Extracting Discriminative Concepts for Domain Adaptation in Text Mining

ABSTRACT

One common predictive modeling challenge occurs in text mining problems is that the training data and the operational (testing) data are drawn from different underlying distributions. This poses a great difficulty for many statistical learning methods. However, when the distribution in the source domain and the target domain are not identical but related, there may exist a shared concept space to preserve the relation. Consequently a good feature representation can encode this concept space and minimize the distribution gap. To formalize this intuition, we propose a domain adaptation method that parameterizes this concept space by linear transformation under which we explicitly minimize the distribution difference between the source domain with sufficient labeled data and target domains with a large amount of unlabeled data, while at the same time minimizing the empirical loss on the labeled data in the source domain. Another characteristic of our method is its capability for considering multiple classes and their interactions simultaneously. We have conducted extensive experiments on two common text mining problems, namely, information extraction and document classification to demonstrate the effectiveness of our proposed method.[1].

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; I.5.2 [**Design Methodology**]: [feature extractions]

General Terms

Algorithms, Text Mining.

Keywords

Domain Adaptation, Feature Extraction

[1]For repeatability test, datasets and binary codes available at: http://www.se.cuhk.edu.hk/∼bchen/kdd09.htm

WOODSTOCK '97 El Paso, Texas USA

1. INTRODUCTION

Traditional statistical learning techniques rely on the basic assumption that the training data and the operational (testing) data are drawn from the same underlying distribution. However, in many text mining applications involving high-dimensional feature space, it is difficult to collect sufficient training data for different domains. For example, consider a text information extraction problem whose objective is to automatically extract precise job information such as job title, duty, requirement, etc. from recruitment Web sites in different industries supporting intelligent analysis of employment information. Usually we may just have few experts who can accurately annotate the information in one specific industry like accounting for preparing the training data. The learnt model deployed obviously cannot perform well in other domains (industries) such as logistic or health care due to the distribution of the terms in each domain is different. One strategy to tackle this problem is to adapt the trained model from one domain known as the source domain with sufficient labeled data to another domain known as the target domain where only a small amount or even no labeled data is available.

It can be observed that domain adaptation is reasonable and practical if the distributions between the source domain and the target domain is related, which is mainly based on the fact that there exists a shared concept space in which the embedded distribution of each domain is close enough. Consequently it is very reasonable to believe that a good feature representation is able to encode this concept space and provide strong adaptive power from the source domain to the target domain. On the other hand, such a changed representation may encode less information leading to an increase of the empirical loss on the labeled data. To cope with this problem, we try to learn the ideal shared concept space with respect to two criteria: the empirical loss in the source domain, and the embedded distribution gap between the source domain and the target domain. Consider again the job information extraction example. For the task of extracting the job requirement information in the domain of accounting, the most representative terms are "qualified", "year", "experience", "CPA", "CA", "ACCA", etc. Similarly for the domain of health care, the corresponding terms shift to "qualified", "degree", "year", "CCP", "Physiology", "experience", etc. If we can extract the shared domain independent features such as "qualified", "year", "experience" for the specific task, then the learnt extractor can be effectively adapted to the domain of health care.

In this paper we propose a domain adaptation method

which directly minimizes both the distribution gap between the source domain and the target domain, as well as the empirical loss on the labeled data in the source domain by extracting the low-rank concept subspace. Maximum Mean Discrepancy (MMD) [5] is adopted to measure the embedded distribution difference between the source domain with sufficient but finite labeled data and the target domain with sufficient unlabeled data. Then our objective is to minimize the empirical loss and the MMD measurement with respect to the parametric family (linear transformation) which parameterizes the embedded feature subspace. Furthermore, we apply the graph Laplacian [1] to exploit the predictive power for some domain dependent representative features in the target domain based on the co-occurrence with the shared features. This technique can help improve the performance especially when the common features are not sufficient in the target domain.

In fact, there have several domain adaptation methods been proposed to learn a reasonable representation so as to make the distributions between the source domain and the target domain more closer [3, 12, 13, 11]. However, none of them can automatically learn the concept space where the prediction power in the source domain and the adaptive power from the source domain to the target domain are both considered.

Our main contributions can be summarized as follows:
(1) We propose a domain adaptation method to extract the low-rank concept space shared by the source domain and the target domain, which can ensure both the predictive power and adaptive power are maximized.
(2) We can transfer the predictive power from the extracted common features to the characteristic features in the target domain by the feature graph Laplacian.
(3) We theoretically analyze the expected error in the target domain showing that the error bound can be controlled by the expected loss in the source domain, and the embedded distribution gap, so as to prove that what we minimize in the objective function is very reasonable for domain adaptation.
(4) Our domain adaptation method is capable of considering multiple classes and their interactions simultaneously. It can be applied to high dimensional text mining applications due to two major properties of text: latent semantic and sparseness. The first property ensures that low-rank concept space can still preserve enough information, and the second property contributes to the computation speed.

We have conducted extensive experiments on two common text mining problems, namely, information extraction and document classification to demonstrate the effectiveness of our proposed method. Experiment results show that our method can get better performance than other existing competitive methods.

2. DOMAIN ADAPTATION

2.1 Related Work

Domain adaptation is a widely studied area. It addresses a common situation when applying the trained model to a different domain. Many works try to learn a new representation which can bridge the source domain and the target domain. Blitzer *et al.* [3] proposed a heuristic method to select some domain independent pivot features to learn an embedded space where the data comeing from both domains can share the same feature structure. Daumé III [4] proposed the Feature Augmentation method to augment features for domain adaptation. The augmented features are used to construct a kernel function for kernel methods. Raina *et al.* [12] learned the sparse basis from the unlabeled data which is not necessary in the same domain as the labeled data. Then it represents the labeled data by those learned high-level basis for further classification. Several domain adaptation methods [6, 14, 15, 8, 2] suggested to apply the instance weighting technique for domain adaption in various applications. Recently, Pan *et al.* [11] applied the Maximum Mean Discrepancy (MMD) to learn the embedded space where the distribution between the source domain and the target domain is minimized.

2.2 Problem Statement and Preliminaries

In this paper, we focus on the setting where the testing samples come from another domain, which is different from the training set. In the sequel, we refer the training set to as the source domain $D_{\mathcal{S}} = \{(x_i, y_i)\}_{i=1}^{n_1}$, where $x_i \in \mathbb{R}^d$ is the d dimensional input space, and y_i is the output label. We also assume that the testing samples are available. Denote the testing set as $D_{\mathcal{T}} = \{x'_i\}_{i=1}^{n_2}$ and $x'_i \in \mathbb{R}^d$ is the input. Let $\mathcal{P}(x)$ and $\mathcal{Q}(x')$ (or $\mathcal{P}$ and $\mathcal{Q}$ for short) be the marginal distributions of the input sets $\{x_i\}$ and $\{x'_i\}$ from the source and target domains, respectively. In general, $\mathcal{P}$ and $\mathcal{Q}$ can be different. The task of domain adaptation is to predict the labels y'_i's corresponding to the inputs x'_i's in the target domain. Note that domain adaptation is different from Semi-Supervised Learning (SSL). SSL methods employ both labeled and unlabeled data for better classification, in which the labeled and unlabeled data are assumed to be drawn from the same domain. Unlike SSL, the key assumption in domain adaptation is that $\mathcal{P} \neq \mathcal{Q}$, but the class conditional distribution of the source and target domains remains unchanged, *i.e.*, $P(y|x) = P(y'|x')$.

