Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

Transfer Feature Representation via Multiple Kernel Learning

Wei Wang[1], **Hao Wang**[1,2], **Chen Zhang**[1], **Fanjiang Xu**[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-

bination 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 geometry of target domain data; 3) 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 converge. 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 $\mathbf{K}$. Given a series of observations $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_{n_1}\}$ drawn from the distribution P and observations $\mathbf{Z} = \{\mathbf{z}_1, ..., \mathbf{z}_{n_2}\}$ drawn from Q, the empirical estimate of MMD is:

$$\|\frac{1}{n_1}\sum_{i=1}^{n_1}\phi(\mathbf{x}_i) - \frac{1}{n_2}\sum_{i=1}^{n_2}\phi(\mathbf{z}_i)\|_{\mathcal{H}}, \quad (1)$$

where $\phi : \mathbb{R} \to \mathcal{H}$ with $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$. 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):

$$\mathbf{K}(x_i, x_j) = \sum_{m=1}^{M} d_m\mathbf{K}_m(x_i, x_j), \quad d_m \geq 0, \sum_{m=1}^{M} d_m = 1, \quad (2)$$

where $\mathbf{K}_m$ 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 $\mathbf{X}_{src}$ as a set of n_1 labeled training samples drawn from the source domain: $\mathbf{X}_{src} = \{(\mathbf{x}_1^s, y_1^s), ..., (\mathbf{x}_{n_1}^s, y_{n_1}^s)\}$, where $\mathbf{x}_i^s \in \mathbb{R}^d$ and $y_i^s \in \mathcal{Y}^s$ is the class label. Denote $\mathbf{X}_{tar}$ as a set of n_2 unlabeled testing samples drawn from the target domain: $\mathbf{X}_{tar} = \{\mathbf{x}_1^t, ..., \mathbf{x}_{n_2}^t\}$, where $\mathbf{x}_i^t \in \mathbb{R}^d$. Denote $\mathbf{X} = \mathbf{X}_{src} \cup \mathbf{X}_{tar} = \{x_1^s, ..., x_{n_1}^s, x_1^t, ..., x_{n_2}^t\} \in \mathbb{R}^{d\times N}$ with $N = n_1 + n_2$. Let $P_t(\mathbf{X}_{tar})$ and $P_s(\mathbf{X}_{src})$ be the marginal probability distributions of $\mathbf{X}_{tar}$ and $\mathbf{X}_{src}$ respectively, and $P_t(\mathbf{X}_{tar}) \neq P_s(\mathbf{X}_{src})$.

Suppose a kernel function $\mathbf{K}$ is constructed by multiple bases kernels as Equation (2). Denote $\phi : \mathbb{R}^d \to \mathcal{H}$ as the nonlinear map induced by $\mathbf{K}$. The task of TFR is to learn the optimal weights d_m and a shared linear transformation $\mathbf{W}$ in $\mathcal{H}$ simultaneously by: 1) explicitly reducing the distribution difference between $P_t(\mathbf{W}\phi(\mathbf{X}_{tar}))$ and $P_s(\mathbf{W}\phi(\mathbf{X}_{src}))$; 2) preserving the geometry of $\mathbf{X}_{tar}$; 3) preserving the label information of $\mathbf{X}_{src}$. 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 $\mathbf{W}$ or a Mahalanobis distance $\mathbf{A}$ across two domains. Following Equation (1), the distance between $P_s(\mathbf{W}\mathbf{X}_{src})$ and $P_t(\mathbf{W}\mathbf{X}_{tar})$ is

measured by the squared distance between the sample means in the two domains (Long et al. 2013; Wang et al. 2014):

$$\| \frac{1}{n_1}\sum_{i=1}^{n_1}\mathbf{W}\mathbf{x}_i^s - \frac{1}{n_2}\sum_{i=1}^{n_2}\mathbf{W}\mathbf{x}_i^t\|^2 = tr(\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{A}), \quad (3)$$

