Domain Invariant Transfer Kernel Learning

Mingsheng Long, Jianmin Wang, Jiaguang Sun, and Philip S. Yu, *Fellow, IEEE*

Abstract—Domain transfer learning generalizes a learning model across training data and testing data with different distributions. A general principle to tackle this problem is reducing the distribution difference between training data and testing data such that the generalization error can be bounded. Current methods typically model the sample distributions in input feature space, which depends on nonlinear feature mapping to embody the distribution discrepancy. However, this nonlinear feature space may not be optimal for the kernel-based learning machines. To this end, we propose a *transfer kernel learning* (TKL) approach to learn a *domain-invariant* kernel by directly matching source and target distributions in the reproducing kernel Hilbert space (RKHS). Specifically, we design a family of spectral kernels by extrapolating target eigensystem on source samples with Mercer's theorem. The spectral kernel minimizing the approximation error to the ground truth kernel is selected to construct domain-invariant kernel machines. Comprehensive experimental evidence on a large number of text categorization, image classification, and video event recognition datasets verifies the effectiveness and efficiency of the proposed TKL approach over several state-of-the-art methods.

Index Terms—Transfer learning, kernel learning, Nyström method, text mining, image classification, video recognition

1 Introduction

The generalization error bound for standard supervised learning, where training data and testing data are sampled from identical probability distribution, is guaranteed by the statistical learning theory [1]. In the big data era, however, exponential growth of heterogenous data generated from different application domains such as texts, images, and videos, has created a compelling requirement for statistical learning models to generalize across different distributions. For instance, an object recognition model trained on manually annotated images is expected to generalize well on testing images under substantial variations in pose, lighting, or illumination. For another instance, a video event recognizer trained with limited labeled videos is inaccurate while expensive manual labeling is prohibitive in web applications. As such, domain transfer learning is the fundamental machine learning technology for cross-domain knowledge discovery [2].

A stream of studies have shown that when standard supervised classifiers are evaluated outside of their training datasets, the performance drops significantly [3], [4]. The mismatched training data and testing data arise in a wide range of applications, e.g., computer vision [5], [6], natural language processing [7], [8], and others. The *domain transfer learning* tasks involve two distinct types of datasets, one from a source domain, and the other from a target domain. The *source domain* contains a large amount of labeled data such that a classifier can be reliably constructed. The *target domain* contains a large amount of unlabeled data sampled from a substantially different but related distribution. The goal is to correct the distribution mismatch such that a standard classifier is adaptive across domains.

A fruitful line of prior works have been focusing on explicitly minimizing the distribution discrepancy via parametric or nonparametric divergences [9], [10], [11], [12], [13]. The main objective is to identify a feature representation or instance re-weighting through which the source and target domains manifest shared characteristics and the distribution mismatch can be formally corrected. For example, Pan et al. [12] proposed to learn the transfer components in the form of parameterized kernel maps, such that the kernel mean of training data is close to that of testing data. Generally, the distribution discrepancy can be formalized by the Kullback-Leibler (KL) divergence [13], Bregman divergence [11], and maximum mean discrepancy (MMD) [9], [10], [12]. However, the applicability of KL and Bregman divergences is hindered by the required density estimation procedure. The limitation of MMD is that it depends on nonlinear kernel map to compute distribution discrepancy, but the kernel function may not be optimal for kernel learning machines, e.g. SVM.

To alleviate the inconsistency of optimal kernels between distribution divergences and kernel machines, another line of recent works have been concentrating on directly learning a domain-invariant kernel matrix [4], [14], [15], [16]. Subject to the joint minimization of empirical loss and MMD, the kernel matrix is learned by semi-definite programming (SDP) [14] or multiple kernel learning (MKL) [15]. Unfortunately, standard SDP solver usually requires $O(n^{6.5})$ complexity and is computationally prohibitive for practical applications, while MKL encodes the distribution discrepancy into a small set of kernel ensemble parameters and may be inadequate for correcting the distribution mismatch. To address this, Zhang et al. [16] proposed a surrogate kernel matching (SKM) approach to directly match

- *M. Long is with the School of Software and the Department of Computer Science, Tsinghua University, Beijing 100084, China. E-mail: longmingsheng@gmail.com.*
- *J. Wang and J. Sun are with the School of Software and Tsinghua National Laboratory for Information Science and Technology, Tsinghua University, Beijing 100084, China. E-mail: {jimwang, sunjg}@tsinghua.edu.cn.*
- *P.S. Yu is with the Department of Computer Science, University of Illinois at Chicago, IL 60607. E-mail: psyu@uic.edu.*

Manuscript received 26 Dec. 2013; revised 15 Oct. 2014; accepted 21 Oct. 2014. Date of publication 20 Nov. 2014; date of current version 27 Apr. 2015.
Recommended for acceptance by I. Davidson.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2014.2373376

training data and testing data in a reproducing kernel Hilbert space (RKHS). This is theoretically attractive, since the distribution discrepancy is directly measured by the error between source kernel and target kernel. A major limitation of SKM is that it linearly transforms the complete source kernel to the eigenspace of target kernel, which leads to inevitably large approximation error of source data. That said, it is somewhat reduced to representing all data using only target eigenspace.

Motivated by the above discussions, in this paper, we propose a *transfer kernel learning* (TKL) approach, to learn a *domain-invariant* kernel by directly matching the source and target distributions in the reproducing kernel Hilbert space. Our key observation is that the distribution divergence can be formalized by the Nyström approximation error [17] between the extrapolated source kernel and the ground truth source kernel. Based on this observation, we design a family of spectral kernels by extrapolating target eigensystem on source samples using the Mercer's theorem [18]. The spectral kernel that minimizes the approximation error to the ground truth kernel is chosen to construct domain-invariant kernel machines, e.g. SVM. In this way, the learned domain-invariant kernel can respect both the target eigensystem and source approximation quality. Our contributions are summarized as follows.

- To correct the distribution mismatch without violating the optimal kernel machines, we propose a novel approach based on Nyström approximation error minimization and spectral kernel design. To our best knowledge, this is the first attempt to domain transfer learning with such a methodology.
- Off-the-shelf kernel machines, e.g. support vector machine (SVM) and kernel ridge regression (KRR), can be applied with the domain-invariant kernel matrix to tackle domain transfer learning.
- Comprehensive experiments on a large number of benchmark text, image, and video datasets with different adaptation difficulty verify that TKL can consistently outperform existing methods.

The subsequent paper is organized as follows. We review related works in Section 2 and preliminaries in Section 3. We present the TKL approach, the learning algorithms, and the analysis on computational complexity and approximation error bound in Section 4. Empirical results are reported in Section 5, while conclusions and future works are discussed in Section 6.

2 Related Work

Transfer learning [2] has been extensively studied in many research areas, including statistics and machine learning [10], [12], [16], [19], natural language processing [7], [8], and recently computer vision [4], [5], [13].

Of the particular relevance to our work is the idea of learning domain-invariant feature representations, which enable transferring classifiers from the source domain to the target domain [11], [12], [13]. As such, the domain-invariant characteristics are encouraged by explicitly minimizing the distribution discrepancy using parametric or nonparametric divergences. Generally, the distribution discrepancy can be formalized by Kullback-Leibler divergence [13], Bregman divergence [11], and maximum mean discrepancy [9], [10], [12]. However, the applicability of KL and Bregman divergences is hindered by the required density estimation procedure. The limitation of MMD is that it depends on nonlinear kernel map to compute distribution discrepancy, but the kernel function may not be optimal for kernel learning machines, e.g. SVM. Our work shares with this line of works the key idea of correcting the distribution mismatch by explicitly minimizing proper distribution divergences. Note the substantial distinction is that our work is reduced to minimizing the Nyström approximation error across different kernel matrices, which is a novel distribution divergence first explored for domain transfer learning. In particular, our work does not require the nontrivial density estimation as an intermediate procedure. Also, it does not violate the optimal kernel for kernel learning machines, since it does not require nonlinear kernel maps to measure the distribution discrepancy.

Another related literature is cross-modality learning, which maximizes the feature correlations across different modalities within a multiple-type domain. This literature includes canonical correlation analysis (CCA), partial least squares (PLS), Kernel CCA (KCCA) [20], and Deep CCA (DCCA) [21]. Cross-modality methods cannot be directly applied to cross-domain learning problems for transferring shared knowledge across different domains under distribution shift. Several recent works extended CCA to cross-domain problems [22], [23], [24]. However, these works usually require prior knowledge to construct domain-invariant and domain-specific features (e.g., shared and distinct words in the natural language processing domains) and maximize the feature correlations, which is nontrivial for general purpose learners [22].

Our work is closely related to a nonlinear learning literature called kernel learning, which directly learns a nonparametric kernel matrix subject to some proper similarity constraints [25], [26], [27]. There are several works which directly learn a domain-invariant kernel matrix for domain transfer learning [4], [14], [15], [16]. Subject to the joint minimization of empirical loss and MMD, the kernel matrix is learned by Semi-Definite Programming (SDP) [14] or multiple kernel learning [15]. However, standard SDP solver usually requires $O(n^{6.5})$ and is computationally prohibitive, while MKL encodes the distribution discrepancy into a small set of kernel ensemble parameters and may be inadequate for correcting the distribution mismatch. Different from these works, our work directly learns a domain-invariant kernel by minimizing the Nyström approximation error, which can correct the distribution mismatch better and scale well to large datasets.

In particular, the surrogate kernel matching proposed by Zhang et al. [16] is the closest to ours. SKM directly matches training data and testing data in the reproducing kernel Hilbert space. A major limitation of SKM is that it linearly transforms the complete source kernel to the eigenspace of target kernel, which leads to inevitably large approximation error of source data. Our work significantly differs from SKM in minimizing the approximation error of source data using spectral kernels extrapolated from target eigenspace. This guarantees better cross-domain generalization performance.