2.3 Maximum Mean Discrepancy

Recall that, in domain adaptation, the fundamental question is how to evaluate the difference in distribution between two domains given finite observations of $\{x_i\}$ and $\{x'_i\}$. There exists many criteria (such as the Kullback-Leibler (KL) divergence) that can be used to measure their distance. However, many of these estimators are parametric and require an intermediate density estimate. To avoid this non-trivial task, a non-parametric distance estimate between distributions is more desirable. Recently, Gretton *et al.* [5] introduced the Maximum Mean Discrepancy (MMD) for comparing distributions based on the Reproducing Kernel Hilbert Space (RKHS) distance. Let the kernel-induced feature map be $\phi : \mathbb{R} \mapsto \mathcal{H}$, where $\mathcal{H}$ is the corresponding feature space. The MMD between the source domain $D_{\mathcal{S}}$ and the target domain $D_{\mathcal{T}}$ is defined as follows:

$$\begin{aligned} \mathrm{MMD}[D_{\mathcal{S}}, D_{\mathrm{T}}] &= \sup_{\|f\|_{\mathcal{H}} \leq 1} \left(\mathrm{E}_{\mathcal{Q}}[f(x')] - \mathrm{E}_{\mathcal{P}}[f(x)]\right]) \\ &= \|\mathrm{E}_{\mathcal{Q}}[\phi(x')] - \mathrm{E}_{\mathcal{P}}[\phi(x)]\|_{\mathcal{H}}\,. \end{aligned} \tag{1}$$

The empirical measure of the MMD in (1) is defined as:

$$\mathrm{MMD}[D_{\mathcal{S}}, D_{\mathrm{T}}] = \left\| \frac{1}{n_2} \sum_{x' \in D_{\mathcal{T}}} \phi(x') - \frac{1}{n_1} \sum_{x \in D_{\mathcal{S}}} \phi(x) \right\|_{\mathcal{H}}. \tag{2}$$

Therefore, the distance between two distributions of two samples is simply the distance between the two mean elements in the RKHS.

2.4 Kernel Mean Matching

Due to the change of distribution from different domains, training with samples from the source domain may degrade the generalization performance in the another target domain. To reduce the mismatch between the two different domains, Huang *et al.* [6] proposed a two-step approach Kernel Mean Matching (KMM). The first step is to diminish the difference of means of samples in RKHS between the two domains by re-weighting the samples $\phi(x_i)$ in the source domain as $\beta_i\phi(x_i)$, where β_i is learned by using the MMD criterion in (2).

Then the second step is to learn a decision classifier $f(x) = w^\top\phi(x) + b$ that separates patterns of opposite classes using the loss function re-weighted by β_i in the objective.

2.5 Maximum Mean Discrepancy Embedding

However, the simple re-weighting scheme may have a limited improvement in the target domain when the dimensionality of the data is high. In particular, some features may cause the data distribution between domains to be different, while others may not. Some features may preserve the structure of data for adaptation, while others may not. To address this problem, Pan *et al.* [11] proposed Maximum Mean Discrepancy Embedding (MMDE) for domain adaptation by embedding both the source and target domain data onto a shared low-dimensional latent space. The key idea is to formulate this as a kernel learning problem using the kernel trick $K_{ij} = K(x_i, x_j) = \phi(x_i)^\top\phi(x_j)$, and to learn the kernel matrix defined on all the data:

$$K = \begin{matrix} K_{S,S} & K_{S,T} \\ K_{T,S} & K_{T,T} \end{matrix} \in \mathbb{R}^{(n_1+n_2)\times(n_1+n_2)}, \tag{3}$$

where $K_{S,S}$, $K_{T,T}$ and $K_{S,T}$ are the Gram matrices defined on the source domain, target domain, and cross domain data, respectively. By minimizing the distance (measured by MMD) between the source and target domain data. The square of the MMD in (2) can be written as

$$\text{trace}(KD), \tag{4}$$

where

$$D_{ij} = \begin{cases} \frac{1}{n_1^2} & \text{when } x_i, x_j \in D_{\mathcal{S}} \\ \frac{1}{n_2^2} & \text{when } x_i, x_j \in D_{\mathcal{T}} \\ \frac{-1}{n_1 n_2} & \text{otherwise.} \end{cases} \tag{5}$$

This leads to a Semi-Definite Programming (SDP) problem.

After that, the embedding of data can be extracted by performing eigen-decomposition on the learned kernel matrix K in (3), and can be used for training classifiers.

3. FEATURE EXTRACTION FOR MULTI-CLASS DOMAIN ADAPTATION

In previous domain adaptation methods [6, 11], the weights β_i's in KMM or the kernel matrix K of samples in MMDE are learned separately using the MMD criterion in (2) defined on the input data only without considering any labels. While the use of labels in linear discriminant analysis usually helps extract more discriminative features, the label information from the source domains may be also useful to learn kernels or extract features for a better domain adaptation.

In addition, there are two main limitations associated with MMDE. First, MMDE is transductive and cannot generalize on unseen patterns. Second, it requires to solve an expensive SDP problem, which takes $O((n_1+n_2)^{6.5})$ time to solve the optimization problem (4). Although polynomial-time solvers are available, current interior-point methods are still too computationally intensive for large-scale SDPs in real applications. Note that only the low dimensional embedding of the data is extracted from the learned kernel matrix K in MMDE, and is then used for the training of the decision classifiers. Therefore, not all components from the learned kernel matrix K are required to train the classifiers for domain adaptation.

3.1 Proposed Framework

Based on the above discussions, instead of using two-step approaches as in [6, 11], we propose a unified domain adaptation learning framework to find the discriminative feature subspace Θ, and to learn decision classifiers $f_l(x)$'s simultaneously. In particular, our proposed method minimizes the distribution difference between the samples of the source and target domains after the projection into the subspace Θ (*i.e.* Θx_i), as well as the structural risk functional of the n_1 labeled data from the source domain $D_{\mathcal{S}}$. Moreover, we suppose that the learning problem is in multiclass setting, and there are m decision classifiers $f_l(x)$'s. Let us denote the label indicator matrix as $Y \in \mathbb{R}^{n_1\times m}$, and $Y_{il} = 1$ if the i-th sample belongs to the l-th class, and 0 if it is labeled as others. Similar to other feature extraction methods, we also suppose Θ is orthogonal on rows so that $\Theta\Theta^T = I$. The optimization problem is then formulated as follows:

$$\min \quad \sum_{l=1}^{m}\sum_{i=1}^{n_1}\ell(f_l, Y_{il}, x_i) + \alpha\sum_{l=1}^{m}\Omega(f_l)\Bigg) + \beta\text{dist}_\Theta(D_{\mathcal{S}}, D_{\mathcal{T}}), \tag{6}$$

subject to $\Theta\Theta^T = I$. Here, the first term is the empirical risk functional of the decision functions f_l's on the labeled data from the source domain $D_{\mathcal{S}}$, and $\ell(\cdot)$ is the empirical loss function. The regularizer $\Omega(\cdot)$ controls the complexity of f_l, and the last term measures the distribution difference between the embedding of $D_{\mathcal{S}}$ and $D_{\mathcal{T}}$. Two tradeoff parameters $\alpha > 0$ and $\beta > 0$ are introduced to control the fitness of the decisions functions, and to balance the difference of distribution from the two domains and the structural risk functional for the labeled patterns, respectively. Hence, using (6), the subspace Θ and the decision functions f_l's can be learned at the same time.

