Heterogeneous Transfer Learning via Deep Matrix Completion with Adversarial Kernel Embedding

Haoliang Li[1]**, Sinno Jialin Pan**[2]**, Renjie Wan**[1]**, Alex C. Kot**[1]
[1]Rapid-Rich Object Search (ROSE) Lab, Nanyang Technological University, Singapore
[2]School of Computer Science and Engineering, Nanyang Technological University, Singapore
{lihaoliang,sinnopan,rjwan,eackot}@ntu.edu.sg

Abstract

Heterogeneous Transfer Learning (HTL) aims to solve transfer learning problems where a source domain and a target domain are of heterogeneous types of features. Most existing HTL approaches either explicitly learn feature mappings between the heterogeneous domains or implicitly reconstruct heterogeneous cross-domain features based on matrix completion techniques. In this paper, we propose a new HTL method based on a deep matrix completion framework, where kernel embedding of distributions is trained in an adversarial manner for learning heterogeneous features across domains. We conduct extensive experiments on two different vision tasks to demonstrate the effectiveness of our proposed method compared with a number of baseline methods.

Introduction

Transfer learning aims to transfer knowledge from one domain with sufficient labeled data to another domain where labeled data is sparse or no labeled data is available. In the literature, many existing transfer learning approaches focus on cross-domain learning problems of homogeneous features, which are referred to as homogeneous transfer learning problems (Pan and Yang 2010). However, there exist many other applications where data from the source domain and the target domain is characterized by different sets of features, which are referred to as heterogeneous transfer learning (HTL) problems. For example, in some computer vision problems, one may extract powerful deep features with a well-trained deep learning network in a domain where sufficient labeled data is available for training. However, in some other domains, training data may be protected by a privacy policy (e.g. EU data protection rule (Carey 2018)). In this case, one cannot employ deep learning but use handcrafted features to represent the data. In some other applications, one domain can be text (e.g. food recipe) and the other can be images (e.g. photos of food) which also leads to heterogeneous feature presentation. Thus, it is highly desired if knowledge extracted from the learning tasks with the public available data can be transferred to help the learning tasks where the corresponding training data is regularized by the privacy policy. In this context, as each data instance, e.g., an image, is represented by heterogeneous features across different domain, HTL techniques are crucial.

Recently, matrix completion based methods have been proposed for HTL problems (Xiao and Guo 2013; Zhou et al. 2016). The key idea is that instead of learning an (or a pair of) explicit feature mapping(s) between feature spaces of the source domain and the target domain, one can directly reconstruct cross-domain heterogeneous features for each instance through matrix completion techniques. For instance, recently, Zhou et al. (2016) proposed a method named Distribution Matching based Matrix Completion (DMMC) to encode the distance in distributions between domains into feature reconstruction in order to reduce the distance between domains. However, existing matrix-completion-based methods have several major problems. First, they directly reduce the distance between domains and learn a classifier based on the recovered feature, which may fail to capture the intrinsic underlying low-dimensional structure, which indeed may benefit knowledge transfer and classifier training. In addition, to make the optimization problem tractable, DMMC adopts the Maximum Mean Discrepancy (MMD) metric (Gretton et al. 2006; 2012) with a linear kernel to measure the distance between distributions, rather than a characteristic kernel (Sriperumbudur et al. 2009). As a result, the distance between distributions may not be measured precisely. Moreover, they require plenty of corresponding instances between the source domain and the target domain when performing matrix completion, which limits their applications in real-world problems as the corresponding data instances may be difficult to collect in some scenarios.

To address the aforementioned issues, in this paper, we propose a novel algorithm named **Deep M**atrix **C**ompletion with **A**dversarial Kernel Embedding (Deep-MCA) to jointly perform matrix completion with distribution matching in Reproducing Kernel Hilbert Space (RKHS) and learn a classifier for the target domain. To be specific, we propose an auto-encoder style architecture for matrix completion and latent feature representation learning. Subsequently, data instances from both the source domain and the target domain are mapped to a RKHS induced by an adversarially trained kernel. As all instances are in the RKHS, using MMD under a proper bounded function can provide more accurate distance measure between distributions, which makes knowledge transfer more effective. Moreover, a classifier is trained in the RHKS in the semi-supervised manner. We conduct extensive experiments on two different vision tasks to verify the effectiveness of our proposed method.

The contributions of our work are three folds.

- We propose a novel deep neural network architecture for HTL problems, where the learned latent features can better benefit distribution matching and classifier training.
- Different from previous on employing linear kernel for measuring MMD (Zhou et al. 2016), our proposed method learns a suitable function under a RKHS in order to obtain more accurate distribution measure. The function can be jointly learned with other components.
- Experimental results on different vision tasks demonstrate that our proposed method achieves superior performance in terms of classification accuracy over some state-of-the-art HTL methods.

Related Work

Traditional transfer learning methods focus on learning problems where the source domain and target domain are represented by the same type of features (Pan et al. 2011; Gong et al. 2012), which cannot be directly applied to HTL problems. Generally, existing HTL approaches can be categorized into two groups. A first group requires a few labeled data and some unlabeled in the target domain for training, which is referred to as semi-supervised HTL. Early works (Wang and Mahadevan 2011; Harel and Mannor 2011; Shi et al. 2010) focused on aligning heterogenous features in a common latent space for knowledge transfer. Kulis, Saenko, and Darrell (2011) proposed to learn a metric for instances from heterogeneous domains. Zhou et al. (2014) and Xiao and Guo (2015) proposed semi-supervised HTL methods for multi-class classification problems by exploiting the Error-Correcting Output Coding (ECOC) scheme, respectively. Chen et al. (2016) proposed a neural network based transfer learning approach for cross-domain feature adaptation. Tsai, Yeh, and Frank Wang (2016) proposed a landmark selection strategy to align MMD and conditional MMD based on label information. More recently, Yan et al. (2018) proposed to use Gromov-Wasserstein discrepancy instead of MMD for distribution matching with semantic consistency regularization.

