Proceedings of Machine Learning Research 95:375-390, 2018 ACML 2018

Unsupervised Heterogeneous Domain Adaptation with Sparse Feature Transformation

Chen Shen CHEN.SHEN@TEMPLE.EDU
Computer and Information Sciences, Temple University, USA

Yuhong Guo YUHONG.GUO@CARLETON.CA
School of Computer Science, Carleton University, Canada

Editors: Jun Zhu and Ichiro Takeuchi

Abstract

Heterogeneous domain adaptation (HDA), which aims to adapt information across domains with different input feature spaces, has attracted a lot of attention recently. However, many existing HDA approaches rely on labeled data in the target domain, which is either scarce or even absent in many tasks. In this paper, we propose a novel unsupervised heterogeneous domain adaptation approach to bridge the representation gap between the source and target domains. The proposed method learns a sparse feature transformation function based on the data in both the source and target domains and a small number of existing parallel instances. The learning problem is formulated as a sparsity regularized optimization problem and an ADMM algorithm is developed to solve it. We conduct experiments on several real-world domain adaptation datasets and the experimental results validate the advantages of the proposed method over existing unsupervised heterogeneous domain adaptation approaches.

1. Introduction

For supervised learning, labels of instances play a key role for classification model training with majority of approaches. However, for many real-world problems, it is difficult or expensive to collect a sufficient amount of labeled data. Domain adaptation (DA), which aims to transfer label knowledge from a source domain to a target domain, hence has attracted a lot of attention recently in computer vision (Niu et al., 2015) and many other fields (Xiao and Guo, 2013).

Much existing DA work assumes a homogeneous cross-domain feature space, which hinders the applicability of domain adaptation in many real world scenarios where a source domain with different input feature space can be readily available for the prediction task in a given target domain. For example, for image classification in a target domain, a source domain can have labeled images across the same set of classes but with different resolutions, which leads to different dimensions of codebook and hence feature space; labeled text data that describe a set of object classes can be a natural heterogeneous source domain for images from the same set of object classes. Heterogeneous domain adaptation (HDA) techniques, which tackle domain adaptation problems with different cross-domain feature spaces, hence are in high demand.

The major challenge for HDA lies in bridging the disjoint cross-domain features spaces. Supervised HDA methods tackle this problem by building cross-domain connections based on the existence of a small set of labeled instances in the target domain. They utilize the labeled data to learn a feature mapping function from one domain to another (Kulis et al., 2011; Hoffman et al., 2013; Zhou et al., 2014), or map the feature spaces of both domains into a common subspace (Duan et al., 2012; Sukhija et al., 2016). The performance of such methods however is highly restricted by the scarcity of the labeled instances in the target domain. A number of semi-supervised HDA approaches hence further exploit the unlabeled instances in the target domain to alleviate this restriction and improve the learning of feature transformation or classifiers (Tsai et al., 2016; Wu et al., 2013; Xiao and Guo, 2015; Yao et al., 2015). Some semi-supervised HDA methods even utilize parallel unlabeled instances to learn cross-domain representations (Platt et al., 2010; Xiao and Guo, 2013). These methods however still depend on the existence of labeled target instances. A few unsupervised HDA approaches overcome this dependence limitation on labeled target data by learning a common latent correlation subspace based only on parallel instances (Hardoon et al., 2004; Yeh et al., 2014). However, these existing methods typically require a large number of parallel instances to achieve reasonable performance.

In this paper, we propose a novel feature transformation method to tackle unsupervised heterogeneous domain adaptation by assuming the existence of a small number of parallel instances. The method uses a linear function to transform the source domain features into the target domain features to match the parallel instances, while minimizing the cross domain distribution divergence by aligning the transformed source domain covariance matrix with the target domain covariance matrix. Under the assumption that only a small number of source domain features are needed to induce a target domain feature, we further exploit two types of sparsity inducing norms to regularize the linear transformation model. We formulate this unsupervised HDA problem as a minimization problem over a sparsity regularized quartic function and develop an alternating direction method of multipliers (ADMM) to solve it efficiently. Experiments are conducted on a few heterogeneous domain adaptation datasets for image classification. The experimental results show that the proposed method outperforms existing unsupervised heterogeneous domain adaptation approaches and achieves promising results even when there are only a very small number of parallel instances.

2. Related Work

In this section, we briefly review the related groups of DA approaches, including unsupervised domain adaptation methods, (semi-)supervised HDA methods, and unsupervised HDA methods.

2.1. Unsupervised Domain Adaptation

Unsupervised DA aims to exploit the labeled data in a source domain to assist a target domain that has no labeled instances at all. Many unsupervised DA techniques have been developed in computer vision field. The work in (Gong et al., 2012) addresses cross-domain object recognition problems. It constructs and computes the geodesic flow kernel of infinite subspaces between the source and target domains to overcome the domain shift problem.

Fernando et al. (2013) propose a visual domain adaptation approach for image classification that uses a linear mapping to align subspaces across domains. Long et al. (2014) introduce a transfer joint matching model that considers both feature matching and instance reweighting for cross-domain digit classification and object recognition. Recently, Sun et al. (2015) propose a simple but effective DA approach based on correlation alignment, which learns a feature transformation by aligning the covariance matrices of the source and target domains. The approach has been applied on cross-domain object recognition problems. Bousmalis et al. (2016) propose to combine feature extraction with domain adaptation. It learns and coordinates both private and shared subspaces with a deep learning model. Recently Cao et al. (2018) propose to extract invariant feature representations and estimate unbiased instance weights for minimizing the cross-domain distribution discrepancy. These methods though share some similarities with our proposed approach in bridging the cross-domain feature gaps with feature transformation and alignment, they are limited to homogeneous domain adaptation problems and do not handle disjoint cross-domain feature spaces.

