Learning with Augmented Features for Supervised and Semi-Supervised Heterogeneous Domain Adaptation

Wen Li, *Student Member, IEEE*, Lixin Duan, Dong Xu, *Senior Member, IEEE*, and Ivor W. Tsang

Abstract—In this paper, we study the heterogeneous domain adaptation (HDA) problem, in which the data from the source domain and the target domain are represented by heterogeneous features with different dimensions. By introducing two different projection matrices, we first transform the data from two domains into a common subspace such that the similarity between samples across different domains can be measured. We then propose a new feature mapping function for each domain, which augments the transformed samples with their original features and zeros. Existing supervised learning methods (*e.g.*, SVM and SVR) can be readily employed by incorporating our newly proposed augmented feature representations for supervised HDA. As a showcase, we propose a novel method called Heterogeneous Feature Augmentation (HFA) based on SVM. We show that the proposed formulation can be equivalently derived as a standard Multiple Kernel Learning (MKL) problem, which is convex and thus the global solution can be guaranteed. To additionally utilize the unlabeled data in the target domain, we further propose the semi-supervised HFA (SHFA) which can simultaneously learn the target classifier as well as infer the labels of unlabeled target samples. Comprehensive experiments on three different applications clearly demonstrate that our SHFA and HFA outperform the existing HDA methods.

Index Terms—Heterogeneous domain adaptation, domain adaptation, transfer learning, augmented features

1 Introduction

In real-world applications, it is often expensive and time-consuming to collect the labeled data. Domain adaptation, as a new machine learning strategy, has attracted growing attention because it can learn robust classifiers with very few or even no labeled data from the target domain by leveraging a large amount of labeled data from other existing domains (a.k.a., source/auxiliary domains).

Domain adaptation methods have been successfully used for different research fields such as natural language processing and computer vision [1]–[7]. According to the supervision information in the target domain, the domain adaptation methods can generally be divided into three categories: supervised domain adaptation by only using the labeled data in the target domain, semi-supervised domain adaptation by using both the labeled and unlabeled data in the target domain, and unsupervised domain adaptation by only using unlabeled data in the target domain. However, most existing methods assume that the data from different domains are represented by the same type of features with the same dimension. Thus, they cannot deal with the problem where the dimensions of data from the source and target domains are different, which is known as heterogeneous domain adaptation (HDA) [8], [9].

In the literature, a few approaches have been proposed for the HDA problem. To discover the connection between different features, some work exploited an auxiliary dataset which encodes the correspondence between different types of features. Dai *et al*. [8] proposed to learn a feature translator between two features from two domains, which is modeled by the conditional probability of one feature given the other one. Such feature translator is learnt from an auxiliary dataset which contains the co-occurrence of these two types of features. A similar assumption was also used in [9], [10] for text-aid image clustering and classification. Others proposed to use an explicit feature correspondence, for example, the bilingual dictionary in cross-language text classification task. Based on the structural correspondence learning (SCL) [1], two methods [11], [12] were recently proposed to extract the so-called *pivot* features from the source and target domains, which are specifically designed for the cross-language text classification task. These pivot features are constructed by text words which have explicit semantic meanings. They either directly translated the pivot features from one language to the other or modified the original SCL to select pairs of pivot words from different languages. However, it is unclear how to build such correspondence for more general HDA tasks such as the object recognition task where only the low-level visual features are provided.

- *W. Li, and D. Xu are with the School of Computer Engineering, Nanyang Technological University, Singapore 639798. E-mail: wli1@e.ntu.edu.sg; dongxu@ntu.edu.sg.*
- *L. Duan is with the Institute for Infocomm Research, Singapore 138632. E-mail: lxduan@gmail.com.*
- *I. W. Tsang is with the Center for Quantum Computation & Intelligent Systems, University of Technology, Sydney, Australia. E-mail:ivor.tsang@gmail.com.*

Manuscript received 21 Jan. 2013; revised 15 June 2013; accepted 2 Aug. 2013. Date of publication 28 Aug. 2013; date of current version 12 May 2014.
Recommended for acceptance by K. Borgwardt.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier 10.1109/TPAMI.2013.167

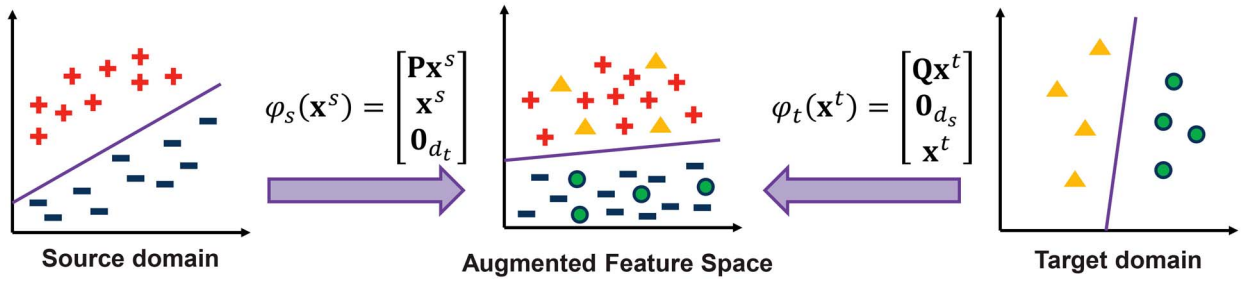


Fig. 1. Samples from different domains are represented by different features, where red crosses, blue strips, orange triangles and green circles denote source positive samples, source negative samples, target positive samples and target negative samples, respectively. By using two projection matrices **P** and **Q**, we transform the heterogenous samples from two domains into an augmented feature space.

For more general HDA tasks, Shi *et al*. [13] proposed a method called Heterogeneous Spectral Mapping (HeMap) to discover a common feature subspace by learning two feature mapping matrices as well as the optimal projection of the data from both domains. Harel and Mannor [14] learnt rotation matrices to match source data distributions to that of the target domain. Wang and Mahadevan [15] used the class labels of training data to learn the manifold alignment by simultaneously maximizing the intra-domain similarity and the inter-domain dissimilarity. By kernelizing the method in [16], Kulis *et al*. [17] proposed to learn an asymmetric kernel transformation to transfer feature knowledge between the data from the source and target domains. However, these existing HDA methods were designed for the supervised learning scenario. For these methods, it is unclear how to learn the projection matrices or transformation metric by utilizing the abundant unlabeled data in the target domain which is usually available in many applications.

In this work, we first propose a new method called Heterogeneous Feature Augmentation (HFA) for supervised heterogeneous domain adaptation. As shown in Fig. 1, considering the data from different domains are represented by features with different dimensions, we first transform the data from the source and target domains into a common subspace by using two different projection matrices **P** and **Q**. Then, we propose two new feature mapping functions to augment the transformed data with their original features and zeros. With the new augmented feature representations, we propose to learn the projection matrices **P** and **Q** by using the standard SVM with the hinge loss function in a linear case. We also describe its kernelization in order to efficiently cope with the data with very high dimension. To simplify the nontrivial optimization problem in HFA, we introduce an intermediate variable **H** called as a transformation metric to combine **P** and **Q**. In our preliminary work [18], we proposed an alternating optimization algorithm to iteratively learn an individual transformation metric **H** and a classifier for each class. However, the global convergence remains unclear and there may be pre-mature convergence. In this work, we equivalently reformulate it into a convex optimization problem by decomposing **H** into a linear combination of a set of rank-one positive semi-definite (PSD) matrices, which shares a similar formulation with the well-known Multiple Kernel Learning (MKL) problem [19]. Therefore, the global solution can be obtained easily by using the existing MKL solvers.

Moreover, we further extend our HFA to semi-supervised HFA or SHFA in short by additionally utilizing the unlabeled data in the target domain. While learning the transformation metric **H**, we also infer the labels for the unlabeled target samples. Considering we need to solve a non-trivial mixed integer programming problem when inferring the labels of unlabeled target training data, we first relax the objective of SHFA into a problem of finding the optimal linear combination of all possible label candidates. Then we also use the linear combination of these rank-one PSD matrices to replace **H** as in HFA. Finally, we further rewrite the problem as a convex MKL problem which can be readily solved by existing MKL solvers.

The remainder of this paper is organized as follows. The proposed HFA method and SHFA are introduced in Section 2 and Section 3, respectively. Extensive experimental results are presented in Section 4, followed by conclusions and future work in Section 5.

2 Heterogeneous Feature Augmentation

In the remainder of this paper, we use the superscript $'$ to denote the transpose of a vector or a matrix. We define $\mathbf{I}_n$ as the $n \times n$ identity matrix and $\mathbf{O}_{n\times m}$ as the $n \times m$ matrix of all zeros. We also define $\mathbf{0}_n, \mathbf{1}_n \in \mathbb{R}^n$ as the $n \times 1$ column vectors of all zeros and all ones, respectively. For simplicity, we also use $\mathbf{I}$, $\mathbf{O}$, $\mathbf{0}$ and $\mathbf{1}$ instead of $\mathbf{I}_n$, $\mathbf{O}_{n\times m}$, $\mathbf{0}_n$ and $\mathbf{1}_n$ when the dimension is obvious. The ℓ_p-norm of a vector $\mathbf{a} = [a_1, \ldots, a_n]'$ is defined as $\|\mathbf{a}\|_p = \left(\sum_{i=1}^n a_i^p\right)^{\frac{1}{p}}$. We also use $\|\mathbf{a}\|$ to denote the ℓ_2-norm. The inequality $\mathbf{a} \leq \mathbf{b}$ means that $a_i \leq b_i$ for $i = 1, \ldots, n$. Moreover, $\mathbf{a} \circ \mathbf{b}$ denotes the element-wise product between the vectors $\mathbf{a}$ and $\mathbf{b}$, *i.e.*, $\mathbf{a} \circ \mathbf{b} = [a_1 b_1, \ldots, a_n b_n]'$. And $\mathbf{H} \succeq 0$ means that $\mathbf{H}$ is a positive semi-definite (PSD) matrix.

In this work, we assume there are only one source domain and one target domain. We are provided with a set of labeled training samples $\{(\mathbf{x}_i^s, y_i^s)|_{i=1}^{n_s}\}$ from the source domain as well as a limited number of labeled samples $\{(\mathbf{x}_i^t, y_i^t)|_{i=1}^{n_t}\}$ from the target domain, where y_i^s and y_i^t are the labels of the samples $\mathbf{x}_i^s$ and $\mathbf{x}_i^t$, respectively, and $y_i^s, y_i^t \in \{1, -1\}$. The dimensions of $\mathbf{x}_i^s$ and $\mathbf{x}_i^t$ are d_s and d_t, respectively. Note that in the HDA problem, $d_s \neq d_t$. We also define $\mathbf{X}_s =$

$[\mathbf{x}_1^s, \ldots, \mathbf{x}_{n_s}^s] \in \mathbb{R}^{d_s \times n_s}$ and $\mathbf{X}_t = [\mathbf{x}_1^t, \ldots, \mathbf{x}_{n_t}^t] \in \mathbb{R}^{d_t \times n_t}$ as the data matrices for the source and target domains, respectively.

2.1 Heterogeneous Feature Augmentation

Daume III [3] proposed Feature Replication (FR) to augment the original feature space $\mathbb{R}^d$ into a larger space $\mathbb{R}^{3d}$ by replicating the source and target data for homogeneous domain adaptation. Specifically, for any data point $\mathbf{x} \in \mathbb{R}^d$, the feature mapping functions φ_s and φ_t for the source and target domains are defined as $\varphi_s(\mathbf{x}) = [\mathbf{x}', \mathbf{x}', \mathbf{0}_d']'$ and $\varphi_t(\mathbf{x}) = [\mathbf{x}', \mathbf{0}_d', \mathbf{x}']'$. Note that it is not meaningful to directly use the method in [3] for the HDA task by simply padding zeros to make the dimensions of the data from two domains become the same, because there would be no correspondences between the heterogeneous features in this case.

To effectively utilize the heterogeneous features from two domains, we first introduce a common subspace for the source and target data so that the heterogeneous features from two domains can be compared. We define the common subspace as $\mathbb{R}^{d_c}$, and any source sample $\mathbf{x}^s$ and target sample $\mathbf{x}^t$ can be projected onto it by using two projection matrices $\mathbf{P} \in \mathbb{R}^{d_c \times d_s}$ and $\mathbf{Q} \in \mathbb{R}^{d_c \times d_t}$, respectively. Note that promising results have been shown by incorporating the original features into the augmented features [3] to enhance the similarities between data from the same domain. Motivated by [3], we also incorporate the original features in this work and then augment any source and target domain samples $\mathbf{x}^s \in \mathbb{R}^{d_s}$ and $\mathbf{x}^t \in \mathbb{R}^{d_t}$ by using the augmented feature mapping functions φ_s and φ_t as follows:

$$\varphi_s(\mathbf{x}^s) = \begin{bmatrix} \mathbf{P}\mathbf{x}^s \\ \mathbf{x}^s \\ \mathbf{0}_{d_t} \end{bmatrix} \text{ and } \varphi_t(\mathbf{x}^t) = \begin{bmatrix} \mathbf{Q}\mathbf{x}^t \\ \mathbf{0}_{d_s} \\ \mathbf{x}^t \end{bmatrix}. \tag{1}$$

After introducing $\mathbf{P}$ and $\mathbf{Q}$, the data from two domains can be readily compared in the common subspace. It is worth mentioning that our newly proposed augmented features for the source and target samples in (1) can be readily incorporated into different methods (*e.g.*, SVM and SVR), making these methods applicable for the HDA problem.

Specifically, we use the standard SVM formulation with the hinge loss as a showcase for the supervised heterogeneous domain adaptation, which is referred as Heterogeneous Feature Augmentation (HFA). To additionally utilize the unlabeled data in the target domain, we also develop the semi-supervised HFA (SHFA) method based on ρ-SVM with the squared hinge loss for the semi-supervised heterogeneous domain adaptation task. Details of the two methods are introduced below.

