Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence

Semi-Supervised Kernel Matching for Domain Adaptation

Min Xiao and **Yuhong Guo**
Department of Computer and Information Sciences
Temple University
Philadelphia, PA 19122, USA
{*minxiao, yuhong*}@*temple.edu*

Abstract

In this paper, we propose a semi-supervised kernel matching method to address domain adaptation problems where the source distribution substantially differs from the target distribution. Specifically, we learn 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. Our empirical results show the proposed kernel matching method significantly outperforms alternative methods on the task of across domain sentiment classification.

Introduction

Domain adaptation addresses the problem of exploiting information in a source domain where we have plenty labeled data to help learn a prediction model in a target domain where we have little labeled data (Daumé III 2007; Ben-David et al. 2006; Duan et al. 2009; Mansour, Mohri, and Rostamizadeh 2009). The need for domain adaptation is prevailing in various applied machine learning areas, such as natural language processing (Blitzer, McDonald, and Pereira 2006; Daumé III 2007; Chen, Weinberger, and Blitzer 2011), computer vision (Saenko et al. 2010) and WiFi localization (Pan et al. 2007; Zheng et al. 2008).

In many practical domain adaptation problems, the data distribution in the source domain is substantially different from the data distribution in the target domain. A key challenge raised in such problems is the *feature divergence* issue. That is, one cannot find support 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 natural language processing, where different genres often use very different vocabulary to describe similar concepts. For example, in sentiment classification data of product reviews, terms like "*harmonious*" or "*melodic*" are positive indicators in *Music* domain, but not in *Books* domain; similarly, terms like "*noise*" or "*yelling*" are negative indicators in *Music* domain, but not in *Books* domain. In this situation, most domain adaptation algorithms seek to bridge the gap between the two domains by re-weighting source instances (Sugiyama et al. 2008; Shimodaira 2000), self-labeling target instances (Tur 2009; Chen, Weinberger, and Blitzer 2011), inducing a new feature representation (Blitzer, Dredze, and Pereira 2007; Daumé III, Kumar, and Saha 2010; Blitzer, Foster, and Kakade 2011) and many other ways.

In this paper, we address the problem of feature representation divergence between the two domains from a novel perspective. We assume we have a source domain that contains a much larger number of labeled instances and unlabeled instances comparing to 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 perform 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. Through the kernel affinity measure, we expect unlabeled target instances to be most likely mapped into corresponding source instances with same labels as well. 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 form. We empirically evaluate two versions of the proposed method on across domain sentiment classification data of Amazon product reviews, where one tries to extract opinion-

oriented or sentiment polarity information from a given review text. Our experimental results suggest the proposed approach significantly outperforms the feature representation based across domain sentiment classification approaches.

Related Work

Domain adaptation has recently been popularly studied in machine learning and related fields. 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. 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 (Shimodaira 2000; Sugiyama et al. 2008). These methods however perform poorly for highly divergent domains characterized by missing features under source distribution for target instances.

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 (Jiang and Zhai 2007) or co-training (Tur 2009; Chen, Weinberger, and Blitzer 2011). Their performances greatly depend on the initial model trained from source labeled data and they are not best suitable for highly divergent domains either.

A number of domain adaptation algorithms address the domain divergence issue directly from feature representation learning perspective, including structural correspondence learning methods (Blitzer, McDonald, and Pereira 2006; Blitzer, Dredze, and Pereira 2007; Tan 2009), coupled subspace methods (Blitzer, Foster, and Kakade 2011) and others. They seek to learn a shared representation and distinguish domain-specific features by exploiting the large amount of unlabeled data from both the source and target domains. The efficacy of these methods nevertheless depends on the existence of a certain amount of pivot features that are used to induce shared feature representations. In addition to these, the transfer learning work in (Wang and Yang 2011) exploits the Hilbert Schmidt Independence Criterion to learn the mapping of selected features from two domains.

The approach we develop in this work is related to the feature representation learning methods. But instead of exploring cross domain feature similarities, we focus on cross domain instance similarities according to kernel representations. Our approach does not need pivot features or feature correspondence information, but needs only a very small set of labeled pivot instances from the target domain. Our empirical study shows the proposed approach is more effective than the feature learning based domain adaptation methods on across domain sentiment classification.

