Transfer Feature Representation via Multiple Kernel Learning

Wei Wang\textsuperscript{1}, Hao Wang\textsuperscript{1,2}, Chen Zhang\textsuperscript{1}, Fanjiang Xu\textsuperscript{1}
1. Science and Technology on Integrated Information System Laboratory
2. State Key Laboratory of Computer Science
Institute of Software, Chinese Academy of Sciences, Beijing 100190, China
weiwangpenny@gmail.com

Abstract

Learning an appropriate feature representation across source and target domains is one of the most effective solutions to domain adaptation problems. Conventional cross-domain feature learning methods rely on the Reproducing Kernel Hilbert Space (RKHS) induced by a single kernel. Recently, Multiple Kernel Learning (MKL), which bases classifiers on combinations of kernels, has shown improved performance in the tasks without distribution difference between domains. In this paper, we generalize the framework of MKL for cross-domain feature learning and propose a novel Transfer Feature Representation (TFR) algorithm. TFR learns a convex combination of multiple kernels and a linear transformation in a single optimization which integrates the minimization of distribution difference with the preservation of discriminating power across domains. As a result, standard machine learning models trained in the source domain can be reused for the target domain data. After rewritten into a differentiable formulation, TFR can be optimized by a reduced gradient method and reaches the convergence. Experiments in two real-world applications verify the effectiveness of our proposed method.

Introduction

Conventional supervised learning has been successfully applied to various fields based on the assumption that there are plenty of labeled training samples following the same distribution of test samples. However, in many real-world applications, label information is expensive or even impossible to be obtained in a target domain. In this case, one may turn to collect labeled data from a related but different domain, i.e., source domain, as prior knowledge. Apparently, classifiers trained only in source domain cannot be directly reused in target domain due to the distribution difference. Recently, there is an increasing interest in developing feature representation methods (Long et al. 2012; Long et al. 2013; Saenko et al. 2010; Pan, Kowok, and Yang 2008; Pan et al. 2011) for knowledge transfer between domains (Caruana 1997; Pan and Yang 2010).

Generally, the key challenge in cross-domain feature learning is to explicitly minimize the distribution difference between domains while preserving the important properties (e.g., variance or geometry) of data. According to whether the feature representation is linear or nonlinear, these methods can be classified into two categories. The first category explores a linear transformation as a bridge for information transfer between domains. For example, Geodesic Flow Kernel (GFK) (Gong et al. 2012) integrates an infinite number of linear subspaces to learn domain invariant feature representation. Joint Distribution Adaptation (JDA) (Long et al. 2013) constructs cross-domain feature subspace under Principal Component Analysis (PCA) (Jolliffe 1986). In (Wang et al. 2014), a shared Mahalanobis distance is optimized based on information theory. Although these methods can be optimized efficiently, they may lose the capability to capture the high order statistics and the underlying structures of complex data spaces. The second category explores a nonlinear transformation. Specifically, a heuristic nonlinear map (Daumé III 2007) is constructed in the supervised case. Maximum Mean Discrepancy Embedding (MMDE) (Pan, Kowok, and Yang 2008) learns a nonparametric kernel matrix by preserving the data variance. However, these methods are limited to the transductive setting and the optimization is computationally expensive. To overcome the drawbacks above, the nonlinear method Transfer Component Analysis (TCA) (Pan et al. 2011) applies the empirical kernel map (Schölkopf, Smola, and GengMüller 1998) of a single predefined kernel function and learns some transfer components. It is shown that the performance of TCA heavily depends on the choice of the single kernel.

Recently, Multiple Kernel Learning (MKL), which bases Support Vector Machine (SVM) or other kernel methods on combinations of kernels, emphasizes the need of learning multiple kernels instead of fixing a single kernel in the literature (Bach, Lanckriet, and Jordan 2004; Lin, Liu, and Fuh 2011; Kim, Magnani, and Boyd 2006; Rakotomamonjy et al. 2008). Although these methods are shown to achieve improved performance and flexibility, they are not explicitly developed for domain adaptation. When training and test data are drawn from different domains, the distribution difference will make the optimal kernels learnt in source domain invalid in target domain.