where $\mathbf{W}^T\mathbf{W} = \mathbf{A} \in \mathbb{R}^{d\times d}$ is PSD. $\mathbf{L} = [\mathbf{L}_{ij}]$ with $\mathbf{L}_{ij} = \frac{1}{n_1^2}$ if $\mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}_{src}$; $\mathbf{L}_{ij} = \frac{1}{n_2^2}$ if $\mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}_{tar}$; otherwise $\mathbf{L}_{ij} = -\frac{1}{n_1 n_2}$. 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 ϕ ($\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$), where the squared distance between the sample means in the two domains is:

$$\| \frac{1}{n_1}\sum_{i=1}^{n_1}\phi(\mathbf{x}_i^s) - \frac{1}{n_2}\sum_{i=1}^{n_2}\phi(\mathbf{x}_i^t)\|^2 = tr(\mathbf{K}\mathbf{L}), \quad (4)$$

where $\mathbf{K} \in \mathbb{R}^{N\times N}$. Although this nonlinear method benefits from finding the inner structure of data and the correlation between data points, it has to solve an expensive semi-definite programming problem (Boyd and Vandenberghe 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 $\mathbf{X}_{src}$ and $\mathbf{X}_{tar}$ are mapped to a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$ by a nonlinear map $\phi : \mathbb{R}^d \to \mathcal{H}$. We focus on learning a linear transformation $\mathbf{W}$ in this RKHS $\mathcal{H}$. In this term, the distance between $P_s(\mathbf{W}\phi(\mathbf{X}_{src}))$ and $P_t(\mathbf{W}\phi(\mathbf{X}_{tar}))$, denoted as $Dist(\mathbf{W}\phi(\mathbf{X}_{src}), \mathbf{W}\phi(\mathbf{X}_{tar}))$, is given as follows:

$$\| \frac{1}{n_1}\sum_{i=1}^{n_1}\mathbf{W}\phi(\mathbf{x}_i^s) - \frac{1}{n_2}\sum_{i=1}^{n_2}\mathbf{W}\phi(\mathbf{x}_i^t)\|^2 = tr(\mathbf{W}\Phi\mathbf{L}\Phi^T\mathbf{W}^T), \quad (5)$$

where $\Phi = \{\phi(\mathbf{x}_1^s), ..., \phi(\mathbf{x}_{n_1}^s), \phi(\mathbf{x}_1^t), ..., \phi(\mathbf{x}_{n_2}^t)\}$. Following (Schölkopf, Smola, and GengMüller 1998), $\mathbf{W}$ in $\mathcal{H}$ can be parameterized as a linear combination of data points, i.e., $\mathbf{W} = \mathbf{U}\Phi^T$. Substituting $\mathbf{W}$ to Equation (5), we obtain

$$\begin{aligned} Dist(\mathbf{W}\phi(\mathbf{X}_{src}), \mathbf{W}\phi(\mathbf{X}_{tar})) &= tr(\mathbf{U}\Phi^T\Phi\mathbf{L}\Phi^T\Phi\mathbf{U}^T) \\ &= tr(\mathbf{K}\mathbf{L}\mathbf{K}\hat{\mathbf{A}}), \end{aligned} \quad (6)$$

where $\hat{\mathbf{A}} = \mathbf{U}^T\mathbf{U} \in \mathbb{R}^{N\times N}$ 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:

$$tr(\mathbf{K}\mathbf{L}\mathbf{K}\hat{\mathbf{A}}) = tr[(\sum_{m=1}^{M} d_m\mathbf{K}_m)\mathbf{L}(\sum_{m=1}^{M} d_m\mathbf{K}_m)\hat{\mathbf{A}}], \quad (7)$$