2.2. (Semi-) Supervised HDA

Most existing HDA methods require the availability of a small number of labeled instances in the target domain. Depending on whether unlabeled target domain instances are utilized, these methods can be divided into two groups: supervised and semi-supervised methods.

Supervised HDA methods exploit the labeled target domain instances in addition to the source domain data to bridge the feature representation gap. For example, Kulis et al. (2011) learn an asymmetric and nonlinear feature transformation matrix for cross-domain image classification by solving an ARC-t problem with the help of the labeled data. Duan et al. (2012) propose a heterogeneous feature augmentation method that transforms the data in both domains into a common subspace and then augments the projected data with original features. Hoffman et al. (2013) propose a max-margin domain transformation method, which combines the learning of an asymmetric cross-domain transformation function and the learning the classification parameters in max-margin framework. Zhou et al. (2014) construct a sparse and class-invariant feature transformation matrix to map the weight vector of classifiers. Recently, Sukhija et al. (2016) propose to use the shared label distributions across domains as pivots for learning a sparse feature transformation in a supervised HDA setting.

Semi-supervised HDA methods further exploit unlabeled target domain instances to help the adaptation. Wu et al. (2013) propose to learn a discriminative common feature space for cross-view action recognition by minimizing canonical correlations of interclass instances and maximizing intraclass instances. Xiao and Guo (2015) develop a semi-supervised kernel matching framework that simultaneously maps the target domain instance into the source domain instances and learns a prediction function on the labeled source instances. Tsai et al. (2016) propose a cross-domain landmark selection method to learn representative landmarks from cross-domain data. Yao et al. (2015) propose a semi-supervised domain adaptation method which learns a subspace that reduces the underlying cross-domain difference and preserves the local structures of domains. More recently, Yan et al. (2017) propose to learn a discriminative correlation subspace and the target domain classifier simultaneously with a

unified objective, which achieves state-of-the-art results. The *requirement* of labeled target instances however remains to be a *limitation* for such (semi-)supervised methods.

2.3. Unsupervised HDA

Unsupervised HDA methods do not require any labeled data from the target domain and mainly use unlabeled instances to bridge the heterogeneous cross-domain feature spaces. Although a lot of approaches have been developed for unsupervised domain adaptation (Wei et al., 2016), unsupervised heterogeneous domain adaptation has received far little attention due to its difficulty. Hardoon et al. (2004) propose a canonical correlation analysis (CCA) method which learns a common semantic representation between the text and image domains. This method can naturally bridge the representation gap across heterogeneous domains with parallel data. Recently, Yeh et al. (2014) propose a novel unsupervised HDA framework that exploits unlabeled cross-domain data pairs to derive a feature transformation model for cross-domain recognition. Similar to CCA, it utilizes the derived correlation subspace as a joint representation for associating data across domains, and advances reduced kernel techniques for kernel CCA (KCCA) for producing nonlinear correlation subspace. It also incorporates the domain adaptation ability into classifier design by employing a SVM with a correlation regularizer. This method however can only exploit the unlabeled cross-domain data pairs (i.e., parallel instances) which can be limited in many domains, while ignoring the large set of unlabeled nonparallel instances in each domain. Our proposed unsupervised HDA approach in this paper can overcome such a drawback by exploiting all existing data in both domains in an unsupervised manner.

3. Unsupervised HDA with Sparse Feature Transformation

In this section, we present a novel sparse feature transformation method for unsupervised heterogeneous domain adaptation (SFT-HDA). It induces a feature transformation by aligning the distributions of the transformed source features and target features in an unsupervised manner.

3.1. Problem Setting

We assume D_s and D_t are the source and target domains respectively with different feature spaces. There are n_s labeled instances (X_s, Y_s) in the source domain D_s, where $X_s \in \mathbb{R}^{n_s \times d_s}$ is the feature matrix and $Y_s \in \{0,1\}^{n_s \times L}$ is the label matrix over L classes with a single 1 indicating its class label in each row. In the target domain D_t, we only have n_t unlabeled instances $X_t \in \mathbb{R}^{n_t \times d_t}$ and need to predict their unknown label matrix $Y_t \in \{0,1\}^{n_t \times L}$ in the same label space as in the source domain. Unlike vast HDA approaches that utilize the labeled instances in the target domain to adapt the heterogeneous domains, we consider a harder unsupervised scenario where there is no labeled data in the target domain. Instead, we assume there are a small number of n_p unlabeled parallel instances, i.e., (X_s^0, X_t^0) with $X_s^0 \in \mathbb{R}^{n_p \times d_s}$ and $X_t^0 \in \mathbb{R}^{n_p \times d_t}$. The unlabeled parallel instances have feature representations in both the source and target domain feature spaces to build cross-domain connections. Compared with the expensive labeled target domain instances, the acquisition of a small number of unlabeled parallel instances can be more convenient – they

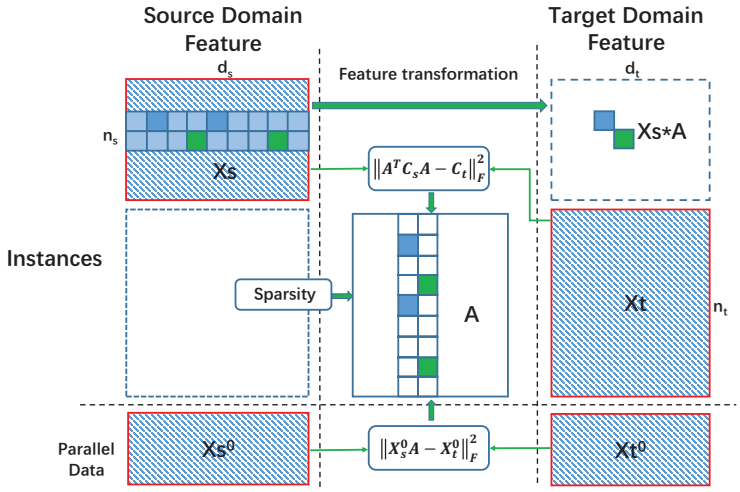

Figure 1: Unsupervised sparse feature transformation. The input data are indicated with blue stripes and red edges.

can be readily available in many applications; e.g., images taken by two cameras on the same set of objects.