2.2 Proposed Method

We define the feature weight vector $\mathbf{w} = [\mathbf{w}_c', \mathbf{w}_s', \mathbf{w}_t']' \in \mathbb{R}^{d_c+d_s+d_t}$ for the augmented feature space, where $\mathbf{w}_c \in \mathbb{R}^{d_c}$, $\mathbf{w}_s \in \mathbb{R}^{d_s}$ and $\mathbf{w}_t \in \mathbb{R}^{d_t}$ are also the weight vectors defined for the common subspace, the source domain and the target domain, respectively. We then propose to learn the projection matrices $\mathbf{P}$ and $\mathbf{Q}$ as well as the weight vector $\mathbf{w}$ by minimizing the structural risk functional of SVM. Formally, we present the formulation of our HFA method as follows:

$$\min_{\mathbf{P},\mathbf{Q}} \min_{\mathbf{w},b,\xi_i^s,\xi_i^t} \frac{1}{2}\|\mathbf{w}\|^2 + C\left(\sum_{i=1}^{n_s}\xi_i^s + \sum_{i=1}^{n_t}\xi_i^t\right), \tag{2}$$

$$\text{s.t.} \quad y_i^s(\mathbf{w}'\varphi_s(\mathbf{x}_i^s) + b) \geq 1 - \xi_i^s,\ \xi_i^s \geq 0; \tag{3}$$

$$y_i^t(\mathbf{w}'\varphi_t(\mathbf{x}_i^t) + b) \geq 1 - \xi_i^t,\ \xi_i^t \geq 0; \tag{4}$$

$$\|\mathbf{P}\|_F^2 \leq \lambda_p,\ \|\mathbf{Q}\|_F^2 \leq \lambda_q,$$

where $C > 0$ is a tradeoff parameter which balances the model complexity and the empirical losses on the training samples from two domains, and $\lambda_p, \lambda_q > 0$ are predefined parameters to control the complexities of $\mathbf{P}$ and $\mathbf{Q}$, respectively.

To solve (2), we first derive the dual form of the inner optimization problem in (2). Specifically, we introduce dual variables $\{\alpha_i^s|_{i=1}^{n_s}\}$ and $\{\alpha_i^t|_{i=1}^{n_t}\}$ for the constraints in (3) and (4), respectively. By setting the derivatives of the Lagrangian of (2) with respect to $\mathbf{w}$, b, ξ_i^s and ξ_i^t to zeros, we obtain the Karush-Kuhn-Tucker (KKT) conditions as: $\mathbf{w} = \sum_{i=1}^{n_s}\alpha_i^s y_i^s \varphi_s(\mathbf{x}_i^s) + \sum_{i=1}^{n_t}\alpha_i^t y_i^t \varphi_t(\mathbf{x}_i^t)$, $\sum_{i=1}^{n_s}\alpha_i^s y_i^s + \sum_{i=1}^{n_t}\alpha_i^t y_i^t = 0$ and $0 \leq \alpha_i^s, \alpha_i^t \leq C$. With the KKT conditions, we arrive at the dual problem as follows:

$$\min_{\mathbf{P},\mathbf{Q}} \max_{\boldsymbol{\alpha}} \quad \mathbf{1}'\boldsymbol{\alpha} - \frac{1}{2}(\boldsymbol{\alpha} \circ \mathbf{y})'\mathbf{K}_{\mathbf{P},\mathbf{Q}}(\boldsymbol{\alpha} \circ \mathbf{y}), \tag{5}$$

$$\text{s.t.} \quad \mathbf{y}'\boldsymbol{\alpha} = 0,\ \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1},$$

$$\|\mathbf{P}\|_F^2 \leq \lambda_p,\ \|\mathbf{Q}\|_F^2 \leq \lambda_q,$$

where $\boldsymbol{\alpha} = [\alpha_1^s, \ldots, \alpha_{n_s}^s, \alpha_1^t, \ldots, \alpha_{n_t}^t]' \in \mathbb{R}^{n_s+n_t}$ is a vector of the dual variables, $\mathbf{y} = [\mathbf{y}_s', \mathbf{y}_t']' \in \{+1, -1\}^{n_s+n_t}$ is the label vector of all training samples, $\mathbf{y}_s = [y_1^s, \ldots, y_{n_s}^s]' \in \{+1, -1\}^{n_s}$ is the label vector of samples from the source domain, $\mathbf{y}_t = [y_1^t, \ldots, y_{n_t}^t]' \in \{+1, -1\}^{n_t}$ is the label vector of samples from the target domain, and $\mathbf{K}_{\mathbf{P},\mathbf{Q}} = \begin{bmatrix} \mathbf{X}_s'(\mathbf{I}_{d_s} + \mathbf{P}'\mathbf{P})\mathbf{X}_s & \mathbf{X}_s'\mathbf{P}'\mathbf{Q}\mathbf{X}_t \\ \mathbf{X}_t'\mathbf{Q}'\mathbf{P}\mathbf{X}_s & \mathbf{X}_t'(\mathbf{I}_{d_t} + \mathbf{Q}'\mathbf{Q})\mathbf{X}_t \end{bmatrix} \in \mathbb{R}^{(n_s+n_t)\times(n_s+n_t)}$ is the derived kernel matrix for the samples from both domains.

To solve the optimization problem in (5), the dimension of the common subspace (*i.e.*, d_c) must be given beforehand. However, it is usually nontrivial to determine the optimal d_c. Observing that in the kernel matrix $\mathbf{K}_{\mathbf{P},\mathbf{Q}}$ in (5), the projection matrices $\mathbf{P}$ and $\mathbf{Q}$ always appear in the forms of $\mathbf{P}'\mathbf{P}, \mathbf{P}'\mathbf{Q}, \mathbf{Q}'\mathbf{P}$ and $\mathbf{Q}'\mathbf{Q}$, we then replace these multiplications by defining an intermediate variable $\mathbf{H} = [\mathbf{P}, \mathbf{Q}]'[\mathbf{P}, \mathbf{Q}] \in \mathbb{R}^{(d_s+d_t)\times(d_s+d_t)}$. Obviously, $\mathbf{H}$ is positive semidefinite, *i.e.*, $\mathbf{H} \succeq 0$. With the introduction of $\mathbf{H}$, we can throw away the parameter d_c. Moreover, the common subspace becomes latent, because we do not need to explicitly solve for $\mathbf{P}$ and $\mathbf{Q}$ any more.

With the definition of $\mathbf{H}$, we reformulate the optimization problem in (5) as follows:

$$\min_{\mathbf{H}\succeq 0} \max_{\boldsymbol{\alpha}} \quad \mathbf{1}'\boldsymbol{\alpha} - \frac{1}{2}(\boldsymbol{\alpha} \circ \mathbf{y})'\mathbf{K}_{\mathbf{H}}(\boldsymbol{\alpha} \circ \mathbf{y}), \tag{6}$$

$$\text{s.t.} \quad \mathbf{y}'\boldsymbol{\alpha} = 0,\ \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1},$$

$$\text{trace}(\mathbf{H}) \leq \lambda,$$

where $\mathbf{K}_{\mathbf{H}} = \mathbf{X}'(\mathbf{H} + \mathbf{I})\mathbf{X}$, $\mathbf{X} = \begin{bmatrix} \mathbf{X}_s & \mathbf{O}_{d_s \times n_t} \\ \mathbf{O}_{d_t \times n_s} & \mathbf{X}_t \end{bmatrix} \in \mathbb{R}^{(d_s+d_t)\times(n_s+n_t)}$ and $\lambda = \lambda_p + \lambda_q$.

Thus far, we have successfully converted our original HDA problem, which learns two projection matrices $\mathbf{P}$ and $\mathbf{Q}$, into a new problem of learning a *transformation metric* $\mathbf{H}$. We emphasize that this new problem has two main advantages: i) it avoids determining the optimal dimension of the common subspace beforehand; and ii) as the common subspace becomes latent after the introduction of $\mathbf{H}$, we only need to optimize $\boldsymbol{\alpha}$ and $\mathbf{H}$ for our proposed method.

However, there are still two major limitations for the current formulation of HFA in (6): i) The transformation metric $\mathbf{H}$ is linear, which may not be effective for some recognition tasks. ii) The size of $\mathbf{H}$ grows with the dimensions of the source and target data (*i.e.*, d_s and d_t). It is computationally expensive to learn the linear metric $\mathbf{H}$ in (6) for some real-world applications (*e.g.*, text categorization) with very high dimensional data. In order to effectively deal with high dimensional data, inspired by [17], in the next subsection we will apply *kernelization* to the data from the source and target domains and show that (6) can be solved in a kernel space by learning a nonlinear transformation metric with its size independent from the feature dimensions.

2.3 Nonlinear Feature Transformation

Note that the size of the linear transformation metric $\mathbf{H}$ is related to the feature dimension, and thus it is computationally expensive for very high dimension data. In this subsection, we will show that by applying kernelization, the transformation metric is independent from the feature dimension and grows only with respect to the number of training data from both domains.

Let us denote the kernel on the source domain samples as $\mathbf{K}_s = \Phi_s'\Phi_s \in \mathbb{R}^{n_s \times n_s}$ where $\Phi_s = [\phi_s(\mathbf{x}_1^s), \ldots, \phi_s(\mathbf{x}_{n_s}^s)]$ and $\phi_s(\cdot)$ is the nonlinear feature mapping function induced by $\mathbf{K}_s$. Similarly, we denote the kernel on the target domain samples as $\mathbf{K}_t = \Phi_t'\Phi_t \in \mathbb{R}^{n_t \times n_t}$ where $\Phi_t = [\phi_t(\mathbf{x}_1^t), \ldots, \phi_t(\mathbf{x}_{n_t}^t)]$ and $\phi_t(\cdot)$ is the nonlinear feature mapping function induced by $\mathbf{K}_t$. As in the linear case, we can correspondingly define the augmented features $\varphi_s(\mathbf{x}^s)$ and $\varphi_t(\mathbf{x}^t)$ in (1) for the nonlinear features of two domains by replacing $\mathbf{x}^s$ and $\mathbf{x}^t$ with $\phi_s(\mathbf{x}^s)$ and $\phi_t(\mathbf{x}^t)$, respectively. Denoting the dimensions of the nonlinear features $\phi_s(\mathbf{x}^s)$ and $\phi_t(\mathbf{x}^t)$ as $\tilde{d}_s$ and $\tilde{d}_t$, we can also derive an optimization problem as in (6) to solve a transformation metric $\mathbf{H} \in \mathbb{R}^{(\tilde{d}_s+\tilde{d}_t)\times(\tilde{d}_s+\tilde{d}_t)}$ which maps the different nonlinear features from two domains into a common feature space. Correspondingly, the kernel can be written as $\mathbf{K}_\mathbf{H} = \Phi'(\mathbf{H}+\mathbf{I})\Phi$ where $\Phi = \begin{bmatrix} \Phi_s & \mathbf{O}_{\tilde{d}_s \times n_t} \\ \mathbf{O}_{\tilde{d}_t \times n_s} & \Phi_t \end{bmatrix} \in \mathbb{R}^{(\tilde{d}_s+\tilde{d}_t)\times(n_s+n_t)}$.

However, we usually do not know about the explicit forms of the nonlinear feature mapping functions $\phi_s(\cdot)$ and $\phi_t(\cdot)$ and hence the dimensions of $\mathbf{H}$ cannot be determined. Even in some special cases that the explicit forms of $\phi_s(\cdot)$ and $\phi_t(\cdot)$ can be derived, the dimensions of the nonlinear features, *i.e.* $\tilde{d}_s$ and $\tilde{d}_t$, are usually very high and hence it is very computationally expensive to solve $\mathbf{H}$.

Inspired by [17], we define a nonlinear transformation matrix $\tilde{\mathbf{H}} \in \mathbb{R}^{(n_s+n_t)\times(n_s+n_t)}$ which satisfies that $\mathbf{H} = \Phi\mathbf{K}^{-\frac{1}{2}}\tilde{\mathbf{H}}\mathbf{K}^{-\frac{1}{2}}\Phi'$ where $\mathbf{K} = \begin{bmatrix} \mathbf{K}_s & \mathbf{O}_{n_s \times n_t} \\ \mathbf{O}_{n_t \times n_s} & \mathbf{K}_t \end{bmatrix} \in \mathbb{R}^{(n_s+n_t)\times(n_s+n_t)}$ and $\mathbf{K}^{\frac{1}{2}}$ is the symmetric square root of $\mathbf{K}$. Now we show that the kernelization version of (6) can be derived as an optimization problem on $\tilde{\mathbf{H}}$ rather than $\mathbf{H}$.

It is easy to verify that $\text{trace}(\tilde{\mathbf{H}}) = \text{trace}(\mathbf{H}) \leq \lambda$. Moreover, the kernel matrix can be written as $\mathbf{K}_\mathbf{H} = \Phi'(\mathbf{H}+\mathbf{I})\Phi = \mathbf{K}^{\frac{1}{2}}(\tilde{\mathbf{H}}+\mathbf{I})\mathbf{K}^{\frac{1}{2}} = \mathbf{K}_{\tilde{\mathbf{H}}}$. Then we arrive at the formulation of our proposed HFA method after applying kernelization as follows:

$$\begin{aligned} \min_{\tilde{\mathbf{H}} \succeq 0} \max_{\boldsymbol{\alpha}} \; & \mathbf{1}'\boldsymbol{\alpha} - \frac{1}{2}(\boldsymbol{\alpha} \circ \mathbf{y})'\mathbf{K}_{\tilde{\mathbf{H}}}(\boldsymbol{\alpha} \circ \mathbf{y}), \\ \text{s.t.} \; & \mathbf{y}'\boldsymbol{\alpha} = 0, \; \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1}, \\ & \text{trace}(\tilde{\mathbf{H}}) \leq \lambda. \end{aligned} \tag{7}$$

Hence, we optimize $\tilde{\mathbf{H}}$ in (7) rather than directly solving $\mathbf{H}$. Note the size of $\tilde{\mathbf{H}}$ is independent from the feature dimensions $\tilde{d}_s$ and $\tilde{d}_t$.