In this paper, we make great efforts to alleviate the limitations discussed above and propose a novel Transfer Feature Representation (TFR) algorithm. TFR selects a convex com-
Combination of multiple kernels to induce a Reproducing Kernel Hilbert Space (RKHS) and then learns a cross-domain linear transformation in this space. Overall, TFR is distinguished by three main contributions. Firstly, to the best of our knowledge, TFR has made the first attempt to incorporate MKL with cross-domain feature learning. Finding an optimal way to combine multiple kernels definitely benefits the exploration of prior knowledge and the description of data characteristics. Moreover, the diversity of kernel functions adopted in TFR will further improve the flexibility and effectiveness. Secondly, in TFR, the linear transformation and the multiple kernels are learnt in a single optimization, such that standard machine learning models (e.g., classification and regression) trained in source domain can be reused in target domain. In the formulation of TFR, three conditions are taken into account: 1) minimizing the distribution difference; 2) preserving the label information of source domain data. Thirdly, instead of using alternate optimization, we rewrite TFR problem into a differentiable formulation with constraints on the multiple kernel weights. The differentiation of this formulation wraps a positively semi-definite (PSD) matrix optimization and we prove that this optimization has a closed-form solution. Following (Rakotomamonjy et al. 2008), reduced gradient method is employed to iteratively update the weights and the PSD matrix simultaneously, leading to rapid convergence. Experimental results verify the effectiveness of TFR compared with state-of-the-art transfer learning methods.

Relative Works

Maximum Mean Discrepancy

To measure the distance between data distributions, many parametric criteria (e.g., Kullback-Leibler divergence) have been proposed by assuming or estimating the detailed distribution formats. To avoid such a nontrivial task, Borgwardt et al. propose Maximum Mean Discrepancy (MMD) (Borgwardt et al. 2006) for comparing distributions in the Reproducing Kernel Hilbert Space (RKHS) \( \mathcal{H} \) induced by a kernel \( K \). Given a series of observations \( X = \{x_1, ..., x_{n_1}\} \) drawn from the distribution \( P \) and observations \( Z = \{z_1, ..., z_{n_2}\} \) drawn from \( Q \), the empirical estimate of MMD is:

\[
\frac{1}{n_1} \sum_{i=1}^{n_1} \phi(x_i) - \frac{1}{n_2} \sum_{i=1}^{n_2} \phi(z_i) \|_{\mathcal{H}},
\]

where \( \phi : \mathbb{R}^{d} \rightarrow \mathcal{H} \) with \( K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle \). Therefore, the distance between \( P \) and \( Q \) can be estimated as the distance between the data means in \( \mathcal{H} \).

Multiple Kernel Learning

Instead of directly learning a nonparametric kernel matrix or using a single predefined kernel function, MKL considers that the learnt kernel function is a convex combination of given (basis) kernels (Bach, Lanckriet, and Jordan 2004):

\[
K(x_i, x_j) = \sum_{m=1}^{M} d_m K_m(x_i, x_j), \quad d_m \geq 0, \sum_{m=1}^{M} d_m = 1,
\]

where \( K_m \) can simply be classical kernels (e.g., RBF kernels) with different parameters. Following Equation (2), many MKL methods have been proposed to learn the weights \( d_m \) and a specific machine learning model (e.g., SVM) simultaneously (Gönen and Alpaydin 2008; Rakotomamonjy et al. 2008). It is shown that the ensemble kernel based on the learnt weights \( d_m \) can achieve better performance than a single basis kernel or the average kernel. However, these methods are limited by the underlying assumption that training data and test data are drawn from the same distribution.