Under this unsupervised domain adaptation setting, our proposed feature transformation framework can be demonstrated in Figure 1. We will present the details below.

3.2. Feature Transformation Model for HDA

To exploit the information in the source domain for our target prediction task, the major challenge is to bridge the heterogeneous cross-domain feature spaces. Although projecting data from both domains into a common subspace has been a typically technique adopted in the literature for DA tasks, such a third space induces transformation loss for both source and target domains. Hence we propose to directly transfer the source domain features into the target domain feature space without seeking a middle ground representation. In particular, we consider a linear transformation function, $f : \mathcal{X}_s \rightarrow \mathcal{X}_t$, that maps the source domain features into the target domain features via a transformation matrix $A \in \mathbb{R}^{d_s \times d_t}$. On the parallel data, we expect the transformed data $f(X_s^0) = X_s^0 A$ can be a good approximation to its counterpart, X_t^0, in the target domain. By minimizing the squared approximation loss, this leads to the following transformation learning problem:

$$\min_{A} \quad \left\| X_s^0 A - X_t^0 \right\|_F^2 \tag{1}$$

where $\|\cdot\|_F$ denotes the Frobenius matrix norm. This learning formulation however entirely relies on the parallel instances, while ignoring the large amount of non-parallel data in the two domains. When the number of parallel instances is small, this learning mechanism

can hardly induce well generalizable transformation functions – there can be distribution divergence between the transformed source feature space and the original target domain feature space.

To bridge the cross-domain divergence, we propose to align the transformed feature distribution with the original target domain feature distribution by adopting a second moment matching strategy (Sun et al., 2015). Specifically, we minimize the distance between the second-order statistics, i.e., covariance, of the transformed source features and target features and formulate the distribution alignment of HDA as:

$$\min_{A} \quad \left\| A^\top C_s A - C_t \right\|_F^2 \tag{2}$$

where the source feature covariance matrix $C_s \in \mathbb{R}^{d_s \times d_s}$ is computed from the non-parallel source domain data X_s, and the target feature covariance matrix $C_t \in \mathbb{R}^{d_t \times d_t}$ is computed from the non-parallel target domain data X_t. The covariance matrix in the transformed source feature space is computed with the linear transformation matrix A as $A^\top C_s A$. With the large amount of non-parallel data in the source and target domains, we expect the empirical covariance matrix matching will reduce the cross-domain feature distribution divergence and facilitate cross-domain information transfer.

To ensure both a meaningful feature transformation and a minimal cross-domain distribution divergence, we finally combine the loss functions in both Eq.(1) and Eq.(2) and formulate a heterogeneous feature transformation model as below by exploiting both the non-parallel data and the parallel-data in the two domains:

$$\min_{A} \quad \left\| A^\top C_s A - C_t \right\|_F^2 + \alpha \left\| X_s^0 A - X_t^0 \right\|_F^2 \tag{3}$$

where α is a trade-off parameter to balance the two losses.

3.3. Sparse Feature Transformation

The cross-domain feature transformation above can be interpreted as constructing each target domain feature as a linear combination of the source domain features. With a large number of source domain features, an unregularized full linear transformation however can easily cause overfitting, encode noise, or capture spontaneous cross-domain feature relations. We hence propose to enforce sparsity on our linear feature transformation model by employing the following form of mixed norm of sparsity regularizers (Kowalski et al., 2009):

$$\|A\|_{p,q} = \left[\sum_{j=1}^{d_t} \left[\sum_{i=1}^{d_s} |A_{i,j}|^p \right]^{q/p} \right]^{1/q} \tag{4}$$

With different (p, q) values, this mixed norm can result in different type of regularizers. This leads to the following sparse feature transformation model for HDA:

$$\min_{A} \ \frac{1}{2} \left\| A^\top C_s A - C_t \right\|_F^2 + \frac{\alpha}{2} \left\| X_s^0 A - X_t^0 \right\|_F^2 + \frac{\gamma}{q} \|A\|_{p,q}^q \tag{5}$$

where γ is a trade-off parameter for the sparsity regularizer. By integrating the three components – distribution alignment of the source and target domains, transformation error

minimization of the parallel data, and the sparsity regularization, we expect to learn a robust and generalizable feature transformation matrix A that can effectively bridge the cross-domain representation gap and facilitate information adaptation from the source domain to the target domain.

In this work, we consider two types of norms with (p=1,q=1) and (p=1,q=2) respectively. First, to filter noise and avoid spontaneous cross-domain feature correlations, we consider enforcing an overall sparsity regularization over the linear coefficients in the transformation matrix A. This can be achieved by using an entrywise ℓ_1 norm regularizer with p=1 and q=1:

$$\|A\|_{1,1} = \sum_{i=1}^{d_s}\sum_{j=1}^{d_t}|A_{i,j}| \tag{6}$$

Second, typically only a small fraction of the source domain features are related to a target domain feature. For instance, for domain adaptation across two types of image feature spaces, a target domain local feature that describes one part/position of image may only related to a small number of source domain features over several related parts/positions. With this motivation, we consider employing an individual sparsity inducing $\ell_{1,2}$ norm regularizer below:

$$\|A\|_{1,2} = \left[\sum_{j=1}^{d_t}\left[\sum_{i=1}^{d_s}|A_{i,j}|\right]^2\right]^{1/2} \tag{7}$$

This $\ell_{1,2}$ norm can enforce individual sparsity on each column of the transformation matrix A separately (Vatashsky and Crammer, 2013), and hence relate each target domain feature to only a few source domain features with the non-zero entries of the corresponding column of matrix A.

