Feature Space Independent Semi-Supervised Domain Adaptation via Kernel Matching

Min Xiao and Yuhong Guo

Abstract—Domain adaptation methods aim to learn a good prediction model in a label-scarce target domain by leveraging labeled patterns from a related source domain where there is a large amount of labeled data. However, in many practical domain adaptation learning scenarios, the feature distribution in the source domain is different from that in the target domain. In the extreme, the two distributions could differ completely when the feature representation of the source domain is totally different from that of the target domain. To address the problems of substantial feature distribution divergence across domains and heterogeneous feature representations of different domains, we propose a novel feature space independent semi-supervised kernel matching method for domain adaptation in this work. Our approach learns a prediction function on the labeled source data while mapping the target data points to similar source data points by matching the target kernel matrix to a submatrix of the source kernel matrix based on a Hilbert Schmidt Independence Criterion. We formulate this simultaneous learning and mapping process as a non-convex integer optimization problem and present a local minimization procedure for its relaxed continuous form. We evaluate the proposed kernel matching method using both cross domain sentiment classification tasks of Amazon product reviews and cross language text classification tasks of Reuters multilingual newswire stories. Our empirical results demonstrate that the proposed kernel matching method consistently and significantly outperforms comparison methods on both cross domain classification problems with homogeneous feature spaces and cross domain classification problems with heterogeneous feature spaces.

Index Terms—Domain adaptation, kernel matching, heterogeneous feature spaces

1 Introduction

In many real-world applications, collecting labeled training data is an expensive and time-consuming process. Meanwhile, statistical machine learning models need sufficient labeled samples in order to be well trained for accurate prediction. Recently, domain adaptation has been proposed and received great attention to learn a robust classifier for the target domain where labeled patterns are scarce or unavailable by exploiting label information from other related source domains where there are plenty of labeled instances [1], [2], [3], [4]. Domain adaptation learning strategies are prevailingly needed and have demonstrated a great success in various practical applications, such as sentiment classification [5], [6], [7], visual concept classification [8], [9], [10], Wi-Fi localization [11], [12], part-of-speech tagging [13], [14], gene name recognition [15], and cross language text classification [16], [17], [18].

One typical issue needs to be addressed in many practical domain adaptation learning settings is that the feature distribution of the source domain substantially differs from that of the target domain. A key challenge raised in such settings is the feature divergence problem. That is, one cannot find supports in the source domain for some critical discriminative features of the target domain while the discriminative features of the source domain are not informative or do not appear in the target domain. This is very common in the natural language processing area, where different genres often use very different vocabularies to describe similar concepts. For example, in sentiment classification data of product reviews, terms like "harmonious" or "melodic" are positive indicators in the Music domain, but not in the Books domain; similarly, terms like "noise" or "yelling" are negative indicators in the Music domain, but not in the Books domain. In this situation, most domain adaptation algorithms seek to bridge the gap between the two domains by re-weighting source instances [19], [20], self-labeling target instances [6], [21], inducing new feature representations [5], [7], [14], etc. These methods nevertheless are limited to situations where the two domains share the same feature representation space.

In many real-world cross domain learning tasks the feature representation space in the source domain can be completely different from that in the target domain, which is known as heterogeneous domain adaptation and has been recently addressed in the literature [16], [22], [23]. Domain adaptation with heterogeneous feature representations exists in many practical applications such as cross language text classification where documents from different language domains are represented by words in different languages [16], [18] and text-aided image classification where the source domain has word features and the target domain has visual features [10]. Some works have been proposed to address heterogeneous feature spaces by finding auxiliary resources to link different feature representations [10], [17]. A few other works have been proposed to directly address the issue of heterogeneous feature spaces by assuming that there is a shared latent feature representation between two domains. They propose to induce the shared representation by spectral mapping [22], feature augmentation [16], or exploiting label correspondence [23].

• *The authors are with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122. E-mail: {minxiao, yuhong}@temple.edu.*

Manuscript received 1 Sept. 2013; revised 29 Apr. 2014; accepted 12 June 2014. Date of publication 25 July 2014; date of current version 5 Dec. 2014. Recommended for acceptance by T. Scheffer.
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 no. 10.1109/TPAMI.2014.2343216

In this work, we propose a semi-supervised kernel matching method to address cross domain learning from a novel perspective. The proposed approach is feature space independent and can perform domain adaptation over both homogeneous and heterogeneous feature spaces across domains. We assume the source domain contains a much larger number of labeled instances and unlabeled instances than the target domain. Instead of focusing on bridging the cross domain feature divergence, we employ kernelized representations for instances in each domain to eliminate the feature representation divergence issue. Specifically, we first produce two kernel matrices with a given kernel function, one over the instances in the source domain and one over the instances in the target domain. Then we learn a prediction function from the labeled source instances while mapping each target instance to a source instance by matching the target kernel matrix to a submatrix of the source kernel matrix based on a Hilbert Schmidt Independence Criterion (HSIC). The labeled instances in the target domain work as pivot points for class separation: Each labeled instance in the target domain is guaranteed to be mapped into a source instance with the same class label; the unlabeled target instances are expected to be mapped into corresponding source instances with the same labels as well guided by the labeled pivot points through the function of kernel affinity measures between instances. Moreover, we perform semi-supervised learning by minimizing the training loss on labeled instances in both domains while using graph Laplacian regularization terms to incorporate geometric information from unlabeled instances. Each graph Laplacian regularizer reflects the intrinsic structure of the instance distribution in each domain. We formulate this simultaneous semi-supervised learning and mapping process as a non-convex integer optimization problem and present a local minimization procedure for its relaxed continuous formulation. We empirically evaluate the proposed kernel matching method with extensive cross domain sentiment classification tasks on Amazon product reviews, where the source feature distribution is substantially different from the target feature distribution, and with cross language text classification tasks on Reuters multilingual newswire stories, where there is no overlap between the source feature representation space and the target feature representation space. Our experimental results suggest the proposed approach significantly outperforms the feature representation based cross domain sentiment classification approaches and a number of heterogeneous domain adaptation techniques for cross language text classification.

1.1 Notation and Setting

In this paper, we consider cross domain prediction model learning in two domains, a source domain $\mathcal{D}_S$ and a target domain $\mathcal{D}_T$. In the source domain, we have l_s labeled instances $\{(X_i^s, \mathbf{y}_i^s)\}_{i=1}^{l_s}$ and u_s unlabeled instances $\{(X_i^s)\}_{i=l_s+1}^{n_s}$, where $n_s = l_s + u_s$. In the target domain, we have l_t labeled instances $\{(X_i^t, \mathbf{y}_i^t)\}_{i=1}^{l_t}$ and u_t unlabeled instances $\{(X_i^t)\}_{i=l_t+1}^{n_t}$, where $n_t = l_t + u_t$. Here we assume that X^s is a $n_s \times d_s$ instance matrix whose ith row X_i^s is the ith instance, $\mathbf{y}^s$ is a $l_s \times 1$ label vector and $\mathbf{y}_i^s$ denotes its ith entry. Similarly, X^t is a $n_t \times d_t$ instance matrix and $\mathbf{y}^t$ is a $l_t \times 1$ label vector. Moreover, we assume the source domain has plenty labeled and unlabeled instances such that l_s is *much larger* than l_t and n_s is *much larger* than n_t.

2 Related Work

Domain adaptation has recently been popularly studied in machine learning and related fields. In this section, we review related works on both homogeneous domain adaptation where the two domains share the same feature representation space, and heterogeneous domain adaptation where the feature representation spaces of the two domains are different.

2.1 Homogeneous Domain Adaptation

Many domain adaptation approaches have been developed in the literature to cope with the feature distribution divergence between the source domain and the target domain. For example, covariate shift methods attempt to bridge the gap between domains by putting more weights on source instances that are in the dense region of the target domain [19], [20]. These methods however perform poorly for highly divergent domains where important features of the target instances are not supported in the source domain distribution.

Self-labeling adaptation methods, on the other hand, focus on target instances. They train an initial model on labeled source instances and then use it to label target instances. The newly labeled target instances will be used to update the initial model through self-training [15] or co-training [6], [21]. Their performances greatly depend on the initial model trained from source labeled data and they are not well suitable for highly divergent domains either.

In addition, a number of domain adaptation algorithms address the domain divergence issue directly from feature representation learning perspective. For example, the structural correspondence learning methods [5], [13], [24] and coupled subspace methods [14] seek to learn shared representations by exploiting the large amount of unlabeled data from both the source and target domains. Their efficacy nevertheless depends on the existence of a certain amount of pivot features that are used to induce shared feature representations. The work in [25] proposed to first select a subset of source instances similar to the target instances and add them to the target domain, and then use an unsupervised domain adaptation method to learn domain-invariant features. The work in [26], [27] proposed to simultaneously map features from the target domain to the source domain via a linear transformation and perform classification model training on the labeled source and target instances based on feature transformation. Instead of projecting features from one domain to another, the work in [28] proposed to use a shared projection matrix to transform features from two domains to a shared space. Some other work has also used deep learning algorithms to extract high-level deep representations for domain adaptation [29].