3 Preliminaries

3.1 Maximum Mean Discrepancy

The difference in the probability distributions between domains is the fundamental challenge to the generalization performance of supervised learning methods. Therefore, it is crucial to formalize the distribution difference and design effective methods to minimize it. Parametric divergences have been adopted for transfer learning, e.g. Kullback-Leibler divergence [13] and Bregman divergence [11]. However, they usually involve a more difficult density estimation procedure, which aggravates the model complicatedness. To this end, Smola et al. proposed a nonparametric divergence, maximum mean discrepancy [9], to compare two distributions p and q based on the distance between the expectations of the two generated datasets $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ and $\mathcal{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_m\}$ in a reproducing kernel Hilbert space $\mathcal{H}$. MMD and the empirical estimate are defined as follows [9]:

$$\begin{aligned} \mathrm{MMD}[p,q] &\triangleq \sup_{\|f\|_{\mathcal{H}}\le 1} \left(\mathbb{E}_{\mathbf{x}\sim p}[f(\mathbf{x})] - \mathbb{E}_{\mathbf{z}\sim q}[f(\mathbf{z})]\right) \\ \mathrm{MMD}[\mathcal{X},\mathcal{Z}] &\triangleq \left\| \frac{1}{n}\sum_{i=1}^{n}\phi(\mathbf{x}_i) - \frac{1}{m}\sum_{j=1}^{m}\phi(\mathbf{z}_j) \right\|_{\mathcal{H}}, \end{aligned} \tag{1}$$

where $f(\cdot) \in \mathcal{H}$ is any arbitrary function in $\mathcal{H}$ and $\phi(\cdot)$ is the nonlinear feature mapping that induces $\mathcal{H}$.

Theorem 1 (MMD [9]). *Let p, q be Borel probability measures, and let $\mathcal{H}$ be a universal reproducing kernel Hilbert space. Then $MMD[p,q] = 0$ if and only if $p = q$.*

By nonlinear mapping $\phi(\cdot)$, MMD can intrinsically capture both first- and high-order statistics. Therefore, reducing MMD suffices to matching both the means and moments of data distributions between domains. MMD has been widely explored for transfer learning through instance reweighting [10], feature extraction [12], and classifier adaptation [15], etc. Unfortunately, previous methods may not sufficiently reduce MMD by imposing MMD as a penalty to standard learning methods, as the other criteria, e.g., squared loss, will drive the feasible region to be out of the MMD minima. In this paper, we present a domain-invariant transfer kernel learning approach to address the above issues.

3.2 Nyström Kernel Approximation

We tackle the limitations of MMD-based distribution matching by learning a domain-invariant kernel. Our approach is related to Nyström kernel approximation, which approximates a new kernel by the extrapolated eigensystem of a target kernel via the Mercer theorem.

Theorem 2 (Mercer [18]). *Let $k(\mathbf{z},\mathbf{x})$ be a continuous symmetric non-negative function which is positive semi-definite and square integrable w.r.t. distribution $p(\mathbf{x})$, then*

$$k(\mathbf{z},\mathbf{x}) = \sum_{i=1}^{\infty} \lambda_i \phi_i(\mathbf{z})\, \phi_i(\mathbf{x}). \tag{2}$$

The eigenvalues λ_i's and the orthonormal eigenfunctions ϕ_i's are the solutions of the following integral equation

$$\int k(\mathbf{z},\mathbf{x})\, \phi_i(\mathbf{x})\, p(\mathbf{x})\, \mathrm{d}\mathbf{x} = \lambda_i \phi_i(\mathbf{z}). \tag{3}$$

This is the fundamental theorem for the theory of reproducing kernel Hilbert space. It states that any positive semi-definite (PSD) kernel can be reconstructed by the eigensystem of a target kernel, and provides a mechanism to generate kernel matrices on any new datasets. In particular, given dataset $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ sampled from distribution $p(\mathbf{x})$ and the corresponding kernel matrix $\mathbf{K}_{\mathcal{X}}$, we can compute via Equation (3) the eigenvalues λ_i's and eigenfunctions $\phi_i(\mathbf{x})$'s. Then we can evaluate the kernel k on any new dataset $\mathcal{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_m\}$. The generated kernel matrix $\mathbf{K}_{\mathcal{Z}}$ relies only on the eigensystem of $\mathbf{K}_{\mathcal{X}}$, $\{\lambda_i, \phi_i(\mathbf{x})\}$.

The idea of Nyström method [28] is to approximate the integral in Equation (3) by its empirical estimate

$$\sum_{j=1}^{n} \frac{k(\mathbf{z},\mathbf{x}_j)\phi_i(\mathbf{x}_j)}{n} \simeq \lambda_i \phi_i(\mathbf{z}). \tag{4}$$

Traversing $\mathbf{z} \in \mathcal{X}$ results in standard eigendecomposition (up to a scaling constant) $\mathbf{K}_{\mathcal{X}}\mathbf{\Phi}_{\mathcal{X}} = \mathbf{\Phi}_{\mathcal{X}}\mathbf{\Lambda}_{\mathcal{X}}$, where $\mathbf{\Phi}_{\mathcal{X}} \in \mathbb{R}^{n\times n}$ are n eigenvectors, and $\mathbf{\Lambda}_{\mathcal{X}} = \mathrm{diag}(\lambda_1, \ldots, \lambda_n) \in \mathbb{R}^{n\times n}$ are n eigenvalues. The eigenfunctions $\phi_i(\mathbf{x})$'s and eigenvalues λ_i's in Equation (3) can be discretely approximated by the columns of $\mathbf{\Phi}_{\mathcal{X}}$ and the diagonal entries of $\mathbf{\Lambda}_{\mathcal{X}}$, respectively. The eigenfunction $\phi_i(\mathbf{z})$ at any new point $\mathbf{z}$ can be *extrapolated* by $\phi_i(\mathbf{z}) = \sum_{j=1}^{n} \frac{k(\mathbf{z},\mathbf{x}_j)\phi_i(\mathbf{x}_j)}{n\lambda_i}$. Evaluating the eigenfunctions $\phi_i(\mathbf{z})$ on new dataset $\mathcal{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_m\}$ with distribution $p(\mathbf{z})$ leads to discrete approximation

$$\mathbf{\Phi}_{\mathcal{Z}} \simeq \mathbf{K}_{\mathcal{Z}\mathcal{X}}\mathbf{\Phi}_{\mathcal{X}}\mathbf{\Lambda}_{\mathcal{X}}^{-1}, \tag{5}$$

where $\mathbf{K}_{\mathcal{Z}\mathcal{X}} \in \mathbb{R}^{m\times n}$ is the cross-dataset similarity matrix between $\mathcal{Z}$ and $\mathcal{X}$, evaluated using kernel k. By the Mercer theorem, $\mathbf{K}_{\mathcal{Z}}$ can be approximated as

$$\mathbf{K}_{\mathcal{Z}} \simeq \mathbf{\Phi}_{\mathcal{Z}}\mathbf{\Lambda}_{\mathcal{X}}\mathbf{\Phi}_{\mathcal{Z}}^{\mathsf{T}} = \mathbf{K}_{\mathcal{Z}\mathcal{X}}\mathbf{K}_{\mathcal{X}}^{-1}\mathbf{K}_{\mathcal{X}\mathcal{Z}}. \tag{6}$$

If we denote $\mathcal{Z}$ and $\mathcal{X}$ the source data and target data, respectively, then $\mathbf{z}$ and $\mathbf{x}$ will follow different probability distributions. Thus the approximation of $\mathbf{K}_{\mathcal{Z}}$ using the eigensystem of $\mathbf{K}_{\mathcal{X}}$ will not be accurate. Zhang et al. [16] does not fully consider this problem when applying the Nyström method across domains. We will present our approach to tackle this problem.

3.3 Spectral Kernel Design

Kernel learning has been extensively explored in the literature, e.g., nonparametric kernel transforms [29], nonparametric kernel learning [27], etc. In this paper, we adopt spectral kernel design [25], which constructs new kernels from the eigenvectors of a target kernel.

Theorem 3 (Spectral Kernel [25]). *If a positive semi-definite kernel matrix $\mathbf{K} \in \mathbb{R}^{n\times n}$ has eigensystem $\{\gamma_i, \boldsymbol{\phi}_i\}_{i=1}^{n}, \gamma_1 \ge \cdots \ge \gamma_n \ge 0$, then the family of matrices*

$$\mathbf{K}_{\lambda} = \sum_{i=1}^{n} \lambda_i \boldsymbol{\phi}_i \boldsymbol{\phi}_i^{\mathsf{T}}, \lambda_1 \ge \cdots \ge \lambda_n \ge 0 \tag{7}$$

will produce PSD kernels with $\mathbf{K}_{\lambda}$ as kernel matrices.

The reconstructed new kernel matrix $\mathbf{K}_{\lambda}$ is a linear ensemble of multiple basic kernels $\{\mathbf{K}_i = \boldsymbol{\phi}_i\boldsymbol{\phi}_i^{\mathsf{T}}\}_{i=1}^{n}$. However, this is

TABLE 1
Notations and Their Descriptions Used in This Paper

Notation	Description	Notation	Description
$\mathcal{Z}, \mathcal{X}$	source/target domain	$\mathbf{X}$	input data matrix
m, n	#source/target examples	$\mathbf{K}$	kernel matrix
d, c	# features/classes	$\mathbf{\Phi}$	eigenvector matrix
r, ζ	#eigenvectors/damping factor	$\mathbf{\Lambda}$	eigenvalue matrix

quite different from multiple kernel learning [26] which ensembles multiple pre-computed kernels. Algorithmically, we can find the nonparametric eigenspectrum $\{\lambda_i\}$ by optimizing proper criteria, e.g., kernel alignment [30] and graph alignment [31]. In this paper, we will present how to learn a domain-invariant kernel by optimizing the Nyström approximation error w.r.t. the source kernel and target kernel.

4 Transfer Kernel Learning

In this section, we first formulate the problem settings. Then we present the TKL approach and the learning algorithms. Finally, we formally analyze the computational complexity and approximation error bound.

4.1 Problem Formulation

We begin with the terminology definitions. For clarity, frequently used notations are summarized in Table 1.

Definition 1 (Domain [2]). *A domain $\mathcal{D}$ is composed of an d-dimensional feature space $\mathcal{F}$ and a marginal probability distribution $P(\mathbf{x})$, i.e., $\mathcal{D} = \{\mathcal{F}, P(\mathbf{x})\}$, $\mathbf{x} \in \mathcal{F}$.*

Generally speaking, two domains $\mathcal{Z}$ and $\mathcal{X}$ will be different if they have different feature spaces or marginal distributions, i.e., $\mathcal{F}_\mathcal{Z} \neq \mathcal{F}_\mathcal{X} \vee P(\mathbf{z}) \neq P(\mathbf{x})$.