Intuitively, one can observe that the main differences between the formulations of the nonlinear HFA in (7) and the linear HFA in (6) are: i) we use $\mathbf{K}^{\frac{1}{2}}$ in the nonlinear HFA to replace $\mathbf{X}$ in the linear case; ii) we also define a new nonlinear transformation metric $\tilde{\mathbf{H}}$ which only depends on the numbers of training samples n_s and n_t instead of using $\mathbf{H}$ which depends on the feature dimensions d_s and d_t. Despite the above differences, the two formulations share the same form from the perspective of optimization. Therefore, we will only discuss the nonlinear case in the remainder of this paper while the linear case can be similarly derived by replacing $\mathbf{K}^{\frac{1}{2}} \in \mathbb{R}^{(n_s+n_t)\times(n_s+n_t)}$ and $\tilde{\mathbf{H}} \in \mathbb{R}^{(n_s+n_t)\times(n_s+n_t)}$ with $\mathbf{X} \in \mathbb{R}^{(d_s+d_t)\times(n_s+n_t)}$ and $\mathbf{H} \in \mathbb{R}^{(d_s+d_t)\times(d_s+d_t)}$, respectively. We also use $\mathbf{H}$ instead of $\tilde{\mathbf{H}}$ below for better presentation.

2.4 A Convex Formulation

To solve the optimization problem in (7), in our preliminary work [18], we proposed an alternating optimization approach in which we iteratively solve an SVM problem with respect to $\boldsymbol{\alpha}$ and a semi-definite programming (SDP) problem with respect to $\mathbf{H}$. However, the global convergence remains unclear and there may be pre-mature convergence. In this subsection, we show that (7) can be equivalently reformulated as a convex MKL problem so that the global solution can be guaranteed by using the existing MKL solvers [19].

As pointed in [19], the Ivanov regularization can be replaced with some Tikhonov regularization and vice verse with the appropriate choice of regularization parameter, which means we can write the trace norm regularization in (7) either as a constraint or as a regularizer term in the objective function. Formally, let us denote $\mu(\mathbf{H}) = \max_{\boldsymbol{\alpha} \in \mathcal{A}} \mathbf{1}'\boldsymbol{\alpha} - \frac{1}{2}(\boldsymbol{\alpha} \circ \mathbf{y})'\mathbf{K}_\mathbf{H}(\boldsymbol{\alpha} \circ \mathbf{y})$ where $\mathcal{A} = \{\boldsymbol{\alpha} | \mathbf{y}'\boldsymbol{\alpha} = 0, \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1}\}$, then the problem in (7) can also be reformulated as:

$$\min_{\mathbf{H} \succeq 0} \; \mu(\mathbf{H}) + \eta \, \text{trace}(\mathbf{H}), \tag{8}$$

where η is a tradeoff parameter. By properly setting η, the above optimization problem yields the same solution as the original problem in (7) [19].

To avoid solving the non-trivial SDP problem as in [18], we propose to decompose $\mathbf{H}$ as a linear combination of a set of positive semi-definite (PSD) matrices. Inspired by [20], in this work, we use the set of rank-one normalized PSD

matrices which is defined as $\mathcal{M} = \{\mathbf{M}_r|_{r=1}^{\infty}\}$ where $\mathbf{M}_r = \mathbf{h}_r\mathbf{h}_r'$, $\mathbf{h}_r \in \mathbb{R}^{(n_s+n_t)}$ and $\mathbf{h}_r'\mathbf{h}_r = 1$. Then, any PSD matrix $\mathbf{H}$ in (8) can be represented as a linear combination of the rank-one PSD matrices in $\mathcal{M}$, *i.e.*, $\mathbf{H} = \mathbf{H}_{\boldsymbol{\theta}} = \sum_{r=1}^{\infty}\theta_r\mathbf{M}_r$ where the linear combination coefficient vector $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_\infty]'$, $\boldsymbol{\theta} \geq \mathbf{0}$. Although there are an infinite number of matrices in $\mathcal{M}$ (*i.e.*, the index r goes from 1 to ∞), only considering the linear combination vector $\boldsymbol{\theta}$ with a finite number of non-zero entries is sufficient to represent $\mathbf{H}$ as shown in [20].

Note that we have $\text{trace}(\mathbf{H}) = \text{trace}(\sum_{r=1}^{\infty}\theta_r\mathbf{M}_r) = \sum_{r=1}^{\infty}\theta_r\text{trace}(\mathbf{M}_r) = \mathbf{1}'\boldsymbol{\theta}$. Instead of directly solving for the optimal $\mathbf{H}$ in (8), we show in the following theorem that it is equivalent to solving for the optimal linear combination coefficient vector $\boldsymbol{\theta}$:

Theorem 1. *Given that $\boldsymbol{\theta}^*$ is the optimal solution to the following optimization problem,*

$$\min_{\boldsymbol{\theta}\geq\mathbf{0}} \ \mu(\mathbf{H}_{\boldsymbol{\theta}}) + \eta\,\mathbf{1}'\boldsymbol{\theta}, \tag{9}$$

$\mathbf{H}_{\boldsymbol{\theta}^*}$ is also the optimum to the optimization problem in (8).

Proof. Let us denote the objective function in (8) as $F(\mathbf{H}) = \mu(\mathbf{H}) + \eta\,\text{trace}(\mathbf{H})$ and the objective function in (9) as $G(\boldsymbol{\theta}) = \mu(\mathbf{H}_{\boldsymbol{\theta}}) + \eta\,\mathbf{1}'\boldsymbol{\theta}$, and denote the optimums to (8) and (9) as $\mathbf{H}^* = \arg\min_{\mathbf{H}\geq 0} F(\mathbf{H})$ and $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}\geq\mathbf{0}} G(\boldsymbol{\theta})$, respectively. To show $\mathbf{H}_{\boldsymbol{\theta}^*}$ is also the optimum of (8), we need to prove $F(\mathbf{H}_{\boldsymbol{\theta}^*}) = F(\mathbf{H}^*)$.

On one hand, we have $F(\mathbf{H}_{\boldsymbol{\theta}^*}) \geq F(\mathbf{H}^*)$, because $\mathbf{H}^*$ is the optimal solution to (8). On the other hand, we will prove it as $F(\mathbf{H}^*) \geq G(\boldsymbol{\theta}^*) = F(\mathbf{H}_{\boldsymbol{\theta}^*})$. Specifically, for any PSD matrix $\mathbf{H}$ and a vector $\boldsymbol{\theta}$ which satisfies $\mathbf{H} = \mathbf{H}_{\boldsymbol{\theta}} = \sum_{r=1}^{\infty}\theta_r\mathbf{M}_r$, we have $F(\mathbf{H}) = \mu(\mathbf{H}) + \eta\,\text{trace}(\mathbf{H}) = \mu(\mathbf{H}_{\boldsymbol{\theta}}) + \eta\,\mathbf{1}'\boldsymbol{\theta} = G(\boldsymbol{\theta}) \geq G(\boldsymbol{\theta}^*)$ in which $G(\boldsymbol{\theta}) \geq G(\boldsymbol{\theta}^*)$ is due to the fact that $\boldsymbol{\theta}^*$ is the optimal solution to (9). Thus we have $F(\mathbf{H}^*) \geq G(\boldsymbol{\theta}^*)$. Moreover, since $G(\boldsymbol{\theta}^*) = \mu(\mathbf{H}_{\boldsymbol{\theta}^*}) + \eta\,\mathbf{1}'\boldsymbol{\theta}^* = \mu(\mathbf{H}_{\boldsymbol{\theta}^*}) + \eta\,\text{trace}(\mathbf{H}_{\boldsymbol{\theta}^*}) = F(\mathbf{H}_{\boldsymbol{\theta}^*})$, we have $F(\mathbf{H}^*) \geq G(\boldsymbol{\theta}^*) = F(\mathbf{H}_{\boldsymbol{\theta}^*})$.

Finally, we conclude that $F(\mathbf{H}_{\boldsymbol{\theta}^*}) = F(\mathbf{H}^*)$, because we have proved $F(\mathbf{H}_{\boldsymbol{\theta}^*}) \geq F(\mathbf{H}^*)$ and $F(\mathbf{H}^*) \geq G(\boldsymbol{\theta}^*) = F(\mathbf{H}_{\boldsymbol{\theta}^*})$. This completes the proof. □

By replacing the Tikhonov regularization (9) with the corresponding Ivanov regularization (*i.e.* the regularizer term $\mathbf{1}'\boldsymbol{\theta}$ is rewritten as the constraint), we reformulate the optimization problem of HFA as:

$$\begin{aligned}\min_{\boldsymbol{\theta}}\max_{\boldsymbol{\alpha}\in\mathcal{A}} \ & \mathbf{1}'\boldsymbol{\alpha} - \frac{1}{2}(\boldsymbol{\alpha}\circ\mathbf{y})'\mathbf{K}^{\frac{1}{2}}(\mathbf{H}_{\boldsymbol{\theta}} + \mathbf{I})\mathbf{K}^{\frac{1}{2}}(\boldsymbol{\alpha}\circ\mathbf{y}), \\ \text{s.t.} \ & \mathbf{H}_{\boldsymbol{\theta}} = \sum_{r=1}^{\infty}\theta_r\mathbf{M}_r, \quad \mathbf{M}_r \in \mathcal{M}, \\ & \mathbf{1}'\boldsymbol{\theta} \leq \lambda, \quad \boldsymbol{\theta} \geq \mathbf{0}.\end{aligned} \tag{10}$$

By setting $\boldsymbol{\theta} \leftarrow \frac{1}{\lambda}\boldsymbol{\theta}$, it can be further rewritten as:

$$\min_{\boldsymbol{\theta}\in\mathcal{D}_{\boldsymbol{\theta}}}\max_{\boldsymbol{\alpha}\in\mathcal{A}} \ \mathbf{1}'\boldsymbol{\alpha} - \frac{1}{2}(\boldsymbol{\alpha}\circ\mathbf{y})'\sum_{r=1}^{\infty}\theta_r\mathbf{K}_r(\boldsymbol{\alpha}\circ\mathbf{y}), \tag{11}$$

where $\mathbf{K}_r = \mathbf{K}^{\frac{1}{2}}(\lambda\mathbf{M}_r + \mathbf{I})\mathbf{K}^{\frac{1}{2}}$ and $\mathcal{D}_{\boldsymbol{\theta}} = \{\boldsymbol{\theta}|\mathbf{1}'\boldsymbol{\theta} \leq 1, \boldsymbol{\theta} \geq 0\}$. It is an Infinite Kernel Learning (IKL) problem with each base kernel as $\mathbf{K}_r$, which can be readily solved with the existing MKL solver [19], [21].

2.5 Solution

One problem in (11) is that there are an infinite number of base kernels because the set $\mathcal{M}$ contains infinite rank-one matrices. However, a finite number of rank-one matrices are sufficient to represent the matrix $\mathbf{H}$ [20]. Inspired by [21], we solve (11) based on a small number of base kernels which are constructed by using the cutting-plane algorithm. Let us introduce a dual variable τ for $\boldsymbol{\theta}$ in (11) and write the dual form as:

$$\begin{aligned}\max_{\tau,\boldsymbol{\alpha}\in\mathcal{A}} \ & \mathbf{1}'\boldsymbol{\alpha} - \tau, \\ \text{s.t.} \ & \frac{1}{2}(\boldsymbol{\alpha}\circ\mathbf{y})'\mathbf{K}_r(\boldsymbol{\alpha}\circ\mathbf{y}) \leq \tau, \quad \forall r,\end{aligned} \tag{12}$$

which has an infinite number of constraints. With the cutting-plane algorithm, we can approximate (12) by iteratively adding a kernel for which the corresponding constraint is violated according to the current solution. The kernel associated with this constraint is called an *active kernel*. To find the most active kernel, we need to maximize the left-hand side of the constraint in (12), which is given as:

$$\max_{\mathbf{M}\in\mathcal{M}} \ \frac{1}{2}(\boldsymbol{\alpha}\circ\mathbf{y})'\mathbf{K}_{\mathbf{M}}(\boldsymbol{\alpha}\circ\mathbf{y}), \tag{13}$$

where $\mathbf{K}_{\mathbf{M}} = \mathbf{K}^{\frac{1}{2}}(\lambda\mathbf{M} + \mathbf{I})\mathbf{K}^{\frac{1}{2}}$. It has a closed form solution as $\mathbf{M} = \mathbf{h}\mathbf{h}' \in \mathbb{R}^{(n_s+n_t)\times(n_s+n_t)}$ with $\mathbf{h} = \frac{\mathbf{K}^{\frac{1}{2}}(\boldsymbol{\alpha}\circ\mathbf{y})}{\|\mathbf{K}^{\frac{1}{2}}(\boldsymbol{\alpha}\circ\mathbf{y})\|}$.

We summarize the proposed algorithm in Algorithm 1. First, we initialize the set of rank-one PSD matrices $\mathcal{M}$ with $\mathbf{M}_1 = \mathbf{h}_1\mathbf{h}_1'$ where $\mathbf{h}_1$ is a unit vector. Based on the current $\mathcal{M}$, we solve the MKL problem in (11) to obtain the optimal $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$. After that, we find the most active kernel which is decided by a rank-one PSD matrix $\mathbf{M}$ as in (13). By using the closed form solution of (13), we obtain a new rank-one PSD matrix and add it into the current set $\mathcal{M}$. Then we solve the MKL problem again. The above steps are repeated until convergence. After obtaining the optimal solution $\boldsymbol{\alpha}$ and $\mathbf{H}$ to (11), we can predict any test sample $\mathbf{x}$ from the target domain by using the following target decision function:

$$f(\mathbf{x}) = (\boldsymbol{\alpha}\circ\mathbf{y})'\mathbf{K}^{\frac{1}{2}}(\mathbf{H} + \mathbf{I})\begin{bmatrix}\mathbf{O}_{n_s\times n_t} \\ \mathbf{K}_t^{-\frac{1}{2}}\end{bmatrix}\mathbf{k}_t + b, \tag{14}$$

where $\mathbf{k}_t = [k(\mathbf{x}_1^t, \mathbf{x}), \ldots, k(\mathbf{x}_{n_t}^t, \mathbf{x})]'$ and $k(\mathbf{x}_i, \mathbf{x}_j) = \phi_t(\mathbf{x}_i)'\phi_t(\mathbf{x}_j)$ is a predefined kernel function for two data samples $\mathbf{x}_i$ and $\mathbf{x}_j$ in the target domain.