3.4. Learning Algorithm

The unsupervised sparse feature transformation learning problem formulated in Eq.(5) is a quartic program with a non-smooth sparsity regularizer. It is difficult to tackle due to the existence of the quartic term. We propose to solve it using an alternating direction method of multipliers (ADMM), which breaks a complex optimization problem into a few simpler subproblems, and solves the simpler subproblems separately (Boyd et al., 2011).

We first rewrite Eq.(5) into the following equivalent formulation by introducing an additional matrix B and an equality constraint

$$\min_{A,\,B}\ \frac{1}{2}\left\|A^\top C_s B - C_t\right\|_F^2 + \frac{\alpha}{2}\left\|X_s^0 B - X_t^0\right\|_F^2 + \frac{\gamma}{q}\|B\|_{p,q}^q \qquad \text{s.t. } A = B \tag{8}$$

The re-expressed problem is a quadratic minimization problem with sparsity regularizer in terms of A and B separately, subjecting to the equality constraint. The augmented Lagrangian function for this problem is

$$\begin{aligned} L_\rho(A,B,\Lambda) = &\frac{1}{2}\left\|A^\top C_s B - C_t\right\|_F^2 + \frac{\alpha}{2}\left\|X_s^0 B - X_t^0\right\|_F^2 + \\ &\frac{\gamma}{q}\|B\|_{p,q}^q + \mathrm{tr}(\Lambda^\top(A-B)) + \frac{\rho}{2}\|A-B\|_F^2 \end{aligned} \tag{9}$$

where Λ is the dual variable matrix associated with the equality constraint and ρ is the penalty parameter for the constraint. In each iteration of the ADMM algorithm, we then minimize this augmented Lagrangian over the primal variable matrices A and B separately, while updating the dual variable matrix. Specifically, in the $(k+1)$-th iteration, given the $(A^{(k)}, B^{(k)}, \Lambda^{(k)})$ from the previous iteration, we perform the following three steps.

(1) Minimization over B. Given the current fixed $A^{(k)}$ and $\Lambda^{(k)}$, B can be updated by minimizing the augmented Lagrangian:

$$B^{(k+1)} := \arg\min_{B} \; L_\rho(A^{(k)}, B, \Lambda^{(k)}) := \arg\min_{B} \; \ell(B) + \frac{\gamma}{q} \|B\|_{p,q}^q \tag{10}$$

where $\ell(B)$ is a smooth function such that

$$\ell(B) = \frac{1}{2} \left\| A^{(k)\top} C_s B - C_t \right\|_F^2 + \frac{\alpha}{2} \left\| X_s^0 B - X_t^0 \right\|_F^2 - \mathrm{tr}(\Lambda^{(k)\top} B) + \frac{\rho}{2} \left\| A^{(k)} - B \right\|_F^2$$

This minimization problem is a convex quadratic programming with a non-smooth sparsity regularizer. We solve it using a fast proximal gradient descent method with a quadratic convergence rate (Beck and Teboulle, 2009), which tackles Eq.(10) by solving a sequence of intermediate problems iteratively with proximity operators. In the t-th iteration, the intermediate problem at the current point $Q^{(t)}$ is in the following form:

$$\mathcal{P}_\eta(Q^{(t)}) = \arg\min_{B} \left\{ \frac{1}{2} \|B - \widehat{Q}^{(t)}\| + \frac{\gamma}{q\eta} \|B\|_{p,q}^q \right\} \tag{11}$$

where $\widehat{Q}^{(t)} = Q^{(t)} - \frac{1}{\eta} \nabla \ell(Q^{(t)})$ is derived from the gradient of $\ell(Q^{(t)})$ at the current point $Q^{(t)}$ and η is the Lipschitz constant of the gradient; we used $\eta = \sigma_{\max}\big(C_s^\top A^{(k)} A^{(k)\top} C_s + \alpha X_s^{0\top} X_s^0 + \rho I\big)$, where $\sigma_{\max}$ denotes the largest singular value of the given matrix. The nice property about this intermediate problem is that it allows us to exploit closed-form solutions for the proximity operator $\mathcal{P}_\eta(Q^{(t)})$ with either the ℓ_1-norm regularizer ($p = 1$ and $q = 1$) or the $\ell_{1,2}$-norm regularizer ($p = 1$ and $q = 2$). According to (Kowalski et al., 2009), we have a closed-form solution $\mathcal{P}_\eta(Q^{(t)}) = \tilde{Q}$ for the proximity operation such that for all (i, j),

$$\tilde{Q}_{i,j} = \begin{cases} \mathrm{sign}(\widehat{Q}_{i,j}^{(t)}) \left(|\widehat{Q}_{i,j}^{(t)}| - \frac{\gamma}{\eta} \right)_+ & \text{If } p = 1, q = 1 \text{ (ℓ_1-norm);} \\ \mathrm{sign}(\widehat{Q}_{i,j}^{(t)}) \left(|\widehat{Q}_{i,j}^{(t)}| - \frac{\gamma \sum_{r=1}^{m_j} \overrightarrow{Q}_{r,j}}{(\eta + \gamma m_j) \|\widehat{Q}_{:j}^{(t)}\|_2} \right)_+ & \text{If } p = 1, q = 2 \text{ ($\ell_{1,2}$-norm);} \end{cases} \tag{12}$$

where $(\cdot)_+ = \max(0, \cdot)$, $\overrightarrow{Q}_{:j}$ denotes a reordered j-th column $|\widehat{Q}_{:j}^{(t)}|$ with a descending order of the entries, and m_j is the number such that

$$\overrightarrow{Q}_{m_j+1,j} \le \frac{\gamma}{\eta} \sum_{r=1}^{m_j+1} \left(\overrightarrow{Q}_{r,j} - \overrightarrow{Q}_{m_j+1,j} \right), \quad \text{and} \quad \overrightarrow{Q}_{m_j,j} > \frac{\gamma}{\eta} \sum_{r=1}^{m_j} \left(\overrightarrow{Q}_{r,j} - \overrightarrow{Q}_{m_j,j} \right). \tag{13}$$

With the proximal operators, the proximal gradient descent method can easily handle the non-smooth mixed-norms and produce desired sparse solutions.