2.2 Heterogeneous Domain Adaptation

For heterogeneous domain adaptation over domains with different feature representation spaces, a number of approaches have been developed in the literature, including heterogeneous spectral mapping [22], feature mapping [30],

feature projection and transformation [8], [9], [16], manifold alignment [23], and exploiting auxiliary resources [10], [18].

In [22], a heterogeneous spectral mapping method is proposed to tackle heterogeneous domain adaptation problems. It projects both domains into a common latent space by using spectral transformation, in which the projected instances from both domains have similar distributions. This learning technique was shown to be effective on drug efficacy prediction and image classification tasks. However, HeMap does not use discriminative label information for inducing the shared feature subspace and the learned latent low dimensional representation is not optimized for the discriminative classification task. In [30], heterogeneous domain adaptation was tackled by sequentially performing feature selection and feature mapping based on HSIC. It first selects features in each domain based on the HSIC between the instance feature kernel matrix and the instance label kernel matrix and then maps the selected features across domains. The work in [16] proposed a heterogeneous feature augmentation (HFA) method to address heterogeneous domain adaptation. It first projects the data from both domains into a common subspace and then uses two feature mapping functions to augment the data. It learns the projection matrices by using the standard SVM on the augmented feature representations. The efficacy of this approach has been empirically demonstrated on cross-lingual text classification tasks on Reuters multilingual data set and object recognition tasks on images from Amazon, DSLR and Webcam domains. Similarly, [8] and [9] learn asymmetric feature transformations for both domains by performing nonlinear metric learning with similarity and dissimilarity constraints.

Different from these feature mapping and transformation approaches, [23] proposes a manifold alignment based approach to tackle the challenge of heterogeneous feature representations, which exploits instance labels to align the manifolds of instances. It projects the source and target domains to a new latent space, while simultaneously separating the instances with different labels, matching instances with the same labels from both domains, and preserving the topology of each input domain.

Moreover, there is a group of approaches that employ auxiliary resources for heterogeneous domain adaptation. For example, [10] used auxiliary image and text data to construct a latent subspace for image low-level features based on collective matrix factorization. Dai et al. [18] proposed a translated learning method, which exploits a translation tool between the source and target domains. These approaches however have limited applicability due to the fact that auxiliary resources are task specific and hence time-consuming to obtain.

2.3 Discussions

The feature space independent approach we developed in this work is different from the feature representation learning based methods. Instead of explicitly mapping features to a shared representation space, we directly map all the target training instances to a subset of the source training instances based on the HSIC with kernel representations. HSIC has been employed in the literature for instance matching in non-domain adaptation scenarios [31], [32]. The direct instance correspondence mapping is beneficial in many domain adaptation learning scenarios such as text classifications. For text classification, the feature space dimension is usually much larger than the number of instances, projecting features needs much more computational effort than mapping instances. Moreover, the kernel matching method does not need any pivot features or feature correspondence information, it only needs a very small set of labeled pivot instances from the target domain. Our empirical results show the proposed method is effective on both homogeneous domain adaptation for sentiment classification and heterogeneous domain adaptation for cross language document classification.

3 Semi-Supervised Kernel Matching for Domain Adaptation

In this section, we present a semi-supervised kernel matching approach to address domain adaptation in a transductive manner by exploiting a large amount of data from a source domain. Our primary idea is to extend kernelized object matching into cross domain learning. Similar to many semi-supervised methods developed in the literature [33], [34], we have one basic manifold assumption in both domains: if two points x_1, x_2 are close in the intrinsic geometry of the marginal distribution $\mathcal{P}_X$, then the conditional distributions $\mathcal{P}(Y \mid x_1)$ and $\mathcal{P}(Y \mid x_2)$ are similar. We utilize properties of Reproducing Kernel Hilbert Spaces (RKHS) to construct our semi-supervised learning objective which has three types of components: a kernel matching criterion, prediction losses, and graph Laplacian regularizers.

3.1 Kernel Matching Criterion

The kernel matching criterion is developed to map each instance in the target domain into one instance in the source domain, according to their geometric similarities expressed in kernel matrices. In particular, we conduct instance mapping by maximizing a Hilbert Schmidt Independence Criterion over the kernel matrix of the target instances and the kernel matrix of the mapped source instances. HSIC [35] originally measures the independence between given random variables based on the eigenspectrum of covariance operators in Reproducing Kernel Hilbert Spaces. Quadrianto [36] proposed an unsupervised kernel sorting method to match object pairs from two sources of observations by maximizing their dependence based on the HSIC. In this work we exploit this criterion in a semi-supervised manner to map pairs of instances to each other without exact correspondence requirement (since we do not have two sets of parallel objects in two domains) but ensuring class separation. We require each labeled instance in the target domain is guaranteed to be mapped into a source instance with the same class label. The labeled instances in the target domain thus perform as *pivot points* for class separation. Through the kernel affinity measures between instances, we expect unlabeled target instances to be most likely mapped into corresponding source instances with same labels as well, following the similar pivot points.

Specifically, we construct two kernel matrices in the two domains $K^s = \Phi(X^s)\Phi(X^s)^\top$ and $K^t = \Phi(X^t)\Phi(X^t)^\top$, where Φ is a feature map function that maps feature vectors

into a Reproducing Kernel Hilbert Space. Then the kernel matching can be conducted by

$$\begin{aligned} &\max_{M} (n_t-1)^{-2} tr(MK^sM^\top HK^tH) \\ &\text{s.t. } M \in \{0,1\}^{n_t,n_s};\ M\mathbf{1}=\mathbf{1};\ M(1:l_t,1:l_s)\mathbf{y}^s=\mathbf{y}^t, \end{aligned} \tag{1}$$

where $H = I - \frac{1}{n_t}\mathbf{1}\mathbf{1}^\top$, I denotes a $n_t \times n_t$ identity matrix, and $\mathbf{1}$ denotes column vectors with all 1 entries. The objective function here is a biased estimate of HSIC. It is known to be sensitive to diagonal dominance. To address this problem, we can modify the biased HSIC objective in (1) to reduce bias by removing the main diagonal terms of the kernel matrices, as suggested in [36], which leads to the following problem

$$\begin{aligned} &\max_{M} (n_t-1)^{-2} tr(M\widehat{K}^sM^\top H\widehat{K}^tH) \\ &\text{s.t. } M \in \{0,1\}^{n_t,n_s};\ M\mathbf{1}=\mathbf{1};\ M(1:l_t,1:l_s)\mathbf{y}^s=\mathbf{y}^t, \end{aligned} \tag{2}$$

where $\widehat{K}^s_{ij} = K^s_{ij}(1-\delta_{ij})$ and $\widehat{K}^t_{ij} = K^t_{ij}(1-\delta_{ij})$ are the kernel matrices with main diagonal terms removed.

3.2 Prediction Losses

Supervised learning is conducted on the labeled instances. We propose to learn a prediction function $f : x \longrightarrow y$ on the labeled instances in the source domain, while minimizing the training losses not only on the labeled source instances, but also on the labeled target instances that have mapped prediction values. That is, giving the mapping matrix M, we conduct supervised training as below

$$\min_{f\in\mathcal{H}} \sum_{i=1}^{l_s} \ell\big(f\big(X^s_i\big),\mathbf{y}^s_i\big) + \eta \sum_{i=1}^{l_t} \ell\big(M(i,:)f(X^s),\mathbf{y}^t_i\big) + \beta\|f\|^2_{\mathcal{H}}, \tag{3}$$

where $\ell(\cdot,\cdot)$ is a loss function, $\mathcal{H}$ is the Reproducing Kernel Hilbert Space associated with the kernel function that produces the kernel matrix K^s; the RKHS norm $\|f\|^2_{\mathcal{H}}$ measures the complexity of f function. Penalizing the RKHS norm imposes smoothness conditions on possible solutions. By the Representer Theorem, the solution to this minimization problem can be written in terms of kernel matrix

$$f\big(X^s_i\big) = \sum_{j=1}^{n_s} \boldsymbol{\alpha}_j K^s(j,i), \quad f = K^s\boldsymbol{\alpha}, \tag{4}$$

where $\boldsymbol{\alpha}$ is a $n_s \times 1$ coefficient parameter vector. Here we used a more general form of representation to take the unlabeled instances into account as well. The RKHS norm of f can then be re-expressed as

$$\|f\|^2_{\mathcal{H}} = \boldsymbol{\alpha}^\top K^s\boldsymbol{\alpha}. \tag{5}$$

Then using a squared loss function, the minimization Problem (3) can be rewritten as

$$\min_{\boldsymbol{\alpha}} \|\mathbf{y}^s - J^sK^s\boldsymbol{\alpha}\|^2 + \beta\boldsymbol{\alpha}^\top K^s\boldsymbol{\alpha} + \eta\|\mathbf{y}^t - J^tMK^s\boldsymbol{\alpha}\|^2, \tag{6}$$

where J^s is an $l_s \times n_s$ matrix whose first l_s columns form an identity matrix and all other entries are 0s; J^t is an $l_t \times n_t$ matrix whose first l_t columns form an identity matrix and all other entries are 0s.