3.1.1 Shared Subspace for Label Dependency

To capture the label dependency, we follow [7] to define the m decision functions:

$$f_l(x) = w_l^\top x = u_l^\top x + v_l^\top\Theta x, \; l \; = \; 1, \cdots, m \tag{7}$$

where $w_l \in \mathbb{R}^d$ is the weight vector for the decision function, $\Theta \in \mathbb{R}^{r\times d}$ is the matrix of the shared subspace for the m decision functions, and $v_l \in \mathbb{R}^r$ is the weight vector defined in the projected subspace Θ, and $u_l \in \mathbb{R}^d$ is the weight vector defined in the original input space. With the parametric form (7) of the m decision classifiers, the learned subspace Θ can capture the intrinsic structure of label dependency in multiclass problems [7], the weight vector v_l is the discriminative direction in the subspace Φ for each class, while the weight vector u_l can be used to fit the residue $w_l - \Theta^\top v_l$ for each class independently.

Though we learn a linear shared subspace in (7), the linear subspace is usually more efficient and also achieves good generalization performance for high dimensional data such as text documents. Moreover, one can simply replace the input x by the feature mapped input $\phi(x)$ in (7) and apply the Representer Theorem for w_l, u_l and Θ, which gives rise to the kernel variant of the proposed framework for the nonlinear generalization performance, which is beyond the

scope of this paper. For simplicity, we use the notation x instead of $\phi(x)$ in the sequel.

3.1.2 Loss Function and Regularization

For the empirical loss on the labeled data, we apply the square loss function:

$$\begin{aligned}\ell(f_l, Y_{il}, x_i) &= (f_l(x_i) - Y_{il})^2 \\ &= (w_l^\top x_i - Y_{il})^2 \\ &= (u_l^\top x_i + v_l^\top \Theta x_i - Y_{il})^2.\end{aligned}$$

Suppose $X_{\mathcal{S}} = [x_1, \cdots, x_{n_1}] \in \mathbb{R}^{d\times n_1}$ is the data matrix of the source domain, $W = [w_1, \cdots, w_m] \in \mathbb{R}^{d\times m}$, $U = [u_1, \cdots, u_m] \in \mathbb{R}^{d\times m}$, and $V = [v_1, \cdots, v_m] \in \mathbb{R}^{r\times m}$, then the first term in (6) can be rewritten as:

$$\begin{aligned}\sum_{l=1}^{m}\sum_{i=1}^{n_1} \ell(f_l, Y_{il}, x_i) &= \left\| W^\top X_{\mathcal{S}} - Y^\top \right\|^2 \\ &= \left\| U^\top X_{\mathcal{S}} + V^\top \Theta X_{\mathcal{S}} - Y^\top \right\|^2.\end{aligned}$$

Based on the parametric form (7) of the decision function f_l, we introduce the following regularizer:

$$\Omega(f_l) = \|u_l\|^2 = \left\| w_l - \Theta^\top v_l \right\|^2,$$

which controls the complexity of each classifier independently. The second term in (6) can be rewritten as:

$$\sum_{l=1}^{m} \Omega(f_l) = \|U\|^2 = \left\| W - \Theta^\top V \right\|^2.$$

3.1.3 Distribution Gap between Domains

Recall that the last term in (6) measures the mismatch between the embedding of the source and target domain. Here, we use the MMD criterion in (4) as the nonparametric measure for the mismatch. Suppose $X_{\mathcal{T}} = [x'_1, \cdots, x'_{n_2}] \in \mathbb{R}^{d\times n_2}$ and $X = [X_{\mathcal{S}}, X_{\mathcal{T}}] \in \mathbb{R}^{d\times(n_1+n_2)}$, are the data matrices defined on the target domain and all input data, respectively, and assume $\phi(x) = \Theta x$, and so $K = X^\top \Theta^\top \Theta X$. Then, the criterion (4) becomes

$$\begin{aligned}\mathrm{MMD}^2[D_{\mathcal{S}}, D_{\mathcal{T}}] &= \mathrm{trace}(X^\top \Theta^\top \Theta X D) \\ &= \mathrm{trace}(\Theta X D X^\top \Theta^\top). \qquad (8)\end{aligned}$$

3.1.4 Final Formulation

Combining all the above, we arrive at the following minimization problem:

$$\begin{aligned}&\min_{\Theta,W,V} \left\| W^\top X_{\mathcal{S}} - Y^\top \right\|^2 + \alpha \left\| W - \Theta^\top V \right\|^2 + \beta \mathrm{trace}(\Theta X D X^\top \Theta^\top) \\ &\text{s.t.} \quad \Theta\Theta^\top = I, \qquad (9)\end{aligned}$$

which learns both the shared subspace Θ, and the parameters W and V in decision functions simultaneously.

3.2 Detailed Algorithm

In this section, we show that the optimization problem (9) can be solved efficiently by alternatively finding the optimal subspace matrix Θ, and the matrices V and W of the weight vectors.

3.2.1 Computing V^*

First, we show that the optimal V^* in the optimization problem (9) can be expressed in term of Θ and W.

PROPOSITION 1. *For the fixed W and Θ, the optimal V^* that solves the optimization problem (9) is*

$$V^* = \Theta W. \qquad (10)$$

PROOF. Setting the derivative of the optimization problem (9) w.r.t. V to zeros, we have:

$$\Theta(W - \Theta^\top V) = 0 \quad \text{or} \quad \Theta\Theta^\top V = \Theta W.$$

Using $\Theta\Theta^\top = I$, this completes the proof. □

3.2.2 Computing W^*

Second, we show that the optimal W^* in the optimization problem (9) has a closed-form solution in term of Θ and V.