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.

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 (Belkin and Niyogi 2002; Belkin, Niyogi, and Sindhwani 2005), 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 $\mathcal{P}(Y|x_1)$ and $\mathcal{P}(Y|x_2)$ 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.

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 (HSIC) over the kernel matrix of the target instances and the kernel matrix of the mapped source instances. HSIC (Gretton et al. 2005) originally measures the independence between given random variables based on the eigenspectrum of covariance operators in Reproducing Kernel Hilbert Spaces. (Quadrianto et al. 2008) 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

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

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 (Quadrianto et al. 2008), which leads to the following problem

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

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

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

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

where $\ell(\cdot,\cdot)$ is a loss function, $\mathcal{H}$ is the Reproducing Kernel Hilbert Space (RKHS) 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

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

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

$$\|f\|^2_{\mathcal{H}} = \boldsymbol{\alpha}^\top K^s\boldsymbol{\alpha} \quad (5)$$

Then using a square loss function, the minimization problem (3) can be rewritten as

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

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.

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 (Belkin and Niyogi 2002; Belkin, Niyogi, and Sindhwani 2005). Specifically, we will incorporate the following graph Laplacian terms which approximate manifold regularization

$$\gamma_s\|f\|^2_{G_s} + \gamma_t\|Mf\|^2_{G_t} \quad (7)$$

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

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

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

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

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.

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

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

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

$$\min_{M} \; g(M) \tag{11}$$
$$\text{s.t. } 0 \le M \le 1;\; M\mathbf{1} = \mathbf{1};\; J^t M J^{s\top}\mathbf{y}^s = \mathbf{y}^t.$$

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,

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

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,

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

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 (Nocedal and Wright 2006) to seek an optimal step size ρ^* to obtain $M^{(k+1)}$ along the direction D in the close neighborhood of $M^{(k)}$: $M^{(k+1)} = M^{(k)} + \rho^* D$. 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 minimizing (11) is given in Algorithm 1.

Algorithm 1: Local Optimization Procedure

Input: $\mathbf{y}^s, \mathbf{y}^t, K^s, K^t; M^{(0)}, \epsilon; \mu, \beta, \gamma_s, \gamma_t$; 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 $k > MaxIters$
$M^* = M^{(k)}$.

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

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.

Experiments

In this section, we present our experimental results on across domain sentiment classifications. We first describe our experimental setting and then present results and discussions.

Experimental Setting

Dataset We used the across domain sentiment classification dataset from (Prettenhofer and Stein 2010) in our experiments. The dataset contains reviews in 3 domains (*Books*, *DVD* and *Music*), and have 4 different language versions (English, German, French and Japanese). Each domain contains 2000 positive views and 2000 negative reviews, each of which is represented as a term-frequency (TF) vector. We used the English version and constructed 6 source-target ordered domain pairs based on the original 3 domains: *B2D* (Books to DVD), *D2B* (DVD to Books), *B2M* (Books to Music), *M2B* (Music to Books), *D2M* (DVD to Music), and

Table 1: Test accuracies for 6 domain adaptation tasks.

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

M2D (Music to DVD). For each pair of domains, we built an unigram vocabulary from combined reviews in both domains. We further preprocessed the data by removing features that appear less than twice in either domain, replacing TF features with TF-IDF features, and normalizing each attribute into $[0, 1]$.

The divergence of each pair of domains can be measured with $A-$distance (Ben-David et al. 2006). We adopted the same method in (Rai et al. 2010) to computed approximate $A-$distance values. We first trained a linear separator to separate source and target domains with all instances from both. The average per-instance hinge-loss for this separator subtracted from 1 was used as an estimate of proxy $A-$distance. It is a number in the interval of $[0, 1]$ with larger values indicating larger domain divergence. Table 2 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.

Table 2: Statistics for different domain pairs.

Domains	**Vocabulary Size**	A**-distance**
Books vs. DVD	10370	0.7221
Books vs. Music	8006	0.8562
DVD vs. Music	8825	0.7831

Approaches In our experiments, we compared the performance of the following approaches.