Complexity Analysis: In our HFA, we first calculate $\mathbf{K}^{\frac{1}{2}}$ once at the beginning, which costs $O(n^3)$ time with $n = n_s + n_t$ being the total number of training samples[1]. After that, we perform the cutting-plane algorithm (*i.e.*, Algorithm 1), in which we iteratively train an MKL classifier and find the most violated rank-one matrix as in (13). As we have an efficient closed form solution for solving (13), the major time cost of Algorithm 1 is from the training of MKL at each iteration. However, the time complexity of MKL has not been theoretically analyzed. Usually, the MKL solver needs to train an SVM classifier for a few iterations. The empirical analysis shows that optimizing the QP problem

1. More accurately, the time complexity for solving $\mathbf{K}^{\frac{1}{2}}$ is $O(n_s^3+n_t^3)$, because the kernel matrix $\mathbf{K}$ is a block-diagonal matrix.

Algorithm 1 Heterogeneous Feature Augmentation

Input: Labeled source samples $\{(\mathbf{x}_i^s, y_i^s)|_{i=1}^{n_s}\}$ and labeled target samples $\{(\mathbf{x}_i^t, y_i^t)|_{i=1}^{n_t}\}$.

1: Set $r = 1$ and initialize $\mathcal{M}_1 = \{\mathbf{M}_1\}$ with $\mathbf{M}_1 = \mathbf{h}_1\mathbf{h}_1'$ and $\mathbf{h}_1 = \frac{1}{\sqrt{n_s+n_t}}\mathbf{1}_{n_s+n_t}$.
2: **repeat**
3: Solve $\boldsymbol{\theta}$ and $\boldsymbol{\alpha}$ in (11) based on $\mathcal{M}_r$ by using the existing MKL solver [19].
4: Obtain a rank-one PSD matrix $\mathbf{M}_{r+1}$ by solving (13).
5: Set $\mathcal{M}_{r+1} = \mathcal{M}_r \bigcup \{\mathbf{M}_{r+1}\}$, and $r = r + 1$.
6: **until** The objective converges.

Output: $\boldsymbol{\alpha}$ and $\mathbf{H} = \lambda \sum_r \theta_r \mathbf{M}_r$.

in SVM is about $O(n^{2.3})$ [22]. Therefore, the complexity of MKL is $O(Ln^{2.3})$ with L being the number of iterations in MKL. Thus, the total time complexity of our HFA is $O(n^3 + TLn^{2.3})$, where T is the number of iterations in Algorithm 1. In practice, both L and T are not very large.

2.6 Convergence Analysis

Let us represent the objective function in (11) as $F(\boldsymbol{\alpha}, \boldsymbol{\theta}) = \mathbf{1}'\boldsymbol{\alpha} - \frac{1}{2}(\boldsymbol{\alpha} \circ \mathbf{y})' \sum_{r=1}^{\infty} \theta_r \mathbf{K}_r (\boldsymbol{\alpha} \circ \mathbf{y})$, and also denote the optimal solution to (11) as $(\boldsymbol{\alpha}^*, \boldsymbol{\theta}^*) = \arg\min_{\boldsymbol{\theta} \in \mathcal{D}_\theta} \max_{\boldsymbol{\alpha} \in \mathcal{A}} F(\boldsymbol{\alpha}, \boldsymbol{\theta})$.

We denote the optimal solution of the MKL problem at the r-th iteration as $(\boldsymbol{\alpha}^r, \boldsymbol{\theta}^r)$. Because there are at most r non-zero elements in $\boldsymbol{\theta}^r$, we assume these non-zero elements are the first r entries in $\boldsymbol{\theta}^r$ for ease of presentation. Then, we show in the following theorem that Algorithm 1 converges to the global optimal solution:

Theorem 2. *With Algorithm 1, $F(\boldsymbol{\alpha}^r, \boldsymbol{\theta}^r)$ monotonically decreases as r increases, and the following inequality holds*

$$F(\boldsymbol{\alpha}^r, \boldsymbol{\theta}^r) \geq F(\boldsymbol{\alpha}^*, \boldsymbol{\theta}^*) \geq F(\boldsymbol{\alpha}^r, \mathbf{e}_{r+1}),$$

where $\mathbf{e}_{r+1} \in \mathcal{D}_\theta$ is the vector with all zeros except the $(r+1)$-th entry being 1. We also have $F(\boldsymbol{\alpha}^r, \boldsymbol{\theta}^r) = F(\boldsymbol{\alpha}^*, \boldsymbol{\theta}^*) = F(\boldsymbol{\alpha}^r, \mathbf{e}_{r+1})$ when Algorithm 1 converges at the r-th iteration.

The theorem can be proved similarly as in [23]. We also give the proof in the Appendix, which is available in the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2013.167. Moreover, as indicated in [24], the cutting-plane algorithm stops in a finite number of steps under some conditions. In our experiments, the algorithm usually takes less than 50 iterations to obtain a sufficient accurate solution.

2.7 Discussion

Our work is related to the existing heterogeneous domain adaptation methods. The pioneering works [8]–[12] are limited to some specific HDA tasks, because they required additional information to transfer the source knowledge to the target domain. For instance, Dai *et al*. [8] and Zhu *et al*. [10] proposed to use either labeled or unlabeled text corpora to aid image classification by assuming images are associated with textual annotations. Such textual annotations can be additionally utilized to mine the word co-occurrence from textual annotations of images and words in text documents, which is served as a bridge to transfer knowledge from the text documents to images.

To handle more general HDA tasks, other methods have been proposed to explicitly discover a common subspace [13], [15], [17] without using additional information, such that original data from the source and target domains can be measured in the common subspace. Specifically, Shi *et al*. [13] proposed to learn feature mapping matrices based on a spectral transformation for domains with different features. Wang *et al*. [15] proposed to learn the feature mapping by using the manifold alignment. However, such manifold assumption may not be satisfied in real-world applications with very diverse data. Recently, Kulis *et al*. [17] proposed a nonlinear metric learning method to learn an asymmetric feature transformation for the source and target data with high dimensions. They assume that if one source sample and one target sample are from the same category, the learned similarity between this pair of samples should be large; otherwise, the similarity should be small.

In contrast to [13], [15], [17], in our proposed HFA, we simultaneously learn the common subspace and a max-margin classifier by solving a convex optimization problem, which shares a similar form with the MKL formulation. We also propose the heterogeneous augmented features by incorporating the original features from two domains, in order to learn a more robust classifier (see Section 4.3 for experimental comparisons). Moreover, our work can also be extended to handle unlabeled samples from the target domain as shown in the next section.

3 Semi-Supervised Heterogeneous Feature Augmentation

The unlabeled data has been demonstrated to be helpful for training a robust classifier in many applications [25]. For the traditional semi-supervised learning, readers can refer to [26] for a comprehensive survey. There are also many works on semi-supervised homogeneous domain adaptation, such as [27]–[29]. However, most existing heterogeneous domain adaptation works [13], [15], [17] were designed for the supervised setting, and cannot utilize the abundant unlabeled data in the target domain. Thus, we further propose semi-supervised HFA to utilize the unlabeled data in the target domain.

We still use $\{(\mathbf{x}_i^s, y_i^s)|_{i=1}^{n_s}\}$ and $\{(\mathbf{x}_i^t, y_i^t)|_{i=1}^{n_t}\}$ to represent the labeled data from the source domain and the target domain, respectively. Let us denote the unlabeled data in the target domain as $\{(\mathbf{x}_i^u, y_i^u)|_{i=1}^{n_u}\}$ where $\mathbf{x}_i^u \in \mathbb{R}^{d_t}$ is an unlabeled sample in the target domain, n_u is the number of unlabeled samples, and the label $y_i^u \in \{-1, +1\}$ is unknown. We also denote $\mathbf{y}_u = [y_1^u, \ldots, y_{n_u}^u]'$ as the label vector of all the unlabeled data. Moreover, the total number of training samples is denoted as $n = n_s + n_t + n_u$.

3.1 Formulation

Since the labels of unlabeled data are unknown, we propose to infer the optimal labeling $\mathbf{y}_u$ for the unlabeled data in the target domain when learning the classifier. Based on the ρ-SVM with the squared hinge loss, we propose the objective

for semi-supervised heterogeneous domain adaptation as follows:

$$\begin{aligned}\min_{\substack{\mathbf{y}_u,\mathbf{w},b,\rho,\\ \mathbf{P},\mathbf{Q},\xi_i^s,\xi_i^t,\xi_i^u}} \quad & \frac{1}{2}\left(\|\mathbf{w}\|^2+b^2\right)-\rho \\ & +\frac{C}{2}\left(\sum_{i=1}^{n_s}(\xi_i^s)^2+\sum_{i=1}^{n_t}(\xi_i^t)^2\right)+\frac{C_u}{2}\sum_{i=1}^{n_u}(\xi_i^u)^2 \qquad (15)\\ \text{s.t.}\quad & y_i^s(\mathbf{w}'\varphi_s(\mathbf{x}_i^s)+b)\geq\rho-\xi_i^s,\\ & y_i^t(\mathbf{w}'\varphi_t(\mathbf{x}_i^t)+b)\geq\rho-\xi_i^t,\\ & y_i^u(\mathbf{w}'\varphi_t(\mathbf{x}_i^u)+b)\geq\rho-\xi_i^u,\\ & \mathbf{1}'\mathbf{y}_u=\delta,\ \|\mathbf{P}\|_F^2\leq\lambda_p,\ \|\mathbf{Q}\|_F^2\leq\lambda_q,\end{aligned}$$

where $\varphi_s(\cdot)$ and $\varphi_t(\cdot)$ are defined in (1) for generating the augmented features, and the constraint $\mathbf{1}'\mathbf{y}_u=\delta$ is used as the prior information on the unlabeled data similarly as in Transductive SVM (T-SVM) [25]. We refer to the above method as Semi-supervised Heterogeneous Feature Augmentation, or SHFA in short.

Similarly as in HFA, we only discuss the nonlinear case for SHFA here, and the linear case can be derived analogously. Let us define a kernel matrix $\mathbf{K}=\begin{bmatrix}\mathbf{K}_s & \mathbf{O}_{n_s\times(n_t+n_u)}\\ \mathbf{O}_{(n_t+n_u)\times n_s} & \mathbf{K}_t\end{bmatrix}\in\mathbb{R}^{n\times n}$ where $\mathbf{K}_s\in\mathbb{R}^{n_s\times n_s}$ is the kernel of source domain samples and $\mathbf{K}_t\in\mathbb{R}^{(n_t+n_u)\times(n_t+n_u)}$ is the kernel of target domain samples. Then, by defining a nonlinear transformation metric $\mathbf{H}\in\mathbb{R}^{n\times n}$, we can derive the dual form of (15) as follows:

$$\begin{aligned}\min_{\mathbf{y}\in\mathcal{Y},\mathbf{H}\succeq\mathbf{0}}\max_{\boldsymbol{\alpha}\in\mathcal{A}}\ & -\frac{1}{2}\boldsymbol{\alpha}'(\mathbf{Q}_{\mathbf{H},\mathbf{y}}+\mathbf{D})\boldsymbol{\alpha} \qquad (16)\\ \text{s.t.}\ & \text{trace}(\mathbf{H})\leq\lambda,\end{aligned}$$

where $\mathbf{Q}_{\mathbf{H},\mathbf{y}}=\left(\mathbf{K}^{\frac{1}{2}}(\mathbf{H}+\mathbf{I})\mathbf{K}^{\frac{1}{2}}+\mathbf{1}\mathbf{1}'\right)\circ(\mathbf{y}\mathbf{y}')\in\mathbb{R}^{n\times n}$, $\mathbf{y}=[\mathbf{y}_s',\mathbf{y}_t',\mathbf{y}_u']'$ is the label vector in which $\mathbf{y}_s$ and $\mathbf{y}_t$ are given and $\mathbf{y}_u$ is unknown, $\mathcal{Y}=\{\mathbf{y}\in\{-1,+1\}^n|\mathbf{y}=[\mathbf{y}_s',\mathbf{y}_t',\mathbf{y}_u']',\ \mathbf{1}'\mathbf{y}_u=\delta\}$ is the domain of $\mathbf{y}$, $\boldsymbol{\alpha}=[\alpha_1^s,\ldots,\alpha_{n_s}^s,\alpha_1^t,\ldots,\alpha_{n_t}^t,\alpha_1^u,\ldots,\alpha_{n_u}^u]'\in\mathbb{R}^n$ with α_i^s's, α_i^t's and α_i^u's are the dual variables corresponding to the constraints for source samples, labeled target samples and unlabeled target samples, respectively, $\mathcal{A}=\{\boldsymbol{\alpha}|\boldsymbol{\alpha}\geq\mathbf{0},\ \mathbf{1}'\boldsymbol{\alpha}=1\}$ is the domain of $\boldsymbol{\alpha}$ and $\mathbf{D}\in\mathbb{R}^{n\times n}$ is a diagonal matrix with the diagonal elements as $\frac{1}{C}$ for the labeled data from both domains and $\frac{1}{C_u}$ for the unlabeled target data.

3.2 Convex Relaxation

Compared with HFA, one major challenge in (16) is that we need to infer the optimal label vector $\mathbf{y}$, which is a mixed integer programming (MIP) problem. It is an NP problem and is computationally expensive to be solved [30]–[32] because there are possibly an exponential number of feasible labeling candidates $\mathbf{y}$'s. Inspired by [30]–[32], instead of directly finding the optimal labeling $\mathbf{y}$, we seek for an optimal linear combination of the feasible labeling candidates $\mathbf{y}$'s, which leads to a lower-bound of the original optimization problem as described below.

Proposition 1. *The objective of (16) is lower-bounded by the optimum of the following optimization problem:*

$$\begin{aligned}\min_{\boldsymbol{\gamma}\in\mathcal{D}_{\boldsymbol{\gamma}},\mathbf{H}\succeq\mathbf{0}}\max_{\boldsymbol{\alpha}\in\mathcal{A}}\ & -\frac{1}{2}\boldsymbol{\alpha}'\left(\sum_l\gamma_l\mathbf{Q}_{\mathbf{H},\mathbf{y}_l}+\mathbf{D}\right)\boldsymbol{\alpha} \qquad (17)\\ \text{s.t.}\ & \text{trace}(\mathbf{H})\leq\lambda,\end{aligned}$$

where $\mathbf{y}_l$ is the l-th feasible labeling candidate, $\boldsymbol{\gamma}=[\gamma_1,\ldots,\gamma_{|\mathcal{Y}|}]'$ is the coefficient vector for the linear combination of all feasible labeling candidates and $\mathcal{D}_{\boldsymbol{\gamma}}=\{\boldsymbol{\gamma}|\boldsymbol{\gamma}\geq 0,\mathbf{1}'\boldsymbol{\gamma}\leq 1\}$ is the domain of $\boldsymbol{\gamma}$.