Feature augmentation, which was originally introduced in (DauméIII 2007) for homogeneous transfer learning, can also be applied to HTL problems based on this setting. The idea was to augment the original feature space $\mathbb{R}^d$ to $\mathbb{R}^{3d}$, where the source domain feature $\mathbf{x}_S$ is augmented as $[\mathbf{x}_S, \mathbf{x}_S, \mathbf{0}]$ and the target domain feature $\mathbf{x}_T$ is augmented as $[\mathbf{x}_T, \mathbf{0}, \mathbf{x}_T]$. Here, $\mathbf{0}$ denotes the vector of all zeros with dimension d. As such, the connection between the source domain and the target domain can be established. In (Duan, Xu, and Tsang 2012), the idea was extended to HTL problems by introducing a common subspace for the source domain and the target domain. In particular, two projection matrices $\mathbf{P}$ and $\mathbf{Q}$ are introduced for the source domain and the target domain, respectively. The common space is then formulated in a feature augmentation manner as $[\mathbf{P}\mathbf{x}_S, \mathbf{x}_S, \mathbf{0}_T]$ and $[\mathbf{Q}\mathbf{x}_T, \mathbf{0}_S, \mathbf{x}_T]$, where $\mathbf{0}_S$ and $\mathbf{0}_T$ denote the vectors of all zeros with the same dimension as $\mathbf{x}_S$ and $\mathbf{x}_T$, respectively. The HTL problem is solved by jointly optimizing $\mathbf{P}$, $\mathbf{Q}$ as well as the parameters of classifier (e.g. SVM). A follow-up semi-supervised method was proposed by Li et al. (2014), where unlabeled target domain data are utilized during the training process. Our proposed method also leverages the advantage of heterogeneous feature augmentation with few labeled target domain data, which has proven to be effective for HTL.

Another group of approaches does not require labeled data in the target domain but a set of unlabeled correspondences between the source domain and the target domain for training. Existing matrix completion based approaches (Xiao and Guo 2013; Zhou et al. 2016), which also leveraged feature augmentation, fall into this group and will be briefly reviewed as preliminary in the next section. Different from matrix completion based approaches, Dai et al. (2008) and Prettenhofer and Stein (2010) proposed to learn a feature mapping between the heterogeneous features using feature-level correspondences, e.g., word-level translations. It is also worth noting that in our problem setting, though we propose a matrix completion based method for HTL, we do not require any correspondences between domains. However, a few labeled data in the target domain need to be provided in advance for training, which is different from existing matrix completion based approaches.

More prior information/assumption can be leveraged to make the heterogeneous transfer learning problem more trackable. Zhuang et al. (2012) proposed to use multiple domains information based on Probabilistic Latent Semantic Analysis to align text distributions caused by different index words. Yang et al. (2016) proposed to learn the transferred weights with the aid of co-occurrence data which contain the same set of instances but in different feature spaces. Luo, Wen, and Tao (2017) proposed to leverage the data from multiple domains to learn high-order statistics in a multitask metric learning manner.

Model Formulation

Problem Statement and Preliminary

In this paper, we focus on semi-supervised transfer learning problems, where besides plenty of source-domain labeled data, there are a few labeled instances and some unlabeled instances in the target domain for training. We denote by $\mathbf{X}_S = [\mathbf{x}_{S_1}^\top, ..., \mathbf{x}_{S_{N_S}}^\top]^\top$ and $\mathbf{Y}_S \in \mathbb{R}^{N_S \times 1}$ the source-domain input matrix with each row being an instance $\mathbf{x}_{S_i} \in \mathbb{R}^{1 \times d_S}$, and the corresponding output vector, respectively. We also denote by $\mathbf{X}_T = [\mathbf{x}_{T_1}^\top, ..., \mathbf{x}_{T_{N_T}}^\top]^\top$ the target-domain input matrix with $\mathbf{x}_{T_i} \in \mathbb{R}^{1 \times d_T}$. In HTL, $\mathbf{x}_{S_i}$ and $\mathbf{x}_{T_i}$ are of heterogeneous features, and thus in general $d_S \neq d_T$. Suppose the first N_{T_l} instances in $\mathbf{X}_T$ are labeled, whose corresponding label vector is denoted by $\mathbf{Y}_{T_l} \in \mathbb{R}^{N_{T_l} \times 1}$, and the rest $N_{T_u} = N_T - N_{T_l}$ are unlabeled. We assume the two domains share the same set of class labels.

Before introducing our proposed algorithm, we first revisit existing matrix-completion-based methods for HTL (Xiao and Guo 2013; Zhou et al. 2016). As discussed in the previous section, heterogeneous features augmentation introduces zero-padding for missing features. Thus, one intuitive idea is to recover such missing features using matrix completion for more effective knowledge transfer. Given the source-domain input matrix $\mathbf{X}_S$ and the target-domain input matrix $\mathbf{X}_T$, one can first augment the data by simply padding zeros, which are considered as missing values, to make the dimensions of the data from the two domains identical[1],

$$\mathbf{X} = \begin{pmatrix} \mathbf{X}_S & \mathbf{0} \\ \mathbf{0} & \mathbf{X}_T \end{pmatrix} \in \mathbb{R}^{(N_S+N_T)\times(d_S+d_T)}. \quad (1)$$

The goal of matrix completion based methods is to reconstruct the missing entries in $\mathbf{X}$ via solving

$$\min_{\mathbf{X}_r} \|\mathbf{P} \circ (\mathbf{X} - \mathbf{X}_r)\|_F^2 + \lambda\Omega(\mathbf{X}_r), \quad (2)$$

where $\mathbf{X}_r \in \mathbb{R}^{(N_S+N_T)\times(d_S+d_T)}$ is the recovered augmented matrix, where each row is an instance with the original features and the learned augmented features, $\mathbf{P}$ is an indicator matrix with $\mathbf{P}_{ij} = 1$ if $\mathbf{X}_{ij}$ is observed, otherwise 0, the operator $\circ$ is the Hadamard product, $\Omega(\mathbf{X}_r)$ is a regularization term on $\mathbf{X}_r$, and $\lambda > 0$ is a tradeoff parameter.