where $\mathbf{K}_m = \begin{pmatrix}\mathbf{K}_m^s \\ \mathbf{K}_m^t\end{pmatrix}$, $\mathbf{K}_m^s \in \mathbb{R}^{n_1\times N}$ and $\mathbf{K}_m^t \in \mathbb{R}^{n_2\times N}$ are the kernel matrices induced by $\mathbf{K}_m$ 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 $\hat{\mathbf{A}}$. Once d_m and $\hat{\mathbf{A}}$ are obtained, the distance between a new test point $\mathbf{x}_p$ and any training point $\mathbf{x}_i$ can be computed as follows:

$$d(\mathbf{W}\phi(\mathbf{x}_p), \mathbf{W}\phi(\mathbf{x}_i)) = (\mathbf{k}_p - \mathbf{k}_i)^T\hat{\mathbf{A}}(\mathbf{k}_p - \mathbf{k}_i), \quad (8)$$

where $\mathbf{k}_i = \Phi^T\phi(\mathbf{x}_i) = \sum_m d_m[k_m(\mathbf{x}_1^s, \mathbf{x}_i), ..., k_m(\mathbf{x}_{n_2}^t, \mathbf{x}_i)]^T$ and $\mathbf{k}_p = \Phi^T\phi(\mathbf{x}_p) = \sum_m d_m[k_m(\mathbf{x}_1^s, \mathbf{x}_p), ..., k_m(\mathbf{x}_{n_2}^t, \mathbf{x}_p)]^T$. In this term, our proposed feature learning method generalizes to out-of-sample patterns.

Preserving Properties of $\mathbf{X}_{tar}$ and $\mathbf{X}_{src}$

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 $\mathbf{X}_{tar}$; 2) preserving the label information of $\mathbf{X}_{src}$.

For preserving the geometry of $\mathbf{X}_{tar}$, a diffusion kernel $\mathbf{K}_T$ (Kondor and Lafferty 2002) is defined on a weighted graph structure $\mathbf{G}^t$ with the adjacency matrix $\mathbf{M}^t = [\mathbf{M}_{ij}^t]$: $\mathbf{M}_{ij}^t = exp(\|\mathbf{x}_i^t - \mathbf{x}_j^t\|^2/2\sigma^2)$ if x_i^t is one of the k nearest neighbor of x_j^t; otherwise $\mathbf{M}_{ij}^t = 0$. Let $\mathbf{D}^t$ be an $n_2 \times n_2$ diagonal matrix with $\mathbf{D}_{ii}^t = \sum_j \mathbf{M}_{ij}^t$. The Laplacian of $\mathbf{G}^t$ can be defined as $\mathbf{L}^t = \mathbf{D}^t - \mathbf{M}^t$, and the Normalized Laplacian is $\tilde{\mathbf{L}}^t = (\mathbf{D}^t)^{-\frac{1}{2}}\mathbf{L}(\mathbf{D}^t)^{-\frac{1}{2}}$. The eigenvalues and eigenvectors of $\tilde{\mathbf{L}}^t$ are denoted as λ_i^t and ϕ_i^t, so that $\tilde{\mathbf{L}}^t = \sum_i^t \lambda_i^t(\phi_i^t)(\phi_i^t)^T$. In this term, $\mathbf{K}_T$ is defined as: $\mathbf{K}_T = \sum_{i=1}^n exp(-\sigma_d^2/2\lambda_i^t)(\phi_i^t)(\phi_i^t)^T$, where σ_d is the width of the diffusion kernel.

We construct a linear kernel $\mathbf{K}_l^t$ for $\mathbf{W}\phi(\mathbf{X}_{tar})$:

$$\mathbf{K}_l^t = \sum_{m=1}^{M} d_m\mathbf{K}_m^t\hat{\mathbf{A}}\sum_{m=1}^{M}(d_m\mathbf{K}_m^t)^T. \quad (9)$$