Proof. The proof is provided in the Appendix, available online. □

Another challenge in (16) or (17) is to solve the positive semi-definite matrix $\mathbf{H}$. We apply a similar strategy here as used in HFA to solve the optimization problem in (17). Specifically, we decompose $\mathbf{H}$ into a linear combination of a set of rank-one PSD matrices, *i.e.*, $\mathbf{H}=\sum_{r=1}^{\infty}\theta_r\mathbf{M}_r$ where $\mathbf{M}_r\in\mathbb{R}^{n\times n}$ is a rank-one PSD matrix and θ_r is the corresponding combination coefficient, which leads to the following optimization problem:

$$\min_{\boldsymbol{\gamma}\in\mathcal{D}_{\boldsymbol{\gamma}}}\min_{\boldsymbol{\theta}\in\mathcal{D}_{\boldsymbol{\theta}}}\max_{\boldsymbol{\alpha}\in\mathcal{A}}-\frac{1}{2}\boldsymbol{\alpha}'\left(\sum_r\sum_l\theta_r\gamma_l\mathbf{Q}_{\mathbf{M}_r,\mathbf{y}_l}+\mathbf{D}\right)\boldsymbol{\alpha} \qquad (18)$$

where $\mathbf{Q}_{\mathbf{M}_r,\mathbf{y}_l}=\left(\mathbf{K}^{\frac{1}{2}}(\lambda\mathbf{M}_r+\mathbf{I})\mathbf{K}^{\frac{1}{2}}+\mathbf{1}\mathbf{1}'\right)\circ(\mathbf{y}_l\mathbf{y}_l')$ and $\mathcal{D}_{\boldsymbol{\theta}}=\{\boldsymbol{\theta}|\boldsymbol{\theta}\geq 0,\ \mathbf{1}'\boldsymbol{\theta}\leq 1\}$.

However, there are three variables, $\boldsymbol{\theta},\boldsymbol{\gamma}$ and $\boldsymbol{\alpha}$ in (18). To efficiently solve this problem, we propose a relaxation by combining $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$ into one variable $\mathbf{d}$. Specifically, let us denote $d_k=\theta_r\gamma_l$ where d_k is the k-th entry of $\mathbf{d}$. After combining the two indices r and l into one index k, we have $\mathbf{1}'\mathbf{d}=\sum_k d_k=\sum_r\sum_l\theta_r\gamma_l=(\mathbf{1}'\boldsymbol{\theta})(\mathbf{1}'\boldsymbol{\gamma})\leq 1$. Then we reformulate the optimization problem in (18) as:

$$\min_{\mathbf{d}\in\mathcal{D}_{\mathbf{d}}}\max_{\boldsymbol{\alpha}\in\mathcal{A}}-\frac{1}{2}\boldsymbol{\alpha}'\left(\sum_k d_k\mathbf{Q}_{\mathbf{M}_k,\mathbf{y}_k}+\mathbf{D}\right)\boldsymbol{\alpha} \qquad (19)$$

where $\mathbf{Q}_{\mathbf{M}_k,\mathbf{y}_k}=\left(\mathbf{K}^{\frac{1}{2}}(\lambda\mathbf{M}_k+\mathbf{I})\mathbf{K}^{\frac{1}{2}}+\mathbf{1}\mathbf{1}'\right)\circ(\mathbf{y}_k\mathbf{y}_k')$ and $\mathcal{D}_{\mathbf{d}}=\{\mathbf{d}|\mathbf{1}'\mathbf{d}\leq\mathbf{1},\mathbf{d}\geq 0\}$.

Hence, we obtain an MKL problem as in (19) where each base kernel is $\mathbf{Q}_{\mathbf{M}_k,\mathbf{y}_k}$, and the primal form of (19) is as follows:

$$\begin{aligned}\min_{\mathbf{d},\mathbf{w}_k,\rho,\xi_i}\ & \frac{1}{2}\left(\sum_k\frac{\|\mathbf{w}_k\|^2}{d_k}+C\sum_{i=1}^n\nu_i(\xi_i)^2\right)-\rho \qquad (20)\\ \text{s.t.}\ & \sum_k\mathbf{w}_k'\psi_k(\mathbf{x}_i)\geq\rho-\xi_i,\\ & \mathbf{1}'\mathbf{d}\leq 1,\quad \mathbf{d}\geq 0,\end{aligned}$$

where $\mathbf{d}$ is the coefficient vector, $\psi_k(\cdot)$ is the k-th feature mapping function induced by the kernel $\mathbf{Q}_{\mathbf{M}_k,\mathbf{y}_k}=\left(\mathbf{K}^{\frac{1}{2}}(\lambda\mathbf{M}_k+\mathbf{I})\mathbf{K}^{\frac{1}{2}}+\mathbf{1}\mathbf{1}'\right)\circ(\mathbf{y}_k\mathbf{y}_k')$, and ν_i is the weight for the i-th sample which is 1 for labeled data from both domains and C_u/C for unlabeled target data.

3.3 Solution

Similar to HFA, there are also an infinite number of base kernels in (19). We therefore employ the cutting-plane algorithm to iteratively select a small set of active kernels. We first write the dual form of (20) as follows:

$$\begin{aligned} \max_{\tau,\boldsymbol{\alpha}\in\mathcal{A}} \quad & -\tau \\ \text{s.t.} \quad & \frac{1}{2}\boldsymbol{\alpha}'(\mathbf{Q}_{\mathbf{M}_k,\mathbf{y}_k}+\mathbf{D})\boldsymbol{\alpha} \le \tau, \quad \forall k \end{aligned} \tag{21}$$

where we have an infinite number of constraints. The subproblem for selecting the most active kernel is:

$$\max_{\mathbf{y}\in\mathcal{Y},\mathbf{M}\in\mathcal{M}} \quad \frac{1}{2}\boldsymbol{\alpha}'\mathbf{Q}_{\mathbf{M},\mathbf{y}}\boldsymbol{\alpha}, \tag{22}$$

where $\mathbf{Q}_{\mathbf{M},\mathbf{y}} = \left(\mathbf{K}^{\frac{1}{2}}(\lambda\mathbf{M}+\mathbf{I})\mathbf{K}^{\frac{1}{2}}+\mathbf{1}\mathbf{1}'\right)\circ(\mathbf{y}\mathbf{y}')$. Note that we do not need to consider the constant term $\boldsymbol{\alpha}'\mathbf{D}\boldsymbol{\alpha}$ in the above formulation when selecting the most active kernel.

Given any $\mathbf{y}$, finding the violated $\mathbf{M}$ is as the same as in HFA. It can be obtained by solving (13) with the closed form solution $\mathbf{M}=\mathbf{h}\mathbf{h}'$ where $\mathbf{h}=\frac{\mathbf{K}^{\frac{1}{2}}(\boldsymbol{\alpha}\circ\mathbf{y})}{\|\mathbf{K}^{\frac{1}{2}}(\boldsymbol{\alpha}\circ\mathbf{y})\|}$. Then we substitute $\mathbf{M}$ back into (22) and obtain

$$\begin{aligned} &\max_{\mathbf{y}\in\mathcal{Y},\mathbf{M}\in\mathcal{M}} \frac{1}{2}\boldsymbol{\alpha}'\mathbf{Q}_{\mathbf{M},\mathbf{y}}\boldsymbol{\alpha}, \\ &= \max_{\mathbf{y}\in\mathcal{Y},\mathbf{M}\in\mathcal{M}} \frac{1}{2}(\boldsymbol{\alpha}\circ\mathbf{y})'\left(\mathbf{K}^{\frac{1}{2}}(\lambda\mathbf{M}+\mathbf{I})\mathbf{K}^{\frac{1}{2}}+\mathbf{1}\mathbf{1}'\right)(\boldsymbol{\alpha}\circ\mathbf{y}), \\ &= \max_{\mathbf{y}\in\mathcal{Y}} \quad \lambda\frac{(\boldsymbol{\alpha}\circ\mathbf{y})'\mathbf{K}(\boldsymbol{\alpha}\circ\mathbf{y})(\boldsymbol{\alpha}\circ\mathbf{y})'\mathbf{K}(\boldsymbol{\alpha}\circ\mathbf{y})}{(\boldsymbol{\alpha}\circ\mathbf{y})'\mathbf{K}(\boldsymbol{\alpha}\circ\mathbf{y})} \\ &\qquad +(\boldsymbol{\alpha}\circ\mathbf{y})'(\mathbf{K}+\mathbf{1}\mathbf{1}')(\boldsymbol{\alpha}\circ\mathbf{y}) \\ &= \max_{\mathbf{y}\in\mathcal{Y}} \quad (\boldsymbol{\alpha}\circ\mathbf{y})'((\lambda+1)\mathbf{K}+\mathbf{1}\mathbf{1}')(\boldsymbol{\alpha}\circ\mathbf{y}), \end{aligned} \tag{23}$$

which indicates that we only need to solve an optimization problem on $\mathbf{y}$. However, it is another MIP problem, and is difficult to be solved. Similar to [30], [32], we employ an approximated solution to (23) for finding the most violated $\mathbf{y}$. Specifically, we first rewrite (23) as:

$$\max_{\mathbf{y}\in\mathcal{Y}} \mathbf{y}'\left(\tilde{\mathbf{K}}\circ(\boldsymbol{\alpha}\boldsymbol{\alpha}')\right)\mathbf{y} = \max_{\mathbf{y}\in\mathcal{Y}} \|\sum_i y_i\alpha_i\tilde{\phi}(\mathbf{x}_i)\|^2 \tag{24}$$

where $\tilde{\mathbf{K}}=(\lambda+1)\mathbf{K}+\mathbf{1}\mathbf{1}'$ and $\tilde{\phi}(\cdot)$ is the feature mapping function induced by $\tilde{\mathbf{K}}$. Following [30], [32], we use the ℓ_∞-norm to approximate the ℓ_2-norm in (24), and the problem becomes

$$\begin{aligned} &\max_{\mathbf{y}\in\mathcal{Y}} \|\sum_i y_i\alpha_i\tilde{\phi}(\mathbf{x}_i)\|_\infty \\ &= \max_{\mathbf{y}\in\mathcal{Y}} \max_{j=1,\ldots,\tilde{d}} \left\{\sum_i y_i\alpha_i z_{ij}, \ -\sum_i y_i\alpha_i z_{ij}\right\} \\ &= \max_{j=1,\ldots,\tilde{d}} \left\{\max_{\mathbf{y}\in\mathcal{Y}} \sum_i y_i\alpha_i z_{ij}, \ \max_{\mathbf{y}\in\mathcal{Y}} -\sum_i y_i\alpha_i z_{ij}\right\} \end{aligned} \tag{25}$$

where z_{ij} is the j-th entry of the feature vector $\tilde{\phi}(\mathbf{x}_i) = [z_{i1},\ldots,z_{i\tilde{d}}]'$ with $\tilde{d}$ as the feature dimension.

To find the optimal $\mathbf{y}$, we first obtain $\tilde{\phi}(\mathbf{x})$ by using SVD decomposition on the kernel matrix $\tilde{\mathbf{K}}$, which is also known as the empirical kernel map [33]. Then we calculate $\alpha_i z_{ij}$ for each feature dimension and each sample. For the j-th dimension, we can respectively obtain two label vectors by a simple sorting operation to solve the two inner problems in (25). Specifically, we first sort the unlabeled samples in descending order according to $\alpha_i z_{ij}$. For $\max_{\mathbf{y}\in\mathcal{Y}}\sum_i y_i\alpha_i z_{ij}$, the optimal label vector can be obtained by setting the first $(\delta+n_u)/2$ unlabeled samples as positive and the remaining unlabeled samples as negative; similarly for $\max_{\mathbf{y}\in\mathcal{Y}}-\sum_i y_i\alpha_i z_{ij}$, the optimal label vector is obtained by setting the last $(\delta+n_u)/2$ unlabeled samples as positive and remaining unlabeled samples as negative. Finally, the most violated $\mathbf{y}$ is the label vector with the maximum objective value among these $2\tilde{d}$ label vectors.

Algorithm 2 Semi-supervised Heterogeneous Feature Augmentation

Input: Labeled source samples $\{(\mathbf{x}_i^s,y_i^s)|_{i=1}^{n_s}\}$, labeled target samples $\{(\mathbf{x}_i^t,y_i^t)|_{i=1}^{n_t}\}$, and unlabeled target samples $\{(\mathbf{x}_i^u,y_i^u)|_{i=1}^{n_u}\}$ with the unknown y_i^u's.

1: Train an SVM classifier f_0 by only using the labeled target samples.
2: Initialize the labeling candidate set $\mathcal{Y}=\{\mathbf{y}_1\}$ where $\mathbf{y}_1=[\mathbf{y}_s',\mathbf{y}_t',\tilde{\mathbf{y}}_u']'$ where $\tilde{\mathbf{y}}_u$ is a feasible label vector obtained by using the prediction from f_0.
3: Initialize the rank-one matrices set $\mathcal{M}=\{\mathbf{M}_1\}$ with $\mathbf{M}_1=\mathbf{h}_1\mathbf{h}_1'$ and $\mathbf{h}_1=\frac{1}{\sqrt{n}}\mathbf{1}_n$ and set $k=1$.
4: **repeat**
5: Set $k=k+1$.
6: Solve $\mathbf{d}$ and $\boldsymbol{\alpha}$ in (19) based on $\mathcal{Y}$ and $\mathcal{M}$ by using the existing MKL solver [19].
7: Find the violated $\mathbf{y}_k$ by solving (25) and obtain $\mathbf{M}_k=\mathbf{h}\mathbf{h}'$ where $\mathbf{h}=\frac{\mathbf{K}^{\frac{1}{2}}(\boldsymbol{\alpha}\circ\mathbf{y}_k)}{\|\mathbf{K}^{\frac{1}{2}}(\boldsymbol{\alpha}\circ\mathbf{y}_k)\|}$.
8: Set $\mathcal{M}=\mathcal{M}\bigcup\{\mathbf{M}_k\}$, $\mathcal{Y}=\mathcal{Y}\bigcup\{\mathbf{y}_k\}$.
9: **until** The objective converges.