Deep Matrix Completion

We reformulate the matrix completion method in (2) with an auto-encoder style framework by jointly reconstructing the missing entries in $\mathbf{X}$ and learning a latent feature representation based on $\mathbf{X}$. To be specific, we introduce an encoder $W(\cdot)$ and decoder $V(\cdot)$, which are multi-layer fully-connected neural networks. Given a sparse augmented feature input $\mathbf{x} \in \mathbb{R}^{1\times(d_S+d_T)}$ drawn from $\mathbf{X}$, where $\mathbf{x} = [\mathbf{x}_S, \mathbf{0}]$ if it is from the source and $\mathbf{x} = [\mathbf{0}, \mathbf{x}_T]$ otherwise, we first map $\mathbf{x}$ through the encoder W to obtain a dense latent feature representation $\mathbf{h}$, where $\mathbf{h} = W(\mathbf{x})$. We then aim to reconstruct the missing entries in $\mathbf{x}$ to obtain the dense reconstructed $\mathbf{x}_r$ through the decoder V. That means $\mathbf{x}_r$ can be represented as $\mathbf{x}_r = V(W(\mathbf{x}))$.

For implementation, we consider the augmented matrix $\mathbf{X}$ in (1) as the input. We further denote the latent feature representation and reconstructed output as $\mathbf{H}$ and $\mathbf{X}_r$, respectively. Thus, equation (2) can be reformulated in a deep learning based framework, which can be represented by

$$\min_{W,V} \|\mathbf{P} \circ (\mathbf{X} - V(W(\mathbf{X})))\|_F^2 + \lambda\|V(W(\mathbf{X}))\|_*. \quad (3)$$

Here we impose low-rank regularization on $V(W(\mathbf{X}))$ induced by the nuclear norm $\|\cdot\|_*$. The motivations behind this are two folds. First, the dimensions of augmented feature are supposed to be linearly dependent as there may be some redundant information among heterogeneous features across domains. Second, in many applications, high-dimensional data is usually controlled by a few latent factors.

Different from (Zhou et al. 2016), which recovers the matrix based on adapted matrix completion techniques, our method is based on deep learning to perform non-linear matrix factorization to reconstruct missing values of the matrix, where the "dictionary" V and learned representation $W(\mathbf{X})$ can be jointly learned. By doing so, the low-dimensional representative information can be explored, which can benefit distribution matching and classifier training.

The gradient of (3) w.r.t W and V can be computed by the chain rule with $\frac{\partial \mathcal{L}_{mc}(W,V)}{\partial \mathbf{X}_r}$, where $\mathcal{L}_{mc}(W, V)$ denotes the objective in (3). As the nuclear norm is non-differentiable, we follow (Bernstein 2005) to compute the subgradient of (3), which is given by

$$\frac{\partial \mathcal{L}_{mc}(W,V)}{\partial \mathbf{X}_r} = \mathbf{P} \circ (\mathbf{X} - \mathbf{X}_r) + \lambda \mathbf{A}\mathbf{B}^\top \quad (4)$$

where $\mathbf{A}$ and $\mathbf{B}^\top$ are the components induced by performing SVD on $\mathbf{X}_r$ as $\mathbf{X}_r = \mathbf{A}\mathbf{\Sigma}\mathbf{B}^\top$, the gradient respect to W and V can be computed through network back-propagation with $\frac{\partial \mathcal{L}_{mc}(W,V)}{\partial \mathbf{X}_r}$.

Distribution Matching

The network described so far can estimate the missing values, i.e., the missing heterogenous features. However, we also expect to extract domain-invariant information between the source domain and the target domain. We aim to achieve this by minimizing MMD (Gretton et al. 2006) between the two domains with the latent representation.

Specifically, with the hidden features $\mathbf{H}$ obtained by the W, we split it as $\mathbf{H} = [\mathbf{H}_S; \mathbf{H}_T]$ where $\mathbf{H}_S = W([\mathbf{X}_S, \mathbf{0}])$ and $\mathbf{H}_T = W([\mathbf{0}, \mathbf{X}_T])$ following two probability distributions $\mathbb{P}_S$ and $\mathbb{P}_T$, respectively. The technique of kernel embedding (Gretton et al. 2006) for representing an arbitrary distribution is to introduce a mean map operation $\mu(\cdot)$ to map instances to a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$, and to compute their mean in the RKHS as follows,

$$\boldsymbol{\mu}_{\mathbb{P}} := \mu(\mathbb{P}) = \mathbb{E}_{\mathbf{x}\sim\mathbb{P}}[\phi(\mathbf{x})] = \mathbb{E}_{\mathbf{x}\sim\mathbb{P}}[k(\mathbf{x}, \cdot)], \quad (5)$$

where $\phi : \mathbb{R}^d \to \mathcal{H}$ is a feature map, and $k(\cdot, \cdot)$ is the kernel function induced by $\phi(\cdot)$. If the condition $\mathbb{E}_{\mathbf{x}\sim\mathbb{P}}(k(\mathbf{x}, \mathbf{x})) < \infty$ is satisfied, then $\boldsymbol{\mu}_{\mathbb{P}}$ is also an element in $\mathcal{H}$. It has been proven that if the kernel $k(\cdot, \cdot)$ is characteristic, then the mapping $\mu : \mathcal{P} \to \mathcal{H}$ is injective (Sriperumbudur et al. 2009). The injectivity indicates an arbitrary probability distribution $\mathbb{P}$ is uniquely represented by an element in a RKHS through the mean map. Based on the MMD theory (Gretton et al. 2006), the distance between the source domain and the target domain (or $\mathbb{P}_S$ and $\mathbb{P}_T$) can be measured by