3.3 Graph Laplacian Regularization

In addition to the kernel matching criterion and supervised prediction losses presented above, we consider to incorporate information about the geometric structures of the marginal distributions, $\mathcal{P}^s_X$ and $\mathcal{P}^t_X$, in each domain, based on the manifold assumption [33], [34]. Specifically, we will incorporate the following graph Laplacian terms which approximate manifold regularization

$$\gamma_s\|f\|^2_{G_s} + \gamma_t\|Mf\|^2_{G_t}. \tag{7}$$

The graphs G_s and G_t denote the affinity graphs constructed on the source domain and target domain respectively. These Laplacian terms work as a smoothness functional to ensure the f function changes smoothly not only on the graph that approximates the manifold in the source distribution, but also on the graph that approximates the manifold in the target distribution.

Let $G = <V, E>$ be a weighted adjacency graph on n vertices. The graph Laplacian L of G is defined as $L = D - W$, where W is the edge weight matrix and D is a diagonal matrix such that $D_{ii} = \sum_j W_{ji}$. It is easy to see that L is a symmetric and positive semidefinite matrix. Following this procedure, the graph Laplacian matrices L^s and L^t associated with G_s and G_t can be generated correspondingly. The graph Laplacian regularization terms in (7) can then be rewritten as

$$\begin{aligned} &\gamma_s\|f\|^2_{G_s} + \gamma_t\|Mf\|^2_{G_t} \\ &\quad= \gamma_s f^\top L^s f + \gamma_t f^\top M^\top L^t Mf \\ &\quad= \gamma_s\boldsymbol{\alpha}^\top K^sL^sK^s\boldsymbol{\alpha} + \gamma_t\boldsymbol{\alpha}^\top K^sM^\top L^tMK^s\boldsymbol{\alpha}. \end{aligned} \tag{8}$$

3.4 Semi-Supervised Kernel Matching

Finally, by combing the three components in (2), (6) and (8) together, we obtain the following joint optimization problem for semi-supervised kernel matching

$$\begin{aligned} \min_{M,\boldsymbol{\alpha}}\ & \|\mathbf{y}^s - J^sK^s\boldsymbol{\alpha}\|^2 + \eta\|\mathbf{y}^t - J^tMK^s\boldsymbol{\alpha}\|^2 \\ &+ \beta\boldsymbol{\alpha}^\top K^s\boldsymbol{\alpha} - \mu tr(M\widehat{K}^sM^\top H\widehat{K}^tH) \\ &+ \gamma_s\boldsymbol{\alpha}^\top K^sL^sK^s\boldsymbol{\alpha} + \gamma_t\boldsymbol{\alpha}^\top K^sM^\top L^tMK^s\boldsymbol{\alpha} \\ \text{s.t. } & M \in \{0,1\}^{n_t,n_s};\ M\mathbf{1}=\mathbf{1};\ J^tMJ^{s\top}\mathbf{y}^s=\mathbf{y}^t. \end{aligned} \tag{9}$$

The goal of this optimization problem is to learn a kernel mapping matrix M as well as a kernelized prediction model parameterized by $\boldsymbol{\alpha}$ to minimize the *regularized* training losses in both domains in a semi-supervised manner. It performs domain adaptation in a transductive manner.

4 Optimization Algorithm

The optimization Problem (9) we formulated above is an integer optimization problem. Moreover, the objective function is not jointly convex in M and $\boldsymbol{\alpha}$. Let $h(M,\boldsymbol{\alpha})$ denote the objective function of (9). We first relax the integer constraints to obtain a continuous relaxation

$$\begin{aligned} &\min_{M,\boldsymbol{\alpha}}\ h(M,\boldsymbol{\alpha}) \\ &\text{s.t. } 0 \le M \le 1;\ M\mathbf{1}=\mathbf{1};\ J^tMJ^{s\top}\mathbf{y}^s=\mathbf{y}^t. \end{aligned} \tag{10}$$

This relaxation also largely relaxes the strict requirement of matching each target instance into one unique source instance. Then we propose a first order local minimization algorithm to solve the relaxed non-convex optimization Problem (10).

First we treat (10) as a non-smooth minimization problem over M, and re-express the optimization problem as

$$\begin{aligned} &\min_{M} \quad g(M) \\ &\text{s.t. } 0 \leq M \leq 1;\ M\mathbf{1} = \mathbf{1};\ J^t M J^{s\top}\mathbf{y}^s = \mathbf{y}^t. \end{aligned} \tag{11}$$

for

$$g(M) = \min_{\boldsymbol{\alpha}} h(M, \boldsymbol{\alpha}). \tag{12}$$

Note $\boldsymbol{\alpha}$ can be viewed as a function of M, i.e., $\boldsymbol{\alpha}(M)$. For a given M, a closed-form solution of $\boldsymbol{\alpha}(M)$ can be obtained by setting the partial derivative of $h(M, \boldsymbol{\alpha})$ with respect to $\boldsymbol{\alpha}$ to 0,

$$\boldsymbol{\alpha}^*(M) = Q^{-1}(K^s J^{s\top}\mathbf{y}^s + \eta K^s M^\top J^{t\top}\mathbf{y}^t) \tag{13}$$

where

$$\begin{aligned} Q = {} & K^s J^{s\top} J^s K^s + \eta K^s M^\top J^{t\top} J^t M K^s + \beta K^s \\ & + \gamma_s K^s L^s K^s + \gamma_t K^s M^\top L^t M K^s. \end{aligned} \tag{14}$$

We then solve the minimization Problem (11) using a first order local minimization algorithm with backtracking line search. The algorithm is an iterative procedure, starting from a feasible initial point $M^{(0)}$. At the $(k+1)$th iteration, we approximate the objective function $g(M)$ in the close neighborhood of point $M^{(k)}$ using the first order Taylor series expansion

$$g(M) \approx g(M^{(k)}) + tr(G(M^{(k)})^\top (M - M^{(k)})), \tag{15}$$

where $G(M^{(k)})$ denotes the gradient of $g(M)$ at point $M^{(k)}$ (i.e. the gradient of $h(M, \alpha^*(M^{(k)}))$

$$\begin{aligned} G(M^{(k)}) = {} & 2\eta J^{t\top} J^t M^{(k)} K^s \boldsymbol{\alpha}\boldsymbol{\alpha}^\top K^s - 2\eta J^{t\top}\mathbf{y}^t \boldsymbol{\alpha}^\top K^s \\ & - 2\mu H \widehat{K}^t H M^{(k)} \widehat{K}^s + 2\gamma_t L^t M^{(k)} K^s \boldsymbol{\alpha}\boldsymbol{\alpha}^\top K^s. \end{aligned} \tag{16}$$

Given the gradient at point $M^{(k)}$, we minimize the local linearization (15) to seek a feasible descending direction of M regarding the constraints,

$$\begin{aligned} &\widehat{M} = \arg\min_{M} \quad tr(G(M^{(k)})^\top M) \\ &\text{s.t. } 0 \leq M \leq 1;\ M\mathbf{1} = \mathbf{1};\ J^t M J^{s\top}\mathbf{y}^s = \mathbf{y}^t. \end{aligned} \tag{17}$$

The optimization problem above is a standard convex linear programming and can be solved using a standard optimization toolbox. The update direction for the $(k+1)$th iteration can be determined as

$$D = \widehat{M} - M^{(k)}. \tag{18}$$

We then employ a standard backtracking line search [37] to seek an optimal step size ρ^* to obtain $M^{(k+1)}$ along the direction D in the close neighborhood of $M^{(k)}$:

$$M^{(k+1)} = M^{(k)} + \rho^* D. \tag{19}$$

The line search procedure will guarantee the $M^{(k+1)}$ leads to an objective value no worse than before in terms of the original objective function $g(M) = h(M, \boldsymbol{\alpha}^*(M))$. The overall algorithm for the minimization Problem (11) is given in Algorithm 1.

Algorithm 1. Local Optimization Procedure

Input: $\mathbf{y}^s, \mathbf{y}^t, K^s, K^t; M^{(0)}, \epsilon; \mu, \beta, \gamma_s, \gamma_t, \eta;\ MaxIters$
Output: M^*
Initialize $k = 0, NoChange = 0$;
Repeat
1. Compute gradient $G(M^{(k)})$ according to Eq. (16).
2. Solve the linear optimization (17) to get $\widehat{M}$.
3. Compute descend direction D using Eq. (18).
4. Conduct backtracking line search to obtain $M^{(k+1)}$.
5. **if** $\|M^{(k+1)} - M^{(k)}\|^2 < \epsilon$ **then** $NoChange = 1$.
6. $k = k + 1$.

Until $NoChange = 1$ or $k > MaxIters$
$M^* = M^{(k)}$.

After obtaining the local optimal solution M^*, we need to round it back to an integer solution satisfying the linear constraints in (9). We use a simple heuristic greedy procedure to conduct the rounding. The procedure is described in Algorithm 2. The quality of the local solution we obtained depends greatly on the initial $M^{(0)}$. In our experiments, we used 100 random initializations to pick the best feasible initial $M^{(0)}$ that minimizes the training objective.