Definition 2 (Task [2]). *Given domain $\mathcal{D}$, a task $\mathcal{T}$ is composed of a c-cardinality label set $\mathcal{Y}$ and a classifier $f(\mathbf{x})$, i.e., $\mathcal{T} = \{\mathcal{Y}, f(\mathbf{x})\}$, where $y \in \mathcal{Y}$, and $f(\mathbf{x}) = P(y|\mathbf{x})$ can be interpreted as the conditional probability distribution.*

Generally speaking, two tasks $\mathcal{T}_\mathcal{Z}$ and $\mathcal{T}_\mathcal{X}$ will be different if they have different label spaces or conditional distributions, i.e., $\mathcal{Y}_\mathcal{Z} \neq \mathcal{Y}_\mathcal{X} \vee P(y|\mathbf{z}) \neq P(y|\mathbf{x})$.

Problem 1 (Transfer Kernel Learning). *Given a labeled source domain $\mathcal{Z} = \{(\mathbf{z}_1, y_1), \ldots, (\mathbf{z}_m, y_m)\}$ and an unlabeled target domain $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ with $\mathcal{F}_\mathcal{Z} = \mathcal{F}_\mathcal{X}$, $\mathcal{Y}_\mathcal{Z} = \mathcal{Y}_\mathcal{X}$, $P(\mathbf{z}) \neq P(\mathbf{x})$, and $P(y|\mathbf{z}) \neq P(y|\mathbf{x})$, learn a domain-invariant kernel $k(\mathbf{z}, \mathbf{x}) = \langle \phi(\mathbf{z}), \phi(\mathbf{x}) \rangle$ such that $P(\phi(\mathbf{z})) \simeq P(\phi(\mathbf{x}))$. Assume $P(y|\phi(\mathbf{z})) \simeq P(y|\phi(\mathbf{x}))$, so kernel machines trained on $\mathcal{Z}$ can generalize well on $\mathcal{X}$.*

Suppose we are given an input kernel function k, e.g., Gaussian kernel $k(\mathbf{z}, \mathbf{x}) = e^{-\gamma\|\mathbf{z}-\mathbf{x}\|^2}$ or Laplacian kernel $k(\mathbf{z}, \mathbf{x}) = e^{-\gamma|\mathbf{z}-\mathbf{x}|}$, we can compute the source kernel $\mathbf{K}_\mathcal{Z}$ and target kernel $\mathbf{K}_\mathcal{X}$. These two kernels are utilized to learn a domain-invariant kernel $\overline{\mathbf{K}}_{\mathcal{Z}\cup\mathcal{X}}$.

Our work shares similar working assumptions with many existing works [12], [16] in that the conditional distribution is similar across domains in the feature-matching subspace. When the conditional distribution further changes, it is often accompanied by the change in the marginal distribution. In this more challenging case, matching the marginal distributions sufficiently is indispensible for effective domain transfer learning.

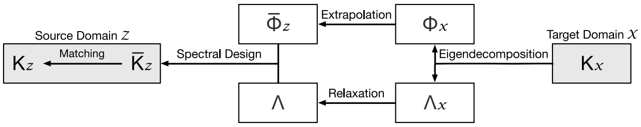


Fig. 1. Complete procedure of transfer kernel learning.

4.2 Learning Approach

Handling the distribution difference in the Hilbert space is nontrivial, since the kernel-induced feature map ϕ often cannot be explicitly represented. In order to require two datasets (e.g., source data $\mathcal{Z}$ and target data $\mathcal{X}$) to follow similar feature-space distributions, i.e., $P(\phi(\mathbf{z})) \simeq P(\phi(\mathbf{x}))$, it just suffices to require them to have similar kernel matrices, i.e., $\mathbf{K}_\mathcal{Z} \simeq \mathbf{K}_\mathcal{X}$ [16]. However, the kernel matrices are data-dependent, and it is even impossible to directly evaluate the closeness between different kernels due to different dimensions, i.e., $\mathbf{K}_\mathcal{Z} \in \mathbb{R}^{m\times m}$, $\mathbf{K}_\mathcal{X} \in \mathbb{R}^{n\times n}$. To solve this problem, we adopt the idea of Nyström kernel approximation [28] to generate an *extrapolated* source kernel $\overline{\mathbf{K}}_\mathcal{Z} \in \mathbb{R}^{m\times m}$ using the eigensystem of the target kernel $\mathbf{K}_\mathcal{X}$. Then $\overline{\mathbf{K}}_\mathcal{Z}$ is comparable to the ground truth source kernel $\mathbf{K}_\mathcal{Z}$ and is matched with spectral kernel design. A complete learning procedure is shown in Fig. 1.

4.2.1 Eigensystem Extrapolation

We adopt Nyström kernel approximation to perform eigensystem extrapolation. To this end, we first apply standard eigendecomposition on the target kernel $\mathbf{K}_\mathcal{X}$ as

$$\mathbf{K}_\mathcal{X}\mathbf{\Phi}_\mathcal{X} = \mathbf{\Phi}_\mathcal{X}\mathbf{\Lambda}_\mathcal{X}, \tag{8}$$

which gives the eigensystem $\{\mathbf{\Lambda}_\mathcal{X}, \mathbf{\Phi}_\mathcal{X}\}$ of the target kernel $\mathbf{K}_\mathcal{X}$. Then we evaluate the eigensystem on the source data $\mathcal{Z}$ using the Mercer theorem, and derive the extrapolated eigensystem of the source kernel $\mathbf{K}_\mathcal{Z}$ as

$$\overline{\mathbf{\Phi}}_\mathcal{Z} \simeq \mathbf{K}_{\mathcal{Z}\mathcal{X}}\mathbf{\Phi}_\mathcal{X}\mathbf{\Lambda}_\mathcal{X}^{-1}, \tag{9}$$

where $\mathbf{K}_{\mathcal{Z}\mathcal{X}} \in \mathbb{R}^{m\times n}$ is the cross-domain kernel matrix between $\mathcal{Z}$ and $\mathcal{X}$, evaluated using kernel function k.

In the original Nyström method, the extrapolated source eigenvectors $\overline{\mathbf{\Phi}}_\mathcal{Z}$ and the target eigenvalues $\mathbf{\Lambda}_\mathcal{X}$ are directly used to approximate the source kernel $\mathbf{K}_\mathcal{Z}$ as Equation (6). We have to note that the Nyström approximation (6) is accurate if and only if the Mercer condition (3) holds, i.e., $P(\mathbf{z}) \simeq P(\mathbf{x})$. In other words, when $P(\mathbf{z}) \neq P(\mathbf{x})$, the Nyström approximation error will be inevitably large, making direct cross-domain kernel approximation inappropriate. Therefore, the original Nyström method [16] cannot correct the distribution mismatch sufficiently for transfer learning.

Nonetheless, the discussion gives us an important observation: Nyström approximation error essentially embodies the distribution divergence across domains, i.e., error approaches zero if and only if $P(\mathbf{z}) \simeq P(\mathbf{x})$. If we can find an extrapolated kernel such that the Nyström approximation error is minimized, then the extrapolated kernel will be invariant to different domains and thus facilitate more effective cross-domain generalization.

4.2.2 Eigenspectrum Relaxation

To reduce the Nyström approximation error, we adopt the idea of spectral kernel design [25] to construct new kernel matrix from extrapolated eigensystem (9). In this way, the generated kernel matrix could preserve the key structure of the target kernel $\mathbf{K}_{\mathcal{X}}$ through the extrapolated eigensystem $\overline{\mathbf{\Phi}}_{\mathcal{Z}}$, yet remain flexible to be reshaped to minimize the distribution divergence.

Specifically, we relax the eigenspectrum $\mathbf{\Lambda}_{\mathcal{X}}$ in the original Nyström method (6) to be learnable parameters $\mathbf{\Lambda}$, which leads to a family of kernels extrapolated from target eigensystem but evaluated on source data

$$\overline{\mathbf{K}}_{\mathcal{Z}} = \overline{\mathbf{\Phi}}_{\mathcal{Z}} \mathbf{\Lambda} \overline{\mathbf{\Phi}}_{\mathcal{Z}}^{\mathsf{T}}. \tag{10}$$

This family of kernels preserve the key structures of the target domain, i.e., eigenvectors $\overline{\mathbf{\Phi}}_{\mathcal{Z}}$, which behave as a proper bridge for knowledge transfer. However, these eigenvectors do not necessarily reduce the distribution divergence unless we explicitly minimize it [12]. Furthermore, the free eigenspectrum $\mathbf{\Lambda}$ remains undetermined. Unlike traditional spectral kernel design [25] which learns parameter $\mathbf{\Lambda}$ by aligning the spectral kernel to a prior kernel computed in the same domain, we perform kernel matching across domains.

4.2.3 Approximation Error Minimization

In this paper, to minimize the distribution divergence explicitly, we propose to minimize the approximation error between the extrapolated source kernel $\overline{\mathbf{K}}_{\mathcal{Z}}$ and the ground truth source kernel $\mathbf{K}_{\mathcal{Z}}$ using squared loss

$$\begin{gathered} \min_{\mathbf{\Lambda}} \left\| \overline{\mathbf{K}}_{\mathcal{Z}} - \mathbf{K}_{\mathcal{Z}} \right\|_F^2 = \left\| \overline{\mathbf{\Phi}}_{\mathcal{Z}} \mathbf{\Lambda} \overline{\mathbf{\Phi}}_{\mathcal{Z}}^{\mathsf{T}} - \mathbf{K}_{\mathcal{Z}} \right\|_F^2 \\ \lambda_i \geq \zeta \lambda_{i+1}, i = 1, \ldots, n-1 \\ \lambda_i \geq 0, i = 1, \ldots, n, \end{gathered} \tag{11}$$

where $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ are the n nonnegative eigenspectrum parameters. As the eigenspectrum of a PSD kernel decays following the power-law distribution [30], [32], we impose a set of eigenvalue damping constraints, where $\zeta \geq 1$ is the eigenspectrum *damping* factor. In principle, the damping constraints are crucial to the performance of TKL, since it can encode the prior decaying trend of $\mathbf{K}_{\mathcal{Z}}$, and more importantly, encourage the larger eigenvectors to contribute more to the knowledge transfer procedure. In this way, the learned kernel is established to be domain-invariant.

The TKL approach can be generalized to match the marginal distributions of multiple source domains with the target domain. This can be implemented by first learning a source-specific eigenspectrum $\mathbf{\Lambda}$ for each source domain respectively, and then utilizing existing multiple-source learning algorithms [33], [34] to perform consensus prediction for the target domain based on the predictions of multiple source domains.