$$\begin{aligned} \text{MMD}(\mathbf{H}_S, \mathbf{H}_T) &= \sup_{f\in\mathcal{H}}\{\langle f, \mu_{\mathbb{P}_S} - \mu_{\mathbb{P}_T}\rangle\} \\ &= \|\mu_{\mathbb{P}_S} - \mu_{\mathbb{P}_T}\|_{\mathcal{H}}, \end{aligned} \quad (6)$$

where f is a measure function defined in RKHS.

Adversarial Kernel Embedding Training

One intuitive solution to align the distribution between source and target domain is imposing MMD with a predefined kernel (Pan et al. 2011) on $\mathbf{H}_S$ and $\mathbf{H}_T$, which leads to an identity function of f, or a combination of kernels (Long et al. 2015), where f is learned through multiple kernel learning. However, it is not easy to identify an optimal kernel or a set of them to measure the MMD term with the hidden features learned by deep neural networks. To bridging

[1] It should be noted that we omit the corresponding data (Xiao and Guo 2013; Zhou et al. 2016) here as the corresopnding data are not available in our setting.

the gap between deep learning and the MMD measurement, Li, Swersky, and Zemel (2015) proposed a multi-layer data space network to parameterize f by mapping the data from one space to another which is suitable for MMD measurement. Though their work was originally proposed for generation tasks, it tackles a similar problem as transfer learning problems we focus on in a high level, as it aims to map the data drawn from one domain (noise) to another domain (data space) where the distributions of the training data and the generated data are aligned in terms of the MMD distance in (6).

Therefore, we introduce an encoding network f_{ϕ_e} to parameterize f, and reformulate (6) as

$$\begin{aligned}&\text{MMD}(\mathbf{H}_S, \mathbf{H}_T)\\ &\quad = \max_{f_{\phi_e}} \left\| \frac{1}{N_S} f_{\phi_e}(\mathbf{H}_S)^\top \mathbf{1}_S - \frac{1}{N_T} f_{\phi_e}(\mathbf{H}_T)^\top \mathbf{1}_T \right\|_F^2 ,(7)\end{aligned}$$

where $\mathbf{1}_S$ and $\mathbf{1}_T$ are all-one vectors with the size N_S and N_T, respectively. $\frac{1}{N_S} f_{\phi_e}(\mathbf{H}_S)^\top \mathbf{1}_S$ and $\frac{1}{N_T} f_{\phi_e}(\mathbf{H}_T)^\top \mathbf{1}_T$ are the empirical measure (Gretton et al. 2006) of $\mu_{\mathbb{P}_S}$ and $\mu_{\mathbb{P}_T}$, respectively.

Aligning the distribution between source and target domain can be achieved by minimizing W as

$$\min_W \max_{f_{\phi_e}} \left\| \frac{1}{N_S} f_{\phi_e}(\mathbf{H}_S)^\top \mathbf{1}_S - \frac{1}{N_T} f_{\phi_e}(\mathbf{H}_T)^\top \mathbf{1}_T \right\|_F^2 . \quad (8)$$

Note that (8) is related to Generative Adversarial Network (GAN) (Goodfellow et al. 2014). In GAN, there are two types of networks: a generative model G that aims to capture the distribution of the training data for data generation, and a discriminative model D that aims to distinguish between the instances drawn from G and the instances sampled from the dataset. In our model, the network f_{ϕ_e} works in a similar manner as D to maximize MMD in order to distinguish distributions, and W is trained to map the input to the latent feature space where MMD is minimized. These two components are jointly trained in a competitive fashion: 1) to train f_{ϕ_e} to distinguish the two feature distributions generated by W, and 2) to train W to fool f_{ϕ_e} with its latent features.

To make the kernel embedding more effective, there are two issues need to be considered. First, the continuous function is supposed to be bounded in the RKHS (Theorem 3.5 in (Muandet et al. 2017)). To achieve this condition, we impose locally Lipschitz constraint by adding weight clipping on f_{ϕ_e}. Second, f needs to be injective (Sriperumbudur et al. 2009; Muandet et al. 2017). We therefore introduce another neural network f_{ϕ_d} to parameterized f^{-1} in order to approximate the injective property by $f^{-1}(f(\mathbf{h})) = \mathbf{h}, \forall \mathbf{h} \in \mathbf{H}$. Similar to (Li et al. 2017), we obtain the final distribution divergence loss term $\mathcal{L}_d(W, f_{\phi_e}, f_{\phi_d})$ as

$$\begin{aligned}\mathcal{L}_d(W, f_{\phi_e}, f_{\phi_d}) = \zeta &\left\| \frac{1}{N_S} f_{\phi_e}(\mathbf{H}_S)^\top \mathbf{1}_S - \frac{1}{N_T} f_{\phi_e}(\mathbf{H}_T)^\top \mathbf{1}_T \right\|_F^2\\ &- \eta \left\| \mathbf{H} - f_{\phi_d}(f_{\phi_e}(\mathbf{H})) \right\|_F^2 . \quad (9)\end{aligned}$$

Label Propagation for Semi-supervised Learning

To make matrix completion more effective, existing approaches (Xiao and Guo 2013; Zhou et al. 2016) assume some cross-domain instance correspondences are given in advance to build a bridge between domains. However, as discussed this assumption is difficult to satisfy in practice. Therefore, instead, we assume only a few target-domain labeled instances are given. Our goal is to introduce a classification network denoted by C based on the latent representation $\mathbf{H}$ to bridge the gap between domains via labels. In particular, the classification network is a full-connected network with the Softmax function,

$$p(y_i = k | C) = \frac{\exp(C_k(\mathbf{H}_i))}{\sum_{k'} \exp(C_{k'}(\mathbf{H}_i))}, \quad (10)$$

where $y_i \in \mathbf{Y}_S$ if the sample is from source domain and $y_i \in \mathbf{Y}_{T_l}$ for labeled target domain, $\mathbf{H}_i$ denotes the i-th instance from $\mathbf{H}$, and $C_k(\cdot)$ denotes the output on the k-th class generated by Softmax.