Algorithm 1 ADMM training algorithm

Input: Covariance matrices C_s and C_t; parallel data X_s^0 and X_t^0; α, γ, ρ, λ and ϵ
Initialize $A^{(1)} = B^{(1)}$ *with Eq.(17), set* $\Lambda^{(1)} = 0$ *and* $k = 1$
repeat
  $B^{(k+1)} := \arg\min_B \ L_\rho(A^{(k)}, B, \Lambda^{(k)})$; $A^{(k+1)} := \arg\min_A \ L_\rho(A, B^{(k+1)}, \Lambda^{(k)})$;
  $\Lambda^{(k+1)} := \Lambda^{(k)} + \rho(A^{(k+1)} - B^{(k+1)})$; Set $k = k + 1$.
until *convergence*;

(2) Minimization over A. With the current values of $B^{(k+1)}$ and $\Lambda^{(k)}$ being fixed, we update A by minimizing the augmented Lagrangian objective:

$$A^{(k+1)} := \arg\min_A \ L_\rho(A, B^{(k+1)}, \Lambda^{(k)}) \tag{14}$$

This is a simple quadratic minimization problem, which yields a closed-form solution:

$$A^{(k+1)} = \left(C_s B^{(k+1)} B^{(k+1)\top} C_s^\top + \rho I\right)^{-1} \left(C_s B^{(k+1)} C_t^\top - \Lambda^{(k)} + \rho B^{(k+1)}\right) \tag{15}$$

(3) Update of the dual variable matrix Λ. Following the standard ADMM method, we update the dual matrix Λ by

$$\Lambda^{(k+1)} := \Lambda^{(k)} + \rho(A^{(k+1)} - B^{(k+1)}) \tag{16}$$

The overall ADMM algorithm is given in Algorithm 1. Given the input data and parameter settings, the algorithm first initializes A, B and Λ before the iterative updates. We initialize A and B by simply setting them as the closed-form solution of the ℓ_2-norm regularized parallel data transformation:

$$\begin{aligned} A^{(1)} = B^{(1)} &= \arg\min_B \ \|X_s^0 B - X_t^0\|_F^2 + \lambda \|B\|_F^2 \\ &= (X_s^{0\top} X_s^0 + \lambda I_{d_s})^{-1} (X_s^{0^\top} X_t^0) \end{aligned} \tag{17}$$

where I_{d_s} denotes an identity matrix of size $d_s \times d_s$; λ is the regularization trade-off parameter and can be used to avoid the numerical problem of matrix inversion in the closed-form solution. We expect this initialization can provide a more informative starting point than random initialization. For the dual matrix Λ, we initialize it with all zero values.

The iterative three step updates of primal and dual variable matrices of the ADMM algorithm aim to minimize the augmented Lagrangian function, which will eventually push A to be close to the sparse B to recover the equality constraint $A = B$. Hence the algorithm eventually solves Eq.(8). It has been shown in (Hong et al., 2016) that even in the presence of non-convex objective, the ADMM algorithm is able to reach the set of stationary solutions for the linearly constrained problem in the form of Eq. (8). In our experiments, we adopt the following stopping criterion for the iterative updates: We stop the iteration loop whenever the distance between $A^{(k+1)}$ and $B^{(k+1)}$ is less than a very small positive constant ϵ or the maximum iterations is reached.

3.5. Cross-Domain Classification with Feature Transformation

After obtaining a cross-domain feature transformation matrix A, we can transform the labeled source domain data X_s into the target domain feature space as X_sA. Then we can train a multi-class classification model over the labeled data (X_sA, Y_s) and use it to predict the class categories of the unlabeled target domain instances X_t. In our experiments, we used one-vs-all SVM as the classification model.

4. Experiments

4.1. Datasets and Settings

We conducted experiments on three datasets, *UCI Multiple Features* (Asuncion and Newman, 2007), *Wikipedia* (Rasiwasia et al., 2010), and *Office-Caltech* (Gong et al., 2012). The *UCI multiple features dataset* contains 2000 images of 10 handwritten digits from '0' to '9', with 200 images per-class. For each image, there are six types of features. We dropped two types of features which have very small dimensions, and used the remaining four types of features: Fourier coefficient (fou), profile correlations (fac), Karhunen-Love coefficients (kar) and pixel averages (pix). By using one feature type as the source domain and another as the target domain, we formed 12 HDA tasks. For each task, 100 instances are pre-selected as the parallel instances. We then randomly sampled 20 instances per-category as the labeled source instances and the rest are used as the target instances. The *Wikipedia dataset* contains 2,866 multimedia documents over 29 categories, and each document consists of one paragraph of text and one related image. The images are represented as 128-dim bag-of-word SIFT features. Latent Dirichlet Allocation (LDA) is used to extract 10-dim text features from the document set. Following (Yeh et al., 2014), we considered five categories: art and architecture, biology, literature, sport, and warfare. In total 200 instances are used as parallel instances. Then 100 instances are selected for each category: 20 instances per-category are used as the labeled source instances and the rest instances are used as the target domain instances. The *Office-Caltech dataset* contains 10 classes of images from four domains: Amazon (A), DSLR (D), Webcam(W) and Caltech-256(C). We excluded the DSLR domain as there are very few instances per class. In addition to the 800-dimensional SURF features, we extracted 4096-dimensional CNN features (VGG19) (Simonyan and Zisserman, 2014). We select one domain from the three domains (A, C, W) as the source domain with one feature type (e.g., SURF), and select another domain as the target domain with a different feature type (e.g., VGG19). Hence in total we have 6 HDA tasks from SURF feature to VGG19 and another 6 HDA tasks from VGG19 feature to SURF. We selected 50 instances from both the source and target domains as the unlabeled parallel data. Then we randomly selected 20 instances (10 for Webcam) per-category as the labeled source domain instances and used the other source instances as unlabeled source instances. The instances in target domain are used as unlabeled target instances.