4.2.4 Domain-Invariant Kernel Machine

With learned optimal eigenspectrum parameters $\mathbf{\Lambda}$, it is straight-forward to construct the domain-invariant kernel $\overline{\mathbf{K}}_{\mathcal{A}}$ on the source and target data $\mathcal{A} = \mathcal{Z} \cup \mathcal{X}$. Based on the spectral kernel design, $\overline{\mathbf{K}}_{\mathcal{A}}$ can be generated from the domain-invariant eigensystem $\{\mathbf{\Lambda}, \overline{\mathbf{\Phi}}_{\mathcal{A}}\}$

$$\overline{\mathbf{K}}_{\mathcal{A}} = \begin{bmatrix} \overline{\mathbf{\Phi}}_{\mathcal{Z}} \mathbf{\Lambda} \overline{\mathbf{\Phi}}_{\mathcal{Z}}^{\mathsf{T}} & \overline{\mathbf{\Phi}}_{\mathcal{Z}} \mathbf{\Lambda} \mathbf{\Phi}_{\mathcal{X}}^{\mathsf{T}} \\ \mathbf{\Phi}_{\mathcal{X}} \mathbf{\Lambda} \overline{\mathbf{\Phi}}_{\mathcal{Z}}^{\mathsf{T}} & \mathbf{\Phi}_{\mathcal{X}} \mathbf{\Lambda} \mathbf{\Phi}_{\mathcal{X}}^{\mathsf{T}} \end{bmatrix} = \overline{\mathbf{\Phi}}_{\mathcal{A}} \mathbf{\Lambda} \overline{\mathbf{\Phi}}_{\mathcal{A}}^{\mathsf{T}}, \tag{12}$$

where $\overline{\mathbf{\Phi}}_{\mathcal{A}} \triangleq [\overline{\mathbf{\Phi}}_{\mathcal{Z}}; \mathbf{\Phi}_{\mathcal{X}}]$ are extrapolated eigenvectors on all data $\mathcal{A}$. The domain-invariant kernel $\overline{\mathbf{K}}_{\mathcal{A}}$ can be directly fed to standard kernel machines, e.g., SVM, to facilitate cross-domain prediction and generalization.

Specifically, kernel SVM classifier is trained on the source partition $\overline{\mathbf{K}}_{\mathcal{Z}} = \overline{\mathbf{\Phi}}_{\mathcal{Z}} \mathbf{\Lambda} \overline{\mathbf{\Phi}}_{\mathcal{Z}}^{\mathsf{T}}$ and then applied to the cross-domain partition $\overline{\mathbf{K}}_{\mathcal{X}\mathcal{Z}} = \mathbf{\Phi}_{\mathcal{X}} \mathbf{\Lambda} \overline{\mathbf{\Phi}}_{\mathcal{Z}}^{\mathsf{T}}$ as below

$$\mathbf{y}_{\mathcal{X}} = \overline{\mathbf{K}}_{\mathcal{X}\mathcal{Z}} (\boldsymbol{\alpha} \odot \mathbf{y}_{\mathcal{Z}}) + b, \tag{13}$$

where $\boldsymbol{\alpha}$ is the Lagrange multipliers and b is the bias. Applying SVM with kernel matrix as input can be easily implemented via off-the-shelf LIBSVM package.

For out-of-sample prediction with testing data $\mathcal{X}_o$ sampled from the same distribution as the target data $\mathcal{X}$, we need to evaluate target eigenvectors $\mathbf{\Phi}_{\mathcal{X}}$ on $\mathcal{X}_o$

$$\mathbf{\Phi}_{\mathcal{X}_o} = \mathbf{K}_{\mathcal{X}_o \mathcal{X}} \mathbf{\Phi}_{\mathcal{X}} \mathbf{\Lambda}_{\mathcal{X}}^{-1}, \tag{14}$$

where $\mathbf{K}_{\mathcal{X}_o \mathcal{X}}$ is the cross-dataset kernel matrix in the target domain evaluated using function k. Then we compute the cross-domain partition as $\overline{\mathbf{K}}_{\mathcal{X}_o \mathcal{Z}} = \mathbf{\Phi}_{\mathcal{X}_o} \mathbf{\Lambda} \overline{\mathbf{\Phi}}_{\mathcal{Z}}^{\mathsf{T}}$. Feeding this matrix to Equation (13) gives out-of-sample predictions on $\mathcal{X}_o$.

4.3 Learning Algorithm

We design learning algorithm with convex optimization and scale to large datasets with Nyström method.

4.3.1 Quadratic Progamming (QP)

We show how to solve the TKL optimization problem (11) using the standard quadratic programming with linear constraints. Denote $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n)$ the n eigenspectrum parameters, i.e., $\mathbf{\Lambda} = \mathrm{diag}(\boldsymbol{\lambda})$. By linear algebra, we reformulate Equation (11) in matrix form

$$\begin{aligned} \min_{\boldsymbol{\lambda}} \ & \boldsymbol{\lambda}^{\mathsf{T}} \mathbf{Q} \boldsymbol{\lambda} - 2\mathbf{r}^{\mathsf{T}} \boldsymbol{\lambda} \\ & \mathbf{C}\boldsymbol{\lambda} \geq \mathbf{0} \\ & \boldsymbol{\lambda} \geq \mathbf{0}. \end{aligned} \tag{15}$$

This is a standard QP with linear constraints, which can be solved efficiently by many off-the-shelf convex optimization packages, e.g., the *quadprog* function in MATLAB. The QP coefficient matrices $\mathbf{Q}$, $\mathbf{r}$ and constraint matrix $\mathbf{C}$ are defined as follows, respectively

$$\begin{aligned} \mathbf{Q} &= \left(\overline{\mathbf{\Phi}}_{\mathcal{Z}}^{\mathsf{T}} \overline{\mathbf{\Phi}}_{\mathcal{Z}}\right) \odot \left(\overline{\mathbf{\Phi}}_{\mathcal{Z}}^{\mathsf{T}} \overline{\mathbf{\Phi}}_{\mathcal{Z}}\right) \\ \mathbf{r} &= \mathrm{diag}\left(\overline{\mathbf{\Phi}}_{\mathcal{Z}}^{\mathsf{T}} \mathbf{K}_{\mathcal{Z}} \overline{\mathbf{\Phi}}_{\mathcal{Z}}\right) \\ \mathbf{C} &= \mathbf{I} - \zeta \overline{\mathbf{I}}, \end{aligned} \tag{16}$$

where $\zeta \geq 1$ is the eigenspectrum damping factor and is the only tunable parameter in TKL, $\mathbf{I} \in \mathbb{R}^{n \times n}$ is the identity matrix, $\overline{\mathbf{I}} \in \mathbb{R}^{n \times n}$ is the 1th-diagonal matrix with the nonzero elements $\overline{I}_{i,i+1} = 1, i = 1, \ldots, n-1$.

Real-world problems usually exhibit the "eigengap" property, i.e., the largest r eigenvalues of the kernel matrix are much larger than the remaining eigenvalues [32]. In such scenarios, it is unnecessary to compute the full eigenspectrum of target kernel $\mathbf{K}_{\mathcal{X}}$. Retaining the largest r eigenvalues and eigenvectors results in significantly reduced computational complexity. In this paper, to speed up computation, we fix $r = \min(500, n)$, i.e. $\overline{\mathbf{\Phi}}_{\mathcal{Z}} \in \mathbb{R}^{m\times r}$, $\boldsymbol{\lambda} \in \mathbb{R}^{r\times 1}$, $\mathbf{Q} \in \mathbb{R}^{r\times r}$, which is an order fast yet accurate for most problems. A complete procedure is summarized in Algorithm 1.

4.3.2 Scalable Implementation

Standard kernel machines are greatly challenged in the "big data" era, since these methods usually require $O(n^2)$ computation and cannot scale to large datasets. The Nyström method is a well-established solution to the large-scale kernel machines [17], [28]. In particular, the Nyström method approximates a large-scale target kernel matrix $\mathbf{K}_{\mathcal{X}} \in \mathbb{R}^{n\times n}$ by (randomly or uniformly) sampling a subset of its columns $\mathbf{K}_{\mathcal{X}\hat{\mathcal{X}}} \in \mathbb{R}^{n\times\hat{n}}, \hat{n} \ll n$. Denote $\mathbf{K}_{\hat{\mathcal{X}}}$ the kernel matrix on the data subset $\hat{\mathcal{X}}$, whose eigensystem is learned via eigendecomposition $\mathbf{K}_{\hat{\mathcal{X}}}\mathbf{\Phi}_{\hat{\mathcal{X}}} = \mathbf{\Phi}_{\hat{\mathcal{X}}}\mathbf{\Lambda}_{\hat{\mathcal{X}}}$. By Nyström approximation (5), the eigensystem of target kernel $\mathbf{K}_{\mathcal{X}}$ is extrapolated as

$$\mathbf{\Phi}_{\mathcal{X}} \simeq \mathbf{K}_{\mathcal{X}\hat{\mathcal{X}}}\mathbf{\Phi}_{\hat{\mathcal{X}}}\mathbf{\Lambda}_{\hat{\mathcal{X}}}^{-1}. \tag{17}$$

Similarly, the large-scale source kernel $\mathbf{K}_{\mathcal{Z}} \in \mathbb{R}^{m\times m}$ can also be approximated by a subset of its columns $\mathbf{K}_{\mathcal{Z}\hat{\mathcal{Z}}} \in \mathbb{R}^{m\times\hat{m}}, \hat{m} \ll m$. Denote $\mathbf{K}_{\hat{\mathcal{Z}}}$ the kernel matrix on the data subset $\hat{\mathcal{Z}}$. By Nyström approximation in Equation (6), source kernel $\mathbf{K}_{\mathcal{Z}}$ is approximated as

$$\mathbf{K}_{\mathcal{Z}} \simeq \mathbf{K}_{\mathcal{Z}\hat{\mathcal{Z}}}\mathbf{K}_{\hat{\mathcal{Z}}}^{-1}\mathbf{K}_{\hat{\mathcal{Z}}\mathcal{Z}}. \tag{18}$$

To scale up QP (15), we also need to approximate the extrapolated eigensystem $\overline{\mathbf{\Phi}}_{\mathcal{Z}}$. We extrapolate the target eigensystem $\mathbf{\Phi}_{\mathcal{X}}$ to the source data subset $\hat{\mathcal{Z}}$ using the cross-domain Nyström approximation in (9) as

$$\overline{\mathbf{\Phi}}_{\hat{\mathcal{Z}}} \simeq \mathbf{K}_{\hat{\mathcal{Z}}\mathcal{X}}\mathbf{\Phi}_{\mathcal{X}}\mathbf{\Lambda}_{\hat{\mathcal{X}}}^{-1}. \tag{19}$$

Then we extrapolate the eigensystem $\overline{\mathbf{\Phi}}_{\hat{\mathcal{Z}}}$ to the full source data $\mathcal{Z}$ using within-domain Nyström method as

$$\overline{\mathbf{\Phi}}_{\mathcal{Z}} \simeq \mathbf{K}_{\mathcal{Z}\hat{\mathcal{Z}}}\overline{\mathbf{\Phi}}_{\hat{\mathcal{Z}}}\mathbf{\Lambda}_{\hat{\mathcal{X}}}^{-1}. \tag{20}$$

Thus, all variables in QP (15) and domain-invariant kernel (12) are turned into scalable implementations. For space limitation, we leave out the detailed scalable procedure since it has similar routines as Algorithm 1.