We would also like to mention that Jie et al. (Jie, Tommasi, and Caputo 2011) and Duan et al. (Duan, Tsang, and Xu 2012) learn MKL-based classifiers to address cross-domain problems. In contrast, our method incorporates MKL with transfer feature learning by optimizing a linear transformation and kernel weights at the same time. Therefore, standard machine learning models trained in the source domain can be directly used for the target domain data.

Transfer Feature Representation via Multiple Kernel Learning

In this section, we introduce the proposed Transfer Feature Representation (TFR) algorithm in detail.

Problem Definition

Denote \( X_{src} \) as a set of \( n_1 \) labeled training samples drawn from the source domain: \( X_{src} = \{(x_1^s, y_1^s), ..., (x_{n_1}^s, y_{n_1}^s)\} \), where \( x_i^s \in \mathbb{R}^{d} \) and \( y_i^s \in Y \) is the class label. Denote \( X_{tar} \) as a set of \( n_2 \) unlabeled testing samples drawn from the target domain: \( X_{tar} = \{(x_1^t, ..., x_{n_2}^t)\} \), where \( x_i^t \in \mathbb{R}^{d} \). Denote \( X = X_{src} \cup X_{tar} = \{x_1^s, ..., x_{n_1}^s, x_1^t, ..., x_{n_2}^t\} \in \mathbb{R}^{d \times N} \) with \( N = n_1 + n_2 \). Let \( P_t(X_{tar}) \) and \( P_s(X_{src}) \) be the marginal probability distributions of \( X_{src} \) and \( X_{tar} \), respectively, and \( P_t(X_{tar}) \neq P_s(X_{src}) \).

Suppose a kernel function \( K \) is constructed by multiple bases kernels as Equation (2). Denote \( \phi : \mathbb{R}^{d} \rightarrow \mathcal{H} \) as the nonlinear map induced by \( K \). The task of TFR is to learn the optimal weights \( d_m \) and a shared linear transformation \( W \) in \( \mathcal{H} \) simultaneously by: 1) explicitly reducing the distribution difference between \( P_t(W_{\phi}(X_{src})) \) and \( P_t(W_{\phi}(X_{src})) \); 2) preserving the geometry of \( X_{tar} \); 3) preserving the label information of \( X_{src} \). This task adopts multiple kernels rather than a single one to precisely characterize data from different aspects, where various existing kernel functions can be applied as bases. The learnt kernel weights and transformation optimally transfer the discriminating power gained from the source domain to the target domain, that is, the same labeled points are kept close and the differently labeled points are pushed far apart.

Reducing Mismatch of Data Distribution

MMD has been widely employed in both linear and nonlinear transfer feature learning methods for measuring the distribution difference. In this section, we first revisit the ideas behind these two kinds of methods, and then propose a linear transformation in RKHS as an integration of these methods.

Linear methods explore a transformation \( W \) or a Mahalanobis distance \( A \) across two domains. Following Equation (1), the distance between \( P_s(WX_{src}) \) and \( P_t(WX_{tar}) \) is...
measured by the squared distance between the sample means in the two domains (Long et al. 2013; Wang et al. 2014):
\[
\| \frac{1}{n_1} \sum_{i=1}^{n_1} W x_i - \frac{1}{n_2} \sum_{i=1}^{n_2} W x_i \|^2 = tr(XX^T A),
\]
(3)
where \( W^T W = A \in \mathbb{R}^{d \times d} \) is PSD. \( L = [L_{ij}] \) with \( L_{ij} = \frac{1}{n} \) if \( x_i, x_j \in X_{src} \); \( L_{ij} = \frac{1}{n} \) if \( x_i, x_j \in X_{tar} \); otherwise \( L_{ij} = -\frac{1}{\sqrt{n_1 n_2}} \). Although these linear subspace learning methods can be efficiently optimized and easily generalized to new data points, they show limitations in capturing the high order statistics and the underlying structures of complex data spaces.