4.2. Comparison Methods

There is not much work on unsupervised HDA. We compared to two CCA based unsupervised HDA methods. Moreover, we also compared to a number of variants of the proposed

model by only considering parts of the three components in Eq.(5). All the comparison methods used in the experiments are listed below.

- **linear CCA:** It uses the linear canonical correlation analysis (Hardoon et al., 2004) to learn a common cross-domain representation with the unlabeled parallel instances.
- **Rd KCCA:** This is a Reduced Kernel CCA method, which is an unsupervised HDA method from (Yeh et al., 2014).
- **SFT-noCov:** A variant of the proposed SFT-HDA method that drops the covariance alignment component.
- **SFT-noPara:** A variant of the proposed SFT-HDA that drops the parallel data.
- **SFT-noSparse:** A variant of the proposed SFT-HDA method that drops the sparse regularization term.
- **SFT**$_{1,1}$ and **SFT**$_{1,2}$**:** Our proposed SFT-HDA approach with the ℓ_1-norm and $\ell_{1,2}$-norm sparisity regularizers respectively.

For both linear CCA and Rd KCCA, we conducted experiments following the work in (Yeh et al., 2014). and set the correlation coefficient as 0.5. But we set the size of reduced set as 30, which leads to better performance than their original setting in our experiments.

For each method, a linear Support Vector Machine (SVM) is trained on the transformed labeled source instances, and tested on the target domain instances. For linear CCA and Rd KCCA, the target instances are also projected to the learned subspace. The hyper-parameter C of the SVM is selected with 5-fold cross-validation on the transformed labeled source instances. We also tried the linear CCA and Rd KCCA with the correlation-transfer SVM (CTSVM) proposed in (Yeh et al., 2014), the improvement of accuracy is about 2% and it has little influence on the conclusions. To provide a fair comparison, we hence reported the SVM classification results for all comparison methods.

4.3. Parameter Selection

There are three hyper parameters in our approach: ρ, α and γ. However, ρ is only related to the ADMM optimization algorithm and it just needs to be set to a reasonable large value to guarantee the recovery of the equality constraint. We used $\rho = 10$ for the first and second experiments, but used $\rho = 1000$ in the third experiment to make the huge sparse feature transformation converge quicker. α is the trade-off parameter to balance the weights of the distribution alignment loss and the parallel mapping loss. We simply gave both losses equal weights and set $\alpha = 1$ in all experiments. As this is an unsupervised approach (no labeled data in the target domain), the traditional hyper-parameter tuning method of cross-validation is not really applicable. Nevertheless, we used 5-fold cross-validation of linear SVM to select the value of γ from $[0.01, 0.1, 1]$ on the transformed labeled source instances $X_s A$. Moreover, the regularization parameter λ in the closed-form initialization of A and B is also set to a reasonable large value – we simply used the same value as ρ, while $\epsilon = 10^{-4}$ is used for detecting the convergence of the training algorithm.

Table 1: Average classification accuracy (%) over 20 runs on UCI Multiple Feature dataset.

Source	fou	fou	fou	fac	fac	fac
Target	fac	kar	pix	fou	kar	pix
linear CC	36.34	32.36	32.31	27.32	50.60	57.45
Rd KCCA	64.03	52.67	58.62	52.33	76.26	87.49
SFT-noCov	66.05	58.28	65.17	55.84	83.79	87.67
SFT-noPara	14.49	12.64	7.37	9.01	18.18	11.86
SFT-noSparse	71.16	52.87	65.37	59.53	85.00	92.96
$SFT_{1,1}$	**73.83**	56.50	**69.53**	**61.53**	**85.97**	93.35
$SFT_{1,2}$	71.06	**60.44**	66.79	61.17	85.50	**93.53**
Source	kar	kar	kar	pix	pix	pix
Target	fou	fac	pix	fou	fac	kar
linear CC	29.72	57.37	74.00	27.14	51.58	66.78
Rd KCCA	53.62	84.85	86.43	53.33	89.65	81.66
SFT-noCov	62.16	81.86	88.86	58.07	83.88	87.40
SFT-noPara	11.54	18.27	17.48	6.99	17.56	16.45
SFT-noSparse	62.47	91.30	93.71	56.51	91.59	89.31
$SFT_{1,1}$	62.66	91.59	**93.82**	58.92	**92.14**	**89.69**
$SFT_{1,2}$	**63.40**	**91.86**	93.79	**60.01**	92.01	88.13

4.4. Classification Results

4.4.1. Digits Classification

We first tested all the comparison methods on the 12 HDA tasks formed on the UCI Multiple Features dataset for digits classification. The average multi-class classification accuracy results over 20 runs are reported in Table 1 – each run is with a different random source and target instance partition. We can see SFT-noPara has very poor performance here. Though this strategy works well on DA tasks (Sun et al., 2015), HDA is apparently a more challenging task. This set of HDA experiments suggest that the parallel data transformation component provides a major contribution to our SFT-HDA model. But nevertheless, the covariance matching and sparsity regularization components can further improve the performance, as the two full versions of the proposed SFT-HDA ($SFT_{1,1}$ and $SFT_{1,2}$) cover the best results across all the 12 HDA tasks. Compared with the baseline variants, the accuracy results of $SFT_{1,1}$ and $SFT_{1,2}$ are in general better than SFT without either the sparsity regularizer or the covariance alignment component. Moreover, our proposed full SFT-HDA methods substantially outperform the linear CCA and Rd KCCA methods, which depend on the parallel data. These results verified the efficacy of our proposed model.