Algorithm 1. TKL: Transfer Kernel Learning

```
  Input: Data $\mathbf{X}$; kernel $k$; eigen-damping factor $\zeta$.
  Output: Domain-invariant kernel $\overline{\mathbf{K}}_{\mathcal{A}}$.
1 begin
2     Compute matrices $\mathbf{K}_{\mathcal{Z}}$, $\mathbf{K}_{\mathcal{X}}$, $\mathbf{K}_{\mathcal{Z}\mathcal{X}}$ by kernel $k$.
3     Eigendecompose $\mathbf{K}_{\mathcal{X}}$ for $\{\mathbf{\Lambda}_{\mathcal{X}}, \mathbf{\Phi}_{\mathcal{X}}\}$ by (8).
4     Extrapolate for source eigensystem $\overline{\mathbf{\Phi}}_{\mathcal{Z}}$ by (9).
5     Solve QP problem (15) for eigenspectrum $\boldsymbol{\lambda}$.
6     Return domain-invariant kernel matrix as (12).
```

4.4 Computational Complexity

We analyze the computational complexity with the big O notation. Denote r the number of retained largest eigenvectors, which is fixed to $r = \min(500, n)$. For original implementation of TKL in Algorithm 1, computing kernel matrices in Line 2 requires $O(d(m+n)^2)$, eigendecomposing the target kernel in Line 3 requires $O(rn^2)$, extrapolating for the source eigensystem in Line 4 requires $O(rmn)$, solving QP in Line 5 requires $O(rn^2 + r^3)$, constructing the domain-invariant kernel in Line 6 requires $O(r(m+n)^2)$. The overall complexity of Algorithm 1 is quadratic order: $O((d+r)(m+n)^2)$.

For scalable implementation, the overall complexity (excluding domain-invariant kernel $\mathbf{K}_{\mathcal{A}}$ construction) is reduced to linear order: $O((d+r)(\hat{m}+\hat{n})(m+n))$, where $\hat{m} \ll m$, $\hat{n} \ll n$ [28]. Since constructing $\mathbf{K}_{\mathcal{A}}$ still requires $O(r(m+n)^2)$, while $\mathbf{K}_{\mathcal{A}}$ is only low-rank, in practice, $\mathbf{K}_{\mathcal{A}}$ is not directly computed. Instead, Equation (12) is fed to standard kernel machines and matrix computation tricks are applied to the decomposed kernel matrix to achieve linear complexity.

4.5 Approximation Error Analysis

We theoretically analyze the relationship between the distribution divergence and Nyström approximation error with the following theorem by Zhang et al. [17].

Theorem 4 (Approximation Error [17]). *Define* $\mathcal{E}_{Nys} \triangleq \left\|\mathbf{K}_{\mathcal{Z}} - \mathbf{K}_{\mathcal{Z}\mathcal{X}}\mathbf{K}_{\mathcal{X}}^{-1}\mathbf{K}_{\mathcal{X}\mathcal{Z}}\right\|_F$ *as the Nyström approximation error, and define* $\mathcal{E}_{TKL} \triangleq \left\|\overline{\mathbf{\Phi}}_{\mathcal{Z}}\mathbf{\Lambda}\overline{\mathbf{\Phi}}_{\mathcal{Z}}^{\mathsf{T}} - \mathbf{K}_{\mathcal{Z}}\right\|_F$ *as the TKL approximation error, then the errors are bounded by*

$$\mathcal{E}_{TKL} \le \mathcal{E}_{Nys} \le 4m\sqrt{C_k mn\epsilon} + C_k mn\epsilon\left\|\mathbf{K}_{\mathcal{X}}^{-1}\right\|_F, \tag{21}$$

where C_k *is constant,* $\epsilon = \sum_{i=1}^{m}\left\|\mathbf{z}_i - \mathbf{x}_{\mathrm{NN}(i)}\right\|^2$ *is the total quantization error of coding each sample* $\mathbf{z}_i \in \mathcal{Z}$ *with its nearest-neighbor* $\mathbf{x}_{\mathrm{NN}(i)} = \arg\min_{\mathbf{x}_j}\left\|\mathbf{z}_i - \mathbf{x}_j\right\| \in \mathcal{X}$.

The first inequality holds since if the eigenspectrum parameters are not relaxed, that is, $\mathbf{\Lambda} = \mathbf{\Lambda}_{\mathcal{X}}$, then by Equation (6), we have $\mathcal{E}_{TKL} = \mathcal{E}_{Nys}$. However, in TKL model (11), $\mathbf{\Lambda}$ are free parameters while the approximation error is directly minimized, hence $\mathcal{E}_{TKL} \le \mathcal{E}_{Nys}$.

This theorem indicates minimizing the approximation error is equivalent to requiring the source points to overlap sufficiently with the target data. In other words, each source point should correspond to at least one target point that minimizes the quantization error. The TKL optimization (11) is a spectral relaxation of the Nyström approximation. Thus minimizing (11) essentially maximizes the overlapping of the cluster structures between the source data and target data, which finally matches source and target distributions.

5 EXPERIMENTS

We perform comprehensive experiments on three real-world cross-domain applications (text classification, visual object recognition, and video event recognition) to evaluate the efficacy of the proposed TKL approach.

TABLE 2
Top Categories and Subcategories in the Text Datasets

Top Category	Subcategory	#Examples	#Features
	comp.graphics	970	
comp	comp.os.ms-windows.misc	963	
	comp.sys.ibm.pc.hardware	979	
	comp.sys.mac.hardware	958	
	rec.autos	987	
rec	rec.motorcycles	993	
	rec.sport.baseball	991	
	rec.sport.hokey	997	25,804
	sci.crypt	989	
sci	sci.electronics	984	
	sci.med	987	
	sci.space	985	
	talk.politics.guns	909	
talk	talk.politics.mideast	940	
	talk.politics.misc	774	
	talk.religion.misc	627	
orgs	many subcategories	1,237	
people	many subcategories	1,208	4,771
place	many subcategories	1,016	

5.1 Data Preparation

5.1.1 Text Datasets

The 222 cross-domain text datasets are generated from 20-Newsgroups and Reuters-21578, which are two benchmark text corpora widely used for transfer learning evaluation [12], [15], [16], [35], [36], [37], [38].

20-Newsgroups[1] has approximately 20,000 documents distributed evenly in 20 different subcategories. The corpus contains four top categories *comp*, *rec*, *sci* and *talk*. Each top category has four subcategories, which are listed in Table 2. In the experiments, we can construct six dataset groups for binary classification by randomly selecting two top categories (one for positive and the other one for negative) from the four top categories. The six dataset groups are *comp vs rec*, *comp vs sci*, *comp vs talk*, *rec vs sci*, *rec vs talk*, and *sci vs talk*. Similar to the approach in [35], we set up one dataset (including source domain and target domain) for cross-domain classification as follows. For each pair of top categories P and Q (e.g., P for positive and Q for negative), their four sub-categories are denoted by P_1, P_2, P_3, P_4 and Q_1, Q_2, Q_3, Q_4, respectively. We randomly select (without replacement) two subcategories from P (e.g., P_1 and P_2) and two subcategories from Q (e.g., Q_1 and Q_2) to form a source domain, then the remaining subcategories in P and Q (i.e., P_3, P_4 and Q_3, Q_4) are selected to form a target domain. This dataset construction strategy ensures that the domains of labeled and unlabeled data are related, since they are under the same top categories. Besides, the domains are also ensured to be different, since they are drawn from different subcategories. In this way, for each dataset group P *vs* Q, we can generate $C_4^2 \cdot C_4^2 = 36$ datasets. Clearly, for each example in the generated dataset group, its class label is either P or Q. In total, we can generate 6 dataset groups consisting of $6 \cdot 36 = 216$ datasets. For fair comparison, the 216 datasets are constructed using a preprocessed version of 20-Newsgroups [36], which contains 25,804 features and 15,033 documents, with each document weighted by *term frequency-inverse document frequency* (TF-IDF).

Fig. 2. Office, Caltech-256, Kodak, YouTube datasets.

TABLE 3
Statistics of Cross-Domain Image and Video Datasets

Dataset	Type	#Examples	#Features	#Classes	Subsets
Office	Object	1,410	800	10	A, W, D
Caltech-256	Object	1,123	800	10	C
Kodak	Event	195	2,500	6	Kodak
YouTube	Event	906	2,500	6	YouTube

Reuters-21578[2] is a difficult text dataset with many top and subcategories. The three largest top categories are *orgs*, *people*, and *place*, each of which is comprised of many subcategories. Using the three largest top categories, we can generate 6 cross-domain text datasets *orgs vs people*, *people vs orgs*, *orgs vs place*, *place vs orgs*, *people vs place*, and *place vs people*. For fair comparison, we directly adopt the preprocessed version of Reuters-21578 [39]. Note that, in previous works, only three out of the six cross-domain datasets were evaluated. In this paper, we conduct a more comprehensive evaluation by using all the configurations of the datasets.

5.1.2 Image Datasets

The 12 cross-domain image datasets are generated from Office and Caltech-256 (detailed in Fig. 2 and Table 3), which are two benchmark datasets widely adopted for visual domain adaptation evaluation [19].

Office [4], [40], [41] is an increasingly popular benchmark database for visual domain adaptation. The database contains three object domains, i.e., *Amazon* (images downloaded from online merchants), *webcam* (low-resolution images by a web camera), and *DSLR* (high-resolution images by a digital SLR camera). It consists of 4,652 images with 31 categories.

Caltech-256 is a standard database for object recognition. It consists of 30,607 images and 256 categories.