Algorithm 2. Heuristic Greedy Rounding Procedure

Input: $M \in R^{n_t \times n_s}, \mathbf{y}^s, \mathbf{y}^t$.
Output: $M^* \in (0,1)^{n_t \times n_s}$.
Initialize: Set M^* as a $n_t \times n_s$ matrix with all 0s.
for $k = 1$ **to** l_t **do**
  Find indices $\mathbf{d}$, s.t. $\mathbf{y}^s(\mathbf{d}) = \mathbf{y}^t(k)$.
  Compute $v = \arg\max_{v \in \mathbf{d}}(M(k, v))$.
  Set $M^*(k, v) = 1$, $M(k, :) = -\inf$.
end for
for $k = l_t$ **to** n_t **do**
  Identify the largest value $v = \max(M(:))$.
  Identify the indices (d, r) of v in M.
  Set $M^*(d, r) = 1, M(d, :) = -\inf$.
end for

5 Experiments

To evaluate the empirical performance of the proposed kernel matching approach, we conducted extensive experiments on two real world data sets, Amazon product reviews and Reuters multilingual newswire stories, for homogeneous and heterogeneous domain adaptation respectively.

5.1 Homogeneous Domain Adaptation

We conducted homogeneous domain adaptation experiments on cross domain sentiment classification tasks over

TABLE 1
Information of Three Domain Pairs

Domain Pairs	Vocabulary Size	A-distance
Books versus DVD	10,370	0.7221
Books versus Music	8,006	0.8562
DVD versus Music	8,825	0.7831

Amazon product reviews. In these experiments, the source domain and the target domain share the same feature representation space, but there is a feature distribution divergence across domains.

5.1.1 Data Set

We used the Amazon data set[1] constructed in [17], which contains Amazon product reviews in three domains (Books, DVD and Music). Each domain of the data set contains 2,000 positive reviews and 2,000 negative reviews, each of which is represented as a term-frequency (TF) vector.

We used the English reviews to construct six source-target ordered domain pairs based on the original three domains: B2D (Books to DVD), D2B (DVD to Books), B2M (Books to Music), M2B (Music to Books), D2M (DVD to Music), and M2D (Music to DVD). For each pair of domains, we built an unigram vocabulary from combined reviews in both domains. We then preprocessed the data by removing features that appear less than twice in either domain, computing TF-IDF features and normalizing each attribute value into $[0, 1]$.

The divergence of each pair of domains can be measured using A-distance [2]. We adopted the same method in [38] to compute approximate A-distance values. We first trained a linear classifier to separate source and target domains with all instances from both domains. The average per-instance hinge-loss for this classifier subtracted from 1 was used as an estimator of the proxy A-distance. It is a number in the interval of [0, 1] with larger values indicating larger domain divergence. Table 1 presents the vocabulary size and proxy A-distance for each pair of domains we used in the experiments. We can see that all three pairs of domains present substantial divergences.

5.1.2 Experimental Setup

We denote the proposed semi-supervised kernel matching method as *SSKMDA1*. We compared the proposed approach with three baseline methods: *TargetOnly* is a baseline method which trains a SVM classifier on labeled data in the target domain; *SourceOnly* is a baseline method that trains a SVM classifier on labeled data in the source domain; *SourceTarget* is a baseline method which trains a SVM classifier on labeled data in both domains. We also compared our approach with two domain adaptation learning methods, the co-regularization based semi-supervised domain adaptation method (*EA++*) [7] and the *Coupled Subspace* leaning method [14]. The *EA++* method exploits labeled data from both domains and unlabeled data from the target domain, and the *Coupled Subspace* learning method exploits all data from the two domains. An alternative version of our proposed semi-supervised kernel matching method for domain adaptation, *SSKMDA2*, is also tested in our experiments, which is obtained from *SSKMDA1* by replacing the unbiased HSIC component in Eq. (2) with the unbiased HSIC used in [39].

For the *Coupled Subspace* method, we used the software package provided by [14] with the same parameters used in [14]. For our proposed approach, we used Gaussian kernels, $K(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\|\mathbf{x}_1 - \mathbf{x}_2\|^2/(2\sigma^2))$ with $\sigma = 0.05$. We used K-Nearest-Neighbor with $K = 20$ and binary weights to construct the Laplacian graphs G_s and G_t for the source and target domains respectively. For the tradeoff parameters, we chose η from {0.5, 1, 2, 5}, β from {0.01, 0.045, 0.1, 0.5}, γ_s from {0.01, 0.05, 0.1, 0.5, 1}, γ_t from {0.01, 0.05, 0.1, 0.5, 1}, and μ from {0.5, 1, 2, 5, 10}. We ran the first task *B2D* for three times based on different random selections of the labeled target instances and chose the parameter setting with the highest average classification accuracy over three runs, which yields $\eta = 1, \beta = 0.045, \gamma_s = 0.05, \gamma_t = 0.05$ and $\mu = 5$.

5.1.3 Classification Results

The semi-supervised learning was conducted in a transductive manner. We performed training with l_s labeled source instances and u_s unlabeled source instances as well as l_t labeled target instances and u_t unlabeled target instances. The performance of the trained classifier is evaluated on the u_t unlabeled target instances.

In the experiments, we used $l_s = 1{,}390, u_s = 10$, $n_s = l_s + u_s = 1{,}400$, $l_t = 10$, $u_t = 990$, and $n_t = l_t + u_t = 1{,}000$. We randomly chose n_s instances from the source domain, with the first l_s instances labeled and the rest unlabeled. Similarly, we randomly choose n_t instances from the target domain, with the first l_t instances labeled and the rest unlabeled. All comparison methods are tested using the same data. Each experiment was repeated 10 times. The average test accuracies and standard deviations for all six experiments are reported in Table 2.

We can see that *SourceOnly* performs much better than *TargetOnly* for all the six cross domain sentiment classification tasks. This is reasonable since there are much more labeled instances in the source domain $(l_s = 1{,}390)$ than the target domain $(l_t = 10)$, and 10 labeled target instances are far from sufficient to produce a robust sentiment classifier in the target domain. This suggests that labeled data from the auxiliary source domain is useful for the target domain. One can also observe that *SourceTarget* outperforms *SourceOnly* in most of the tasks, but the improvements are small. The largest performance gain for *SourceTarget* over *SourceOnly* in terms of accuracy is about 1.08 percent on the B2D task. Moreover, *SourceTarget* performs worse than *SourceOnly* on the D2B task. Those results demonstrate that simply combining labeled data from both domains is far from ideal to obtain an effective prediction model in the target domain. On the other hand, all the domain adaptation methods demonstrate improvements over the three baselines on the six tasks. The improvements produced by *EA++* are small. *Coupled Subspace* produces larger performance gains over the three baselines, but its performance is not as good as the proposed approaches (two

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

TABLE 2
Classification Accuracies (Percent) for Six Cross Domain Sentiment Classification Tasks on Amazon Product Reviews

Tasks	TargetOnly	SourceOnly	SourceTarget	EA++	Coupled Subspace	SSKMDA1	SSKMDA2
B2D	52.40 ± 0.96	71.77 ± 0.43	72.85 ± 0.65	73.63 ± 0.61	74.36 ± 0.47	79.27 ± 0.32	$\mathbf{79.34} \pm 0.36$
D2B	51.23 ± 0.52	72.27 ± 0.50	72.15 ± 0.46	72.85 ± 0.52	76.03 ± 0.55	$\mathbf{80.04} \pm 0.26$	79.93 ± 0.23
B2M	52.43 ± 0.75	71.16 ± 0.57	71.30 ± 0.57	71.44 ± 0.54	76.75 ± 0.54	$\mathbf{78.14} \pm 0.46$	77.97 ± 0.50
M2B	51.23 ± 0.52	68.25 ± 1.30	68.90 ± 0.39	69.38 ± 0.65	75.70 ± 0.52	$\mathbf{77.47} \pm 0.28$	77.34 ± 0.28
D2M	52.43 ± 0.75	71.86 ± 0.39	72.44 ± 0.46	72.49 ± 0.37	77.80 ± 0.45	$\mathbf{79.70} \pm 0.34$	79.63 ± 0.29
M2D	52.40 ± 0.96	72.12 ± 0.45	72.89 ± 0.50	73.44 ± 0.49	74.59 ± 0.42	$\mathbf{78.54} \pm 0.32$	77.85 ± 0.37

TABLE 3
Time Analysis (Seconds/Minutes) on the Cross Domain Sentiment Classification Tasks

	TargetOnly	SourceOnly	SourceTarget	EA++	Coupled Subspace	SSKMDA1
Training Time	2.26 s	4.62 s	4.11 s	18.20 s	1.51 m	40.30 m

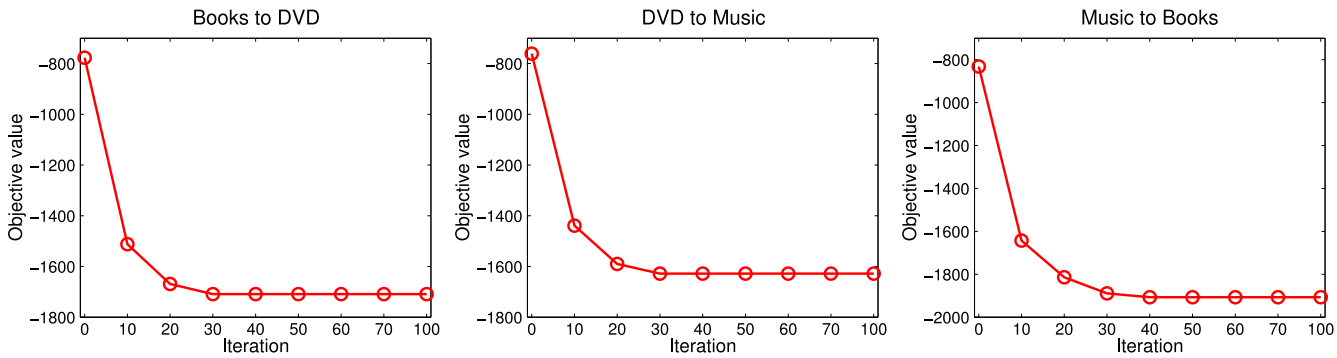

Fig. 1. The convergence process of *SSKMDA1* training over three cross domain sentiment classification tasks.

versions). Both versions of our proposed domain adaptation method perform consistently and significantly better than all the other comparison methods over all six tasks. The two versions of the proposed approach achieved very similar results, though *SSKMDA1* performs slightly better than *SSKMDA2*. Comparing to *Coupled Subspace* method, *SSKMDA1* increases the accuracy by more than 5 percent on the task of B2D, by about 4 percent on the task of D2B and M2D, and by about 2 percent on the tasks B2M, M2D and D2M.