As we only have a few target-domain labeled instances, we employ graph-based semi-supervised learning to propagate label information. In particular, a Laplacian matrix (He et al. 2005) given as $\mathbf{L}_g = \begin{pmatrix} \mathbf{L}_{g_S} & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_{g_T} \end{pmatrix}$, is introduced by adding tr$(\mathbf{H}^\top \mathbf{L}_g \mathbf{H})$ as the manifold regularization term, where $\mathbf{L}_{g_S}$ and $\mathbf{L}_{g_T}$ are the Laplacian matrices on $\mathbf{X}_S$ and $\mathbf{X}_T$, respectively. The loss on the semi-supervised classifier is then defined as

$$\begin{aligned}\mathcal{L}_{cg}(W, C) = &-\frac{1}{N_S} \sum_{i=1}^{N_S} \mathbf{1}(y_i = k) \log p(y_i = k | W, C)\\ &-\frac{1}{N_{T_l}} \sum_{j=N_S+1}^{N_S+N_{T_l}} \mathbf{1}(y_j = k) \log p(y_j = k | W, C)\\ &+\gamma \text{tr}(\mathbf{H}^\top \mathbf{L}_g \mathbf{H}), \quad (11)\end{aligned}$$

where $\mathbf{1}$ is an indicator function which returns 1 when argument is true, otherwise 0. In this work, we use 5-NN with 0/1 weights to construct the Laplacian matrices.

Model Optimization

The final objective of our proposed Deep-MCA method is formulated as

$$\mathcal{L} = \mathcal{L}_{mc}(W, V) + \mathcal{L}_{cg}(W, C) + \mathcal{L}_d(W, f_{\phi_e}, f_{\phi_d}) \quad (12)$$

where the first term defined in (3) serves as a matrix completion component, the second term defined in (11) is the semi-supervised classifier on the learned feature representation, and the third term defined in (9) is a distribution divergence component which captures the difference in distribution between domains.

Algorithm 1 illustrates our training algorithm based on the adaptive moment estimation (ADAM) algorithm (Kingma and Ba 2014). We first initialize the network parameters by Gaussian distribution with $\mathcal{N} \sim (0, 0.02)$. The source and target features are augmented by zero padding as discribed in (1). Similar to GAN, the network can be trained in a minimax manner. During maximization step, we adopt weight clipping to satisfy the Lipschitz constraint condition. Finally, we follow the generative adversarial networks (GAN) (Goodfellow et al. 2014) by training f_{ϕ_d} for few iterations followed by training f_{ϕ_e}, which is the parameter B in Algorithm 1 where $B = 4$.

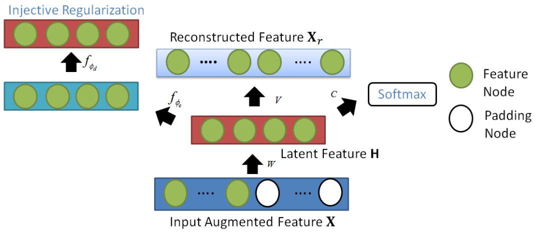


Figure 1: Proposed framework for heterogeneous transfer learning. We first augment the data through zero padding. Then we aim to learn feature representation which capture shareable information across heterogeneous domain through our proposed method. In this figure, we use the augmented feature from source domain for visualization.

Algorithm 1 Deep-MCA

```
Input: $\mathbf{X}_S$, $\mathbf{X}_T$, $\mathbf{Y}_S$ and $\mathbf{Y}_{T_l}$ initialized parameters $W$, $V$, $C$, $f_{\phi_e}$ and $f_{\phi_d}$.
Output: Learned parameters $W^*$, $V^*$, $C^*$, $f^*_{\phi_e}$ and $f^*_{\phi_d}$.
1: Augment the source and target domain as $\mathbf{X}$.
while Stopping criterion is not met do
  for t=1:B do
    2: Compute the gradient of (9) w.r.t. $f_{\phi_e}$ and $f_{\phi_d}$ based on augmented data $\mathbf{X}$.
    3: Take a gradient step to update $f_{\phi_e}$ and $f_{\phi_d}$ to maximize the objective (9).
    4: Conduct weight clipping based on $f_{\phi_e}$ and $f_{\phi_d}$.
  end for
  5: Compute the SVD of $\mathbf{X}_r$ to obtain $\mathbf{A}$ and $\mathbf{B}$.
  6: Compute the gradient of (12) w.r.t. $W$, $V$, $C$ on $\mathbf{X}$ with $\mathbf{A}$ and $\mathbf{B}$, respectively.
  7: Take a gradient step to update $W$, $V$, $C$ to minimize the objective (12), alternatingly.
end while
```

Experiment

In the section, we conduct experiments on two different vision tasks, cross-domain object recognition and text-to-image classification with heterogeneous features to verify the effectiveness of our proposed method compared with some state-of-the-art HTL baselines. For the experimental setup, we assume that we only have one source domain and one target domain. We use a limited number of target-domain labeled instance and some target-domain unlabeled instances for training in our HTL problems. We report the averaged results over 10 random splits on the target-domain unlabeled instances for both cross-domain object recognition and text-to-image classification tasks.