Output: $\boldsymbol{\alpha}$, $\mathbf{d}$, $\mathcal{Y}$ and $\mathcal{M}$.

We summarize the algorithm for solving SHFA in Algorithm 2. We first initialize the set of rank-one PSD matrices $\mathcal{M}$ with $\mathbf{M}_1=\mathbf{h}_1\mathbf{h}_1'$, and also initialize the labeling candidate set $\mathcal{Y}$ by using a feasible label vector $\mathbf{y}_1$. To obtain $\tilde{\mathbf{y}}_u$ in $\mathbf{y}_1$, we first sort the unlabeled training samples in descending order according to the prediction of the classifier trained on the labeled target samples. Then $\tilde{\mathbf{y}}_u$ is obtained by setting the first $(\delta+n_u)/2$ unlabeled samples as positive and the remaining samples as negative. Next, we solve the MKL problem in (19) based on $\mathcal{Y}$ and $\mathcal{M}$. After that, we find a violated $\mathbf{y}$ and calculate the corresponding $\mathbf{M}=\mathbf{h}\mathbf{h}'$ where $\mathbf{h}=\frac{\mathbf{K}^{\frac{1}{2}}(\boldsymbol{\alpha}\circ\mathbf{y})}{\|\mathbf{K}^{\frac{1}{2}}(\boldsymbol{\alpha}\circ\mathbf{y})\|}$. We respectively add $\mathbf{y}$ and $\mathbf{M}$ into $\mathcal{Y}$ and $\mathcal{M}$ and solve the MKL problem again. This process is repeated until convergence. The time complexity can be analyzed similarly as in HFA, which is $O(n^3+TLn^{2.3})$ with $n=n_s+n_t+n_u$ being the total number of training samples[2].

2. The time complexity of the sorting operation for $\tilde{d}$ times in finding the optimal $\mathbf{y}$ is $\tilde{d}n_u\log(n_u)$, which is less than $n^2\log(n)$. When the number of training samples (*i.e.*, n) is large as in our experiments, it can be ignored when compared with the time complexity $O(Ln^{2.3})$ for solving the MKL problem.

After obtaining the optimal solution $\boldsymbol{\alpha}$, $\mathbf{d}$, $\mathcal{Y}$ and $\mathcal{M}$, we can predict any test sample $\mathbf{x}$ from the target domain by using the following target decision function:

$$f(\mathbf{x})=\sum_k d_k(\boldsymbol{\alpha}\circ\mathbf{y}_k)'\mathbf{K}^{\frac{1}{2}}(\lambda\mathbf{M}_k+\mathbf{I})\begin{bmatrix}\mathbf{O}_{n_s\times(n_t+n_u)}\\ \mathbf{K}_t^{-\frac{1}{2}}\end{bmatrix}\mathbf{k}_t+b, \quad (26)$$

where $\mathbf{k}_t=[k(\mathbf{x}_1^t,\mathbf{x}),\ldots,k(\mathbf{x}_{n_t}^t,\mathbf{x}),k(\mathbf{x}_1^u,\mathbf{x}),\ldots,k(\mathbf{x}_{n_u}^u,\mathbf{x})]'$ and $k(\mathbf{x}_i,\mathbf{x}_j)=\phi_t(\mathbf{x}_i)'\phi_t(\mathbf{x}_j)$ is a predefined kernel function for two data samples $\mathbf{x}_i$ and $\mathbf{x}_j$ in the target domain.

3.4 ℓ_p-MKL Extension

Recall that we have formulated our SHFA as an MKL problem in (20), in which the ℓ_1-norm constraint on the kernel coefficient vector $\mathbf{d}$ (*i.e.* $\|\mathbf{d}\|_1\leq 1$) is adopted. However, the optimization problem in (20) can be extended to more general ℓ_p-MKL by using ℓ_p-norm on $\mathbf{d}$ (*i.e.* $\|\mathbf{d}\|_p\leq 1$) as follows:

$$\begin{aligned}\min_{\mathbf{d},\mathbf{w}_k,\rho,\xi_i}\ & \frac{1}{2}\left(\sum_k\frac{\|\mathbf{w}_k\|^2}{d_k}+C\sum_{i=1}^n\nu_i(\xi_i)^2\right)-\rho \quad (27)\\ \text{s.t.}\ & \sum_k\mathbf{w}_k'\psi_k(\mathbf{x}_i)\geq\rho-\xi_i,\\ & \|\mathbf{d}\|_p\leq 1,\quad \mathbf{d}\geq 0,\end{aligned}$$

where $\mathbf{d}$, $\psi_k(\mathbf{x}_i)$ and ν_i are as the same as defined in (20). Thus, the original SHFA is a special case of (27) when $p=1$. The ℓ_p-MKL problem in (27) can also be solved by Algorithm 2. The only difference is that we solve an ℓ_p-MKL problem instead of ℓ_1-MKL in Step 6.

4 Experiments

In this section, we evaluate our proposed HFA and SHFA methods for object recognition, multilingual text categorization and cross-lingual sentiment classification. We focus on the heterogeneous domain adaptation problem with only one source domain and one target domain. For the supervised heterogeneous domain adaptation setting, we only utilize a limited number of labeled training samples in the target domain; for the semi-supervised heterogeneous domain adaptation setting, we additionally employ a large number of unlabeled training samples in the target domain.

4.1 Setup

Object recognition: We employ a recently released Office dataset[3] used in [16], [17] for this task. This dataset contains a total of 4106 images from 31 categories collected from three sources: `amazon` (object images downloaded from Amazon), `dslr` (high-resolution images taken from a digital SLR camera) and `webcam` (low-resolution images taken from a web camera). We follow the same protocols in the previous work [17]. Specifically, SURF features [34] are extracted for all the images. The images from `amazon` and `webcam` are clustered into 800 visual words by using k-means. After vector quantization, each image is represented as a 800 dimensional histogram feature. Similarly, we represent each image from `dslr` as a 600-dimensional histogram feature.

3. http://www.icsi.berkeley.edu/~saenko/projects.html

TABLE 1
Summarization of the Object Dataset Including 31 Categories

Domain	Source		Target
	`amazon`	`webcam`	`dslr`
# dimension	800	800	600
# total samples	2,813	795	498
# labeled samples per class	20	8	3
# unlabeled samples	–	–	405

In the experiments, `dslr` is used as the target domain, while `amazon` and `webcam` are considered as two individual source domains. We strictly follow the setting in [16], [17] and randomly select 20 (resp., 8) training images per category for the source domain `amazon` (resp., `webcam`). For the target domain `dslr` , 3 training images are randomly selected from each category, and the remaining `dslr` images are used for testing, which are also used as the unlabeled training samples in the semi-supervised setting. See Table 1 for a summarization of this dataset.

Text categorization: We use the Reuters multilingual dataset[4] [35], which is collected by sampling parts of the Reuters RCV1 and RCV2 collections. It contains about 11K newswire articles from 6 classes (*i.e.*, C15, CCAT, E21, ECAT, GCAT and M11) in 5 languages (*i.e.*, `English` , `French` , `German` , `Italian` and `Spanish`). While each document was also translated into the other four languages in this dataset, we do not use the translated documents in this work. All documents are represented by using the TF-IDF feature.

We take `Spanish` as the target domain in the experiment and use each of the other four languages as an individual source domain. For each class, we randomly sample 100 training documents from the source domain and m training documents from the target domain, where $m=5,10,15$ and 20. And the remaining documents in the target domain are used as the test data, among which $3,000$ documents are additionally sampled as the unlabeled training data in the semi-supervised setting. Note that the method in [15] cannot handle the original high dimensional TF-IDF features. In order to fairly compare our HFA method [15], for documents written in each language, we perform PCA based on the TF-IDF features with 60% energy preserved. We summarize this dataset in Table 2.

Sentiment Classification: We use the Cross-Lingual Sentiment (CLS) dataset[5] [36], which is an extended version of the Multi-Domain Sentiment Dataset [2] widely used for domain adaptation. It is collected from Amazon and contains about 800,000 reviews of three product categories: Books, DVDs and Music, and written in four languages: English, German, French, and Japanese. The English reviews were sampled from the Multi-Domain Sentiment Dataset and reviews in other languages are crawled from Amazon. For each category and each language, the dataset is officially partitioned into a training set, a test set and an unlabeled set, where the training set

4. http://multilingreuters.iit.nrc.ca/ReutersMultiLingualMultiView.htm

5. http://www.uni-weimar.de/cms/medien/webis/research/corpora/corpus-webis-cls-10.html

TABLE 2
Summarization of the Reuters Multilingual Dataset Including 6 Classes

Domain	Source				Target
	English	French	German	Italian	Spanish
# dim after PCA	1,131	1,230	1,417	1,041	807
# total samples	18,758	25,468	29,953	24,039	11,547
# labeled samples per class	100	100	100	100	5/10/15/20
# unlabeled samples	–	–	–	–	3,000

and test set consist of 2,000 reviews, and the numbers of unlabeled reviews vary from 9,000 to 170,000.

We take English as the source domain and each of the other three languages as an individual target domain in the experiment. We randomly sample 500 reviews from the training set of the source domain and 100 reviews from the training set of the target domain as the labeled data. The test set is the official test set for each category and each language. We also sample 1, 000 reviews from the unlabeled set as the unlabeled target training data. Similarly as for text categorization, we extracted the TF-IDF features and perform PCA with 60% energy preserved. The complete information of this dataset is summarized in Table 3.

Baselines: To evaluate our proposed methods, HFA and SHFA, we compare them with a number of baselines under two settings. The first setting (*i.e.*, the supervised HDA setting) is as the same as [18], in which there are sufficient labeled source samples and a limited number of labeled target samples. As the source and target data have different dimensions, they cannot be directly combined to train any classifiers for the target domain. So the baseline algorithms in this setting are listed as follows:

- **SVM_T:** It utilizes the labeled samples only from the target domain to train a standard SVM classifier for each category/class. This is a naive approach without considering the information from the source domain.
- **HeMap [13]:** It finds the projection matrices for a common feature subspace as well as learns the optimally projected data from both domains. We align the samples from different domains according to their labels. Since HeMap requires the same number of samples from the source and target domains, we randomly select $\min\{n_s, n_t\}$ samples from each domain for learning the subspace.
- **DAMA [15]:** It learns a common feature subspace by utilizing the class labels of the source and target training data for manifold alignment.
- **ARC-t [17]:** It uses the labeled training data from both domains to learn an asymmetric transformation metric between different feature spaces.

TABLE 3
Summarization of the Cross-Lingual Sentiment Dataset Including 3 Categories and 2 Classes

Domain	Source	Target		
	English	French	German	Japanese
# dim after PCA	715	964	929	874
# labeled samples	500	100	100	100
# unlabeled samples	–	1,000	1,000	1,000

In the second setting (*i.e.* the semi-supervised HDA setting), we addtionally employ the unlabeled samples in the target domain. To evaluate our SHFA, we report the results of one more baseline, transductive SVM (T-SVM) [25], which utilizes both the labeled data and unlabeled data to train the classifier. Note that the labeled samples in the source domain cannot be used in T-SVM because they have different features with the samples in the target domain. Moreover, all the above heterogenous domain adaptation methods [13], [15], [17] were designed for the supervised heterogeneous domain adaptation scenario, so it is unclear how to utilize the unlabeled target data to learn the projection matrices or transformation metric for these methods.

For HeMap and DAMA, after learning the projection matrices, we apply SVM to train the final classifiers by using the projected training data from both domains for a given category/class. For ARC-t, we construct the kernel matrix based on the learned asymmetric transformation metric, and then SVM is also applied to train its final classifier. The RBF kernel is used for all methods with the bandwidth parameter as the mean distance of all training samples. As we only have a very limited number of labeled training samples in the target domain, the cross-validation technique cannot be effectively employed to determine the optimal parameters. Therefore, we set the tradeoff parameter in SVM as the default value $C = 1$ for all methods. For our HFA and SHFA methods, we empirically fix the parameter λ as 100 in the vision application (*i.e.* the object recognition) and 1 in the text applications (*i.e.*, document classification and sentiment classification). And we also empirically set the weight of unlabeled data C_u in SHFA as 10^{-3} for all experiments. Moreover, we additionally report the results of our SHFA with the ℓ_p-MKL extension (see Section 3.4) where we empirically set $p = 1.5$ for all the datasets which generally achieves better results.

For other methods, we report their best results on the test data by varying their parameters in a wide range on each dataset. Specifically, we validate the parameters β in HeMap (see Equation (1) in [13]), μ in DAMA (see Theorem 1 in [15]) and λ in ARC-t (see Equation (1) in [17]) from $\{0.01, 0.1, 1, 10, 100\}$. For T-SVM, we validate the weight of unlabeled data C_u from $\{0.001, 0.01, 0.1, 1\}$ and the parameter s for the ramp loss from $[-0.9, 0]$ with the step size as 0.1. For both T-SVM and our SHFA, we set the parameter δ for the balance constraint on unlabeled samples using the prior information.

Evaluation metric: Following [17], for each method we measure the classification accuracy over all categories/classes on three datasets. We randomly sample the training data for a fixed number of times (*i.e.*, 20 for the

TABLE 4
Means and Standard Deviations of Classification Accuracies (%) of All Methods on the Object Dataset by Using 3 Labeled Training Samples Per Class from the Target Domain `dslr`

Methods	Source Domain	
	`amazon`	`webcam`
SVM_T	52.9 ± 3.1	
HeMap	42.8 ± 2.4	42.2 ± 2.6
DAMA	53.3 ± 2.3	53.2 ± 3.2
ARC-t	53.1 ± 2.4	53.0 ± 3.2
HFA	55.4 ± 2.9	54.3 ± 3.6
T-SVM	53.5 ± 2.0	
SHFA ($p = 1$)	$\mathbf{56.1 \pm 2.9}$	$\mathbf{55.1 \pm 3.4}$
SHFA ($p = 1.5$)	$\mathbf{56.6 \pm 2.4}$	$\mathbf{55.9 \pm 3.3}$

Results in boldface are significantly better than the others, judged by the t-test with a significance level at 0.05.