PROPOSITION 2. *For the fixed Θ and V, the optimal W^* has a closed-form solution:*

$$W = (\alpha I + X_{\mathcal{S}} X_{\mathcal{S}}^\top)^{-1}(X_{\mathcal{S}} Y + \alpha \Theta^\top V). \qquad (11)$$

PROOF. As shown in the optimization problem (9), the last term does not depend on W, so we can simplify the objective function as follows:

$$\begin{aligned}&\left\| W^\top X_{\mathcal{S}} - Y^\top \right\|^2 + \alpha \left\| W - \Theta^\top V \right\|^2 \\ &= \mathrm{trace}(W^\top X_{\mathcal{S}} - Y^\top)(W^\top X_{\mathcal{S}} - Y^\top)^\top + \alpha \mathrm{trace}(W - \Theta^\top V)(W - \Theta^\top V)^\top \\ &= \mathrm{trace}(Y^\top Y - 2W^\top(Y^\top X_{\mathcal{S}}^\top + \alpha V^\top \Theta)^\top + W^\top(\alpha I + X_{\mathcal{S}} X_{\mathcal{S}}^\top) W) \qquad (12)\end{aligned}$$

Setting the derivatives of (12) w.r.t. W to zeros, we have:

$$-(Y^\top X_{\mathcal{S}}^\top + \alpha V^\top \Theta)^\top + (\alpha I + X_{\mathcal{S}} X_{\mathcal{S}}^\top) W = 0.$$

This completes the proof. □

Since the matrix inversion $(\alpha I + X_{\mathcal{S}} X_{\mathcal{S}}^\top)^{-1}$ can be precomputed, and the data matrix $X_{\mathcal{S}}$ is usually sparse for text documents, this inversion can be computed by performing Singular Value Decomposition (SVD) on the data matrix $X_{\mathcal{S}}$ in $O(dn_1 \min(d, n_1))$ time. Using (11) and (10), the update of W can be computed in $O(d^2 m)$ time.

3.2.3 Computing Θ^*

Moreover, we can show that the optimal Θ^* in (9) can be solved efficiently by performing SVD on a matrix in term of W.

PROPOSITION 3. *For the fixed W and V, the optimal Θ^* can be obtained by solving the following SVD problem:*

$$\begin{aligned}&\min_{\Theta} \ \Theta(\beta X D X^\top - \alpha W W^T)\Theta^\top \\ &\text{s.t.} \quad \Theta\Theta^\top = I, \qquad (13)\end{aligned}$$

and the matrix Θ^* has the rank at most $\min(d, m+1)$.

PROOF. As shown in the optimization problem (9), the first term does not depend on Θ, and using (10), we can rewrite the objective function as follows:

$$\alpha \left\| W - \Theta^\top \Theta W \right\|^2 + \beta \mathrm{trace}\left(\Theta X D X^\top \Theta^\top \right).$$

Moreover, using $\Theta\Theta^\top = I$, the objective is simplified as:

$$\alpha \mathrm{trace}\left(W^\top W - W^\top \Theta^\top \Theta W \right) + \beta \mathrm{trace}\left(\Theta X D X^\top \Theta^\top \right),$$

so that we can arrive at the optimization problem (13).

Note that D in (5) can be decomposed as $D = ee^\top$, where $e \in \mathbb{R}^{n_1+n_2}$ is a vector with the first n_1 entries equal $1/n_1$ and the remaining entries equal $-1/n_2$, and so $XDX^\top$ is of rank one. Moreover, the matrix $WW^\top$ has rank at most $\min(d, m)$. Thus, the matrix $\beta XDX^\top - \alpha WW^\top$ has rank at most $\min(d, m+1)$. □

Combining all of the above, the optimization problem (9) can be solved by updating the matrices W, Θ, and V iteratively until convergence. The detailed algorithm to solve the optimization problem (9) is summarized in Algorithm 1.

Moreover, based on the Proposition 3, one can perform SVD on the low rank matrix $\beta XDX^\top - \alpha WW^\top$ to obtain the optimal Θ^* efficiently. Assuming that the input dimension is very high, *i.e.* $d \gg m$, the time complexity is $O(d^2m)$ only. The update of V takes $O(dm^2)$ time. Therefore, the overall time complexity of Algorithm 1 is only $O(d^2m)$ assuming the inverse of the matrix $(\alpha I + X_S X_S^\top)$ is pre-computed.

Algorithm 1: The Algorithm of Our Proposed Domain Adaptation

Input: labeled patterns $\{(x_i, y_i)\}_{i=1}^{n_1}$ in D_S, unlabeled patterns $\{(x'_i)\}_{i=1}^{n_2}$ in D_T, regularization parameters α and β.
Output: The optimal projection matrix Θ for feature subspace, the matrix V of weight vectors in the embedded space, and the matrix W of weight vectors of m decision classifiers.
Initialize $\Theta \leftarrow I$, $V \leftarrow 0$.
repeat
1 Update W using (11).
2 Compute Θ by solving SVD in (13).
3 Set $V = \Theta W$.
until convergence

3.3 Prediction

After extracting the shared subspace Θ, and the weight vectors w_l and v_l for each class, one can perform prediction using (7). However, the weight vector w_l is learned to minimize the empirical loss of the labeled data in the source domain D_S, and may not be the discriminative direction for the testing data in the target domain D_T.

Recall that the subspace Θ is learned to minimize the MMD criterion in (8), and captures the intrinsic structure of data for domain adaptation. Moreover, the weight vector v_l is the discriminative direction defined on the projected subspace Φ, so the prediction on the testing data in the target domain D_T can be performed by a decision classifier $f_{Tl}(x') = v_l^\top \Theta x'$ instead of $f_l(x')$ in (7), and $\Theta^\top v_l$ is the discriminative direction for the l-th class in the target domain.

3.4 Discriminative Features Propagation

However, one major problem in text mining is the sparsity of features in the high dimensional space. Specifically, some discriminative features occur frequently in the target domain D_T but seldom appear or even are absent in the source domain D_S. For example, for the task of extracting sentences corresponding to job requirements from job Web sites, some common terms may be "qualified", "year", "experience" and so on. However, some characteristic words are dependent of the job nature. For instance, "CPA", "CA", "ACCA" are the discriminative term for the "accounting" domain whereas "CCP", "physiology" are discriminative terms for the domain of "health care". To address this issue, we develop the following propagation strategy.

According to the discussion in Section 3.2, we can extract a common feature set $\mathcal{F}$ from the both domains for each specific task l by selecting the features with high weight in $\Theta' v_l$. Based on the co-occurrence information in the target domain, we can compute the similarity between the common features in the set $\mathcal{F}$ and the remaining features (non-common features) in another set $\bar{\mathcal{F}}$. For each non-common features, we can sum up its similarity with all the common features. Finally we rank all the non-common features by its similarity weight with the common feature set in the descending order. By selecting the top K high similarity non-common terms, combined with all the existing common features, we can get a set of characteristic features $\mathcal{F}_c \subset \mathcal{F} \cup \bar{\mathcal{F}}$ for the target domain.