Moreover, we recorded the average training time of all the comparison methods on the first task *B2D*. The results are collected on a workstation with 10 GB RAM and Xeon 3.20 GHz CPU, and reported in Table 3. We can see that though the proposed method *SSKMDA1* runs slower than the other comparison methods due to its iterative optimization procedure, its training time is affordable for solving problems with a reasonable size.

5.1.4 Convergence

Next, we studied the convergence property of the training algorithm of the proposed kernel matching approach *SSKMDA1* using three tasks, B2D, D2M, M2B. We used the same experimental setting as before. We plot the objective function value of Eq. (11) with respect to the number of iterations of the training algorithm into Fig. 1. We can see that the objective function value converges within 40 iterations. We have similar observations for the other three tasks.

5.1.5 Classification Results versus Label Complexity

As introduced before, in our proposed approach the labeled target instances perform as pivot points for kernel matching. Then we may ask: how sensitive is the proposed approach to the number of pivot points? To answer this question and investigate the label complexity of the proposed approach in the target domain, we conducted another set of experiments with varying number of labeled target instances. We tested a set of values $l_t \in \{10, 50, 100, 200, 500\}$, with the other settings same as before.

We report the average classification results over 10 runs in Fig. 2 for the two versions of the proposed approach and the four others: *TargetOnly*, *SourceTarget*, *EA++* and *Coupled Subspace*. Note that the varying number of the labeled target instances does not affect the performance of *SourceOnly* and we hence did not consider it. From Fig. 2 we can see that both versions of the proposed approach consistently outperform all the other methods over all six domain adaptation tasks and across the set of different l_t values. Moreover, the increasing of the number of labeled target instances leads to significant performance improvements for the *TargetOnly* method. The performances of *SourceTarget*, *EA++* and *Couple Subspace* vary in a small degree due to the fact that there are many more labeled source instances, and the labeled source instances and the labeled target instances have to work out a compatible solution between them. The performances of the proposed *SSKMDA1* and *SSKMDA2* are quite stable across different l_t values. This suggests the proposed

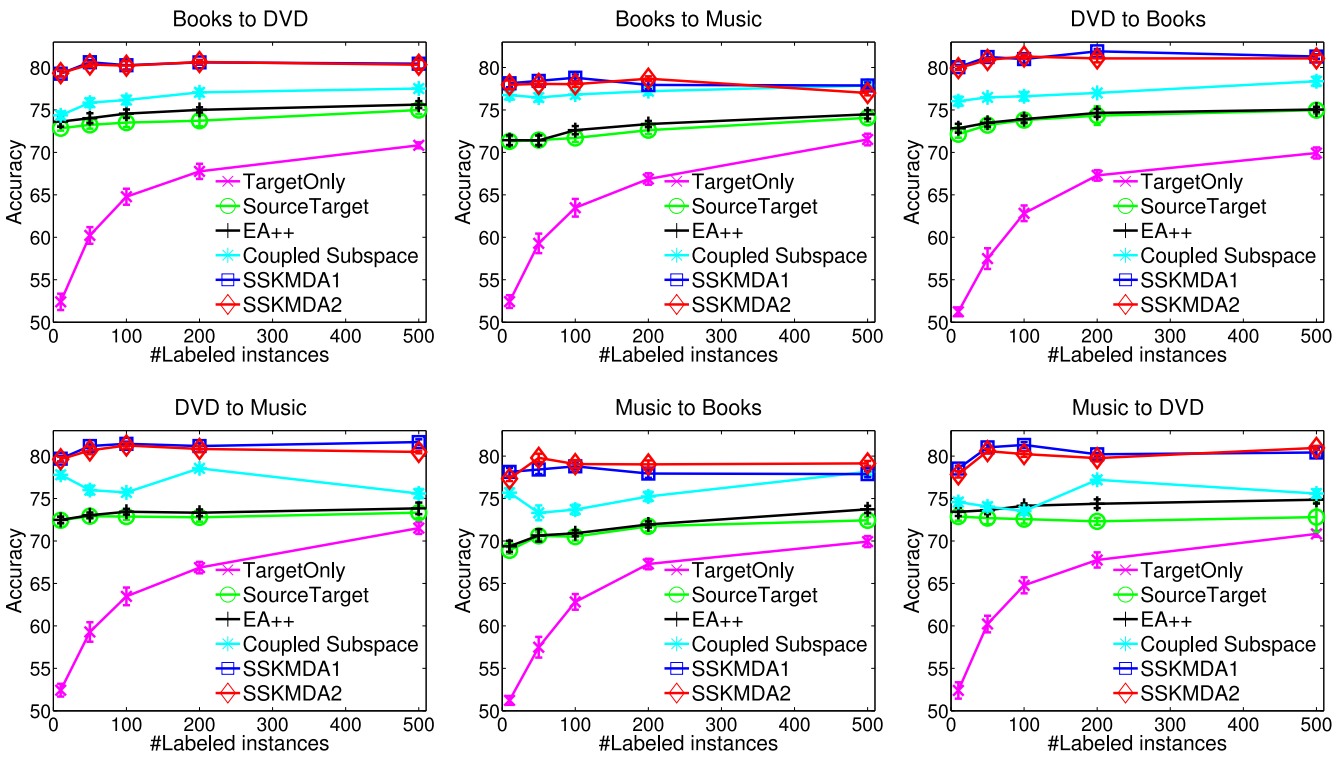

Fig. 2. The accuracy versus the number of labeled target instances for six cross domain sentiment classification tasks.

method only requires a very small number of pivot points to produce a good prediction model for the target instances. The empirical label complexity of the proposed approach in the target domain is very small from this perspective. All these results suggest our proposed method is more effective to handle domain divergence than the feature representation based methods and require much less labeled data from the target domain.

5.1.6 Parameter Sensitivity Analysis

The proposed semi-supervised domain adaptation method contains a number of tradeoff parameters between the model components: $\mu, \eta, \beta, \gamma_s, \gamma_t$. We experimentally studied how does each of these tradeoff parameters affect the prediction performance of the proposed approach in the target domain using three cross domain sentiment classification tasks: *B2D*, *D2M* and *M2B*. The HSIC based kernel matching component in Eq. (2) plays a very important role in the proposed kernel matching approach. We first investigated how does the tradeoff parameter μ associated with the HSIC component affect the performance of the proposed approach. We used the same experimental setting as before with $l_s = 1{,}390$, $u_s = 10, l_t = 10$, and $u_t = 990$. We used the same values as introduced before for all the other tradeoff parameters while conducting experiments with a set of different μ values, $\mu \in \{1, 2, 5, 10, 20\}$. The average classification results over 10 runs are reported in Fig. 3. We can see that though the performance of the *SSKMDA1* method varies a little bit with different μ values, the changes are very small. This suggests that the *SSKMDA1* is not very sensitive to μ within the range of values we studied.

Similarly, we conducted analysis for the other tradeoff parameters. The parameter η balances the prediction losses on labeled data between the two domains. In previous experiments, we empirically set it to be 1 since we treat the two prediction losses equally. Here, we adjusted this value to place more weight on either the source domain or the target domain. We conducted tests with a set of values $\eta \in \{0.1, 1, 2, 5\}$ and reported the prediction accuracies over 10 runs. From Fig. 3, we can see that our approach is not that sensitive to η. Within the considered range of values, there is only a small change (less than 1 percent) in terms of the classification accuracy for all the three tasks. The parameters γ_s and γ_t weight the graph Laplacian regularizers on the source and the target domains respectively. We tested $\gamma_s \in \{0.01, 0.05, 0.1, 1\}$ and $\gamma_t \in \{0.01, 0.05, 0.1, 1\}$ and present the results in Fig. 3. These results show that our approach is insensitive to both γ_s and γ_t and achieves good performance within the considered range of values. For the weight parameter β over the RKHS norm regularizer, we tested $\beta \in \{0.01, 0.045, 0.1, 0.5\}$. From Fig. 3, we can see the performance of *SSKMDA1* varies in a small degree with respect to different values of β. We observed similar results for the other three cross domain classification tasks as well.