A nonlinear transformation method is proposed by Pan et al. (Pan, Kowok, and Yang 2008) to learn a kernel matrix \( \mathbf{K} \) instead of explicitly finding the corresponding \( \phi \) (\( \mathbf{K}(x_i, x_j) = \phi(x_i)^T \phi(x_j) \)), where the squared distance between the sample means in the two domains is:
\[
\| \frac{1}{n_1} \sum_{i=1}^{n_1} \Phi(x_i^s) - \frac{1}{n_2} \sum_{i=1}^{n_2} \Phi(x_i^d) \|^2 = tr(KL),
\]
(4)
where \( \Phi \in \mathbb{R}^{N \times N} \). Although this nonlinear method benefits from finding the inner structure of data and the correlation between different data points, it has to solve an expensive semi-definite programming problem (Boyd and Vandenberge 2004) and is limited to the transductive setting.

It is a natural idea that Equation (3) and Equation (4) can be combined together to take use of their advantages and alleviate their limitations. Suppose \( X_{src} \) and \( X_{tar} \) are mapped to a Reproducing Kernel Hilbert Space (RKHS) \( \mathcal{H} \) by a nonlinear map \( \phi : \mathbb{R}^d \rightarrow \mathcal{H} \). We focus on learning a linear transformation \( W \) in this RKHS \( \mathcal{H} \). In this term, the distance between \( P_x(W \phi(x_{src})) \) and \( P_y(W \phi(x_{tar})) \), denoted as \( Dist(W \phi(x_{src}), W \phi(x_{tar})) \), is given as follows:
\[
\| \frac{1}{n_1} \sum_{i=1}^{n_1} W \phi(x_i^s) - \frac{1}{n_2} \sum_{i=1}^{n_2} W \phi(x_i^d) \|^2 = tr(W \Phi L \Phi^T W^T),
\]
(5)
where \( \Phi = \{ \phi(x_1^s), ..., \phi(x_{n_1}^s), \phi(x_1^d), ..., \phi(x_{n_2}^d) \} \). Following (Schölkopf, Smola, and Gengmüller 1998), \( W \) in \( \mathcal{H} \) can be parameterized as a linear combination of data points, i.e., \( W = U \Phi^T \). Substituting \( W \) to Equation (5), we obtain
\[
Dist(W \phi(x_{src}), W \phi(x_{tar})) = tr(U \Phi^T \Phi L \Phi^T U^T) = tr(KL \hat{A}),
\]
where \( \hat{A} = U^T U \in \mathbb{R}^{N \times N} \) is PSD and \( \mathbf{K} = \Phi^T \Phi \). Instead of specifying \( \mathbf{K} \) in Equation (6) as a single predefined kernel function, we define \( \mathbf{K} \) as a convex combination of \( M \) positive-definite kernels. The adoption of multiple kernels is a feasible way for improving performance and robustness in transfer learning. Following the definition of \( \mathbf{K} \) in Equation (2), the problem in Equation (6) can be rewritten as:
\[
tr(KL \hat{A}) = tr(\sum_{m=1}^{M} d_m K_m^T L(\sum_{m=1}^{M} d_m K_m) \hat{A}],
\]
(7)
where \( K_m = \left( K_m^s, K_m^d \right), K_m^s \in \mathbb{R}^{n_1 \times N} \) and \( K_m^d \in \mathbb{R}^{n_2 \times N} \) are the kernel matrices induced by \( K_m \) in the source and the target domains. Within this framework, the problem of minimizing the distance between data distributions defined in Equation (5) is now reduced to the choice of weights \( d_m \) and the learning of PSD matrix \( \hat{A} \). Once \( d_m \) and \( \hat{A} \) are obtained, the distance between a new test point \( x_p \) and any training point \( x_i \) can be computed as follows:
\[
d(W \phi(x_p), W \phi(x_i)) = (k_p - k_i)^T \hat{A}(k_p - k_i),
\]
(8)
where \( k_i = \Phi^T \phi(x_i) = \sum_m d_m [k_m(x_i^s, x_i), ..., k_m(x_i^d, x_i)]^T \) and \( k_p = \Phi^T \phi(x_p) = \sum_m d_m [k_m(x_p^s, x_p), ..., k_m(x_p^d, x_p)]^T \). In this term, our proposed feature learning method generalizes to out-of-sample patterns.