Based on the assumption that similar features should have similar prediction power in the target domain, we can construct a feature similarity graph $\mathcal{G}$. In $\mathcal{G}$, each vertex v represents a feature, and edge weights are given by a symmetric matrix $E \in \mathbb{R}^{d \times d}$, whose entries $E_{uv} = \langle \pi_u, \pi_v \rangle \geq 0$, where $\langle \cdot, \cdot \rangle$ means the inner product, π_i represents the vector of normalized occurrence in the target domain. Define the degree of vertex v as $d_v = \sum_{u \sim v} E_{uv}$, then we can define the normalized graph Laplacian matrix:

$$\mathcal{L}_{uv} = \begin{cases} 1 - E_{uv}/d_u & \text{if } u = v \text{ and } d_u \neq 0 \\ -E_{uv}/\sqrt{d_u d_v} & \text{if } u \text{ and } v \text{ are } \textit{adjacent} \\ 0 & \text{otherwise.} \end{cases} \quad (14)$$

We also define a column vector $\rho = [\rho_1, \ldots, \rho_d]^\top \in \mathbb{R}^d$ representing the discriminative weight vector of characteristic features. Intuitively, similar features should have similar weights. Therefore, we introduce a manifold regularizer using the feature graph Laplacian matrix in (14) as:

$$\rho^T \mathcal{L} \rho = \sum_{u,v} E_{uv} \left(\frac{\rho_u}{\sqrt{d_u}} - \frac{\rho_v}{\sqrt{d_v}} \right)^2,$$

which propagates the weight of the common features to other characteristic features via the manifold structure of the feature graph.

Moreover, we also require the discriminative weight vector ρ be close to the discriminative direction learned for each class in the target domain. Thus, we arrive at the following optimization problem:

$$\min \quad \rho_l - \Theta^\top v_l \ ^2 + \gamma \rho_l^T \mathcal{L}_l \rho_l, \quad (15)$$

where the first term minimizes the difference between ρ_l and $\Theta^\top v_l$, and the second term enforces that the assignment of the weight of the characteristic features is propagated from the common features. In addition, the optimization problem (15) can be solved according to the following lemma:

PROPOSITION 4. Let v_l be the classifier for the class l on the shared feature subspace Θ, therefore the corresponding optimal ρ_l has a closed form in term of Θ and v_l.

PROOF. We first rewrite the objective function as follows:

$$\begin{aligned} & \rho_l - \Theta^\top v_l \ ^2 + \gamma \rho_l^\top \mathcal{L}_l \rho_l \\ = \ & \rho_l^\top \rho_l - 2 v_l^\top \Theta \rho_l + v_l^\top v_l + \gamma \rho_l^\top \mathcal{L}_l \rho_l \\ = \ & \rho_l^\top (I + \gamma \mathcal{L}_l) \rho_l - 2 v_l^\top \Theta \rho_l + v_l^\top v_l \end{aligned} \quad (16)$$

Setting the derivation of (16) with respect to ρ_l to zeros, we have:

$$\rho_l = (I + \gamma \mathcal{L}_l)^{-1} \Theta^\top v_l.$$

This completes the proof. □

Therefore, the prediction on the testing patterns in the target domain can be performed by:

$$f_{Tl}(x') = v_l^\top (I + \gamma \mathcal{L}_l)^{-1} \Theta x'.$$

However, computing the matrix inversion $(I + \gamma \mathcal{L}_l)^{-1}$ is still computational intensive (with complexity $O(d^3)$). Note that when the predefined parameter γ satisfies $0 < \gamma < 1$, we have the following Taylor expansion:

$$(I + \gamma \mathcal{L}_l)^{-1} = I - \gamma \mathcal{L}_l + \gamma^2 \mathcal{L}_l^2 - \gamma^3 \mathcal{L}_l^3 + ...$$

As $\mathcal{L}_l$ is usually very sparse, especially when γ is small, one can approximate $(I + \gamma \mathcal{L}_l)^{-1}$ as $I - \gamma \mathcal{L}_l$ and the revised discriminative direction is:

$$\rho_l = \Theta^\top v_l - \gamma \mathcal{L}_l \Theta^\top v_l,$$

Then the decision function on the testing patterns in the target domain becomes:

$$f_{Tl}(x') = v_l^\top (I - \gamma \mathcal{L}_l) \Theta x',$$

As a result, the computation of the prediction is much reduced.

As discussed above, $\Theta^\top v_l$ is the optimal discriminative direction of the l-th class in (9). From the propagation of the feature graph $\mathcal{G}$, the discriminative information from other characteristic features $\mathcal{F}_c$ can be used to compute the weight vector $-\gamma \mathcal{L}_l \Theta^\top v_l$ to correct the discriminative direction.

3.5 Error Analysis on Domain Adaptation

In this section, we study the error analysis of our proposed domain adaptation method in the target domain. First, we denote the labeling function in $D_\mathcal{T}$ as follows:

$$g_\mathcal{T}(x) = \begin{cases} v^T \Theta x & \text{if } 0 \le v^T \Theta x \le 1, \\ 1 & \text{if } 1 < v^T \Theta x, \\ 0 & \text{if } v^T \Theta x < 0, \end{cases}$$

and $h(x) : \mathcal{X} \to \{0, 1\}$ is the truth labeling function. Let $sigma(x)$ be a continuous loss function defined as:

$$\sigma(x) = |h(x) - g_\mathcal{T}(x)|. \tag{17}$$

The expected loss of $g_\mathcal{T}$ in $D_\mathcal{T}$ is defined as:

$$\epsilon_\mathcal{T}(h, g_\mathcal{T}(x')) = \mathrm{E}_{x' \sim D_\mathcal{T}}[|h(x') - g_\mathcal{T}(x')|] = \mathrm{E}_{x' \sim D_\mathcal{T}}[\sigma(x')].$$

Note that $f_\mathcal{S}(x) = u^\top x + v^\top \Theta x$ is the proposed decision function in (7) for the labeled data in the source domain, then we also define the expected loss of $f_\mathcal{S}$ in $D_\mathcal{S}$ as:

$$\epsilon_\mathcal{S}(h, f_\mathcal{S}(x)) = \mathrm{E}_{x \sim D_\mathcal{S}}[|h(x) - u^\top x - v^\top \Theta x|].$$

For simplicity, we denote $\mathrm{E}_{x \sim D_\mathcal{S}} = \mathrm{E}_\mathcal{P}$ and $\mathrm{E}_{x' \sim D_\mathcal{T}} = \mathrm{E}_\mathcal{Q}$. Based on the definition of $\sigma(x)$ in (17), we know that $0 \le \sigma(x) \le 1$. With a mild assumption that $\|\sigma\|_\mathcal{H}$ is bounded by a finite number C, where $\mathcal{H}$ is a RKHS, we obtain the following theorem:

THEOREM 1. *Suppose $\|x\| = 1$, the expected loss of $g_\mathcal{T}$ in $D_\mathcal{T}$ is bounded by*

$$\epsilon_\mathcal{T}(h, g_\mathcal{T}(x')) \le \epsilon_\mathcal{S}(h, f_\mathcal{S}(x)) + \mathrm{MMD}[D_\mathcal{S}, D_\mathcal{T}]C + \|u\|. \tag{18}$$

PROOF.