- **TargetOnly**: trained on labeled data in target domain.
- **SourceOnly**: trained on labeled data in source domain.
- **SourceTarget**: trained on labeled data in both domains.
- **EA++**: the domain adaptation method proposed in (Daumé III, Kumar, and Saha 2010).
- **Coupled Subspace**: the domain adaptation method proposed in (Blitzer, Foster, and Kakade 2011).
- **SSKMDA1**: the proposed semi-supervised kernel matching for domain adaptation.
- **SSKMDA2**: in addition to SSKMDA1, we also tested another version of semi-supervised kernel matching method for domain adaptation by replacing the unbiased HSIC component in Eq.(2) with the unbiased HSIC used in (Song et al. 2007).

We used Matlab SVM toolbox for the first three baselines with default hyper-parameters. For Coupled Subspace, we used the software package provided by (Blitzer, Foster, and Kakade 2011)[1]. There are 2 parameters to set in this package, the top k representative features, and the size of source and target projectors. We used the same values that are used in (Blitzer, Foster, and Kakade 2011): 1000 for the top representative features and 100 for the dimension of projectors.

For our proposed approach, we used Gaussian kernels to construct the kernel matrices, $K(x_1, x_2) = \exp(-|x_1 - x_2|^2/(2\sigma^2))$, where the parameter σ was set to 0.05. We used K-nearest-neighbors (KNN) with binary weights to construct Laplacian graphs G_s and G_t for the source and target domains respectively. We used 20 as the number of nearest neighbors in our experiments. For the tradeoff parameters in our formulated optimization (9), we used $\beta = 0.045$, $\gamma_s = 0.05$, $\gamma_t = 0.05$, $\eta = 1$, and $\mu = 5$.

Across Domain Classification Results

As we introduced before, our semi-supervised learning is actually a transductive learning. We conducted 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 was evaluated on the u_t unlabeled target instances.

In the experiments, we used $l_s = 1390, u_s = 10, n_s = l_s + u_s = 1400, l_t = 10, u_t = 990, n_t = l_t + u_t = 1000$. We randomly chose n_s instances from the source domain, with the first l_s instances labeled and the rest unlabeled. Similarly, we randomly chose n_t instances from the target domain, with the first l_t instances labeled and the rest unlabeled. All approaches were tested using the same data. Each experiment was repeated 10 times. The average test accuracies and standard deviations for all 6 experiments are reported in Table 1. We can see that neither a few labeled target instances nor a large amount of labeled source instances alone are enough to train a good sentiment classifier for the target domain. By simply training over both labeled source instances and target instances can have very limited improvement. The EA++ approach demonstrates improvements over the three baselines, but the improvement is not significant. The Coupled Subspace domain adaptation method however presents significant improvement over the first three baselines. Nevertheless, it is not as good as our proposed approach (two