Office dataset as in [17], and 10 for the Reuters dataset and the Cross-Lingual Sentiment dataset) and report the mean classification accuracies of all methods over all rounds of experiments.

4.2 Classification Results

Object recognition: We report the means and standard deviations of classification accuracies for all methods on the Office dataset [16] in Table 4. From the results, we have the following observations in terms of the mean classification accuracy. SVM_T outperforms HeMap by using only 3 labeled training samples from the target domain. The explanation is that HeMap does not explicitly utilize the label information of the target training data to learn the feature mapping matrices. As a result, the learned common subspace cannot well preserve a similar data structure as in the original feature spaces of the source and target data, which results in poor classification performances. DAMA performs only slightly better that SVM_T, possibly due to the lack of strong manifold structure on this dataset. Both results of ARC-t implemented by ourselves and reported in [17] are only comparable with those of SVM_T, which shows that ARC-t is less effective for HDA on this dataset. Our HFA outperforms the other methods for both cases, which clearly demonstrate the effectiveness of our proposed method for HDA by learning with augmented features. Moreover, we also observe that it is beneficial to additionally use unlabeled data in the target domain to learn a more robust classifier. Specifically, when setting the parameter p in the ℓ_p-norm regularizer of ℓ_p-MKL as $p = 1$, our SHFA outperforms HFA on both cases when amazon and webcam are used as the source domain. When setting $p = 1.5$, the improvements of SHFA over HFA are 1.2% and 1.6%, respectively. SHFA also outperforms T-SVM which demonstrates we can train a better classifier by learning the transformation metric $\mathbf{H}$ to effectively exploit the source data in SHFA.

Text categorization: Table 5 shows the means and standard deviations of classification accuracies for all methods on the Reuters multilingual dataset [35] by using $m = 10$ and $m = 20$ labeled training samples per class from the target domain. We have the following observations in terms of the mean classification accuracy. SVM_T still outperforms HeMap. DAMA and ARC-t perform better than SVM_T for most cases. Our proposed HFA method is better than other supervised HDA methods on this dataset. For the semi-supervised setting, T-SVM is even worse than SVM_T although we have tuned all the parameters in a wide range. One possible explanation is that T-SVM cannot effectively utilize these target unlabeled data on this dataset. However, our SHFA can effectively handle the unlabeled data in the target domain and the performance improvements of SHFA ($p = 1.5$) over HFA are 3.5%, 3.2%, 3.1%, 3.1% and 1.1%, 1.1%, 1.0%, 1.1% for these four different source domains when $m = 10$ and $m = 20$, respectively.

We also plot the classification results of SVM_T, DAMA, ARC-t and our methods HFA and SHFA by using different numbers of target training samples per class (*i.e.*, $m = 5, 10, 15$ and 20) for each source domain in Fig. 2. We do not report the results of HeMap, as they are much worse than the other methods. From the results, the performances of all methods increase when using a larger m. And the two HDA methods DAMA and ARC-t generally achieve better mean classification accuracies than SVM_T except for the setting using `English` as the source domain. Our HFA method generally outperforms all other baseline methods according to mean classification accuracy. When using the unlabeled data in the target domain, our SHFA ($p = 1$) outperforms all existing HDA methods and the performance can be further improved when setting $p = 1.5$. We also observe that SHFA has large improvements over HFA when the number of labeled data in the target domain is very small (see $m = 5$ in Fig. 2). When the number of labeled data in the target domain increases, the unlabeled

TABLE 5
Means and Standard Deviations of Classification Accuracies (%) of All Methods on the Reuters Multilingual Dataset by Using 10 and 20 Labeled Training Samples Per Class from the Target Domain `Spanish`

Methods	#target labeled samples per class = 10				#target labeled samples per class = 20			
	`English`	`French`	`German`	`Italian`	`English`	`French`	`German`	`Italian`
SVM_T	66.6 ± 3.7				72.6 ± 2.3			
HeMap	54.7 ± 7.4	55.0 ± 9.4	58.0 ± 7.9	59.4 ± 3.7	65.7 ± 3.1	64.2 ± 4.2	64.6 ± 3.6	65.8 ± 2.3
DAMA	65.0 ± 2.9	66.9 ± 2.1	67.5 ± 2.1	68.5 ± 2.8	72.4 ± 2.4	72.8 ± 2.0	72.9 ± 2.3	73.3 ± 2.1
ARC-t	65.7 ± 2.7	66.9 ± 1.7	68.7 ± 2.9	67.9 ± 2.8	72.9 ± 2.0	73.5 ± 1.8	74.7 ± 1.6	74.0 ± 2.0
HFA	68.6 ± 2.3	69.5 ± 1.9	69.8 ± 2.7	69.8 ± 2.5	75.3 ± 1.7	75.7 ± 1.6	76.1 ± 1.5	75.8 ± 1.8
T-SVM	63.3 ± 3.8				69.2 ± 2.2			
SHFA ($p = 1$)	$\mathbf{70.7 \pm 2.3}$	$\mathbf{71.6 \pm 2.3}$	$\mathbf{72.0 \pm 2.9}$	$\mathbf{71.8 \pm 2.6}$	$\mathbf{75.8 \pm 1.7}$	$\mathbf{76.4 \pm 1.5}$	$\mathbf{76.8 \pm 1.4}$	$\mathbf{76.4 \pm 1.8}$
SHFA ($p = 1.5$)	$\mathbf{72.1 \pm 2.2}$	$\mathbf{72.7 \pm 2.1}$	$\mathbf{72.9 \pm 2.5}$	$\mathbf{72.9 \pm 2.2}$	$\mathbf{76.4 \pm 1.6}$	$\mathbf{76.8 \pm 1.4}$	$\mathbf{77.1 \pm 1.3}$	$\mathbf{76.9 \pm 1.6}$

Results in boldface are significantly better than the others, judged by the t-test with a significance level at 0.05.

TABLE 6
Means and Standard Deviations of Classification Accuracies (%) of All Methods on the Cross-Lingual Sentiment Dataset by Using 100 Labeled Training Samples from the Target Domain

Methods	Target Domain		
	German	French	Japanese
SVM_T	58.3 ± 2.8	60.4 ± 3.9	57.4 ± 2.0
HeMap	50.4 ± 0.6	49.8 ± 0.6	51.3 ± 1.0
DAMA	64.6 ± 1.9	65.7 ± 1.8	64.4 ± 1.8
ARC-t	58.3 ± 3.0	59.4 ± 4.3	57.5 ± 1.9
HFA	66.5 ± 2.2	66.9 ± 2.1	64.2 ± 2.5
T-SVM	65.6 ± 2.4	67.8 ± 2.3	63.9 ± 2.6
SHFA ($p = 1$)	$\mathbf{70.2 \pm 1.9}$	$\mathbf{70.5 \pm 1.1}$	$\mathbf{67.8 \pm 1.2}$
SHFA ($p = 1.5$)	$\mathbf{70.9 \pm 1.7}$	$\mathbf{71.6 \pm 1.2}$	$\mathbf{68.6 \pm 1.3}$

Results in boldface are significantly better than the others, judged by the t-test with a significance level at 0.05.

data in the target domain is less helpful, but SHFA is still better than HFA.

Sentiment classification: Table 6 summarizes the means and standard deviations of classification accuracies for all methods on the Cross-Lingual Sentiment dataset by using $m = 100$ labeled training samples in the target domain. As in each domain there are three categories (*i.e.*, Books, DVDs, Music), each mean accuracy in Table 6 is the mean accuracy over three categories and ten rounds. We have the following observations in terms of the mean classification accuracy. We observe that HeMap is worse than SVM_T which again indicates it cannot learn good feature mappings on this dataset. ARC-t is only comparable with SVM_T, and DAMA outperform SVM_T for all cases. Our HFA is better than other basline methods, except one exceptional case that HFA is worse than DAMA when using `Japanese` as the target domain. A possible explanation is the reviews in `Japanese` have good manifold correspondence with that in `English` . However, our HFA is still comparable with DAMA in this case. Moreover, we also have the similar observation as on the Office dataset and Reuters dataset, our SHFA achieves better results than HFA by additionally exploiting the unlabeled data in the target domain. With setting $p = 1$, the performance improvements of SHFA over HFA are 3.7%, 3.6% and 3.6% when using `German` , `French` and `Japanese` as the target domain, respectively. With setting $p = 1.5$, the performance improvements of SHFA over HFA are further increased to 4.4%, 4.7% and 4.4%, respectively.

4.3 Augmented Features *v.s.* Common Features

We defined two augmented feature mapping functions $\varphi_s(\mathbf{x}^s) = [(\mathbf{P}\mathbf{x}^s)', \mathbf{x}^{s'}, \mathbf{0}'_{d_t}]'$ and $\varphi_t(\mathbf{x}^t) = [(\mathbf{Q}\mathbf{x}^t)', \mathbf{0}'_{d_s}, \mathbf{x}^{t'}]'$ in (1) by concatenating the feature representation in the learnt common subspace (referred to as *common features* here) with the original features and zeros. However, our methods are also applicable by only using the common feature representations $\mathbf{P}\mathbf{x}^s$ and $\mathbf{Q}\mathbf{x}^t$ for the samples from source and target domains without using the original features and zeros. We take SHFA when setting $p = 1.5$ as an example to evaluate our work by only using the feature representation in the common space, which is referred as *SHFA_commFeat* . The results on the Reuters multilingual dataset are shown in Table 7, where we use the same settings as described in Section 4.1. We observe that SHFA_commFeat still outperforms the existing HDA methods HeMap, DAMA, ARC-t, and HFA on all settings in terms of mean accuracy, which clearly demonstrates the effectiveness of our proposed learning scheme. Moreover, SHFA using the augmented features are consistently better than SHFA_commFeat in terms of mean accuracy, which demonstrates it is beneficial to use our proposed new learning methods with the augmented features for HDA.

4.4 Performance Variations Using Different Parameters

We conduct experiments on the Reuters multilingual dataset to evaluate the performance variations of our SHFA by using different parameters (*i.e.*, λ, p, and C_u). As described in Section 4.1, we still use 100 labeled samples per class from the source domain, as well as 20 labeled samples per class and 3000 unlabeled samples from the target domain. The results of our SHFA ($p = 1$) and SHFA ($p = 1.5$) by using the default values $\lambda = 1$ and $C_u = 0.001$ have been reported in Table 5. To evaluate the performance variations, at each time we vary one parameter and set the other parameters as the default values (*i.e.*, $\lambda = 1$, $C_u = 0.001$, and $p = 1.5$). The means of classification accuracies by varying different parameters on the four settings are plotted in Fig. 3.

From Fig. 3, we observe that our SHFA is quite stable to these parameters in certain ranges. Specifically, by changing λ in the range of $[0.01, 100]$, the performances of SHFA ($p = 1.5$) vary within 1% in terms of mean classification accuracy, which are still better than these baseline methods reported in Table 5. Also, by changing the parameter p of

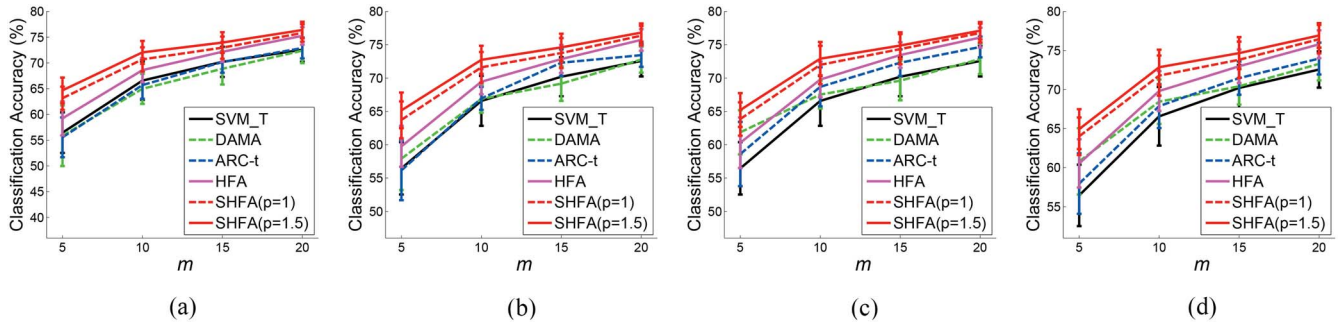


Fig. 2. Classification accuracies of all methods with respect to different number of target training samples per class (*i.e.*, $m = 5, 10, 15$ and 20) on the Reuters multilingual dataset. `Spanish` is considered as the target domain, and in each subfigure the results are obtained by using one language as the source domain. (a) English. (b) French. (c) German. (d) Italian.

TABLE 7
Means and Standard Deviations of Classification Accuracies (%) of Our SHFA ($p = 1.5$) and SHFA_commFeat ($p = 1.5$) on the Reuters Multilingual Dataset

	#target labeled samples per class = 10				#target labeled samples per class = 20			
	English	French	German	Italian	English	French	German	Italian
SHFA ($p = 1.5$)	72.1 ± 2.2	72.7 ± 2.1	72.9 ± 2.5	72.9 ± 2.2	76.4 ± 1.6	76.8 ± 1.4	77.1 ± 1.3	76.9 ± 1.6
SHFA_commFeat ($p = 1.5$)	70.8 ± 2.2	71.6 ± 2.2	72.0 ± 2.8	71.8 ± 2.4	75.9 ± 1.7	76.5 ± 1.5	76.8 ± 1.4	76.5 ± 1.9

the ℓ_p-norm in the range of $\{1, 1.2, 1.5, 2\}$, we observe that with a larger p, SHFA can achieve better results. However, our initial experiments show that a larger p usually leads to a slower convergence. We empirically set $p = 1.5$ as the default value in all our experiments for a good tradeoff between the effectiveness and efficiency. Moreover, we also evaluate our SHFA ($p = 1.5$) by varying C_u in the range of $[10^{-5}, 10^{-1}]$. The parameter C_u controls the weights of unlabeled samples. Intuitively, it should not be too large because the inferred labels for the unlabeled samples are not accurate, which is also supported by our experiment as shown in Fig. 3(c). While we empirically set $C_u = 10^{-3}$ in all our experiments, we observe that SHFA ($p = 1.5$) using a larger value (*i.e.*, $C_u = 10^{-2}$) can achieve better results on this dataset. However, the performances drop dramatically when setting it to a much larger value (say, $C_u = 10^{-1}$). Nevertheless, our SHFA is generally stable and better than these baseline methods reported in Table 5 when setting $C_u \in [10^{-5}, 10^{-2}]$. For the domain adaptation problem, it is difficult to perform cross-validation to choose the optimal parameters, because we usually only have a limited number of labeled samples in the target domain. We would like to study how to automatically decide the optimal parameters in the future.