$$\begin{aligned} & \epsilon_\mathcal{T}(h, g_\mathcal{T}(x')) \\ = \; & \epsilon_\mathcal{S}(h, f_\mathcal{S}(x)) + \epsilon_\mathcal{T}(h, g_\mathcal{T}(x')) - \epsilon_\mathcal{S}(h, f_\mathcal{S}(x)) \\ = \; & \epsilon_\mathcal{S}(h, f_\mathcal{S}(x)) + \epsilon_\mathcal{T}(h, g_\mathcal{T}(x')) - \mathrm{E}_\mathcal{P}[|h(x) - u^\top x - v^\top \Theta x|] \\ \le \; & \epsilon_\mathcal{S}(h, f_\mathcal{S}(x)) + \epsilon_\mathcal{T}(h, g_\mathcal{T}(x')) - \mathrm{E}_\mathcal{P}[|h(x) - v^\top \Theta x|] + \mathrm{E}_\mathcal{P}[|u^\top x|] \\ \le \; & \epsilon_\mathcal{S}(h, f_\mathcal{S}(x)) + \mathrm{E}_\mathcal{Q}[|h(x') - g_\mathcal{T}(x')|] - \mathrm{E}_\mathcal{P}[|h(x) - g_\mathcal{T}(x)|] + \mathrm{E}_\mathcal{P}[|u^\top x|]. \end{aligned} \tag{19}$$

The second last inequality holds because of the triangle inequality

$$|h(x) - v^\top \Theta x| \le |h(x) - u^\top x - v^\top \Theta x| + |u^\top x|,$$

and the last inequality holds due to

$$|h(x) - g_\mathcal{T}(x)| \le |h(x) - v^\top \Theta x|.$$

Moreover, using the Cauchy-Schwarz inequality, we have:

$$\mathrm{E}_\mathcal{P}[|u^\top x|] \le \mathrm{E}_\mathcal{P}[\|u\| \|x\|] = \|u\| \mathrm{E}_\mathcal{P}[\|x\|].$$

Since $\|x\| = 1$, so that

$$\mathrm{E}_\mathcal{P}(|u^\top x|) \le \|u\|. \tag{20}$$

By the virtual of RKHS property, for any function $\sigma(x)$ in the RKHS can be expressed as $\sigma(x) = \langle \sigma, \phi(x) \rangle_\mathcal{H}$. Then, we can obtain the following bound:

$$\begin{aligned} & \mathrm{E}_\mathcal{Q}[|h(x') - g_\mathcal{T}(x')|] - \mathrm{E}_\mathcal{P}[|h(x) - g_\mathcal{T}(x)|] \\ = \; & \mathrm{E}_\mathcal{Q}[\sigma(x')] - \mathrm{E}_\mathcal{P}[\sigma(x)] \\ = \; & \mathrm{E}_\mathcal{Q}[\langle \phi(x'), \sigma \rangle_\mathcal{H}] - \mathrm{E}_\mathcal{P}[\langle \phi(x), \sigma \rangle_\mathcal{H}] \\ = \; & \langle \mathrm{E}_\mathcal{Q}[\phi(x')] - \mathrm{E}_\mathcal{P}[\phi(x)], \sigma \rangle_\mathcal{H}. \end{aligned}$$

Assume $\|\sigma\|_\mathcal{H} \le C$, similar to (1), we have:

$$\begin{aligned} \langle \mathrm{E}_\mathcal{Q}[\phi(x')] - \mathrm{E}_\mathcal{P}[\phi(x)], \sigma \rangle_\mathcal{H} & \le \|\mathrm{E}_\mathcal{Q}[\phi(x')] - \mathrm{E}_\mathcal{P}[\phi(x)]\|_\mathcal{H} C \\ & = \mathrm{MMD}[D_\mathcal{S}, D_\mathcal{T}]C. \end{aligned} \tag{21}$$

Substitute (20) and (21) into (19). This completes the proof. □

Based on the expected error bound in (18), we can conclude that minimizing the MMD in (8), the empirical loss of labeled data in the source domain $D_\mathcal{S}$, and the regularizer $\|u\|$ simultaneously as in (9) can also minimize the expected loss in the target domain $D_\mathcal{T}$.

4. EXPERIMENTS

We demonstrate the effectiveness of our proposed domain adaptation method by conducting experiments on various data sets covering two common text mining problems: document classification and information extraction.

4.1 Document Classification

4.1.1 Experiment Setup

We use the 20-Newsgroup corpus to conduct experiments on document classification. This corpus consists of 18,846 newsgroup articles harvested from 20 different Usenet newsgroups. It can be observed that the marginal distributions of the articles among different newsgroups are not identical. There exists distribution shift from one newsgroup to any other newsgroups. However, we observe that some newsgroups are related. For example, the newsgroups *rec.autos* and *rec.motorcycles* are related to *car*. The newsgroups *comp.sys.mac.hardware* and *comp.sys.ibm.pc.hardware* are related to *hardware*, etc. Table 1 depicts the detailed information of the data sets, derived from 20-Newsgroup, used in our experiments. There are four class labels, namely, *car*, *ball game*, *hardware*, and *OS*. For each class label, there are two related newsgroups, and we can select the articles in one newsgroup as labeled data in the source domain and the articles in the other newsgroup as unlabeled data in the target domain. The data sets NG1-2class, NG2-2class, and NG3-2class have only two class labels. For example, the NG1-2class data set has the class labels *car* and *ball game*. The

source domain contains 400 random articles selected from the newsgroup *rec.auto* and *rec.baseball* for the class label *car* and *ball game* respectively. There are 800 articles in total for the source domain. The target domain contains 400 random articles selected from the newsgroup *rec.motorcycle* and *rec.hockey* for the corresponding class label *car* and *ball game* respectively. There are also 800 articles in the target domain. The datasets NG4-4class and NG5-4class both have 4 class labels, namely, *car*, *ball game*, *hardware*, and *OS*. The composition of articles in each label in the source and target domains is clearly shown in Table 1. Pay attention that all the articles are represented by the vector space model and normalized to unit length.

In order to verify the effectiveness of our method, we compare with three typical classification methods: SVM, Transductive SVM, and CDSC as presented in [10]. They represent supervised classification, semi-supervised classification, and a recent domain adaptation method respectively. SVM and TSVM [9]are implemented by[2] SVM^{light} and the parameters are all set as default in the package. The parameters setting in CDSC is the same as those reported in the paper. For those three comparison algorithms, since they can only handle binary classification, we transform the multiclass problems to the 1-VS-rest problem setting for training. For each data set, we repeated all the algorithms 10 times by randomly sampling the articles in each run and calculate the average performance, so as to decrease the sampling bias.

4.1.2 Result and Discussion

We adopt the recall, precision, and F1-measure as the evaluation metrics. Recall is defined as the number of articles that are correctly classified, divided by the actual number of articles in each class. Precision is defined as the number of articles that are correctly classified, divided by the number of all the articles predicted as the same class. F1-measure is defined as the harmonic mean of recall and precision. Results of all the methods on all data sets depicted in Table 1 are summarized in Table 2 with the best results shown in bold font. It can be observed that the supervised method, namely, SVM, which trains only in the source domain and tests in the target domain always gets the worst performance among the four algorithms. Semi-supervised learning method TSVM outperforms the supervised learning method SVM by take advantages of the unlabeled data in the target domain. Because the articles in the source domain and target domain are related, then the unlabeled data in target domain will supply some distribution information for the training so as to improve the prediction in the target domain. CDSC has been reported for the good performance in two-class cross-domain adaptation. Those results are verified again in our experiments especially when the two classes in the target domain are well separated such as the data set NG3-2class. However, for multiclass problems especially when the multiple classes in the target domain are not very easy to separate such as the data set NG4-4class and NG5-4class, the performance of CDSC is not as good as that in two-class problems. On the other hand, our domain adaptation method can get comparable results with CDSC for the well separated two-class problems and achieve better performance for all the other data sets.