[1] http://john.blitzer.com/software.html

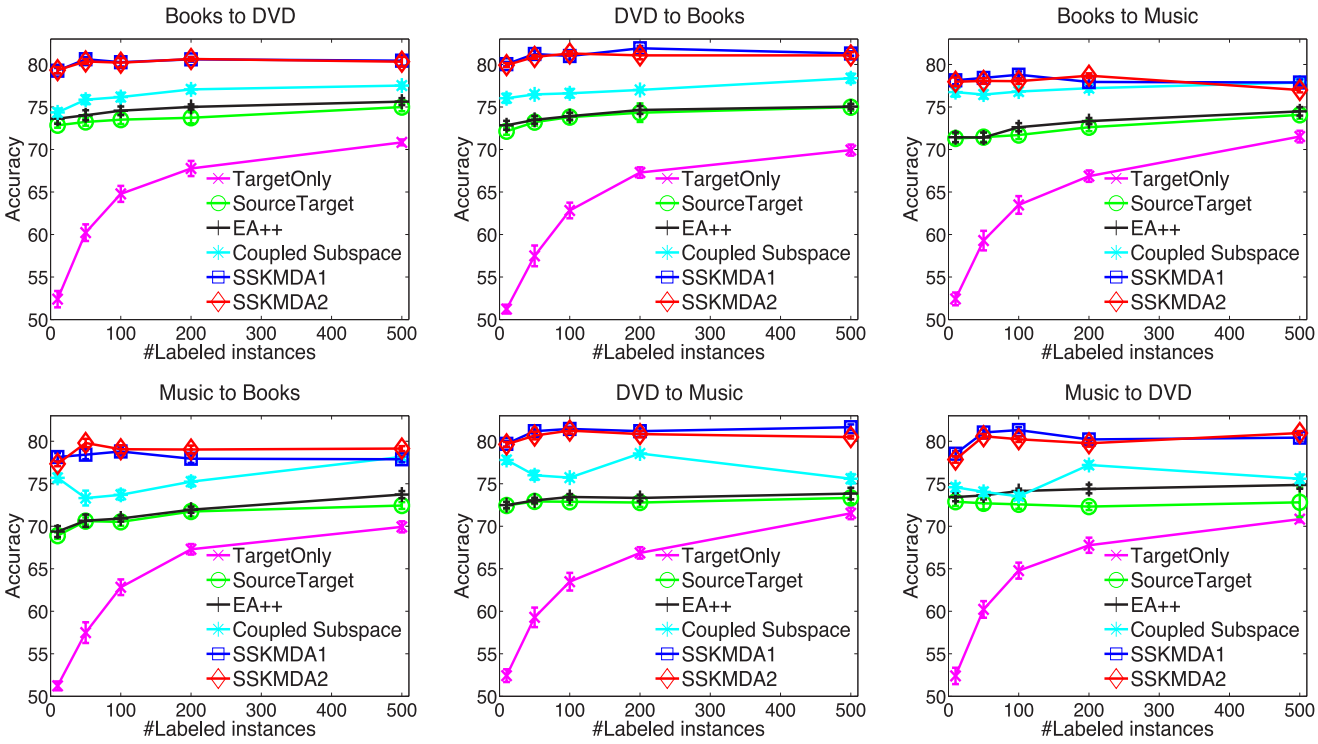

Figure 1: Test accuracies with varying number of labeled instances in the target domain for 6 domain adaptation tasks.

versions). Both versions of our proposed domain adaptation method perform consistently and significantly better than all other approaches over all 6 tasks. For the task B2D, our approach increases the accuracy by more than 5% comparing to Coupled Subspace. For tasks D2B and M2D, our approach increases the accuracy by about 4%; and about 2% for tasks B2M, M2D and D2M. The two versions of the proposed approach achieved very similar results, although SSKMDA1 is slightly better than SSKMDA2.

Classification Results vs Label Complexity

As we introduced before, the labeled target instances perform as pivot points for kernel matching in our proposed approach. Then we may ask: is the proposed approach sensitive to the number of pivot points? To answer this question and study the target domain label complexity of the proposed approach, we conducted another sets of experiments with varying number of labeled target instances. In the experiments above, we used $l_t = 10$ which is a reasonably small number. We thus conducted tests with a set of values $l_t = \{10, 50, 100, 200, 500\}$ here. We still used 1390 labeled instances and 10 unlabeled instances from the source domain, and used 990 unlabeled instances from the target domain. The classification results are reported on the unlabeled 990 instances from the target domain as well.

We reported the average results over 10 times' repeats in Figure 1 for the versions of the proposed approach and four others: *TargetOnly, SourceTarget*, *EA++* and *Coupled Subspace*. We can see that both versions of the proposed approach consistently outperform all the other methods over all 6 domain adaptation tasks and across a set of different l_t values. Moreover, increasing the number of labeled target instances leads to significant performance improvement for the *TargetOnly* method. The performances of *SourceTarget, EA++* and *Couple Subspace* vary in a small degree due to the fact there are a lot more labeled source instances, and these 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 method only requires a very few pivot points to produce a good prediction model for the target instances. The empirical label complexity of the proposed approach 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.