Based on the criterion defined in (Wang and Jin 2009), the distance between $\mathbf{K}_T$ and $\mathbf{K}_l^t$ can be expressed as:

$$\begin{aligned} d(\mathbf{K}_l^t\|\mathbf{K}_T) &= \frac{1}{2}(tr(\mathbf{K}_T^{-1}\mathbf{K}_l^t) - log|\mathbf{K}_l^t| + log|\mathbf{K}_T| - n_2) \\ &= \frac{1}{2}(tr((\sum_{m=1}^{M} d_m\mathbf{K}_m^t)^T\mathbf{K}_T^{-1}(\sum_{m=1}^{M} d_m\mathbf{K}_m^t)\hat{\mathbf{A}}) \\ &- log|(\sum_{m=1}^{M} d_m\mathbf{K}_m^t)\hat{\mathbf{A}}(\sum_{m=1}^{M} d_m\mathbf{K}_m^t)^T| + log|\mathbf{K}_T| - n_2). \end{aligned} \quad (10)$$

For preserving discriminating information of $\mathbf{X}_{src}$, a diffusion kernel $\mathbf{K}_S$ is defined on a weighted graph structure $\mathbf{G}^s$ with the adjacency matrix $\mathbf{M}^s = [\mathbf{M}_{ij}^s]$: $\mathbf{M}_{ij}^s = exp(\|\mathbf{x}_i^s - \mathbf{x}_j^s\|^2/2\sigma^2)$ if $y_i^s = y_j^s$; otherwise $\mathbf{M}_{ij}^s = 0$. In this term, $\mathbf{K}_S = \sum_{i=1}^m exp(-\sigma_d^2/2\lambda_i^s)(\phi_i^s)(\phi_i^s)^T$, where λ_i^s and ϕ_i^s are eigenvalues and eigenvectors of the Normalized Laplacian of $\mathbf{G}^s$. The distance between $\mathbf{K}_S$ and $\mathbf{K}_l^s$ (the lin-

ear kernel of $\mathbf{W}\phi(\mathbf{X}_{src})$) is obtained as:

$$
\begin{aligned}
d(\mathbf{K}_l^s \| \mathbf{K}_S) &= \frac{1}{2}(tr(\mathbf{K}_S^{-1}\mathbf{K}_l^s) - log|\mathbf{K}_l^s| + log|\mathbf{K}_S| - n_1) \\
&= \frac{1}{2}(tr((\sum_{m=1}^{M} d_m \mathbf{K}_m^s)^T \mathbf{K}_S^{-1} (\sum_{m=1}^{M} d_m \mathbf{K}_m^s)\hat{\mathbf{A}}) \\
&- log|(\sum_{m=1}^{M} d_m \mathbf{K}_m^s)\hat{\mathbf{A}}(\sum_{m=1}^{M} d_m \mathbf{K}_m^s)^T| + log|\mathbf{K}_S| - n_1).
\end{aligned} \tag{11}
$$

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 optimization algorithm, lack the convergence guarantees and may lead to numerical problems. Therefore, we rewrite the overall cost function as follows:

$$
\min_{\mathbf{d}} J(\mathbf{d}) \quad \text{with } d_m \geq 0, \sum_{m=1}^{M} d_m = 1, \tag{12}
$$

where $\mathbf{d} = (d_1, ..., d_M)$ and

$$
\begin{aligned}
J(\mathbf{d}) = \min_{\hat{\mathbf{A}} \succeq 0} & tr((\sum_{m=1}^{M} d_m \mathbf{K}_m)(\mathbf{K}' + \lambda \mathbf{L})(\sum_{m=1}^{M} d_m \mathbf{K}_m)\hat{\mathbf{A}}) \\
& - log|(\sum_{m=1}^{M} d_m \mathbf{K}_m^t)\hat{\mathbf{A}}(\sum_{m=1}^{M} d_m \mathbf{K}_m^t)^T| \\
& - log|(\sum_{m=1}^{M} d_m \mathbf{K}_m^s)\hat{\mathbf{A}}(\sum_{m=1}^{M} d_m \mathbf{K}_m^s)^T|, \\
\mathbf{K}' = & \begin{pmatrix} \mathbf{K}_S^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_T^{-1} \end{pmatrix}.
\end{aligned} \tag{13}
$$

In the following section, we will firstly prove that $J(\mathbf{d})$ has a closed-form solution. Secondly, the existence and the calculation of the gradient of $J(\cdot)$ is discussed. Finally, problem (12) is optimized by a reduced gradient method and converges at the local minimum.