4.5 Time Analysis

We take the Cross-Lingual Sentiment dataset as an example to evaluate the running time of all methods. The experimental setting is as the same as described in Section 4.1. The average per class training times of all methods are reported in Table 8. All the experiments are performed on a workstation with Xeon 3.33 GHz CPU and 16 GB of RAM. From Table 8, we observe that the supervised methods (*i.e.*, SVM_T, HeMap, DAMA, ARC-t and HFA) are generally faster than the semi-supervised methods (*i.e.*, T-SVM and our SHFA), because additional unlabeled samples are used in the semi-supervised methods. SVM_T is very fast because it only utilizes the labeled training data from the target domain. HeMap is fast since it only needs to solve the eigen-decomposition problem in a very small size due to the limited number of labeled samples in the target domain. The training time of HFA is comparable to that of DAMA and ARC-t. For the semi-supervised methods, we observe that our SHFA ($p = 1$) is faster than T-SVM, and SHFA ($p = 1.5$) is slower than SHFA ($p = 1$). Moreover, the warm start strategy can be used to further accelerate our SHFA , which will be studied in the future.

5 Conclusion and Future Work

We have proposed a new method called Heterogeneous Feature Augmentation (HFA) for heterogeneous domain adaptation. In HFA, we augment the heterogeneous features from the source and target domains by using two newly proposed feature mapping functions, respectively. With the augmented features, we propose to find the two projection matrices for the source and target data and simultaneously learn the classifier by using the standard SVM with the hinge loss in both linear and nonlinear cases. Then we convert the learning problem into an MKL formulation which is convex and thus the global solution can be guaranteed. Moreover, to utilize the abundant unlabeled data in the target domain, we further extend our HFA method to semi-supervised HFA (SHFA). Promising results have demonstrated the effectiveness of HFA and SHFA on three real-world datasets for object recognition, text classification and sentiment classification.

In the future, we will investigate how to incorporate other kernel learning methods such as [37] into our heterogeneous feature augmentation framework. Another important direction is to analyze the generalization bound for heterogeneous domain adaptation.

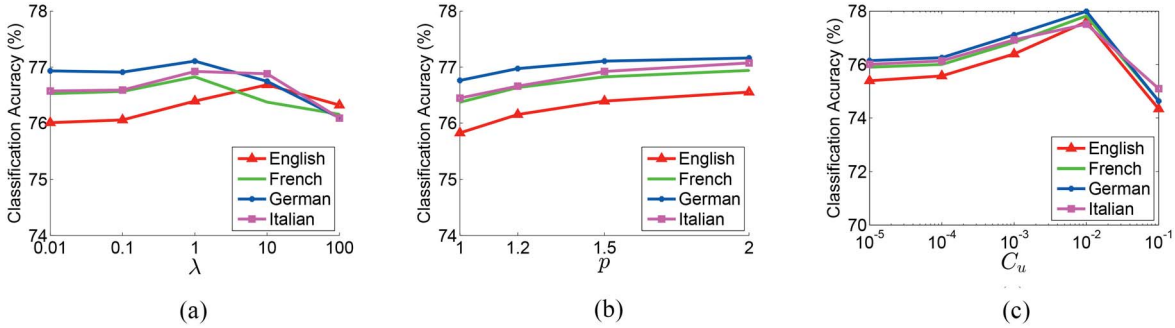


Fig. 3. Performances of our SHFA using different parameters on the Reuters multilingual dataset. (a) Performances w.r.t. λ. (b) Performances w.r.t. p in l_p-norm. (c) Performance w.r.t. C_u.

TABLE 8
Average Per Class Training Time (in Seconds) Comparisons of All Methods on the Cross-Lingual Sentiment Dataset

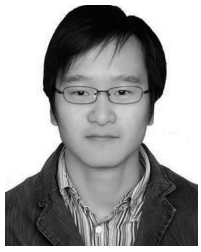
	SVM_T	HeMap	DAMA	ARC-t	HFA	T-SVM	SHFA ($p = 1$)	SHFA ($p = 1.5$)
Training Time	0.01	0.32	9.68	1.98	3.16	20.70	18.13	43.34

ACKNOWLEDGMENTS

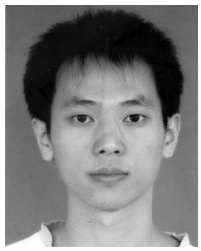
This work is supported by the Singapore MOE Tier 2 Grant (ARC42/13).

REFERENCES

[1] J. Blitzer, R. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Proc. EMNLP*, Sydney, NSW, Australia, 2006.
[2] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification," in *Proc. 45th ACL*, Prague, Czech Republic, 2007.
[3] H. Daumé, III, "Frustratingly easy domain adaptation," in *Proc. ACL*, 2007.
[4] L. Duan, D. Xu, I. W. Tsang, and J. Luo, "Visual event recognition in videos by learning from web data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1667–1680, Sep. 2012.
[5] L. Duan, I. W. Tsang, and D. Xu, "Domain transfer multiple kernel learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 465–479, Mar. 2012.
[6] L. Duan, D. Xu, and S.-F. Chang, "Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach," in *Proc. CVPR*, Providence, RI, USA, 2012, pp. 1338–1345.
[7] L. Chen, L. Duan, and D. Xu, "Event recognition in videos by learning from heterogeneous web sources," in *Proc. CVPR*, Portland, OR, USA, 2013, pp. 2666–2673.
[8] W. Dai, Y. Chen, G.-R. Xue, Q. Yang, and Y. Yu, "Translated learning: Transfer learning across different feature spaces," in *Proc. NIPS*, 2009.
[9] Q. Yang, Y. Chen, G.-R. Xue, W. Dai, and Y. Yu, "Heterogeneous transfer learning for image clustering via the social web," in *Proc. ACL/IJCNLP*, Singapore, 2009.
[10] Y. Zhu *et al.*, "Heterogeneous transfer learning for image classification," in *Proc. AAAI*, 2011.
[11] B. Wei and C. Pal, "Cross-lingual adaptation: An experiment on sentiment classifications," in *Proc. ACL*, 2010.
[12] P. Prettenhofer and B. Stein, "Cross-language text classification using structural correspondence learning," in *Proc. ACL*, 2010.
[13] X. Shi, Q. Liu, W. Fan, P. S. Yu, and R. Zhu, "Transfer learning on heterogenous feature spaces via spectral transformation," in *Proc. ICDM*, Sydney, NSW, Australia, 2010.
[14] M. Harel and S. Mannor, "Learning from multiple outlooks," in *Proc. 28th ICML*, Bellevue, WA, USA, 2011.
[15] C. Wang and S. Mahadevan, "Heterogeneous domain adaptation using manifold alignment," in *Proc. 22nd IJCAI*, 2011.
[16] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. ECCV*, Heraklion, Greece, 2010.
[17] B. Kulis, K. Saenko, and T. Darrell, "What you saw is not what you get: Domain adaptation using asymmetric kernel transforms," in *Proc. CVPR*, Providence, RI, USA, 2011.
[18] L. Duan, D. Xu, and I. W. Tsang, "Learning with augmented features for heterogeneous domain adaptation," in *Proc. 29th ICML*, Edinburgh, Scotland, U.K., 2012, pp. 711–718.
[19] M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien, "ℓ_p-norm multiple kernel learning," *JMLR*, vol. 12, pp. 953–997, Mar. 2011.
[20] M. Dudik, Z. Harchaoui, and J. Malick, "Lifted coordinate descent for learning with trace-norm regularization," in *Proc. 15th AISTATS*, La Palma, Spain, 2012.
[21] P. V. Gehler and S. Nowozin, "Infinite kernel learning," Max Planck Institute for Biological Cybernetics, Tech. Rep. 178, 2008.
[22] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods*. Cambridge, MA, USA: MIT Press, 1999, pp. 185–208.
[23] M. Tan, L. Wang, and I. W. Tsang, "Learning sparse SVM for feature selection on very high dimensional datasets," in *Proc. 27th ICML*, Haifa, Israel, 2010.
[24] A. Mutapcic and S. Boyd, "Cutting-set methods for robust convex optimization with pessimizing oracles," *Optim. Meth. Softw.*, vol. 24, no. 3, pp. 381–406, Jun. 2009.
[25] R. Collobert, F. Sinz, J. Weston, and L. Bottou, "Large scale transductive SVMs," *JMLR*, vol. 7, pp. 1687–1712, Dec. 2006.
[26] X. Zhu, "Semi-supervised learning literature survey," University of Wisconsion-Madison, Tech. Rep. 1530, 2005.
[27] L. Duan, D. Xu, and I. W. Tsang, "Domain adaptation from multiple sources: A domain-dependent regularization approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 3, pp. 504–518, Mar. 2012.
[28] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.
[29] H. Daumé, III, A. Kumar, and A. Saha, "Co-regularization based semi-supervised domain adaptation," in *Proc. NIPS*, 2010.
[30] Y.-F. Li, I. W. Tsang, J. T. Kwok, and Z.-H. Zhou, "Tighter and convex maximum margin clustering," in *Proc. AISTATS*, Clearwater Beach, FL, USA, 2009.
[31] W. Li, L. Duan, D. Xu, and I. W. Tsang, "Text-based image retrieval using progressive multi-instance learning," in *Proc. ICCV*, Barcelona, Spain, 2011, pp. 2049–2055.
[32] W. Li, L. Duan, I. W. Tsang, and D. Xu, "Batch mode adaptive multiple instance learning for computer vision tasks," in *Proc. CVPR*, Providence, RI, USA, 2012, pp. 2368–2375.
[33] B. Schölkopf *et al.*, "Input space versus feature space in kernel-based methods," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 1000–1017, Sep. 1999.
[34] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," in *Proc. ECCV*, Graz, Austria, 2006.
[35] M. Amini, N. Usunier, and C. Goutte, "Learning from multiple partially observed views – An application to multilingual text categorization," in *Proc. NIPS*, 2009.
[36] P. Prettenhofer and B. Stein, "Cross-language text classification using structural correspondence learning," in *Proc. 48th ACL*, Uppsala, Sweden, 2010.
[37] B. Kulis, M. Sustik, and I. Dhillon, "Low-rank kernel learning with Bregman matrix divergences," *JMLR*, vol. 10, pp. 341–376, Feb. 2009.



Wen Li received the B.S. and M.Eng. degrees from the Beijing Normal University, Beijing, China, in 2007 and 2010, respectively. Currently, he is pursuing the Ph.D. degree with the School of Computer Engineering, Nanyang Technological University, Singapore. His current research interests include ambiguous learning, domain adaptation, and multiple kernel learning.



Lixin Duan received the B.E. degree from the University of Science and Technology of China, Hefei, China, in 2008 and the Ph.D. degree from the Nanyang Technological University, Singapore, in 2012. Currently, he is a research scientist with the Institute for Infocomm Research, Singapore. He was a recipient of the Microsoft Research Asia Fellowship in 2009 and the Best Student Paper Award at the IEEE Conference on Computer Vision and Pattern Recognition 2010. His current research interests include transfer learning, multiple instance learning, and their applications in computer vision and data mining.

Dong Xu (M'07–SM'13) received the B.E. and Ph.D. degrees from the University of Science and Technology of China, Hefei, China, in 2001 and 2005, respectively. While pursuing the Ph.D. degree, he was with Microsoft Research Asia, Beijing, China, and the Chinese University of Hong Kong, Shatin, Hong Kong, for more than two years. He was a post-doctoral research scientist with Columbia University, New York, NY, USA, for one year. Currently, he is an associate professor with Nanyang Technological University, Singapore. His current research interests include computer vision, statistical learning, and multimedia content analysis. He was the co-author of a paper that won the Best Student Paper Award in the IEEE International Conference on Computer Vision and Pattern Recognition in 2010.

Ivor W. Tsang is an Australian Future Fellow and Associate Professor with the Centre for Quantum Computation & Intelligent Systems (QCIS), at the University of Technology, Sydney (UTS). Before joining UTS, he was the Deputy Director of the Centre for Computational Intelligence, Nanyang Technological University, Singapore. He received his PhD degree in computer science from the Hong Kong University of Science and Technology in 2007. He has more than 100 research papers published in refereed international journals and conference proceedings, including JMLR, TPAMI, TNN/TNNLS, NIPS, ICML, UAI, AISTATS, SIGKDD, IJCAI, AAAI, ACL, ICCV, CVPR, ICDM, etc. In 2009, Dr Tsang was conferred the 2008 Natural Science Award (Class II) by Ministry of Education, China, which recognized his contributions to kernel methods. In 2013, Dr Tsang received his prestigious Australian Research Council Future Fellowship for his research regarding Machine Learning on Big Data. Besides this, he had received the prestigious IEEE Transactions on Neural Networks Outstanding 2004 Paper Award in 2006, and a number of best paper awards and honors from reputable international conferences, including the Best Student Paper Award at CVPR 2010, the Best Paper Award at ICTAI 2011, the Best Poster Award Honorable Mention at ACML 2012, the Best Student Paper Nomination at the IEEE CEC 2012, and the Best Paper Award from the IEEE Hong Kong Chapter of Signal Processing Postgraduate Forum in 2006. He was also awarded the Microsoft Fellowship 2005, and the ECCV 2012 Outstanding Reviewer Award.