In experiments, we adopt the smaller Office+Caltech datasets released by Gong et al. [4]. SURF features are extracted and quantized into an 800-bin histogram with codebooks computed with Kmeans on a subset of images from *Amazon*. Then the histograms are standardized by z-score. More specifically, we have four domains, **C** (Caltech-256), **A** (Amazon), **W** (webcam), and **D** (DSLR). By randomly selecting two different domains as source domain and target domain respectively, we can construct $4 \times 3 = 12$ cross-domain object datasets, e.g., $C \rightarrow A$, $C \rightarrow W$, $C \rightarrow D, \ldots, D \rightarrow W$.

[1] http://people.csail.mit.edu/jrennie/20newsgroups

[2] http://www.daviddlewis.com/resources/testcollections/reuters21578

5.1.3 Video Datasets

We adopt the public cross-domain video event recognition dataset released by Duan et al. [6] (detailed in Fig. 2 and Table 3). The dataset is comprised of both high-quality Kodak videos and low-quality YouTube videos. The Kodak videos were derived from the Kodak Consumer Video Benchmark Dataset [6], which was originally collected and annotated by Kodak from 100 real users over the period of one year. The YouTube videos were collected using keyword-based search. To build a stronger benchmark, Duan et al. further extended the dataset by including more Kodak videos and YouTube videos on their team's efforts. The extended dataset is very challenging for transfer learning algorithms due to two reasons: 1) the quality of YouTube videos is much lower than that of Kodak videos directly collected from real users; 2) the YouTube videos are only loosely labeled while manual labeling may be prohibitive in the web applications.

In the experiments, six events, "birthday", "picnic", "parade", "show", "sport", "wedding", are selected for evaluation. The total numbers of Kodak videos and YouTube videos are 195 and 906, respectively. In real-world scenarios, labeled examples in the target domain (i.e., consumer video domain) are often much fewer than those in the source domain (i.e., web video domain). To meet this requirement, all the 906 loosely labeled YouTube videos are used as labeled training data in the source domain, while only three consumer videos from each event (18 videos in total) are randomly sampled as the labeled training videos in the target domain, with the remaining videos in the target domain used as the test data. Duan et al. released configuration files indicating how the labeled target training videos were sampled in their experiments. Thus it is easy to exactly reproduce the results in [6].

The dataset is preprocessed as follows. Each video clip is sampled at a rate of two frames per second, with 65 frames per video on average. For each frame, 128D SIFT features from salient regions are extracted, with 4,144 features per video on average. The features of each video are quantized into an 2,500-bin histogram with codebook computed from Kmeans clustering. To effectively match two videos across different space-time locations, the bag-of-word (BOW) histograms are further processed by the aligned space-time pyramid matching (ASTPM) proposed also by Duan et al. [6].

5.2 Comparison Methods

To fully evaluate the robustness of the proposed TKL approach to different configurations of datasets, we conduct comprehensive evaluation on all the 235 text, image, and video datasets, and compare TKL with a large number of competitive state-of-the-art methods

- support vector machine (SVM) [20]
- Laplacian support vector machine (LapSVM) [42]
- partial least squares (PLS) [20]
- canonical correlation analysis (CCA) [22]
- kernel canonical correlation analysis (KCCA) [22]
- cross-domain spectral classification (CDSC) [43]
- spectral feature alignment (SFA) [8]
- kernel mean matching (KMM) [10]
- transfer component analysis (TCA) [12]
- transfer multiple kernel learning (DTMKL) [15]
- geodesic flow kernel (GFK) [4]
- surrogate kernel matching [16]

In particular, DTMKL, GFK, and SKM are the kernel learning methods for domain adaptation. DTMKL is a multiple kernel learning framework, which integrates multiple parametric kernels via linear ensemble, and reduces the domain difference underlying all kernels. GFK extracts infinite number of subspaces that transit smoothly along the geodesic flow from the source domain to the target domain, and computes the cross-domain similarity kernel using all the subspaces. SKM extrapolates a surrogate kernel for the source domain from the eigensystem of the target domain kernel, and aligns it with the source domain kernel using feature transformation. SKM is the most closely related work to TKL, however, TKL differs significantly from SKM in two important perspectives: 1) TKL directly learns a domain-invariant spectral kernel, while SKM simply aligns pre-computed parametric kernels; 2) TKL minimizes the Nyström approximation error between the source and target kernels, while SKM minimizes a traditional kernel alignment objective.

5.3 Implementation Details

For fair comparison, we adopt the same evaluation protocols as [2], [12], [15], [16]. Specifically, SVM and PLS are trained on the labeled source data and tested on unlabeled target data; LapSVM and DTMKL are trained on all data to induce transfer classifiers; CCA, KCCA, CDSC, SFA, KMM, TCA, GFK, SKM, and TKL are trained on all data to learn the domain-invariant feature representation, instance selection, or similarity kernel, then an SVM classifier is trained on the labeled source data to discriminate the unlabeled target data.

Under our experimental setup, it is impossible to select the optimal parameters for the transfer learners using cross validation, since we have no labeled target data. Therefore, we fairly evaluate all the comparison methods on our datasets by grid-searching the parameter space for their respective optimal parameter settings, and report the best results of each method. For SVM,[3] we set the trade-off parameter C by searching $C \in \{0.1, 0.5, 1, 5, 10, 50, 100\}$. For LapSVM,[4] we set regularization parameters γ_A and γ_I by searching $\gamma_A, \gamma_I \in \{0.01, 0.05, 0.1, 0.5, 1, 5, 10\}$. For DTMKL, we set the trade-off parameter θ between the structural risk functional and the distribution matching term by searching $\theta \in \{0.01, 0.1, 1, 10, 100\}$. For PLS, CCA, KCCA, CDSC, SFA, TCA, and GFK, we set the optimal dimension k by searching $k \in \{10, 20, \ldots, 100\}$. For all the kernel-based methods, we use linear kernel, i.e., $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^{\mathsf{T}}\mathbf{x}_j$ on the text datasets, and RBF kernel i.e., $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2}$ on the image and video datasets. We use the default kernel parameter $\gamma = \frac{1}{A}$, where A is the median value of the square distances between all the training samples as suggested in [6].

TKL is an elegant approach since it involves only two model parameters: eigenspectrum damping factor ζ and SVM trade-off parameter C. In the subsequent sections, we

[3] http://www.csie.ntu.edu.tw/~cjlin/libsvm

[4] http://vikas.sindhwani.org/manifoldregularization.html

TABLE 4
Average Classification Accuracy (Percent) on the Nine Cross-Domain Text Dataset Groups Comprising of 222 Datasets

Dataset Group	Standard Learning					Non-Kernel Learning				Kernel Learning			
	SVM	LapSVM	PLS	CCA	KCCA	CDSC	SFA	KMM	TCA	DTMKL	GFK	SKM	TKL
comp vs rec	87.51	81.93	88.20	88.87	89.68	87.95	89.73	93.64	95.12	95.08	93.74	91.31	**96.01**
comp vs sci	75.38	68.96	76.25	77.14	76.60	75.72	78.07	77.45	77.32	81.87	80.62	77.63	**88.14**
comp vs talk	95.44	95.40	96.19	95.34	95.29	97.33	95.85	96.06	97.20	97.16	96.61	**97.75**	**97.74**
rec vs sci	73.82	74.21	75.40	78.98	78.47	77.53	79.25	80.27	82.31	82.97	84.31	77.21	**91.29**
rec vs talk	83.27	87.44	83.12	87.03	86.63	82.14	86.98	85.57	86.58	88.35	92.73	86.83	**93.74**
sci vs talk	76.85	80.22	76.18	78.63	79.67	80.97	79.27	77.05	79.30	75.77	80.91	78.30	**87.52**
Average	82.05	81.36	82.56	84.33	84.39	83.62	84.86	85.01	86.31	86.87	88.15	84.84	**92.41**
orgs vs people	78.55	82.68	79.40	75.90	76.47	80.97	77.20	80.48	81.58	81.19	81.00	78.63	**83.76**
orgs vs place	66.71	68.67	67.87	73.47	73.90	70.62	74.59	68.47	68.15	69.20	76.31	68.06	**80.85**
people vs place	59.94	60.68	60.96	65.73	66.14	64.53	67.08	57.33	57.61	57.80	58.50	59.33	**68.48**
Average	68.40	70.68	69.41	71.70	72.17	72.04	72.96	68.76	69.11	69.40	71.94	68.67	**77.70**

will conduct detailed sensitivity analysis, which validates that TKL can achieve superior performance under a wide range of parameter values. In the comparative study, we empirically fix $C = 10.0$ and set 1) $\zeta = 2.0$ for all the 222 text datasets, and 2) $\zeta = 1.1$ for all the 13 image and video datasets. Since C is more stable than ζ, we can simplify model selection by first determining C and then choosing ζ.

We use the classification *Accuracy* on the test data (unlabeled target data) as the evaluation metric, since it is widely adopted in the literature [8], [12], [15], [16]

$$Accuracy = \frac{|\mathbf{x} : \mathbf{x} \in \mathcal{X} \wedge f(\mathbf{x}) = y(\mathbf{x})|}{|\mathbf{x} : \mathbf{x} \in \mathcal{X}|},$$

where $y(\mathbf{x})$ is the ground truth label of $\mathbf{x}$ while $f(\mathbf{x})$ is the label predicted by the classification algorithm.

5.4 Experimental Results

In this section, we extensively evaluate TKL and the nine competitive methods on the 235 text, image, and video datasets in terms of the classification accuracy.

5.4.1 Results of Text Categorization

As 20-Newsgroups and Reuters-21578 are different in hierarchical structure, we report the results separately.