Experimental Setup

Object Recognition We follow the setting (Tsai, Yeh, and Frank Wang 2016; Yan et al. 2018) by using images collected from Amazon dataset (A), DSLR dataset (D), webcam dataset (W) and Caltech-256 dataset (C), where ten common categories in all these datasets are used for conduct experiments. For the HTL setting, considering that it is difficult to train a robust deep neural network with only few data, we then consider deep feature representation for source domain and hand-crafted feature for target domain. In particular, we use instances of the DeCAF$_6$ features (Jia et al. 2014) with dimension 4,096 to construct a source domain and instances of the bag-of-words based SURF features (Gong et al. 2012) with dimension 800 to construct a target domain. We randomly select 20 labeled instances per category from the source domain and 3 labeled instances per category from the target domain for training, and the remaining instances in target domain are used for testing. For the source domains constructed on DSLR, we randomly choose 5 source instances per category as suggested in (Yan et al. 2018) since the number of instances in D is much smaller.

Figure 2: Example images of Amazon, DSLR, Webcam and Caltech-256 datasets.

Text-to-Image Classification We apply NUS-WIDE (Chua et al. 2009) and ImageNet (Deng et al. 2009) as the datasets for text-to-image classification task. NUS-WIDE contains 269,648 images with tag information collected from Flickr and ImageNet wis with 5247 synsets and 3.2 million images in total. We follow (Chen et al. 2016) by extracting the 64-dimensional feature representation from a five-layer neural network as the feature for tag data. We consider DeCAF$_6$ feature for image data. Eight overlapping categories from these two datasets are considered, which include airplane, birds, buildings, cars, dog, fish, flowers, horses. All tag data are used as source domain. Three images per category from ImageNet dataset are randomly selected as labeled data, and another 100 images for prediction.

Network Architecture and Parameter Setting In our proposed method, we consider to adopt a two-layer architechture for encoder (W), decoder (V), feature mapping network (f_{ϕ_e}), injection network (f_{ϕ_d}) as well as the classification network (C) for all the task. The details of the architechture are summarized in Table 1. The learning rate of our algorithm is set as 0.0001 for all experiments. Regarding the parameter setting for objective, one can use a tuning strategy by training on source domain and testing on labeled target domain. In our experiment, we fix the parameters for all experiments for simplicity. In particular, we set $\lambda = 0.001, \zeta = 10$ and all others as 1. We set the dimension of hidden layer as 100 for fair comparison with other baseline methods.

Table 1: Details of our proposed network architechture. D indicates the dimension of input augmented feature.

W	FC-(D,D), ReLU
	FC-(D,100)
V	FC-(100,100), ReLU
	FC-(100,D)
f_{ϕ_e}, f_{ϕ_d}	FC-(100,100), ReLU
	FC-(100,100)
C	FC-(100,100), ReLU
	FC-(100,$\#class$)

Baseline Methods We compare our proposed method with the following baselines: (SVM_t) that simply employs labeled data from target domain to train a model, and some state-of-the-art HTL methods, including SHFA (Li et al. 2014), DMMC (Zhou et al. 2016), CDLS (Tsai, Yeh, and Frank Wang 2016), SGW (Yan et al. 2018) and the deep learning based TNT (Chen et al. 2016). We follow the previous works to search the parameters in spaces recommended by the original papers and report their best results.

Experimental Results

For object recognition, we report experimental results by constructing HTL tasks on the same dataset and cross datasets. The results are shown in Table 2. From the results, we can see that the SVM_t baseline method performs poorly for all different object recognition tasks. This is reasonable because there are only a few labeled in the target domain for training. By exploiting the labeled training data of heterogeneous features from the source domain, we notice that the learning performance in terms of prediction accuracy improvements. Among all the HTL methods, our proposed method can achieve the best performance among eight out of nine cases, which indicates the effectiveness of kernel adaptation compared with linear adaptation. Another possible explanation is that, while other HTL methods focus on either MMD alignment (Tsai, Yeh, and Frank Wang 2016) or classifier exploiting (Li et al. 2014), we focus on both as regularization. Although TNT method (Chen et al. 2016) is based on neural network, it is lack of distribution alignment term thus the shareable cross-domain information may not be properly extrated. SGW was the most recent proposed method for HTL problem. However, only linear mapping was learned based on the data, which may not model the distribution distance accurately since linear kernel is not characteristic. We also investigate linear matrix completion, DMMC, without correspondences. We observe that DMMC performs poorly when there are no cross-domain correspondences, which is consistent with the results reported in (Zhou et al. 2016). This suggests that to use matrix completion based method, when there are no correspondences, utilizing target-domain label information is crucial for effective knowledge transfer.

Table 2: Classification accuracy (in %) for heterogeneous object recognition (DeCAF$_6$ for source domain, SURF for target domain), where A, C and W denote Amazon, Caltech and Webcam respectively.

S/T	SVM_t	SHFA	DMMC	CDLS	SGW	TNT	Deep-MCA
A/A	42.1	44.9	34.9	43.7	46.4	43.1	**47.4**
C/C	30.1	31.1	25.7	32.3	34.1	30.7	**34.7**
W/W	57.3	62.7	50.3	63.1	63.5	60.0	**66.4**

S/T	SVM_t	SHFA	DMMC	CDLS	SGW	TNT	Deep-MCA
A/D	57.9	58.1	49.1	59.4	59.7	56.5	**60.1**
W/D		62.7	48.6	60.2	59.4	57.2	**63.0**
D/A	42.1	42.7	32.7	43.5	44.1	40.5	**44.7**
W/A		44.2	35.1	45.1	**50.8**	41.8	48.6
A/W	57.3	62.5	44.2	63.5	63.9	59.9	**66.8**
D/W		60.8	46.4	63.7	64.4	60.1	**66.4**

For text-to-image task, we can also observe that our proposed method can outperforms the other baseline methods. DMMC without correspondence achieves a relatively poor performance which is consistent with the results on object recognition. Another interesting result we find is that the TNT method can achieve a relatively better performance compared with the results obtained for object recognitiont task. One possible explanation is that tree based neural network can better deal with inbalanced heterogeneous feature representation scenario better compared with other baseline methods (64 dim for TAG feature extracted by a five-layer neural network compared with the 4096 dim for DeCAF$_6$).