Preserving Properties of \( X_{tar} \) and \( X_{src} \)

For discriminating power transfer from source domain to target domain, some important properties of data should be preserved. Inspired by (Wang et al. 2014), we combine minimizing the distribution difference with: 1) preserving the geometry of \( X_{tar} \); 2) preserving the label information of \( X_{src} \).

For preserving the geometry of \( X_{tar} \), a diffusion kernel \( K_T \) (Kondor and Lafferty 2002) is defined on a weighted graph structure \( G' \) with the adjacency matrix \( M' = [M'_{ij}] \):
\[
M'_{ij} = \exp(||x_i^s - x_j^s||^2/2\sigma^2) \text{ if } x_i^s \text{ is one of the } k \text{ nearest neighbor of } x_j^s; \text{ otherwise } M'_{ij} = 0. \text{ Let } D' \text{ be an } n_2 \times n_2 \text{ diagonal matrix with } D'_{ii} = \sum_j M'_{ij}. \text{ The Laplacian of } G' \text{ can be defined as } L' = D' - M', \text{ and the Normal-}
\]
ized Laplacian is \( L' = L'D'^{-1/2}, D'^{-1/2}L'D'^{-1/2} \). The eigenvalues and eigenvectors of \( L' \) are denoted as \( \lambda' \) and \( \phi' \), so that \( L' = \sum_{i=1}^{n_2} \lambda'_i (\phi'_i)^T (\phi'_i)^T \). In this term, \( K_T \) is defined as:
\[
K_T = \sum_{i=1}^{n_2} \exp(-\sigma_d^2/2\lambda'_i)(\phi'_i)^T (\phi'_i)^T, \text{ where } \sigma_d \text{ is the width of the diffusion kernel.}
\]
We construct a linear kernel \( K_T^s \) for \( W \phi(X_{src}) \):
\[
K_T^s = \sum_{m=1}^{M} d_m K_m^s \hat{A} \sum_{m=1}^{M} (d_m K_m^s)^T
\]
(9)
Based on the criterion defined in (Wang and Jin 2009), the distance between \( K_T \) and \( K_T^s \) can be expressed as:
\[
d(K_T^s||K_T) = \frac{1}{2}(tr(K_T^{-1} K_T^s) - log|K_T^s| + log|K_T| - n_2)
\]
\[
= \frac{1}{2}(tr(\sum_{m=1}^{M} d_m K_m^T K_m^{-1} K_m^T \hat{A} \sum_{m=1}^{M} d_m K_m) - log(\sum_{m=1}^{M} d_m K_m^T \hat{A} \sum_{m=1}^{M} d_m K_m) + log|K_T| - n_2).
\]
(10)
For preserving discriminating information of \( X_{src} \), a diffusion kernel \( K_S \) is defined on a weighted graph structure \( G' \) with the adjacency matrix \( M' = [M'_{ij}] \):
\[
M'_{ij} = \exp(||x_i^s - x_j^s||^2/2\sigma^2) \text{ if } y_i^s = y_j^s; \text{ otherwise } M'_{ij} = 0. \text{ In this term, } K_S = \sum_{i=1}^{n_2} \exp(-\sigma_d^2/2\lambda'_i)(\phi'_i)^T (\phi'_i)^T, \text{ where } \lambda'_i \text{ and } \phi'_i \text{ are eigenvalues and eigenvectors of the Normalized Laplacian of } G'. \text{ The distance between } K_S \text{ and } K_T \text{ (the lin-}