Conclusion

In this paper, we addressed a key challenge in domain adaptation, the problem of feature representation divergence between two domains, from a novel perspective. We developed a semi-supervised kernel matching method for domain adaptation based on a Hilbert Schmidt Independence Criterion (HSIC). By mapping the target domain points into corresponding source domain points in a transductive (semi-supervised) way, the classifier trained in the source domain

can reasonably classify the instances in the target domain as well. The two versions of the proposed method both achieved superior results on across domain sentiment classification tasks comparing to other domain adaptation methods. The empirical results also suggest the proposed method has a low label complexity in the target domain, and can greatly reduce human annotation effort.

References

Belkin, M., and Niyogi, P. 2002. Using manifold structure for partially labeled classification. In *Advances in Neural Information Processing Systems (NIPS)*.

Belkin, M.; Niyogi, P.; and Sindhwani, V. 2005. On manifold regularization. In *Proceedings of the Conference on Artificial Intelligence and Statistic (AISTATS)*.

Ben-David, S.; Blitzer, J.; Crammer, K.; and Pereira, F. 2006. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems (NIPS)*.

Blitzer, J.; Dredze, M.; and Pereira, F. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Blitzer, J.; Foster, D.; and Kakade, S. 2011. Domain adaptation with coupled subspaces. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Blitzer, J.; McDonald, R.; and Pereira, F. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Chen, M.; Weinberger, K.; and Blitzer, J. 2011. Co-training for domain adaptation. In *Advances in Neural Information Processing Systems (NIPS)*.

Daumé III, H. 2007. Frustratingly easy domain adaptation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Daumé III, H.; Kumar, A.; and Saha, A. 2010. Co-regularization based semi-supervised domain adaptation. In *Advances in Neural Information Processing Systems (NIPS)*.

Duan, L.; Tsang, I.; Xu, D.; and Chua, T. 2009. Domain adaptation from multiple sources via auxiliary classifiers. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Gretton, A.; Bousquet, O.; Smola, A.; and Schölkopf, B. 2005. Measuring statistical dependence with hilbert-schmidt norms. In *Proceedings of Algorithmic Learning Theory*.

Jiang, J., and Zhai, C. 2007. A two-stage approach to domain adaptation for statistical classifiers. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management (CIKM)*.

Mansour, Y.; Mohri, M.; and Rostamizadeh, A. 2009. Domain adaptation with multiple sources. In *Advances in Neural Information Processing Systems (NIPS)*.

Nocedal, J., and Wright, S. J. 2006. *Numerical Optimization*. Springer, New York, 2nd edition.

Pan, S.; Kwok, J.; Yang, Q.; and Pan, J. 2007. Adaptive localization in a dynamic WiFi environment through multi-view learning. In *Proceedings of the national conference on Artificial intelligence (AAAI)*.

Prettenhofer, P., and Stein, B. 2010. Cross-language text classification using structural correspondence learning. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Quadrianto, N.; Smola, A.; Song, L.; and Tuytelaars, T. 2008. Kernelized sorting. In *Advances in Neural Information Processing Systems (NIPS)*.

Rai, P.; Saha, A.; DauméIII, H.; and Venkatasubramanian, S. 2010. Domain adaptation meets active learning. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Saenko, K.; Kulis, B.; Fritz, M.; and Darrell, T. 2010. Adapting visual category models to new domains. In *Proceedings of the 11th European conference on Computer vision (ECCV)*.

Shimodaira, H. 2000. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference* 90(2):227–244.

Song, L.; Smola, A.; Gretton, A.; Borgwardt, K.; and Bedo, J. 2007. Supervised feature selection via dependence estimation. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Sugiyama, M.; Nakajima, S.; Kashima, H.; von Bünau, P.; and Kawanabe, M. 2008. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in Neural Information Processing Systems (NIPS)*.

Tan, S. 2009. Improving SCL model for sentiment-transfer learning. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Tur, G. 2009. Co-adaptation: Adaptive co-training for semi-supervised learning. In *Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Wang, H., and Yang, Q. 2011. Transfer learning by structural analogy. In *Proceedings of the national conference on Artificial intelligence (AAAI)*.

Zheng, V.; Pan, S.; Yang, Q.; and Pan, J. 2008. Transferring multi-device localization models using latent multi-task learning. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*.