Optimization

Proposition 1. *The optimal solution of $J(\mathbf{d})$ is:*

$$
\hat{\mathbf{A}}^* = 2((\sum_{m=1}^{M} d_m \mathbf{K}_m)(\mathbf{K}' + \lambda \mathbf{L})(\sum_{m=1}^{M} d_m \mathbf{K}_m))^{-1}. \tag{14}
$$

Proof. The derivative of Equation (13) w.r.t. $\hat{\mathbf{A}}$ is:

$$
((\sum_{m=1}^{M} d_m \mathbf{K}_m)(\mathbf{K}' + \lambda \mathbf{L})(\sum_{m=1}^{M} d_m \mathbf{K}_m)) - 2\hat{\mathbf{A}}^{-1}.
$$

Since $\mathbf{K}_m \succ 0$, $\mathbf{K}' \succ 0$ and $\mathbf{L} \succeq 0$, Proposition 1 now follows by setting the derivative to 0. □

Proposition 2. *$J(\cdot)$ is differentiable and we have:*

$$
\begin{aligned}
\frac{\partial J}{\partial d_m} =& 2\sum_{i=1}^{M} d_i tr(\mathbf{K}_m(\mathbf{K}' + \lambda \mathbf{L})\mathbf{K}_i \hat{\mathbf{A}}^*) \\
& - \sum_{i=1}^{n_1} \frac{1}{|\widetilde{\mathbf{K}}^s|}(|\mathbf{V}_i^s|) - \sum_{i=1}^{n_2} \frac{1}{|\widetilde{\mathbf{K}}^t|}(|\mathbf{V}_i^t|),
\end{aligned} \tag{15}
$$

where $\widetilde{\mathbf{K}}^s$ and $\widetilde{\mathbf{K}}^t$ are defined as:

$$
\widetilde{\mathbf{K}}^\dagger = (\sum_{i=1}^{M} d_i \mathbf{K}_i^\dagger)\hat{\mathbf{A}}^*(\sum_{i=1}^{M} d_i \mathbf{K}_i^\dagger)^T, \dagger \in \{s, t\},
$$

$\mathbf{V}_i^s$ and $\mathbf{V}_i^t$ are defined as:

$$
\mathbf{V}_i^\dagger = \begin{pmatrix} \widetilde{\mathbf{K}}^\dagger(1,1) & ... & \frac{\partial \widetilde{\mathbf{K}}^\dagger(1,i)}{\partial d_m} & ... & \widetilde{\mathbf{K}}^\dagger(1,n) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \widetilde{\mathbf{K}}^\dagger(n,1) & ... & \frac{\partial \widetilde{\mathbf{K}}^\dagger(n,i)}{\partial d_m} & ... & \widetilde{\mathbf{K}}^\dagger(n,n) \end{pmatrix},
$$

where n refers to n_1 when $\dagger = s$, otherwise $n = n_2$. Denote $\bar{\mathbf{K}}^\dagger$ as $2\sum_{k=1}^{M} d_k \mathbf{K}_m^\dagger \hat{\mathbf{A}}^*(\mathbf{K}_k^\dagger)^T$, then we have $\frac{\partial \widetilde{\mathbf{K}}^\dagger(i,j)}{\partial d_m} = \bar{\mathbf{K}}^\dagger(i,j)$.