4.2 Information Extraction

[2]http://svmlight.joachims.org

Domain Label	Domain Name	# of Job Advertisements	# of Text Fragments
D1	Accounting	273	7462
D2	Logistic	202	5636
D3	Health	201	6402

Table 3: The details of the data collected for the information extraction experiments.

4.2.1 Experiment Setup

We conducted a set of experiments in the area of information extraction. The objective of information extraction is to extract precise chunks of consecutive tokens for each field of interest from a semi-structured text document. In our experiments, we target at extracting the job related information from Web pages in some employment Web sites. The fields of interest are *job title*, *company*, *location*, *salary*, *post-date*, *education*, *experience*, and *duty*. We have collected online job advertisement documents from recruitment Web sites in 3 different domains (or industries). Table 3 shows the details of the collected data. The first, second, and third columns refer to the domain label, domain name, and the number of job advertisements collected in the domain respectively. For each online job advertisement collected, we automatically segment the document into a number of text fragments by applying the document object model (DOM)[3] and extract the text contained in the text nodes of the DOM structure. The fourth column of the table shows the number of text fragments in the domain after segmentation. Each text fragment should be labeled as one of the eight job fields mentioned above, or the "not-a-field" label. We can observe that the distribution of the text fragments in one domain is related to the distribution in the other domains. In our experiments, For evaluation purpose, all text fragments in the three domains are manually labeled by two human accessors. If there is a disagreement on the judgment of the two human accessors, it is resolved by a discussion among them.

In each domain, we have conducted different sets of experiments to demonstrate the performance and compare with existing methods. The first set of experiment is to use the labeled training example in the source domain and the unlabeled data in the target domain to learn the extraction model using our domain adaptation method. The learned model is then applied to the testing data in the target domain and the performance is measured. For example, let D1 and D2 be the source and target domains respectively. We use the labeled training fragments in D1 and the unlabeled fragments in D2 to learn a model. Then the learned model is applied to predict the fields of the text fragments in the testing data. The other sets of experiments are designed in a similar manner as the first set. In the second set of experiments, we use transductive support vector machine for model training. As can be seen, in each training, the total number of text fragments in the source domain and target domain is larger than 10,000. Since CDSC needs to compute and store the pairwise similarity for any two fragments, it cannot handle this information extraction data set. Then we do not compare with it because of out of memory. Note that each text fragment is represented by the vector space model and normalized to unit length.

[3]The details of the document object model can be found in *http://www.w3.org/DOM*.

Data set	Domain	class label				.# doc.
		car	ball game	hardware	OS	
NG1-2class	source	rec.auto	rec.baseball	N/A	N/A	800
	target	rec.motorcycle	rec.hockey	N/A	N/A	800
NG2-2class	source	N/A	N/A	comp.sys.ibm.pc.hardware	comp.windows.x	800
	target	N/A	N/A	comp.sys.mac.hardware	comp.os.ms-windows.misc	800
NG3-2class	source	rec.auto	N/A	comp.sys.ibm.pc.hardware	N/A	800
	target	rec.motorcycle	N/A	comp.sys.mac.hardware	N/A	800
NG4-4class	source	rec.auto	rec.baseball	comp.sys.ibm.pc.hardware	comp.windows.x	1600
	target	rec.motorcycle	rec.hockey	comp.sys.mac.hardware	comp.os.ms-windows.misc	1600
NG5-4class	source	rec.motorcycle	rec.hockey	comp.sys.mac.hardware	comp.os.ms-windows.misc	1600
	target	rec.auto	rec.baseball	comp.sys.ibm.pc.hardware	comp.windows.x	1600

Table 1: The details of the data collected for the document classification experiments.

Data set	class label	SVM			TSVM			CDSC			Our approach		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
NG1-2class	car	0.788	0.960	0.867	0.812	0.966	0.884	0.841	0.985	0.907	0.912	0.985	0.947
	ball game	0.949	0.744	0.833	0.957	0.778	0.863	0.982	0.815	0.891	0.984	0.905	0.943
	average	0.869	0.852	0.850	0.884	0.869	0.774	0.912	0.900	0.899	0.948	0.945	**0.945**
NG2-2class	hardware	0.652	0.760	0.702	0.743	0.641	0.693	0.767	0.810	0.788	0.855	0.835	0.845
	OS	0.707	0.589	0.643	0.762	0.782	0.772	0.799	0.755	0.776	0.840	0.859	0.849
	average	0.680	0.674	0.672	0.753	0.713	0.732	0.783	0.783	0.782	0.847	0.847	**0.847**
NG3-2class	car	0.876	0.884	0.880	0.934	0.912	0.923	0.984	0.910	0.945	0.984	0.907	0.944
	hardware	0.874	0.885	0.879	0.916	0.937	0.927	0.916	0.985	0.949	0.914	0.985	0.948
	average	0.880	0.880	0.880	0.925	0.925	0.925	0.950	0.948	**0.947**	0.949	0.946	0.946
NG4-4class	car	0.710	0.845	0.771	0.803	0.854	0.828	0.730	0.890	0.802	0.773	0.903	0.833
	ball game	0.818	0.899	0.854	0.873	0.905	0.889	0.955	0.955	0.955	0.920	0.917	0.918
	hardware	0.637	0.630	0.634	0.669	0.633	0.650	0.633	0.700	0.665	0.819	0.792	0.805
	OS	0.623	0.441	0.517	0.666	0.633	0.649	0.815	0.550	0.657	0.796	0.692	0.741
	average	0.696	0.704	0.694	0.753	0.756	0.754	0.783	0.774	0.770	0.827	0.826	**0.824**
NG5-4class	car	0.743	0.435	0.549	0.745	0.628	0.682	0.862	0.750	0.802	0.750	0.916	0.825
	ball game	0.832	0.819	0.825	0.856	0.760	0.805	0.913	0.835	0.872	0.891	0.843	0.866
	hardware	0.552	0.715	0.623	0.550	0.697	0.615	0.577	0.820	0.678	0.763	0.779	0.771
	OS	0.507	0.574	0.538	0.579	0.577	0.578	0.623	0.495	0.552	0.734	0.596	0.658
	average	0.658	0.636	0.634	0.683	0.665	0.670	0.743	0.725	0.726	0.785	0.783	**0.780**
Average		0.757	0.749	0.746	0.800	0.786	0.791	0.834	0.826	0.825	0.871	0.870	**0.869**

Table 2: The classification performance of different sets of experiments. P, R, and F1 refer to the precision, recall, and F1-measure respectively.