4.4.2. Multimedia Classification

On the multimedia *Wikipedia dataset*, we have two HDA tasks, one performs adaptation from image to text and the other adapts from text to image. On this dataset, instead of using a fixed number of parallel instances, we conducted evaluations with different numbers of parallel instances, i.e., with n_p varies from 100 to 200. The mean values and standard deviations of the multi-class classification accuracy results over 20 runs for the 2 HDA tasks are reported in Table 2. We can see for Image→Text HDA task, $SFT_{1,2}$ achieves a high

Table 2: Average classification accuracy (%) over 20 runs on Wikipedia dataset.

HDA task	Image → Text		
# parallel instances	100	150	200
linear CCA	58.23±2.50	49.60±3.61	40.80±2.27
Rd KCCA	47.67±4.83	64.58±2.06	67.40±2.55
SFT-noCov	75.70±2.11	81.47±1.44	84.35±1.58
SFT-noPara	22.10±1.99	22.10±1.99	22.10±1.99
SFT-noSparse	61.02±2.28	69.53±2.78	78.88±2.02
$SFT_{1,1}$	71.47±2.74	79.03±2.11	83.78±1.37
$SFT_{1,2}$	**76.17±2.81**	**83.92±1.56**	**88.05±1.44**
HDA task	Text → Image		
# parallel instances	100	150	200
linear CCA	27.48±0.72	20.88±0.43	26.88±0.48
Rd KCCA	35.92±1.95	38.85±1.23	**46.33±0.80**
SFT-noCov	41.92±0.43	**41.08±0.66**	42.98±0.46
SFT-noPara	20.25±1.18	20.25±1.18	20.25±1.18
SFT-noSparse	**43.35±0.91**	40.33±0.78	43.55±0.66
$SFT_{1,1}$	42.83±0.67	40.23±0.68	42.80±0.55
$SFT_{1,2}$	42.42±0.77	40.58±0.76	42.85±0.55

accuracy of 76.17% even if there is only 100 parallel instances and it outperforms linear CCA and Rd KCCA by about 18% and 29% respectively. The performance of Rd KCCA is not good when there is not much parallel data available. Linear CCA performs worse as n_p increases to 200, which might be caused by the small dimension of the subspace and the high variety of the covariance with more instances. With the increasing of the number of parallel instances, the performance of our proposed approaches increases dramatically, which again shows the importance of parallel data. We can also see that with $\ell_{1,2}$-norm $SFT_{1,2}$ produces the best results on Image→Text and outperforms the ℓ_1-norm variant $SFT_{1,1}$ with notable margins. This suggests each text feature can be explained by a small fraction of image features. For the Text→Image HDA task, the sparsity regularizers for our approach however are not effective, and the impact of increasing parallel instance number is negligible. The reason is that the feature dimension of the text domain is quite small – only 10 features. It is hence not beneficial to have a sparse mapping to the image features or increase the number of parallel instances. Overall, our proposed full approaches again outperform both linear CCA and Rd KCCA.

4.4.3. Cross-domain Image Classification

The experimental results on 12 HDA tasks of the *Office-Caltech* dataset are reported in Table 3. The 4096-dim deep features are challenging. Rd KCCA fails to work on this dataset because it relies on k-means to cluster the instances into several splits with similar sizes and k-means fails to work on centralized deep features. The sparsity regularizers in our model are also affected by the deep features. The performance of SFT-noSparse, $SFT_{1,1}$ and $SFT_{1,2}$ are quite close. Nevertheless, our proposed approaches still outperform the linear CCA method, while our approach is the first to achieve promising results under unsupervised HDA setting for deep features.

Table 3: Average classification accuracy (%) over 20 runs on Office-Caltech dataset.

HDA task	SURF → VGG					
Source	A	A	W	W	C	C
Target	W	C	A	C	A	W
linear CCA	33.96	43.09	40.14	40.36	44.63	28.96
SFT-noCov	57.31	63.04	65.46	63.89	64.68	49.73
SFT-noPara	14.55	8.14	2.37	7.03	19.48	13.53
SFT-noSparse	**72.69**	70.05	**78.57**	**72.30**	72.64	**50.04**
$SFT_{1,1}$	72.59	69.99	78.06	**72.30**	**73.58**	48.33
$SFT_{1,2}$	71.57	**70.26**	76.66	70.90	70.79	46.55
HDA task	VGG → SURF					
Source	A	A	W	W	C	C
Target	W	C	A	C	A	W
linear CCA	29.41	29.91	24.28	24.81	31.15	25.61
SFT-noCov	33.51	31.10	37.21	**33.36**	43.47	43.82
SFT-noPara	9.67	11.64	7.84	9.55	18.86	16.41
SFT-noSparse	**37.12**	**34.51**	37.26	32.03	**48.87**	53.37
$SFT_{1,1}$	36.71	34.32	38.45	32.05	48.19	**54.86**
$SFT_{1,2}$	35.61	33.85	**38.56**	33.10	46.58	52.84

4.4.4. Parameter Sensitivity Analysis

We conducted sensitivity analysis for the trade-off parameters α and γ with the HDA tasks on the Wikipedia dataset. We used 150 parallel instances, $\rho = 10$ and $\lambda = 10$. For the proposed methods, we conducted experiments first with $\gamma = 1$ and $\alpha \in [0.1, 0.5, 1, 2, 10]$, and then with $\alpha = 1$ and $\gamma \in [0.01, 0.05, 0.1, 0.5, 1]$. The average accuracy and standard deviation of 20 rounds for each HDA task are reported in Figure 2. From the two sub-figures on the left side, we can see that with a fixed γ value, $\alpha = 1$ leads to test performance that is among the best for both HDA tasks, which suggests that it is reasonable to give equal weights to the distribution alignment loss and the parallel mapping loss. The parameter γ controls the sparsity regularization term. The best value choice for γ depends on the properties of cross-domain features. As shown in the two sub-figures on the right side, γ should be set to a relative large value for mapping 128-dimensional image features to 10-dimensional text features. We also tested a much larger γ value than the range in the figure, but found the performance could be unstable and drop down rapidly, while selecting γ from $[0.01, 0.1, 1]$ in the previous experiments is reasonable for both the mapping from high to low dimensional feature space and the reverse mapping.