Proof. Following Proposition 1, $\hat{\mathbf{A}}^*$ is unique for any admissible value of $\mathbf{d}$. This unicity ensures the differentiability of $J(\cdot)$ based on Theorem 4.1 in (Bonnaus and Shaoiro 1998). Therefore, Equation (15) is obtained by simple differentiation of problem (12) with respect to d_m. □

To optimize problem (12), we develop an efficient and effective procedure which performs reduced gradient descent on $J(\cdot)$ with the constraint $\{\mathbf{d} | \sum_m d_m = 1, d_m > 0\}$. This procedure does converge to the local minimum of differentiable function $J(\cdot)$ (Luenberger 1984). Specifically, once the gradient in Equation (15) is obtained, $\mathbf{d}$ is updated in a descent direction to ensure the equality and the positivity constraints. Denote u as the index of the largest component of $\mathbf{d}$ (i.e., $u = \arg\max_m d_m$). Following (Rakotomamonjy et al. 2008), the reduced gradient descent for updating $\mathbf{d}$ is:

$$
D_m = \begin{cases} 0 & if \ \ d_m = 0 \ \ \& \ \ \frac{\partial J}{\partial d_m} - \frac{\partial J}{\partial d_u} > 0 \\ \frac{\partial J}{\partial d_u} - \frac{\partial J}{\partial d_m} & if \ \ d_m > 0 \ \ \& \ \ m \neq u \\ \sum\limits_{\substack{v \neq u \\ d_v > 0}} \frac{\partial J}{\partial d_v} - \frac{\partial J}{\partial d_u} & if \ \ m = u. \end{cases} \tag{16}
$$

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: $\mathbf{d} \leftarrow \mathbf{d} + \gamma \mathbf{D}$, where γ is the step size.
5: **end while**

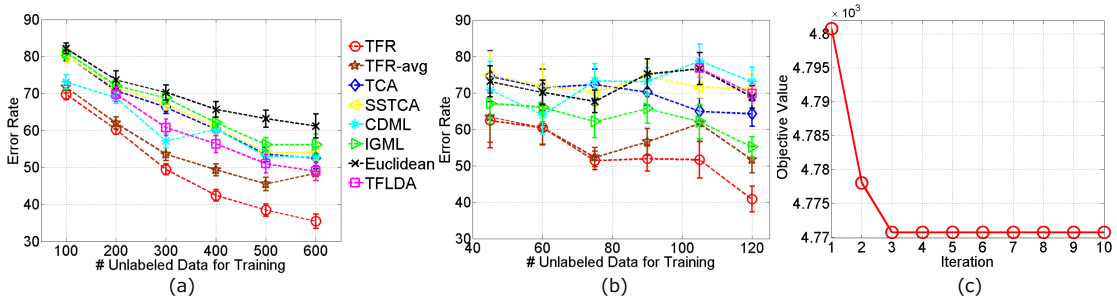

Figure 1: (a) Classification error rates on *Y vs F* data set. (b) Classification error rates on *F vs Y* data set. (c) Convergence evaluation of TFR.

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: σ_d, σ, λ and k. Specifically, we search σ_d based on the validation set in the range $\{0.1, 1, 10\}$, σ in the range $\{0.01, 0.1, 1, 10, 100\}$ and λ 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 vs F*: the source domain set is YALE, and the target domain set consists of 100 individuals randomly selected from FERET. 2) *F vs 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 point 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 vs F*, metric learning algorithm IGML and baseline Euclidean show their limits for cross-domain tasks. Secondly, on *F vs 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.

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.