ear kernel of $\mathbf{W}\phi(\mathbf{X}_{src})$ is obtained as:
\[
d(\mathbf{K}^*_i || \mathbf{K}_S) = \frac{1}{2}(tr(\mathbf{K}^*_i^{-1}\mathbf{K}^*_i) - log|\mathbf{K}^*_i| + log|\mathbf{K}_S| - n_1)
\]
\[
= \frac{1}{2}(tr((\sum_{m=1}^M d_m\mathbf{K}^*_m)^T \mathbf{K}^*_S^{-1}(\sum_{m=1}^M d_m\mathbf{K}^*_m)\tilde{\mathbf{A}})
\]
\[
- \sum_{m=1}^M d_m\mathbf{K}^*_m\tilde{\mathbf{A}}(\sum_{m=1}^M d_m\mathbf{K}^*_m)^T | + log|\mathbf{K}_S| - n_1).
\]
(11)

**Learning Algorithm**

**Cost Function** By minimizing Equation (8), Equation (10) and Equation (11) simultaneously, we propose a novel nonlinear projection method for domain adaptation. Optimizing this overall objective function is challenging since it is not convex. Standard approaches, such as the alternate calculation of the gradient of $J$ and the existence and the calculation of the gradient of $\mathbf{K}$ with the constraint $\{d_i \sum_{m=1}^M d_m = 1, d_m > 0\}$. This procedure does converge to the local minimum of differentiable function $J(\cdot)$ (Luenberger 1984). Specifically, once the gradient in Equation (15) is obtained, $d$ is updated in a descent direction to ensure the equality and the positivity constraints. Denote $u$ as the index of the largest component of $d$ (i.e., $u = \arg \max_m d_m$). Following (Rakotomamonjy et al. 2008), the reduced gradient descent for updating $d$ is:
\[
D_m = \begin{cases}
0 & \text{if } d_m = 0 \& \sum_{i \neq u} \frac{\partial J}{\partial d_i} - \frac{\partial J}{\partial d_u} > 0 \\
\sum_{i \neq u} \frac{\partial J}{\partial d_i} - \frac{\partial J}{\partial d_u} & \text{if } m = u.
\end{cases}
\]
(16)

The overall procedure of the proposed method is summarized in Algorithm 1. In practice, $M$ with a relatively small value (e.g., $M = 11$ in our experiments) can generally guarantee satisfying results. The computational complexity of TFR is $O(T_{max} \times N^3)$, where $T_{max}$ is the number of iterations in Algorithm 1. Experiments show that TFR converges rapidly (generally converges less than five iterations).

**Algorithm 1 Transfer Feature Representation**

**Initialization:**
set $\{d_m\}_{m=1}^M$ with random admissible values.

**Iteration:**
1: while not convergence do
2: Compute $\hat{\mathbf{A}}^*$ with $\mathbf{K} = \sum_{m=1}^M d_m\mathbf{K}_m.$
3: Compute $\frac{\partial J}{\partial d_m}$ and the descent direction $D_m.$
4: $d \leftarrow d + \gamma D,$ where $\gamma$ is the step size.
5: end while
Experiments

In this section, we evaluate our method TFR in two cross-domain learning related applications: 1) face classification and 2) text classification.

Experiment Setup

The proposed method TFR is systematically compared with five state-of-the-art feature-based transfer learning methods including: 1) linear ones: Joint Distribution Adaptation (JDA) (Long et al. 2013), Transferred Fisher’s Linear Discriminant Analysis (TrFLDA) (Si, Tao, and Geng 2010) and Cross-Domain Metric Learning (CDML) (Wang et al. 2014); 2) nonlinear ones: Transfer Component Analysis (TCA) (Pan et al. 2011) and Semi-supervised Transfer Component Analysis (SSTCA) (Pan et al. 2011). Meanwhile, we report the results of metric learning method Information Geometry Metric Learning (IGML) (Wang and Jin 2009). In the experiments, 1-nearest neighbor classifier (1-NN) is used as the base classifier without parameters tuning.