5. Conclusions

In this paper, we proposed a novel sparse feature transformation method for unsupervised heterogeneous domain adaptation. The method transforms the source domain features into the target domain feature space by matching the parallel instances and aligning the empirical second moments of the transformed source feature distribution and target domain feature distribution. To encode the assumption that only a small fraction of source domain features are related to a target domain feature and increase the robustness of transformation, we

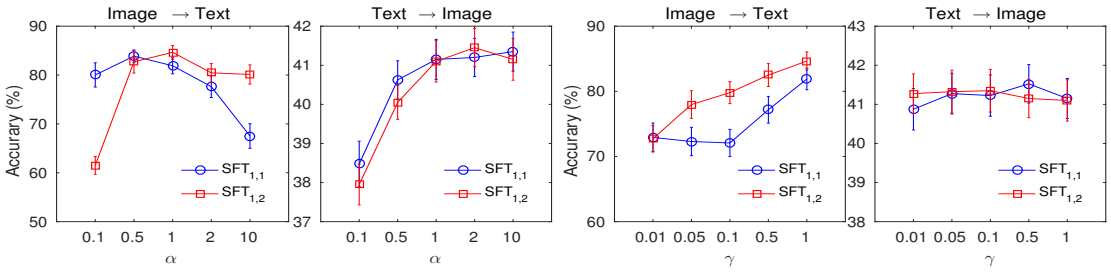

Figure 2: Performance of $SFT_{1,1}$ and $SFT_{1,2}$ with respect to trade-off parameters α and γ for two HDA tasks on Wikipedia dataset

further exploited two types of sparsity inducing norms to regularize the linear transformation model. We developed an ADMM based optimization algorithm to solve the induced problem and conducted experiments for heterogeneous cross-domain classification. The experimental results demonstrated the benefits of our proposed approach.

Acknowledgments

This research was supported in part by NSF grant (1546480), NSERC discovery grant and Canada Research Chairs program.

References

A. Asuncion and D. Newman. UCI machine learning repository. http://archive.ics.uci.edu/, 2007.

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *J. on Imaging Sciences*, 2(1):183–202, 2009.

K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. Domain separation networks. In *NIPS*, 2016.

S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning (FTML)*, 3(1):1–122, 2011.

Y. Cao, M. Long, and J. Wang. Unsupervised domain adaptation with distribution matching machines. In *AAAI*, 2018.

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

B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *ICCV*, 2013.

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

D. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computat.*, 16(12):2639–2664, 2004.

J. Hoffman, E. Rodner, J. Donahue, T. Darrell, and K. Saenko. Efficient learning of domain-invariant image representations. *arXiv preprint arXiv:1301.3224*, 2013.

M. Hong, Z. Luo, and M. Razaviyayn. Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. *J. on Optimizat.*, 26(1):337–364, 2016.

M. Kowalski, M. Szafranski, and L. Ralaivola. Multiple indefinite kernel learning with mixed norm regularization. In *ICML*, 2009.

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

M. Long, J. Wang, G. Ding, J. Sun, and P. S Yu. Transfer joint matching for unsupervised domain adaptation. In *CVPR*, 2014.

L. Niu, W. Li, and D. Xu. Multi-view domain generalization for visual recognition. In *ICCV*, 2015.

J. C Platt, K. Toutanova, and W. Yih. Translingual document representations from discriminative projections. In *EMNLP*, 2010.

N. Rasiwasia, J. C. Pereira, E. Coviello, G. Doyle, G. RG Lanckriet, R. Levy, and N. Vasconcelos. A new approach to cross-modal multimedia retrieval. In *ACM MM*, 2010.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

S. Sukhija, N. C Krishnan, and G. Singh. Supervised heterogeneous domain adaptation via random forests. In *IJCAI*, 2016.

B. Sun, J. Feng, and K. Saenko. Return of frustratingly easy domain adaptation. *arXiv preprint arXiv:1511.05547*, 2015.

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

B. Vatashsky and K. Crammer. Multi class learning with individual sparsity. In *IJCAI*, 2013.

P. Wei, Y. Ke, and C.K. Goh. Deep nonlinear feature coding for unsupervised domain adaptation. In *IJCAI*, 2016.

X. Wu, H. Wang, C. Liu, and Y. Jia. Cross-view action recognition over heterogeneous feature spaces. In *ICCV*, 2013.

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

M. Xiao and Y. Guo. Feature space independent semi-supervised domain adaptation via kernel matching. *PAMI*, 37(1):54–66, 2015.

Y. Yan, W. Li, M. KP Ng, M. Tan, H. Wu, H. Min, and Q. Wu. Learning discriminative correlation subspace for heterogeneous domain adaptation. In *IJCAI*, 2017.

T. Yao, Y. Pan, C. Ngo, H. Li, and T. Mei. Semi-supervised domain adaptation with subspace learning for visual recognition. In *CVPR*, 2015.

Y. Yeh, C. Huang, and Y.F. Wang. Heterogeneous domain adaptation and classification by exploiting the correlation subspace. *TIP*, 23(5):2009–2018, 2014.

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