4.2.2 Result and Discussion

We adopt the recall, precision, and F1-measure as the evaluation metrics. Recall is defined as the number of text fragments that are correctly labeled by our framework, divided by the actual number of text fragments. Precision is defined as the number of text fragments that are correctly labeled by our framework, divided by the number of predicted text fragments using our framework. F1-measure is defined as the harmonic mean of recall and precision.

In each set of experiments, we have conducted 6 runs using different combination of the source and target domains. Table 4 depicts the performance of the experiments. In each run, we measure the recall, precision, and F1-measure for each field. The figure in each cell of Table 4 is the average performance among the 8 fields of interest in the corresponding experiment. For example, our approach achieves an average precision, recall, and F1-measure of 0.814, 0.845, and 0.825 respectively in the target domain when the source and target domains are D1 and D2 respectively. Our approach achieves an average precision, recall, and F1-measure of 0.820, 0.802, and 0.799. It outperforms TSVM which obtains a F1-measure of 0.744.

Figure 1 depicts the detailed comparison between our method and TSVM. The x-axis denotes the eight job fields and the y-axis denotes the extraction performance measured by F-measure. In each plot, we show the F-measure on each job field when training in one domain and adapt to the other two domains. For example, "TSVM-D1-D2" and "Our-D1-D2" represent the result of TSVM and our method respectively on the data set in which D1 is the source domain and D2 is the target domain. It can be observed that our domain adaptation method can get better extraction performance than TSVM in almost all of the fields in each data set.

Experiment Setting		TSVM			Our Approach		
Source Domain	Target Domain	P	R	F1	P	R	F1
D1	D2	0.730	0.815	0.759	0.814	0.845	**0.825**
D1	D3	0.717	0.771	0.731	0.813	0.804	**0.800**
D2	D1	0.782	0.772	0.766	0.866	0.789	**0.807**
D2	D3	0.782	0.796	**0.770**	0.830	0.762	0.765
D3	D1	0.742	0.739	0.731	0.790	0.789	**0.779**
D3	D2	0.727	0.784	0.737	0.793	0.791	**0.786**
Average		0.743	0.775	0.744	0.820	0.800	**0.799**

Table 4: The extraction performance of different sets of experiments. P, R, and F1 refer to the precision, recall, and F1-measure respectively.

5. CONCLUSIONS

In this paper, we present a domain adaptation method by extracting the shared concept space between the source domain with sufficient labeled data and the target domain with a large amount of unlabeled data. In our method, we parameterize the shared space by a linear transformation and finding the optimal solution by considering the combination of the two criteria: the empirical loss on the source domain, and the embedded distribution gap between the

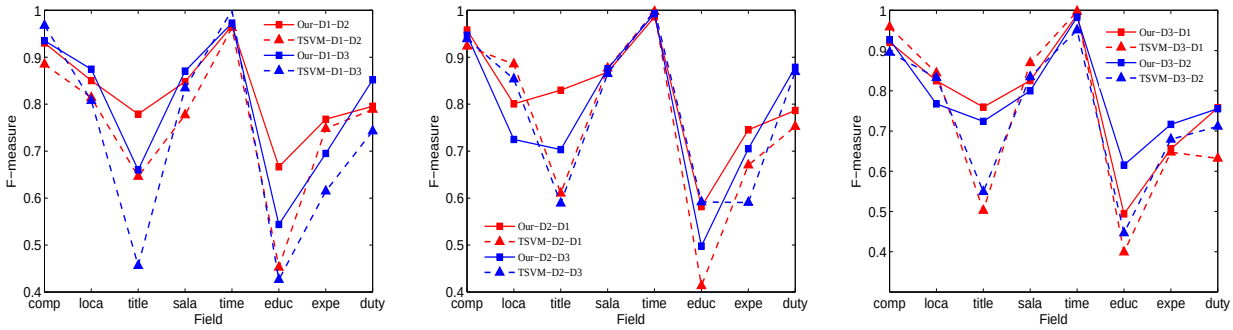

Figure 1: Comparison of the extraction performance of each job field with different source domain and target domain. From left to right: the source domain is D1, D2, and D3 respectively. The fields in the x-axis from left to right are company, location, job title, salary, post-time, education, experience and duty.

source domain and the target domain. Theoretical analysis of the adaption error bound in the target domain shows that it can be well controlled by the criteria in our objective function. Experimental results on document classification and information extraction demonstrate that our method can outperform other competitive methods in the domain adaptation setting.

In the future, we will extend our method to extract discriminative concepts in multiple source domain adaptation problems. Exploration of other domain knowledge for extracting the more discriminative concepts is also one of major directions to our domain adaptation method.

6. ACKNOWLEDGMENTS

The work described in this paper is substantially supported by grants from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project No: CUHK4128/07) and the Direct Grant of the Faculty of Engineering, CUHK (Project Codes: 2050391 and 2050442). This work is also affiliated with the Microsoft-CUHK Joint Laboratory for Human-centric Computing and Interface Technologies.

7. REFERENCES

[1] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 12:2399–2434, 2006.

[2] S. Bickel, C. Sawade, and T. Scheffer. Transfer learning by distribution matching for targeted advertising. In *Advances in Neural Information Processing Systems 21*, 2009.

[3] J. Blitzer, R. McDonald, and F. Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 120–128, 2006.

[4] H. Daumé III. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 256–263, June 2007.

[5] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkolpf, and A. Smola. A kernel method for the two-sample problem. In *Advances in Neural Information Processing Systems 19*, pages 513–520, 2007.

[6] J. Huang, A. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems 19*, pages 601–608, 2007.

[7] S. Ji, L. Tang, S. Yu, and J. Ye. Extracting shared subspace for multi-label classification. In *The Fourteenth ACM SIGKDD International Conference On Knowledge Discovery and Data Mining*, pages 381–389, 2008.

[8] J. Jiang and C. Zhai. Instance weighting for domain adaptation in NLP. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 264–271, 2007.

[9] T. Joachims. *Learning to Classify Text Using Support Vector Machines – Methods, Theory, and Algorithms.* Kluwer/Springer, 2002.

[10] X. Ling, W. Dai, G.-R. Xue, Q. Yang, and Y. Yu. Spectral domain-transfer learning. In *The Fourteenth ACM SIGKDD International Conference On Knowledge Discovery and Data Mining*, pages 488–496, 2008.

[11] S. J. Pan, J. T. Kwok, and Q. Yang. Transfer learning via dimensionality reduction. In *Proceedings of the 23rd AAAI conference on Artificial Intelligence*, pages 677–682, 2008.

[12] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng. Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th Annual International Conference on Machine Learning*, pages 759–766, 2007.

[13] S. Satpal and S. Sarawagi. Domain adaptation of conditional probability models via feature subsetting. In *Proceedings of European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 224–235, 2007.

[14] A. Storkey and M. Sugiyama. Mixture regression for covariate shift. In *Advances in Neural Information Processing Systems 19*, pages 1337–1344, 2007.

[15] M. Sugiyama, S. Nakajima, H. Kashima, P. von Bunau, and M. Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in Neural Information Processing Systems 20*, 2008.