Data Set	*orgs vs people*					*orgs vs place*					*people vs place*				
Size (%)	50	60	70	80	90	50	60	70	80	90	50	60	70	80	90
Eucliden	50.50	51.16	51.10	52.24	53.33	48.91	49.32	50.65	50.50	49.27	50.56	51.04	51.68	53.02	57.41
TCA	48.51	**43.58**	46.48	47.93	50.07	51.44	49.16	50.80	53.59	51.92	47.40	50.12	50.80	53.59	54.41
SSTCA	48.84	45.96	49.54	**42.15**	49.24	**42.99**	**42.93**	46.05	47.45	44.23	**43.63**	45.71	**45.05**	45.45	45.37
CDML	46.34	46.07	46.49	47.24	49.33	47.22	47.32	46.67	45.50	42.69	47.04	45.48	46.67	47.50	43.52
JDA	47.52	46.17	48.90	48.24	**46.28**	48.56	47.80	52.40	50.24	54.81	48.80	48.33	52.40	50.24	47.22
IGML	47.68	47.62	46.41	47.63	50.71	47.50	49.12	48.37	48.67	45.31	48.03	48.72	49.65	50.60	51.85
TFR	**43.71**	44.93	**45.07**	47.93	50.89	44.72	45.24	**44.60**	**44.37**	**42.31**	49.19	**45.03**	45.60	**44.37**	**43.52**
TFR-avg	47.85	45.13	46.38	48.17	52.37	46.83	49.40	45.73	47.85	50.00	50.44	45.71	48.61	52.92	44.64

Data Set	*comp vs rec*			*comp vs sci*			*comp vs talk*			*rec vs sci*			*rec vs talk*			*sci vs talk*		
Size (%)	30	40	50	30	40	50	30	40	50	30	40	50	30	40	50	30	40	50
Eucliden	52.40	53.71	56.20	61.41	59.62	62.97	55.24	58.47	54.23	57.06	55.33	56.29	45.10	43.33	46.73	52.75	53.08	55.79
TCA	50.64	49.66	**49.54**	50.35	49.72	50.70	**43.76**	44.90	45.23	58.86	**49.18**	49.87	43.74	44.02	43.87	50.56	50.50	50.61
SSTCA	51.92	51.11	51.54	50.64	50.75	50.41	44.83	45.21	43.35	49.26	51.41	50.83	43.21	44.79	43.35	48.41	**47.75**	48.95
CDML	51.27	51.90	51.54	49.14	**48.27**	51.59	46.52	49.06	43.17	50.52	50.87	4**9.54**	45.72	51.44	46.25	53.52	49.59	52.17
JDA	52.61	51.28	52.87	48.88	48.70	**47.85**	46.14	47.55	45.39	53.29	52.49	49.67	47.85	48.43	48.69	50.92	50.37	51.02
IGML	55.21	52.04	53.85	53.35	50.56	53.12	47.97	49.72	50.99	56.08	52.59	56.09	**43.12**	**42.02**	**43.18**	53.39	50.63	53.85
TFR	**49.36**	**48.98**	50.00	**48.37**	49.03	48.21	46.32	**44.41**	**42.45**	**48.79**	49.39	50.09	45.32	46.87	43.45	**48.32**	47.95	**48.51**
TFR-avg	51.37	50.09	50.26	48.92	50.32	49.38	47.09	48.36	43.10	50.09	49.67	51.23	48.74	51.38	48.29	50.09	49.87	50.29

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: *orgs vs people*, *orgs 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 leaning 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 *orgs 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

Bach, F. R.; Lanckriet, G. R. G.; and Jordan, M. 2004. Multiple kernel learning, conic duality, and the SMO algorithm. In *Proceedings of the 21st International Conference on Machine learning* (ICML), 6-13.

Belhumeur, P. N.; Hespanha, J. P.; and Kriegman, D. J. 1997. Eigenfaces versus fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19(7):711-720.

Borgwardt, K. M.; Gretton, A.; Rasch, M. J.; Kriegel, H.-P.; Schölkopf, B.; and Smola, A. J. 2006. Integrating structured biological data by Kernel Maximum Mean Discrepancy. *Bioinformatics* 22(14):49-57.

Bonnaus, J. F, and Shaoiro, A. 1998. Optimization problems with perturbation: A guided tour. *SIAM Review* 40(2):202-227.

Boyd, S., and Vandenberghe, L. 2004. *Convex Optimization*. Cambridge University Press, Cambridge.