5.1.7 Impact of the Laplacian Regularizers

For the proposed approach (*SSKMDA1*), there are two Laplacian regularization component, one for each domain. We further investigated the performance of the proposed approach by completely dropping the Laplacian regularization components from our model. Specifically we considered three cases: (1) *LapSource-* we removed the Laplacian regularzer in the target domain and only considered the source Laplacian regularizer. (2) *LapTarget-* we only considered the target Laplacian regularizer and removed the source part. (3) *LapNo-* we removed both Laplacian regularizers.

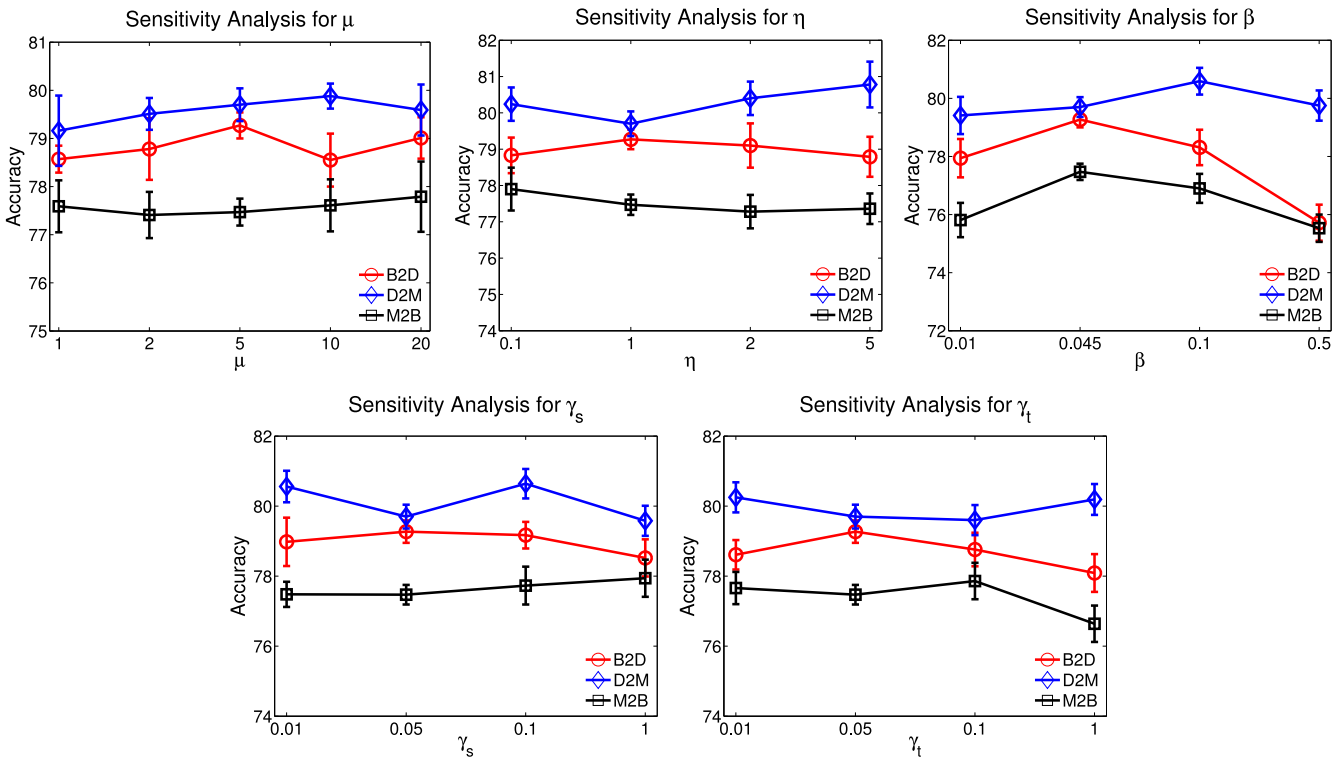

Fig. 3. Performance of *SSKMDA1* with respect to tradeoff parameters for three cross domain sentiment classification tasks.

For all cases, we used 1,390 labeled instances and 10 unlabeled instances in the source domain, and used 10 labeled and 990 unlabeled instances in the target domain. The classification results are reported on the unlabeled 990 instances from the target domain. For all approaches, we chose μ from $\{0.5, 1, 2, 5, 10\}$, η from $\{0.5, 1, 2, 5\}$, β from $\{0.01, 0.045, 0.1, 0.5\}$, chose γ_s from $\{0.01, 0.05, 0.1, 0.5, 1\}$ for *LapSource* and chose γ_t from $\{0.01, 0.05, 0.1, 0.5, 1\}$ for *LapTarget*. We used the same procedure to select parameter values for all approaches on the first task *B2D*. After parameter selection, we reran each task 10 times based on different random selections of the labeled target instances. The empirical results are reported in Table 4. For comparison, we also included the results for the full version of our approach, *SSKMDA1*. From Table 4, we can see that by removing either one of the Laplacian regularizers, the performance of the model degrades, comparing to the full model, *SSKMDA1*. Moreover, if we continue to remove the remaining Laplacian regularizer for *LapSource* or *LapTarget*, the model performance will continue to degrade. These results suggest that the Laplacian regularization terms are important components of the proposed semi-supervised learning model.

TABLE 4
Classification Accuracy Results (Percent) for the Empirical Investigation over the Impact of the Laplacian Regularizers

Tasks	LapNo	LapSource	LapTarget	SSKMDA1
B2D	77.55 ± 0.35	78.14 ± 0.24	78.34 ± 0.34	**79.27** ± 0.32
D2B	78.48 ± 0.32	78.79 ± 0.30	78.97 ± 0.32	**80.04** ± 0.26
B2M	76.91 ± 0.40	77.16 ± 0.46	77.44 ± 0.51	**78.14** ± 0.46
M2B	76.14 ± 0.42	76.75 ± 0.39	76.89 ± 0.44	**77.47** ± 0.28
D2M	78.07 ± 0.30	78.54 ± 0.26	78.73 ± 0.38	**79.70** ± 0.34
M2D	76.89 ± 0.30	77.13 ± 0.31	77.56 ± 0.27	**78.54** ± 0.32

5.2 Heterogeneous Domain Adaptation

We conducted heterogeneous domain adaptation experiments on cross language text classification tasks of Reuters multilingual newswire articles. In these experiments, the source domain and the target domain have completely different feature representation spaces.

5.2.1 Data Set

The experiments are conducted on a multilingual data set constructed from a comparable multilingual corpus used in [40], which contains newswire articles written in five languages (English(E), French(F), German(G), Italian(I), Spanish(S)), and distributed over six classes (C15, CCAT, E21, ECAT, GCAT, M11). We used the original documents in each language in our experiments. By taking each ordered pair of languages as the source and target language respectively and using the two large classes, CCAT and ECAT, we constructed 20 cross language binary text classification tasks. For example, we use *E2F* to denote the task that uses English as the source language and uses French as the target language. For each task, we randomly selected 2,000 documents from each language to form the data set.

5.2.2 Experimental Setup

We compared the proposed kernel matching method *SSKMDA1* with a baseline method and four heterogeneous domain adaptation learning methods: *TB* is a baseline approach that trains a SVM classifier on the labeled target instances; *HeMap* is a heterogeneous spectral mapping method developed in [22], which learns a latent low-dimensional representation across domains; *DAMA* is a manifold alignment based approach developed in [23], which seeks a latent space by utilizing labels from both domains; *ARC-t* is