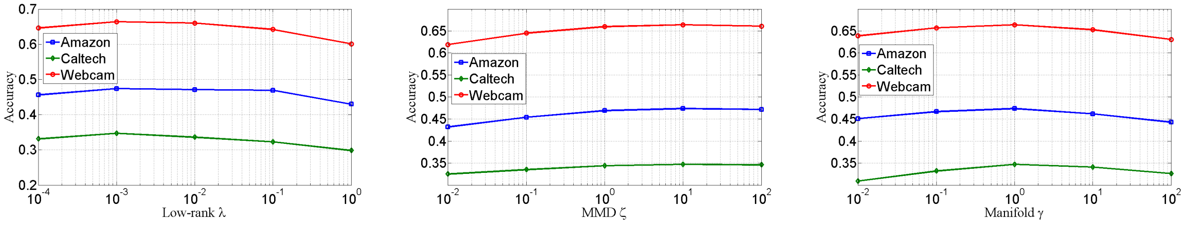
Table 3: Classification accuracy (in %) for adapting text (NUS-WIDE) to image data (ImageNet).

S/T	SVM_t	SHFA	DMMC	CDLS	SGW	TNT	Ours
Tag/Image	67.5	67.3	60.7	69.0	68.3	70.4	**71.7**

Impact on Different Components

In this section, we further conduct experiments on object recognition to understand the impact of different components of our proposed algorithm with object classification task by considering the same dataset setting from Amazon, DSLR and Webcam dataset with DeCAF$_6$ for source domain and SURF as target domain. Experimental results are shown in Table 4. "No Low-rank" means that we remove the low-rank constraint $\|\mathbf{X}_r\|_*$ for the final objective. "No Injective" means that we remove the subnetwork f_{ϕ_d} and the term $\|\mathbf{H} - f_{\phi_d}(f_{\phi_e}(\mathbf{H}))\|_F^2$. "No MMD" means that we remove the MMD regularization term as well as the "injective" regularization term from the objective (as injective regularization is meaningless without MMD) and "No Manifold" means that we remove the manifold constraint on the latent feature representation.

From the table, we observe that removing the "low-rank constraint", "injective constraint", MMD regularization



(a) (b) (c)

Figure 3: Parameter sensitivity analysis. Accuracy under varying (a) λ, (b) ζ, (c) γ

or the graph regularization component leads to poor performance. This verifies the effectiveness of our proposed framework: 1) Considering "low-rank" constraint benefits latent feature representation learning regarding augmented input 2) Using MMD-based regularization is helpful to reduce difference between the seen source domains, and thus able to learn invariant features across source domains. 3) Imposing "injective" regularization helps to learn a better representation for distribution matching. 4) Incorporating manifold regularization helps to learn a more robust classifier based on target domain. Moreover, compared with the SVM_t baseline, it is observed that we still can achieve better performance under all scenarios.

Table 4: Impact of different components on performance, where A, C and W denote Amazon, Caltech and Webcam respectively.

S/T	No Low-rank	No Injective	No MMD	No Manifold	Ours
A/A	43.7	47.0	43.0	44.6	**47.4**
C/C	30.2	34.5	32.0	30.4	**34.7**
W/W	62.3	63.1	58.4	62.7	**66.4**

We further observe that, "low-rank", "MMD" and "Manifold" constraints play more important roles in the final results. Therefore, we also aim to analyze the sensitivity of the parameters λ, ζ and γ which control "low-rank", "MMD" and "Manifold" regularization. Experimental results are shown in Figure 3. From the figure, we notice that the parameters we choose are reasonable based on all scenarios. For λ, we can achieve good performances when the parameter is relatively small. For ζ, we observe that the performances are stable when $\zeta \geq 1$. However, for γ, we only achieve the best performance when $\gamma = 1$.

Convergence Analysis

Finally, we are interested in the convergence property of our proposed method. We therefore conduct experiment based on object classification task by considering the same dataset setting from Amazon, DSLR and Webcam dataset with DeCAF$_6$ for source domain and SURF as target domain. We report the classification accuracy respect to the iteration number. The results are shown in Figure 4. Based on the results, we observe that our method can converge after 20 iterations for all cases despite the fact that the neural network is nonlinear and our objective is non-convex.

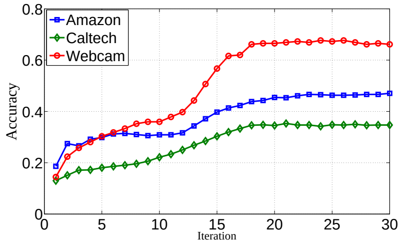


Figure 4: Convergence analysis of our proposed algorithm on Amazon, Caltech and Webcam dataset.

Conclusion

In this paper, we presented a new framework Deep-MCA for HTL. The main idea is to develop an end-to-end solution based on a deep architecture with adversarial kernel training to perform nonlinear matrix factorization for learning heterogeneous features for HTL problems. We conduct experiments on two vision tasks to demonstrate that Deep-MC generally outperforms other state-of-the-art baselines. We have also conduct experiments to test the impact of different components of Deep-MC as well as empirical convergence analysis on Deep-MCA.