For the comparison methods, their parameters spaces are empirically searched by cross validation and the best results are reported. TFR involves four parameters: $\sigma_d$, $\sigma$, $\lambda$ and $k$. Specifically, we search $\sigma_d$ based on the validation set in the range $\{0.1, 1, 10\}$, $\sigma$ in the range $\{0.01, 0.1, 1, 10, 100\}$ and $\lambda$ in the range $\{0.1, 1, 10\}$. Across the experiments, the performance of TFR is stable for a wide range of these parameters. The neighborhood size $k$ for TFR is 3. Basis kernel functions are predetermined for TFR: linear kernel and Gaussian kernels with 10 different bandwidths, i.e., 0.5, 1, 2, 5, 7, 10, 12, 15, 17, 20. SSTCA and TCA are evaluated with these 11 basis kernels respectively and the best results are reported. To evaluate the effectiveness of kernel weights learning in TFR, TFR also applies these 11 basis kernels with the average weights (denoted as TFR-avg).

Cross-Domain Face Classification

Data Preparation

FERET (Phillips et al. 2000) and YALE (Belhumeur, Hespanha, and Kriegman 1997) are two public face data sets. FERET data set contains 13,539 face images from 1,565 individuals, where each image is collected with different poses, illuminations and facial expressions. YALE data set has 165 images from 15 individuals. Some example face images are shown in Figure 2.

Following the previous works (Si, Tao, and Geng 2010; Wang et al. 2014), two cross-domain data sets are constructed: 1) $Y \leftrightarrow F$: the source domain set is YALE, and the target domain set consists of 100 individuals randomly selected from FERET. 2) $F \leftrightarrow Y$: the source set contains 100 individuals randomly selected from FERET, and the target set is YALE. The dimensionality of each image is reduced to 100 by PCA, where 99.99% energy is preserved.

Results of Face Classification

The training data set contains all the labeled source domain data and some randomly selected unlabeled target domain data, while the test data set contains the rest of unlabeled target domain data. FERET and YALE have different classes. At the test stage, the label of the test point is predicted as that of the nearest target domain training data using the learnt transformation. Note that the label information of target domain data is available only at the test stage. Figure 2 shows the misclassification rates versus a varying number of unlabeled target domain training data. The results are obtained by averaging over 10 runs. JDA is inapplicable which requires that source and target domains share the same class. TrFLDA has only part of the results, since it suffers from numerical problems when there is not enough target domain training data.

Some observations can be concluded from Figure 2. Firstly, on $Y \leftrightarrow F$, metric learning algorithm IGML and baseline Euclidean show their limits for cross-domain tasks. Secondly, on $F \leftrightarrow Y$ which has a small target domain data set, IGML performs much better than TCA, SSTCA, CDML and TrFLDA. That is, when source domain and target domain have different classes, these methods fail to separate different classes for target domain data by learning a transformation from labeled source domain data and a few unlabeled

![Figure 2: Image examples from (a) FERET and (b) YALE.](image-url)
Table 1: 1-NN classification errors (in percent) with varying size of target domain data in the training process on (from top to bottom) Reuters-21578 and 20-Newsgroups data sets.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>ors vs people</th>
<th>ors vs place</th>
<th>people vs place</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size (%)</td>
<td>30</td>
<td>60</td>
<td>70</td>
</tr>
<tr>
<td>Euclidean</td>
<td>50.50</td>
<td>51.16</td>
<td>51.10</td>
</tr>
<tr>
<td>TCA</td>
<td>48.57</td>
<td>45.58</td>
<td>40.49</td>
</tr>
<tr>
<td>SSTCA</td>
<td>48.84</td>
<td>45.96</td>
<td>49.54</td>
</tr>
<tr>
<td>CDML</td>
<td>51.00</td>
<td>46.07</td>
<td>46.49</td>
</tr>
<tr>
<td>TDA</td>
<td>47.52</td>
<td>46.17</td>
<td>48.90</td>
</tr>
<tr>
<td>IGML</td>
<td>47.08</td>
<td>47.62</td>
<td>46.31</td>
</tr>
<tr>
<td>TFR</td>
<td><strong>43.71</strong></td>
<td><strong>44.93</strong></td>
<td><strong>45.07</strong></td>
</tr>
<tr>
<td>TFR-avg</td>
<td>47.85</td>
<td>45.13</td>
<td>46.38</td>
</tr>
</tbody>
</table>