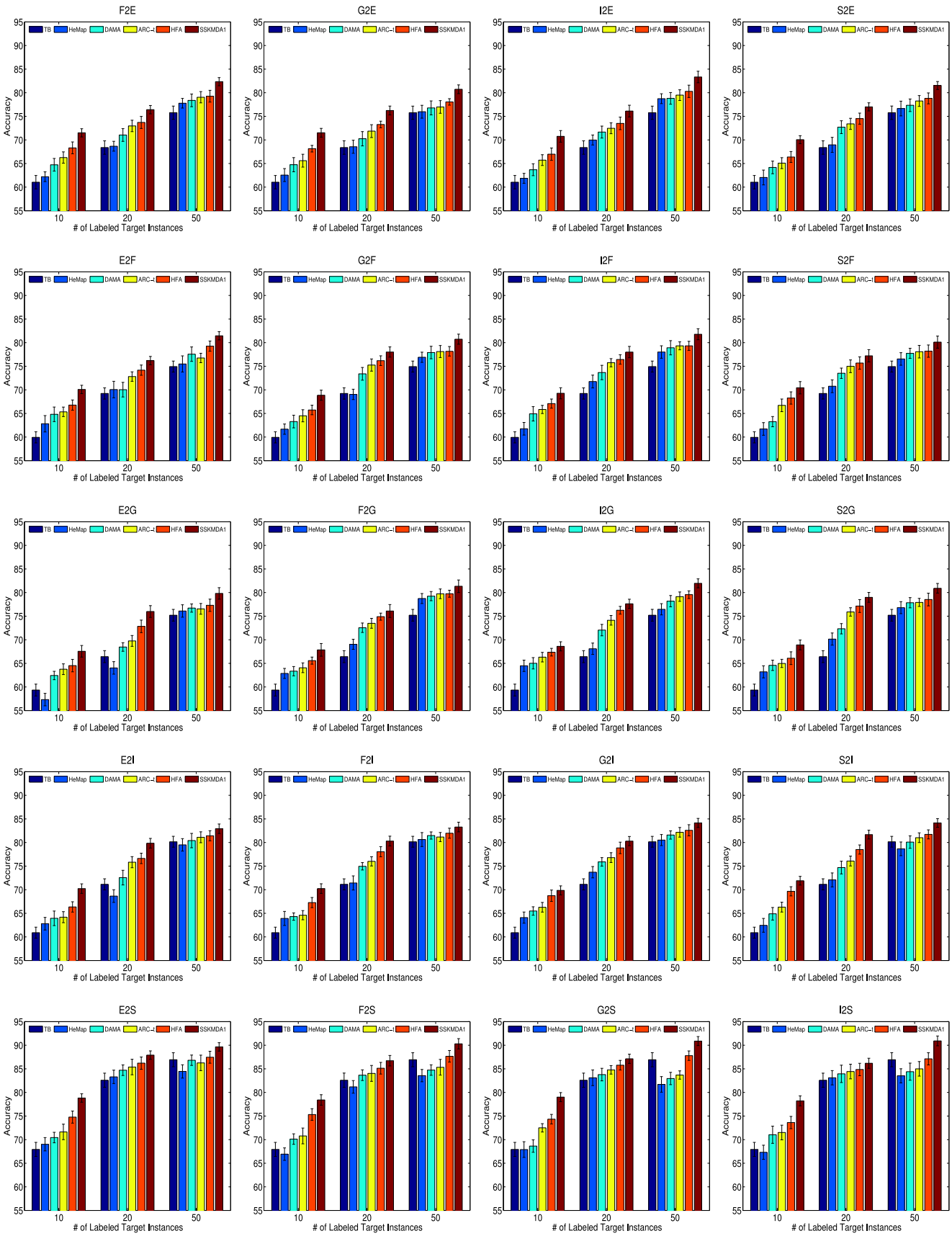

Fig. 4. The cross language classification results for the six comparison methods on 20 cross language text classification tasks with different numbers of labeled target instances, $\ell_t \in \{10, 20, 50\}$.

TABLE 5
The Training Time (Seconds/Minutes) for all Methods on the Cross Language Text Classification

	TB	HeMap	DAMA	ARC-t	HFA	SSKMDA1
Training Time	0.25 s	43.93 s	7.65 s	50.78 s	62.45 s	52.04 m

a domain adaptation method developed in [9], which learns an asymmetric transformation metric between different feature spaces by exploiting labeled data from both domains; *HFA* is a heterogeneous feature augmentation method developed in [16].

Note that since the source domain and the target domain have completely different feature spaces. *TB* can only exploit labeled data from the target domain. *ARC-t* and *HFA* are fully supervised learning methods, and they can only exploit labeled data from both domains. *HeMap*, *DAMA* and *SSKMDA1* can exploit both labeled and unlabeled data from the two domains. For our proposed approach *SSKMDA1*, we constructed the kernel matrices and the Laplacian graphs in the same way as we described in the previous section for cross domain sentiment classification tasks. For the tradeoff parameters, we used $\beta = 0.045$, $\gamma_s = 0.05$, $\gamma_t = 0.05$, $\eta = 1$, and $\mu = 5$.

5.2.3 Classification Results

For each of the constructed cross language text classification tasks, we selected $l_s = 1{,}000$ documents in the source domain as labeled and kept the rest $u_s = 1{,}000$ documents as unlabeled. In the target domain, we kept $u_t = 1{,}950$ documents as unlabeled, while choosing l_t labeled training documents from the remaining 50 documents. We conducted experiments with varying l_t values to study the empirical label complexity of the comparison approaches in the target domain. We tested a range of values, $l_t \in \{10, 20, 50\}$, in our experiments. We ran each experiment for 10 times based on different random selections of the instances and recorded the classification accuracies and standard deviations.

The experimental results on the 20 cross language text classification tasks are presented in Fig. 4. We can see that with limited number of labeled documents (l_t = 10, 20, 50) from the target domain, *TB* performs poorly for all the 20 tasks. Increasing the number of labeled training instances in the target domain can significantly improve the performance of *TB*. *HeMap* has improved the cross language classification accuracy on most tasks, but the improvements are not significant. Moreover, it performs even worse than *TB* on the *E2G* (English to German) and *I2S* (Italian to Spanish) tasks. The inferior performance of *HeMap* might be due to the fact that *HeMap* does not exploit discriminative label information when inducing latent low dimensional representations. Hence the latent representations induced by *HeMap* can bridge the difference across domains, but are less relevant to the classification tasks. Both *DAMA* and *ARC-t* outperform *TB* on most of the tasks, and the improvements achieved by them are much larger comparing to *HeMap*. This can be explained by the fact that both *DAMA* and *ARC-t* explicitly exploit discriminative label information. The other method, *HFA*, integrates both feature augmentation and representation learning. It outperforms *TB* on all the 20 tasks and outperforms *HeMap*, *DAMA*, and *ARC-t* on most tasks. Nevertheless, the proposed method *SSKMDA1* achieves the highest accuracies across the range of different l_t values and outperforms all the other comparison methods on all the 20 tasks. These results clearly demonstrate the efficacy of the kernel matching method in addressing heterogeneous domain adaptation problems.

We have also recorded the average training time for all the comparison approaches on the the first cross language text classification task *E2F* with 50 labeled documents in the target domain. The experiments were conducted on a workstation with 10 GB RAM and Xeon 3.20 GHz CPU. The training times for all the six approaches are reported in Table 5. The results are similar to that in the homogeneous domain adaptation experiments: Our proposed approach though has longer training time, it is still be able to solve cross language classification problems within a reasonable amount of time.

5.2.4 Convergence

We studied the convergence property of the training algorithm for the proposed kernel matching approach empirically. We used the same experimental setting as before, and

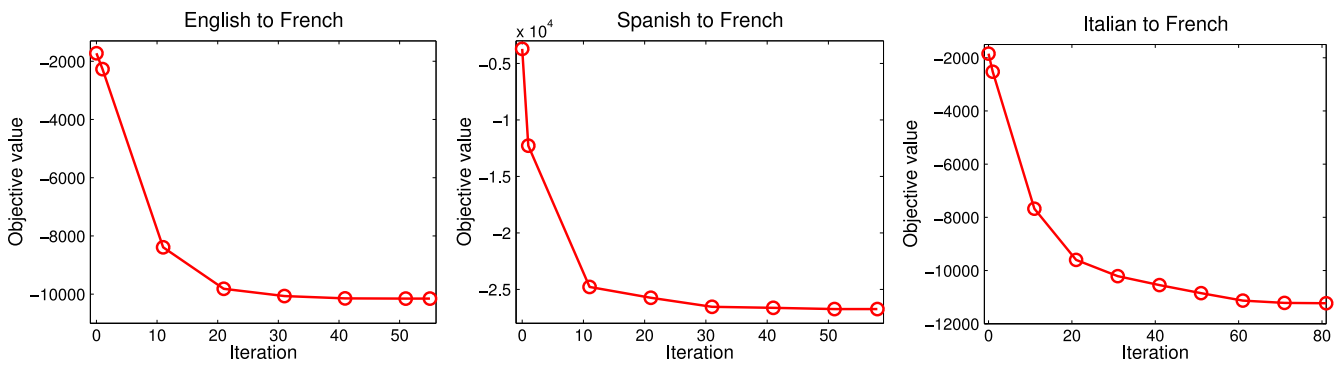

Fig. 5. The convergence process of *SSKMDA1* training on cross language text classification tasks.

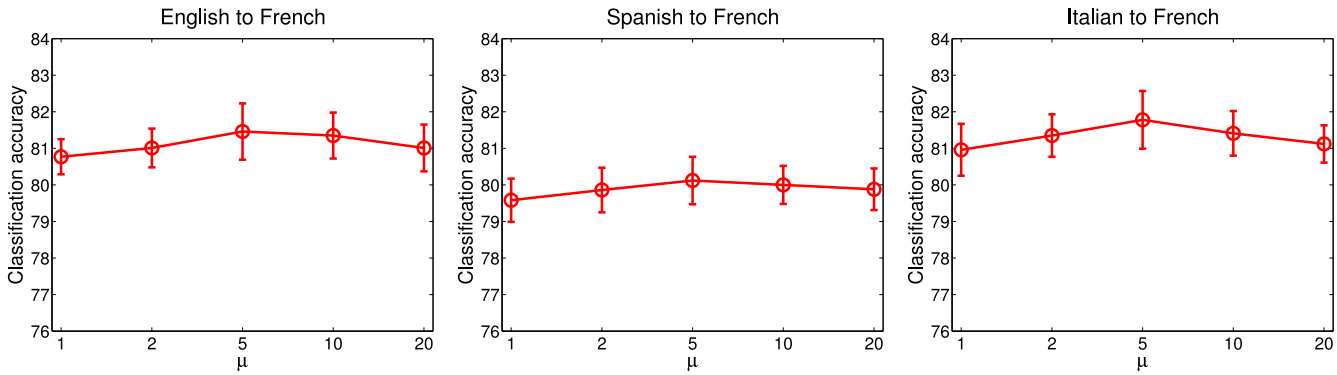

Fig. 6. Performance of *SSKMDA1* with respect to the trade-off parameter μ for cross language text classification tasks.

recorded the objective function value in Eq. (11) with respect to the number of iterations of the training algorithm. We reported the results for three tasks, *E2F* (English to French), *S2F* (Spanish to French), and *I2F* (Italian to French), in Fig. 5. We can see that the objective function value of *SSKMDA1* converges very quickly. We have similar observations for the other tasks.