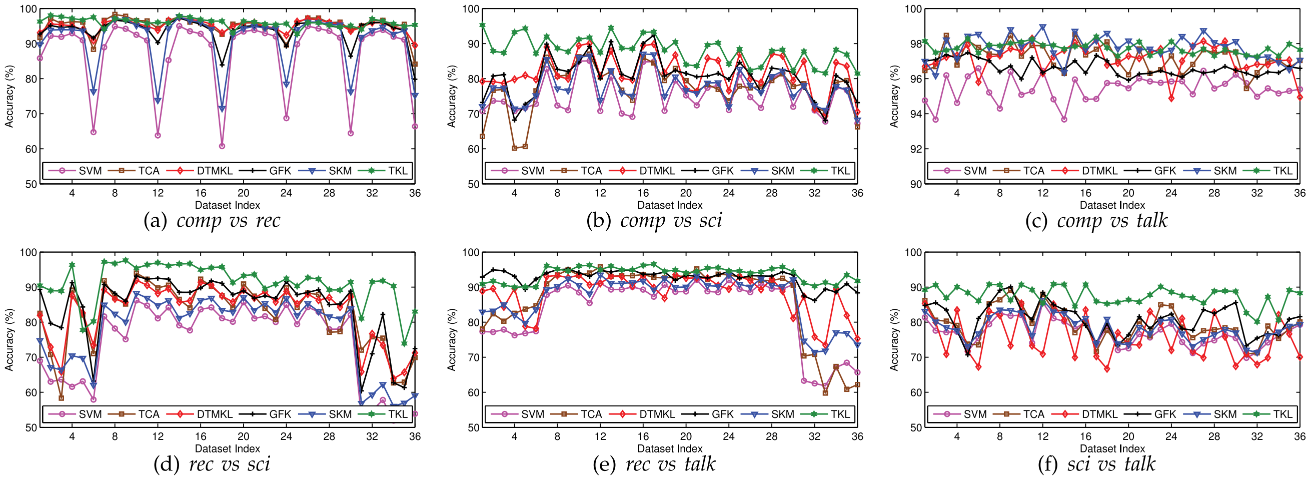
20-Newsgroups. From this dataset, we constructed six cross-domain dataset groups, i.e., *comp vs rec*, *comp vs sci*, *comp vs talk*, *rec vs sci*, *rec vs talk*, and *sci vs talk*, where each dataset group consists of 36 cross-domain text datasets. The classification accuracy is computed on each dataset while the average classification accuracy is computed on each dataset group by averaging the classification accuracies on the corresponding 36 datasets. We list the average classification accuracies of TKL and the nine comparison methods in Table 4. For completeness, we illustrate the detailed results on all 216 datasets in Figs 3a, 3b, 3c, 3d, 3e, and 3f, where each figure shows the results on the 36 datasets of each group. From the results, we can make several observations.

Firstly, TKL can achieve impressively better performance than the twelve comparison methods on most of (171 out of 216) datasets with statistical significance. The average classification accuracy of TKL on the 216 datasets is *92.41 percent* and the performance improvement is *4.26 percent* compared to the best competitive method GFK, which means a great error reduction of *35.95 percent*. Note that, the adaptation difficulty in the 216 datasets varies substantially, since the standard SVM classifier can only achieve an average classification accuracy of 82.05 percent, and may perform very poorly on many of the datasets. Although TKL cannot perform the best on all 216 datasets, it is still established as an effective and robust approach due to these important perspectives: 1) it performs particularly better on difficult datasets where other competitive methods may have to struggle; 2) it performs comparably well on easy datasets where most methods can succeed. The consistent performance gains on a large number of datasets strongly verify that TKL can successfully build robust domain-invariant kernels for cross-domain text classification.

Secondly, standard supervised (SVM and PLS) and semi-supervised (LapSVM) methods cannot achieve good performance on many cross-domain datasets. The major limitation of standard learning methods is that they require the training data and test data to follow identical probability distributions. However, this requirement is inevitably violated in the cross-domain learning problems, resulting in a severe performance degradation. It is a bit surprising to observe that, even by exploring the target data in a transductive way, LapSVM still performs worse than SVM and PLS. This reveals the importance of minimizing the distribution discrepancy between domains to mitigate overfitting.

Thirdly, transfer learning methods generally outperform standard learning methods. For clarity, we categorize existing transfer learning methods into two categories, *non-kernel learning* and *kernel learning*, and discuss the advantages and disadvantages in details.

Non-kernel learning. CCA, CDSC, SFA, KMM, and TCA are non-kernel learning methods. CDSC extracts a shared spectral embedding across domains by jointly preserving in-domain geometric structure and out-of-domain discriminative information. CCA and SFA learn a shared representation by maximizing the correlations between domain-specific features and domain-invariant features. A major limitation of CCA, CDSC and SFA is that they do not explicitly minimize the distribution discrepancy between domains, hence their generalization error is not theoretically bounded. KMM and TCA overcome this limitation by explicitly minimizing the



(a) *comp vs rec* (b) *comp vs sci* (c) *comp vs talk*

(d) *rec vs sci* (e) *rec vs talk* (f) *sci vs talk*

Fig. 3. Classification accuracy of SVM, TCA, DTMKL, GFK, SKM, TKL on the 216 cross-domain text datasets.

maximum mean discrepancy [9] between different distributions in the reproducing kernel Hilbert space. Nonetheless, minimizing MMD can only match first- and high-order moments (mean, variance, etc) of different distributions. The distribution matching is far from perfect and the thus generalization error bound will be too loose [10].

Kernel learning. DTMKL, GFK, and SKM are the kernel learning methods. DTMKL reduces the domain difference by explicitly minimizing MMD underlying multiple different kernels. The advantage of DTMKL over KMM and TCA is that it jointly matches multiple kernels across domains, thus it can reduce the distribution difference to a greater extent. However, due to the limitation of MMD, DTMKL still cannot match different distributions sufficiently. GFK extracts infinite number of subspaces that transit smoothly along the geodesic flow from the source domain to the target domain, and computes the cross-domain similarity kernel using all the subspaces. A limitation of GFK is that the subspace dimension should be small enough to ensure that different subspaces can transit smoothly along the geodesic flow, but the low-dimensional subspaces cannot represent input data accurately. SKM extrapolates a surrogate kernel for the source domain from the eigensystem of the target domain kernel, and aligns it with the source domain kernel using linear feature transformation. A great disadvantage of SKM is that it only aligns pre-computed kernels instead of learning a data-dependent domain-invariant kernel, thus it cannot match different distributions perfectly.

In particular, we design TKL to address the above limitations of the non-kernel learning or kernel learning methods: 1) TKL learns a domain-invariant kernel by explicitly minimizing the difference between the source domain kernel and target domain extrapolated kernel, which is a more accurate criterion than MMD to measure the distribution difference; 2) TKL explores the entire eigenspectrum of the input space, thus it can construct an accurate kernel to represent input data; 3) TKL is a spectral learning method which learns data-dependent kernels instead of pre-computed kernels, thus it can match different distributions better. These advantages guarantee superior performances of TKL.

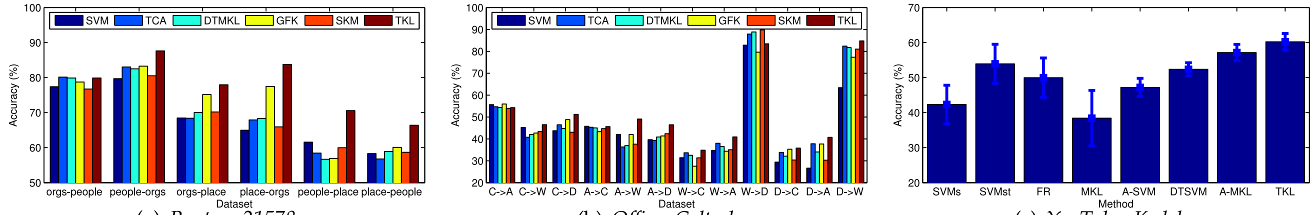
Reuters-21578. From this dataset, we constructed three cross-domain dataset groups, each of which consists of two text datasets. The average classification accuracy on each dataset group is listed in Table 4. The detailed classification accuracy on each dataset is illustrated in Fig. 4a. We observe that TKL has significantly outperformed all the comparison methods on all the datasets. Specifically, the overall classification accuracy of TKL is *77.70 percent*, and the performance improvement from the best competitive method SFA is *4.74 percent*. Note that, our results are the first public reports on all six configurations of the Reuters-21578 released by [39].

It is noteworthy that, Reuters-21578 is much more challenging than 20-Newsgroups, since each of its top categories consists of many diverse subcategories, i.e., clusters or subclasses. Therefore, it is more difficult for CCA, CDSC, SFA, GFK, and SKM to extract shared latent structures for knowledge transfer. It is also more difficult for KMM, TCA, and DTMKL to reduce the distribution discrepancy using MMD, since MMD can only match the first- and high-order statistics (mean and moments) of different distributions but may not match the diverse cluster structures between domains. This explains the incremental performance gains of the comparative transfer methods over standard SVM.

In particular, TKL can naturally explore the diverse cluster structures and match them between domains. TKL achieves this goal by matching the target domain extrapolated kernel with the source domain kernel and then increasing (decreasing) the domain-invariant (domain-dependent) eigenspectrum simultaneously. This advantage explains TKL's superior performance.

5.4.2 Results of Image Classification

The classification (recognition) accuracies of TKL and the twelve comparison methods on the 12 cross-domain visual object datasets are listed in Table 5. For better interpretation, the results are visualized in Fig. 4b. TKL achieves impressively better performance than the nine comparison methods on most (eight out of 12) datasets. The average classification accuracy of TKL on the 12 datasets is *51.12 percent*, which gains a significant performance improvement of *3.09 percent* compared to the best competitive method TCA. Note that, the object recognition datasets are very challenging in computer vision, since most transfer learning methods can only



(a) *Reuters-21578* (b) *Office+Caltech* (c) *YouTube+Kodak*

Fig. 4. Classification accuracy of SVM, TCA, DTMKL, GFK, SKM, TKL on the text, image, and video datasets.

achieve a small margin of improvement over standard SVM, e.g., 2.99 percent performance improvement by TCA.

Visual domain adaptation is more challenging than textual domain adaptation. In visual domain adaptation, there exists the so-called "semantic gap" between low-level image features and high-level categorical information, which poses a great challenge to visual recognition models. In this case, it is difficult to define the relevance of low-level image features to different domains, leading to *negative transfer* for CCA and SFA.

Furthermore, the domain difference in visual domain adaptation is more significant than that of textual domain adaptation. Therefore, it is indispensable to explicitly formalize and minimize the distribution difference between domains. CDSC does not explicitly match different distributions, thus it can only perform comparably as standard SVM. TCA and DTMKL significantly outperform SVM, indicating the importance of distribution matching in visual domain adaptation.

However, the limitation of MMD-based distribution matching is more severe in visual domain adaptation. Theoretically, reducing MMD is equivalent to performing the *translation* operation on the target images by an offset to the source images in the reproducing kernel Hilbert space [9]. Therefore, it cannot effectively reshape the visual datasets across domains. Though GFK and SKM partially alleviated this limitation, they introduced new issues: 1) GFK is limited by the low-dimensional subspaces which are inaccurate for data representation; 2) SKM is limited by the pre-computed kernels which are not adaptive to domains.

