29; New Star Team of Xian University of Posts & Telecommunications; Provincial Key Disciplines Construction Fund of General Institutions of Higher Education in Shaanxi.

References

[1] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[2] S. Wang, Z. Ding, and Y. Fu, "Coupled marginalized auto-encoders for cross-domain multi-view learning," in *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI '16)*, pp. 2125–2131, New York, NY, USA, July 2016.

[3] K. J. Geras and C. Sutton, "Scheduled denoising autoencoders," https://arxiv.org/abs/1406.3269.

[4] M. Chen, K. Weinberger, F. Sha, and Y. Bengio, "Marginalized denoising auto-encoders for nonlinear representations," in *Proceedings of the 31st International Conference on Machine Learning (ICML '14)*, pp. 3342–3350, June 2014.

[5] W. Jiang, H. Gao, F.-L. Chung, and H. Huang, "The $l_{2,1}$-norm stacked robust autoencoders for domain adaptation," in *Proceedings of the 13th AAAI Conference on Artificial Intelligence (AAAI '16)*, pp. 1723–1729, Phoenix, Ariz, USA, 2016.

[6] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[7] M. Ranzato, Y.-L. Boureau, and Y. LeCun, "Sparse feature learning for deep belief networks," in *Advances in Neural Information Processing Systems*, pp. 1185–1192, 2008.

[8] J. T. Zhou, I. W. Tsang, S. J. Pan, and M. Tan, "Heterogeneous domain adaptation for multiple classes," in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS '14)*, pp. 1095–1103, Reykjavik, Iceland, 2014.

[9] P.-T. de Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," *Annals of Operations Research*, vol. 134, pp. 19–67, 2005.

[10] J. Liang and R. Liu, "Stacked denoising autoencoder and dropout together to prevent overfitting in deep neural network," in *Proceedings of the 8th International Congress on Image and Signal Processing (CISP '15)*, pp. 697–701, IEEE, Shenyang, China, October 2015.

[11] K. Cho, "Simple sparsification improves sparse denoising autoencoders in denoising highly corrupted images," in *Proceedings of the 30th International Conference on Machine Learning (ICML '13)*, pp. 432–440, 2013.

[12] H. Wang, X. Shi, and D.-Y. Yeung, "Relational stacked denoising autoencoder for tag recommendation," in *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI '15)*, pp. 3052–3058, Austin, Tex, USA, January 2015.

[13] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, Md, USA, 3rd edition, 1996.

[14] E. Cambria, G.-B. Huang, L. L. C. Kasun et al., "Extreme learning machines [trends & controversies]," *IEEE Intelligent Systems*, vol. 28, no. 6, pp. 30–59, 2013.

[15] J. T. Zhou, S. J. Pan, I. W. Tsang, and Y. Yan, "Hybrid heterogeneous transfer learning through deep learning," in *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, vol. 3, pp. 2213–2219, 2014.

[16] X. Shi, Q. Liu, W. Fan, P. S. Yu, and R. Zhu, "Transfer learning on heterogenous feature spaces via spectral transformation," in *Proceedings of the 10th IEEE International Conference on Data Mining (ICDM '10)*, pp. 1049–1054, IEEE, Sydney, Australia, December 2010.

[17] L. Duan, D. Xu, and I. Tsang, "Learning with augmented features for heterogeneous domain adaptation," https://arxiv.org/abs/1206.4660.

[18] B. Kulis, K. Saenko, and T. Darrell, "What you saw is not what you get: domain adaptation using asymmetric kernel transforms," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 1785–1792, IEEE, Colorado Springs, Colo, USA, 2011.

[19] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: a deep learning approach," in *Proceedings of the 28th International Conference on Machine Learning (ICML '11)*, vol. 27, pp. 97–110, 2011.

[20] M. Chen, Z. Xu, K. Weinberger, and F. Sha, "Marginalized denoising autoencoders for domain adaptation," https://arxiv.org/abs/1206.4683.

[21] N. Rasiwasia, J. Costa Pereira, E. Coviello et al., "A new approach to cross-modal multimedia retrieval," in *Proceedings of the 18th ACM International Conference on Multimedia (MM '10)*, pp. 251–260, ACM, 2010.

[22] J. C. Pereira, E. Coviello, G. Doyle et al., "On the role of correlation and abstraction in cross-modal multimedia retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, pp. 521–535, 2014.

[23] J. B. Tenenbaum and W. T. Freeman, "Separating style and content with bilinear models," *Neural Computation*, vol. 12, no. 6, pp. 1247–1283, 2000.

[24] K. Wang, R. He, W. Wang, L. Wang, and T. Tan, "Learning coupled feature spaces for cross-modal matching," in *Proceedings of the 14th IEEE International Conference on Computer Vision (ICCV '13)*, pp. 2088–2095, IEEE, Sydney, Australia, December 2013.

[25] A. Sharma, A. Kumar, H. Daume, and D. W. Jacobs, "Generalized Multiview Analysis: a discriminative latent space," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 2160–2167, June 2012.

[26] H. Wang, F. Nie, and H. Huang, "Robust distance metric learning via simultaneous ll-norm minimization and maximization," in *Proceedings of the 31st International Conference on Machine Learning (ICML '14)*, pp. 1836–1844, Beijing, China, June 2014.

[27] S.-D. Wei and S.-H. Lai, "Fast template matching based on normalized cross correlation with adaptive multilevel winner update," *IEEE Transactions on Image Processing*, vol. 17, no. 11, pp. 2227–2235, 2008.

[28] F. Emmert-Streib and M. Dehmer, *Information Theory and Statistical Learning*, Springer, Berlin, Germany, 2010.

[29] G. Navarro, "A guided tour to approximate string matching," *ACM Computing Surveys*, vol. 33, no. 1, pp. 31–88, 2001.

[30] N. Rasiwasia, P. J. Moreno, and N. Vasconcelos, "Bridging the gap: query by semantic example," *IEEE Transactions on Multimedia*, vol. 9, no. 5, pp. 923–938, 2007.

[31] P. Prettenhofer and B. Stein, "Cross-language text classification using structural correspondence learning," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL '10)*, pp. 1118–1127, Association for Computational Linguistics, Stroudsburg, Pa, USA, 2010, http://dl.acm.org/citation.cfm?id=1858681.1858795.

[32] A. Vinokourov, N. Cristianini, and J. Shawe-Taylor, "Inferring a semantic representation of text via cross-language correlation analysis," in *Proceedings of the 16th Annual Neural Information Processing Systems Conference (NIPS '02)*, pp. 1473–1480, December 2002.

[33] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th International Conference on Machine Learning (ICML '11)*, pp. 689–696, Bellevue, Wash, USA, 2011.