5.2.5 Parameter Sensitivity Analysis

We conducted sensitivity analysis for *SSKMDA1* with respect to the tradeoff parameter μ in the cross language text classification experiments The experiments are conducted with $l_s = 1{,}000, u_s = 1{,}000, n_s = l_s + u_s = 2{,}000$, $l_t = 50, u_t = 1{,}950, n_t = l_t + u_t = 2{,}000$. We tested a range of μ values such that $\mu \in \{1, 2, 5, 10, 20\}$, and report the average classification accuracies over 10 runs for three tasks, *E2F* (English to French), *S2F* (Spanish to French), and *I2F* (Italian to French), in Fig. 6. We can see that *SSKMDA1* is not very sensitive to the μ value in the considered range.

6 Conclusion

In this work, we proposed a semi-supervised kernel matching method for domain adaptation to address the feature distribution divergence problem and the heterogeneous feature representation problem across domains. The proposed approach simultaneously learns a prediction function on the source domain and maps the target domain points to similar source domain points based on a Hilbert Schmidt Independence Criterion. We empirically evaluated the proposed learning method with extensive experiments on both homogeneous domain adaptation learning tasks of Amazon product reviews and heterogeneous domain adaptation learning tasks of Reuters newswire stores. The proposed approach achieves superior performance over a number of comparison methods on both cross domain sentiment classification of Amazon product reviews with substantially different feature distribution divergence and cross language text categorization of Reuters newswire articles with heterogeneous feature representations across domains. In the future, we plan to extend the proposed semi-supervised method to address domain adaptation with multiple source domains. Moreover, extending the proposed framework to perform active domain adaptation is also one direction we will pursue.

References

[1] H. Daumé III, "Frustratingly easy domain adaptation," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2007, pp. 256–263.
[2] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 137–144.
[3] L. Duan, I. Tsang, D. Xu, and T. Chua, "Domain adaptation from multiple sources via auxiliary classifiers," in *Proc. 26th Int. Conf. Mach. Learn.*, 2009, pp. 289–296.
[4] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation with multiple sources," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1041–1048.
[5] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2007, pp. 440–447.
[6] M. Chen, K. Weinberger, and J. Blitzer, "Co-training for domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 2456–2464.
[7] H. Daumé III, A. Kumar, and A. Saha, "Co-regularization based semi-supervised domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 478–486.
[8] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, pp. 213–226.
[9] B. Kulis, K. Saenko, and T. Darrell, "What you saw is not what you get: Domain adaptation using asymmetric kernel transforms," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 1785–1792.
[10] Y. Zhu, Y. Chen, Z. Lu, S. Pan, G. Xue, Y. Yu, and Q. Yang, "Heterogeneous transfer learning for image classification," in *Proc. 24th Nat. Conf. Artif. Intell.*, 2011, pp. 1304 –1309.
[11] S. Pan, J. Kwok, Q. Yang, and J. Pan, "Adaptive localization in a dynamic Wifi environment through multi-view learning," in *Proc. Nat. Conf. Artif. Intell.*, 2007, pp. 1108–1113.
[12] V. Zheng, S. Pan, Q. Yang, and J. Pan, "Transferring multi-device localization models using latent multi-task learning," in *Proc. Nat. Conf. Artif. Intell.*, 2008, pp. 1427–1432.
[13] J. Blitzer, R. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2006, pp. 120–128.
[14] J. Blitzer, D. Foster, and S. Kakade, "Domain adaptation with coupled subspaces," in *Proc. Int. Conf. Artif. Intell. Stat.*, 2011, pp. 173–181.
[15] J. Jiang and C. Zhai, "A two-stage approach to domain adaptation for statistical classifiers," in *Proc. 16th ACM Conf. Conf. Inf. Knowl. Manag.*, 2007, pp. 401–410.
[16] L. Duan, D. Xu, and I. Tsang, "Learning with augmented features for heterogeneous domain adaptation," in *Proc. 29th Int. Conf. Mach. Learn.*, 2012, pp. 711–718.
[17] P. Prettenhofer and B. Stein, "Cross-language text classification using structural correspondence learning," in *Proc. 48th Annu. Meeting Assoc. Comput. Linguistics*, 2010, pp. 1118–1127.
[18] W. Dai, Y. Chen, G. Xue, Q. Yang, and Y. Yu, "Translated learning: Transfer learning across different feature spaces," presented at the Adv. Neural Inf. Process. Syst., Vancouver, BC, Canada, 2008.

[19] M. Sugiyama, S. Nakajima, H. Kashima, P. von Bünau, and M. Kawanabe, "Direct importance estimation with model selection and its application to covariate shift adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1433–1440.
[20] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *J. Statist. Planning Inference*, vol. 90, no. 2, pp. 227–244, 2000.
[21] G. Tur, "Co-adaptation: Adaptive co-training for semi-supervised learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2009, pp. 3721–3724.
[22] X. Shi, Q. Liu, W. Fan, P. Yu, and R. Zhu, "Transfer learning on heterogenous feature spaces via spectral transformation," in *Proc. IEEE Int. Conf. Data Mining*, 2010, pp. 1049–1054.
[23] C. Wang and S. Mahadevan, "Heterogeneous domain adaptation using manifold alignment," in *Proc. Int. Joint Conf. Artif. Intell.*, 2011, pp. 1541–1546.
[24] S. Tan, "Improving SCL model for sentiment-transfer learning," in *Proc. North Amer. Chapter Assoc. Comput. Linguistics*, 2009, pp. 181–184.
[25] B. Gong, K. Grauman, and F. Sha, "Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 222–230.
[26] J. Hoffman, E. Rodner, J. Donahue, T. Darrell, and K. Saenko, "Efficient learning of domain-invariant image representations," in *Proc. Int. Conf. Learn. Representations*, 2013.
[27] J. Donahue, J. Hoffman, E. Rodner, K. Saenko, and T. Darrell, "Semi-supervised domain adaptation with instance constraints," in *Proc. Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 668–675.
[28] F. Mirrashed and M. Rastegari, "Domain adaptive classification," in *Proc. Int. Conf. Comput. Vis.*, 2013, pp. 2608–2615.
[29] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 647–655.
[30] H. Wang and Q. Yang, "Transfer learning by structural analogy," in *Proc. Nat. Conf. Artif. Intell.*, 2011, pp. 513–518.
[31] N. Quadrianto, K. Kersting, T. Tuytelaars, and W. Buntine, "Beyond 2d-grids: A dependence maximization view on image browsing," in *Proc. ACM Int. Conf. Multimedia Inf. Retrieval*, 2010, pp. 339–348.
[32] J. Jagaralamudi, S. Juarez, and H. Daumé III, "Kernelized sorting for natural language processing," in *Proc. Nat. Conf. Artif. Intell.*, 2010, pp. 1020–1025.
[33] M. Belkin and P. Niyogi, "Using manifold structure for partially labeled classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, pp. 953–960.
[34] M. Belkin, P. Niyogi, and V. Sindhwani, "On manifold regularization," in *Proc. Conf. Artif. Intell. Stat.*, 2005, pp. 17–24.
[35] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf, "Measuring statistical dependence with Hilbert-Schmidt norms," in *Proc. Int. Conf. Algorithmic Learn. Theory*, 2005, pp. 63–77.
[36] N. Quadrianto, A. Smola, L. Song, and T. Tuytelaars, "Kernelized sorting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1289–1296.
[37] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer, 2006.
[38] P. Rai, A. Saha, H. DauméIII, and S. Venkatasubramanian, "Domain adaptation meets active learning," in *Proc. North Amer. Chapter Assoc. Comput. Linguistics*, 2010, pp. 27–32.
[39] L. Song, A. Smola, A. Gretton, K. Borgwardt, and J. Bedo, "Supervised feature selection via dependence estimation," in *Proc. Int. Conf. Mach. Learn.*, 2007, pp. 823–830.
[40] M. Amini, N. Usunier, and C. Goutte, "Learning from multiple partially observed views-An application to multilingual text categorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 28–36.

Min Xiao received the BS degree from the Dalian University of Technology, China, in 2010. He is currently working toward the PhD degree in the Department of Computer and Information Sciences, Temple University. His main research interests include domain adaptation, transfer learning and their applications in natural language processing.

Yuhong Guo received the PhD from the University of Alberta in 2007. She has been a research fellow at the Australian National University and is currently an assistant professor in the Department of Computer and Information Sciences at Temple University. Her primary research area is machine learning, with applications in natural language processing, bioinformatics, and computer vision. She has published more than 40 refereed papers in top venues in these areas. She has received a number of awards for her research, including best paper prizes at IJCAI and AAAI.