Acknowledgement

This research was carried out at the Rapid-Rich Object Search (ROSE) Lab at the Nanyang Technological University, Singapore. This research is supported by the National Research Foundation, Prime Minister's Office, Singapore, under the NRF-NSFC grant NRF2016NRF-NSFC001-098. Sinno J. Pan thanks the support from NTU Singapore Nanyang Assistant Professorship (NAP) grant M4081532.020.

References

Bernstein, D. S. 2005. *Matrix mathematics: Theory, facts, and formulas with application to linear systems theory*, volume 41. Princeton university press.

Carey, P. 2018. *Data protection: a practical guide to UK and EU law*. Oxford University Press, Inc.

Chen, W.-Y.; Hsu, T.-M. H.; Tsai, Y.-H. H.; Wang, Y.-C. F.; and Chen, M.-S. 2016. Transfer neural trees for heterogeneous domain adaptation. In *ECCV*.

Chua, T.-S.; Tang, J.; Hong, R.; Li, H.; Luo, Z.; and Zheng, Y. 2009. Nus-wide: a real-world web image database from national university of singapore. In *CIVR*, 48.

Dai, W.; Chen, Y.; Xue, G.-R.; Yang, Q.; and Yu, Y. 2008. Translated learning: Transfer learning across different feature spaces. In *NIPS*.

DauméIII, H. 2007. Frustratingly easy domain adaptation. In *ACL*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*.

Duan, L.; Xu, D.; and Tsang, I. W. 2012. Learning with augmented features for heterogeneous domain adaptation. In *ICML*.

Gong, B.; Shi, Y.; Sha, F.; and Grauman, K. 2012. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NIPS*.

Gretton, A.; Borgwardt, K. M.; Rasch, M.; Schölkopf, B.; and Smola, A. J. 2006. A kernel method for the two-sample-problem. In *NIPS*.

Gretton, A.; Borgwardt, K. M.; Rasch, M. J.; Schölkopf, B.; and Smola, A. 2012. A kernel two-sample test. *Journal of Machine Learning Research* 13(Mar):723–773.

Harel, M., and Mannor, S. 2011. Learning from multiple outlooks. In *ICML*.

He, X.; Cai, D.; Yan, S.; and Zhang, H.-J. 2005. Neighborhood preserving embedding. In *ICCV*.

Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; and Darrell, T. 2014. Caffe: Convolutional architecture for fast feature embedding. In *MM*.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kulis, B.; Saenko, K.; and Darrell, T. 2011. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *CVPR*.

Li, W.; Duan, L.; Xu, D.; and Tsang, I. W. 2014. Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation. *IEEE transactions on pattern analysis and machine intelligence* 36(6):1134–1148.

Li, C.-L.; Chang, W.-C.; Cheng, Y.; Yang, Y.; and Póczos, B. 2017. Mmd gan: Towards deeper understanding of moment matching network. In *NIPS*.

Li, Y.; Swersky, K.; and Zemel, R. 2015. Generative moment matching networks. In *ICML*.

Long, M.; Cao, Y.; Wang, J.; and Jordan, M. 2015. Learning transferable features with deep adaptation networks. In *ICML*.

Luo, Y.; Wen, Y.; and Tao, D. 2017. Heterogeneous multi-task metric learning across multiple domains. *IEEE transactions on neural networks and learning systems*.

Muandet, K.; Fukumizu, K.; Sriperumbudur, B.; Schölkopf, B.; et al. 2017. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning* 10(1-2):1–141.

Pan, S. J., and Yang, Q. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22(10):1345–1359.

Pan, S. J.; Tsang, I. W.; Kwok, J. T.; and Yang, Q. 2011. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks* 22(2):199–210.

Prettenhofer, P., and Stein, B. 2010. Cross-language text classification using structural correspondence learning. In *ACL*.

Shi, X.; Liu, Q.; Fan, W.; Yu, P. S.; and Zhu, R. 2010. Transfer learning on heterogenous feature spaces via spectral transformation. In *ICDM*.

Sriperumbudur, B. K.; Fukumizu, K.; Gretton, A.; Lanckriet, G. R. G.; and Schölkopf, B. 2009. Kernel choice and classifiability for RKHS embeddings of probability distributions. In *NIPS*.

Tsai, H. Y.-H.; Yeh, Y.-R.; and Frank Wang, Y.-C. 2016. Learning cross-domain landmarks for heterogeneous domain adaptation. In *CVPR*.

Wang, C., and Mahadevan, S. 2011. Heterogeneous domain adaptation using manifold alignment. In *IJCAI*.

Xiao, M., and Guo, Y. 2013. A novel two-step method for cross language representation learning. In *NIPS*.

Xiao, M., and Guo, Y. 2015. Semi-supervised subspace co-projection for multi-class heterogeneous domain adaptation. In *ECML/PKDD*.

Yan, Y.; Li, W.; Wu, H.; Min, H.; Tan, M.; and Wu, Q. 2018. Semi-supervised optimal transport for heterogeneous domain adaptation. In *IJCAI*.

Yang, L.; Jing, L.; Yu, J.; and Ng, M. K. 2016. Learning transferred weights from co-occurrence data for heterogeneous transfer learning. *IEEE transactions on neural networks and learning systems* 27(11):2187–2200.

Zhou, J. T.; Tsang, I. W.; Pan, S. J.; and Tan, M. 2014. Heterogeneous domain adaptation for multiple classes. In *AISTATS*.

Zhou, J. T.; Pan, S. J.; Tsang, I. W.; and Ho, S.-S. 2016. Transfer learning for cross-language text categorization through active correspondences construction. In *AAAI*.

Zhuang, F.; Luo, P.; Shen, Z.; He, Q.; Xiong, Y.; Shi, Z.; and Xiong, H. 2012. Mining distinction and commonality across multiple domains using generative model for text classification. *IEEE Transactions on Knowledge and Data Engineering* 24(11):2025–2039.