Target domain data. Thirdly, although TFR-avg is quite effective here, the ensemble kernel based on the learnt weights in TFR provides higher accuracy. In general, TFR achieves the lowest error rate across all the data sizes, which illustrates its effectiveness in separating different target domain classes even when source domain and target domain has different class numbers.

Cross-Domain Text Classification

Data Preparation  Reuters-21578 and 20-Newsgroups are two benchmark text data sets which are widely used for evaluating the transfer learning algorithms (Dai et al. 2007b; Li, Jin, and Long 2012; Pan et al. 2011). These data sets are organized in a hierarchical structure by different top categories and different subcategories. Data from different subcategories under the same top category is related. Following this strategy, three cross-domain data sets are constructed based on Reuters-21578: ors vs people, ors vs place and people vs place; six cross-domain data sets are constructed based on 20-Newsgroups: comp vs rec, comp vs sci, comp vs talk, rec vs sci, rec vs talk and sci vs talk.

Results of Text Classification  For Reuters-21578, training data set contains all the labeled source domain data and randomly selected (50%, 60%, 70%, 80% or 90%) unlabeled target domain data. For 20-Newsgroups, training data set contains randomly selected 50% labeled source domain data and randomly selected (30%, 40% or 50%) unlabeled target domain data. At the test stage, the remaining unlabeled target instances are compared to the points in the labeled source domain using the learnt transformation. We compare our proposed TFR, TFR-avg with TCA, SSTCA, CDML and JDA, where TrFLDA is inapplicable for this binary classification task. Euclidean is used as the baseline. The classification results across different training data sizes are shown in Table 1 by averaging over 10 runs.

Some observations can be concluded from the results. The first general trend is that the kernel-based nonlinear transformation methods TCA, SSTCA, TFR and TFR-avg always outperform the linear methods CDML and JDA, showing the advantage of nonlinear learning in domain adaptation problems. The second general trend is that the results of non-transfer metric learning method IGML are better than that of the transfer algorithms on rec vs talk. A possible explanation is that the distributions of source and target data are not significantly varied on this data set. But we would like to mention that the transfer methods perform well on other cross-domain data sets. The third general trend is that TFR achieves the minimal error rate on most of the data sets, which illustrates its reliable and effective performance by optimally combining multiple predefined kernel functions.

Convergence  We have proven that TFR does converge to the local minimum. In this section, we employ data set ors vs people to evaluate the convergence efficiency. As shown in Figure 2(c), the objective values of TFR converge after less than five iterations. We have similar observations for other data sets, details are not given due to the lack of space.

Conclusion  In this paper, we have proposed a novel feature representation algorithm to address domain adaptation problem based on multiple kernel learning. It differs from the existing approaches in that using a convex combination of basis kernels can better explore prior knowledge and describe underlying data characteristics. An efficient learning algorithm, based on reduce gradient, is employed to simultaneously learn the linear transformation and the kernel weights. As a result, the discriminating power gained from the source domain is optimally transferred to the target domain. Experimental results in two real-world applications demonstrate the advantages of our method. In future work, we plan to add low-rank constraint to the linear transformation and find the optimal low dimensional space for domain adaptation problem.
Acknowledgments
This work is supported by Natural Science Foundation of China (61303164) (61402447), National Basic Research Program of China (2013CB329305) and Beijing Natural Science Foundation (9144037). This work is also sponsored by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry.

References