Caruana, R. 1997. Multitask learning, *Machine Learning* 28(1):41-75.

Dai, W.; Xue, G.-R.; Yang, Q.; and Yu, Y. 2007. Coclustering based classification for out-of-domain documents. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD).

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics* (ACL), 256-263.

Duan, L.; Tsang, I.W.; and Xu, D. 2012. Domain transfer multiple kernel learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34(3):465-479.

Gönen, M., and Alpaydin, E. 2008. Localized multiple kernel learning. In *Proceedings of the 25th International Conference on Machine Learning* (ICML), 352-359.

Gong, B.; Shi, Y.; Sha, F.; and K, Grauman. 2012. Geodesic flow kernel for unsupervised domain adaptation. In *Proceedings of the 25th IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), 2066-2073.

Multiclass transfer learning from unconstrained priors. Jie, L.; Tommasi, T.; and Caputo, B. 2011. In *Proceedings of the 13th IEEE International Conference on Computer Vision* (ICCV), 1863-1870.

Jolliffe, I. 1986. *Principal Component Analysis*. Springer-Verlag.

Kim, S.-J.; Magnani, A.; and Boyd, S. 2006. Optimal kernel selection in kernel fisher discriminant analysis In *Proceedings of the 23rd International Conference on Machine Learning* (ICML), 465-472.

Kondor, R. S., and Lafferty, J. 2002. Diffusion kernels on graphs and other discrete input spaces. In *Proceedings of the 19th International Conference on Machine Learning* (ICML), 315-322.

Li, L.; Jin, X.; and Long, M. 2012. Topic correlation analysis for cross-domain text classification. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence* (AAAI).

Lin. Y. Y.; Liu, T. L.; and Fuh, C. S. 2011. Multiple kernel learning for dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33(6):1147-1160.

Long, M.; Wang, J.; Ding, G.; Sun, J.; and Yu, P. S. 2013. Transfer feature learning with joint distribution adaptation. In *Proceedings of the 14th IEEE International Conference on Computer Vision* (ICCV), 2200-2207.

Long, M.; Wang, J.; Ding, G.; Shen, D.; and Yang, Q. 2012. Transfer learning with graph co-regularization. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence* (AAAI).

Luenberger, D. 1984. *Linear and Nonlinear Programming*. Addison-Wesley.

Pan, S. J.; Kwok, J. T.; and Yang, Q. 2008. Transfer learning via dimensionality reduction. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence* (AAAI).

Pan, S. J., and Yang, Q. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* 22:1345-1359.

Pan, S. J.; Tsang, I. W.; Kwok, J. T.; and Yang, Q. 2011. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks* 22(2):199-210.

Phillips, J. P.; Moon, H.; Rizvi, S. A.; and Rauss, P. J. 2000. The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(10):1090-1104.

Rakotomamonjy, A.; Bach, F. R.; Canu, S.; and Grandvalet, Y. 2008. SimpleMKL. *Journal of Machine Learning Research* 9:2491-2521.

Saenko, K.; Kulis, B.; Fritz, M.; and Darrell, T. 2010. Adapting visual category models to new domains. In *Proceedings of the 11st European Conference on Computer Vision* (ECCV), 213-226.

Si, S.; Tao, D.; and Geng, B. 2010. Bregman divergence-based regularization for transfer subspace learning. *IEEE Transactions on Knowledge and Data Engineering* 22(7):929-942.

Schölkopf, B.; Smola, A.; and Müller, K.-R. 1998. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation* 10(5):1299-1319.

Wang, H., Wang, W., Zhang C., and Xu, F. J. 2014. Cross-domain metric learning based on information theory. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence* (AAAI), 2099-2105.

Wang, S., and Jin, R. 2009. An information geometry approach for distance metric learning. In *Proceedings of the 12nd International Conference on Artificial Intelligence and Statistics* (AISTATS), 591-598.