In particular, by fully exploring the eigenspaces of the source and target domain data to learn a domain-invariant kernel, TKL can successfully reshape visual datasets across domains for visual domain adaptation.

5.4.3 Results of Video Event Recognition

This experiment falls into the recent research trend of "Internet Vision", where massive web data including images and videos with rich and valuable contextual information (e.g., tags, categories, and captions) are employed for various computer vision applications [6]. To be simple and fair, we directly compare TKL with the methods reported in Duan et al. [6] by using their experimental settings. Specifically, we compare TKL with SVM$_T$ (the labeled training data are from the target domain), SVM$_{ST}$ (the labeled training data are from both the source and target domains), FR [44], A-SVM [45], DTSVM [5], MKL [5], and A-MKL [6].

The recognition accuracies of TKL and the seven comparison methods are listed in Table 6. For better interpretation, the results are visualized in Fig. 4c. TKL significantly outperforms the seven competitive methods on the video event dataset. As pointed out by Duan et al. [6], the video event dataset is particularly challenging: the domain difference becomes so large that many transfer learning methods, e.g., FR, A-SVM, and DTSVM, have suffered from the *negative transfer* problem, i.e., performed worse than standard SVM$_{ST}$. Duan et al. proposed A-MKL to fuse the information from multiple features and cope with the considerable variation in feature distributions. By using this complicated framework, A-MKL achieved *positive transfer* on the challenging dataset, i.e., +3.21 percent improvement. However, A-MKL still relies on the MMD-based distribution matching, which severely limits its capability to reduce the video variation across different domains.

TABLE 5
Average Classification Accuracy (Percent) on the 12 Cross-Domain Visual Object Recognition Datasets (10 Categories)

Dataset	Standard Learning					Non-Kernel Learning				Kernel Learning			
	SVM	LapSVM	PLS	CCA	KCCA	CDSC	SFA	KMM	TCA	DTMKL	GFK	SKM	TKL
C → A	55.64	**56.27**	57.31	47.95	49.06	52.16	49.32	48.32	54.70	54.33	55.95	53.97	54.28
C → W	45.22	45.80	46.50	38.31	38.42	38.54	39.31	45.78	40.76	42.04	42.68	43.31	**46.50**
C → D	43.73	43.73	44.75	41.51	42.12	43.64	41.96	**53.53**	46.44	44.74	48.81	43.05	51.19
A → C	**45.77**	44.23	44.43	41.25	42.25	42.28	42.33	42.21	45.33	45.01	43.28	44.70	45.59
A → W	42.04	42.74	42.04	34.33	35.71	34.94	34.94	42.38	36.31	36.94	42.04	37.58	**49.04**
A → D	39.66	39.79	41.02	35.43	36.95	37.81	36.86	42.72	39.32	40.85	41.36	42.37	**46.44**
W → C	31.43	31.99	30.90	32.46	33.15	32.28	32.50	29.01	33.66	32.50	27.52	31.34	**34.82**
W → A	34.76	34.77	31.84	34.26	34.14	35.73	34.72	31.94	38.00	36.53	34.34	35.07	**40.92**
W → D	82.80	83.43	84.71	81.72	81.62	81.80	83.38	71.98	87.90	88.85	79.62	**89.81**	83.44
D → C	29.39	29.49	30.72	28.76	28.87	33.33	30.50	31.61	33.84	32.10	35.26	30.37	**35.80**
D → A	26.62	27.37	29.54	28.07	29.35	35.88	29.41	32.20	37.79	34.03	37.68	30.27	**40.71**
D → W	63.39	64.31	73.55	66.86	66.97	80.76	68.14	72.88	82.37	81.69	77.29	81.02	**84.75**
Average	45.04	45.32	46.44	42.58	43.22	45.76	43.61	45.38	48.03	47.47	47.15	46.91	**51.12**

TABLE 6
Average Classification Accuracy (Percent) on Cross-Domain Video Event Recognition Dataset (Five Random Experiments)

Method	SVM$_T$	SVM$_{ST}$	FR	MKL	A-SVM	DTSVM	A-MKL	TKL
Accuracy	42.32 ± 5.50	53.93 ± 5.58	49.98 ± 5.63	38.42 ± 7.93	47.19 ± 2.59	52.36 ± 1.88	57.14 ± 2.34	$\mathbf{60.22 \pm 2.35}$

The average classification accuracy of TKL on this challenging dataset is *60.22 percent*. The performance improvement from the best competitive method A-MKL is *3.08 percent*. It is worth noting that TKL is much simpler than A-MKL, yet achieves much better performance. This verifies that learning a data-dependent domain-invariant kernel is more effective than ensembling the multiple pre-computed kernels for video adaptation.

5.5 Effectiveness Analysis

We analyze the effectiveness of TKL by inspecting the distribution divergence computed using the learned domain-invariant kernel. We first perform SVM, TCA, SKM, and TKL on all the 12 image datasets with their optimal parameter settings, respectively. We compute the MMD divergence of each method on its induced embedding or kernel by Equation (1). As theoretically analyzed in [9], smaller distribution divergence implies better cross-domain generalization performance.

The distribution divergences and the classification accuracies of the four methods on the image datasets are illustrated in Figs 5a and 5b, respectively. We can make several observations. Without learning a new feature representation or kernel, the distribution divergence of SVM in the original feature space is the largest, with the corresponding classification accuracy being the lowest. SKM can learn a new kernel to implicitly reduce the distribution divergence, thus it can achieve positive transfer. However, since SKM directly aligns the pre-computed kernels instead of learning a domain-invariant kernel, it cannot sufficiently reduce the distribution difference. TCA can explicitly reduce the distribution difference by minimizing the MMD divergence across domains. Thus it can achieve even better classification accuracy than SKM. However, the MMD-based matching can only match the first- and high-order statistics of the distributions, while the differences underlying the data variances from different domains are not properly matched. TKL can explicitly reduce the distribution divergence by minimizing the error between the source domain kernel and the target domain extrapolated kernel. Thus it can learn a data-dependent kernel which is invariant to the differences in both the means and variances of the distributions. This guarantees TKL to achieve the best performance.

5.6 Parameter Sensitivity

TKL is an elegant approach with only two tunable parameters: eigenspectrum damping factor ζ and SVM trade-off parameter C. We conduct extensive analysis for parameter sensitivity on all the text, image, and video datasets. The average classification accuracy is computed on each of the four dataset types: *20NG* (the 216 text datasets from 20-Newsgroups), *RT* (the 6 text datasets from Reuters-21578), *Image* (the 12 visual object datasets), and *Video* (the video event dataset). The detailed results are listed in Figs 5c and 5d, with the best baseline results shown as dashed lines.

We run TKL with varying values of $\zeta \in [1, 5]$ by fixing C. Since the eigenspectrum of a PSD kernel matrix usually follows the power-law distribution, $\zeta > 1$ can be naturally introduced to control the decaying trend of the eigenspectrum. In principle, the configuration of ζ can be guided by the decaying trend of the input kernel. We plot the average accuracy w.r.t. different values of ζ in Fig. 5c, which indicates $\zeta \in [1.2, 5.0]$ for the textual datasets, and $\zeta \in [1.1, 1.4]$ for the visual datasets. Note that, the decaying trend of the visual datasets is slower than that of the textural datasets, so the optimal range ζ for the visual datasets is narrower.

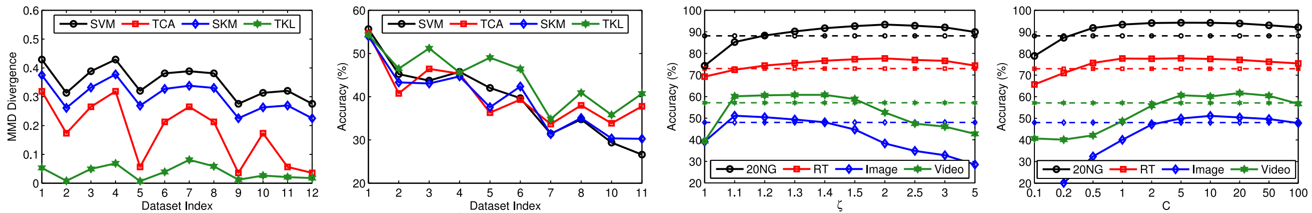
We run TKL with varying values of $C \in [0.1, 100]$ by fixing ζ. In practice, TKL is less sensitive to C than to ζ, so we can determine C first. We plot the average accuracy w.r.t. different values of C in Fig. 5d, which indicates wide ranges of $C \in [0.5, 100]$ for the textual datasets, and $C \in [2, 50]$ for the visual datasets.

5.7 Time Complexity

We empirically investigate the time complexity of all algorithms by running them on *comp vs rec 1* dataset with about 25,800 features and 8,000 documents. The results in Table 7 show that TKL is generally much more efficient than the comparison methods. This establishes the value of TKL for real-world applications.

6 Conclusion and Future Work

In this paper, we propose a transfer kernel learning approach to learn a domain-invariant kernel by directly matching the source and target distributions in the reproducing kernel Hilbert space. The domain-invariant kernel is



(a) MMD Divergence (b) Classification Accuracy (c) Damping Parameter ζ (d) SVM Parameter C

Fig. 5. Analysis for TKL: 1) effectiveness verification on image datasets, 2) parameter sensitivity on all datasets.

TABLE 7
Time Complexity of TKL and the Comparison Methods

Method	Runtime (s)	Method	Runtime (s)	Method	Runtime (s)
SVM	6.79	CDSC	25.37	SFA	20.82
KMM	705.69	TCA	126.79	DTMKL	293.90
GFK	189.93	SKM	147.23	TKL	28.31

learned from a family of spectral kernels extrapolated by the target eigenspace, and the one minimizing the Nyström approximation error is selected to train domain-invariant kernel machines. Our approach consistently outperforms competing algorithms on a variety of benchmark datasets.

In the future, we plan to explore alternative eigenspectrum damping constraints, so that non-power-law distributed eigenspectrum can be learned accurately.

Acknowledgments

This work was supported by National HeGaoJi Key Project (No. 2010ZX01042-002-002-01), National Science Fund for Distinguished Young Scholars (No. F020204), and Tsinghua National Laboratory Fund for Big Data Science and Technology. Philip S. Yu is supported in part by the US National Science Foundation (NSF) through grants OISE-1129076, CNS-1115234, DBI-0960443, and US Department of Army through grant W911NF-12-1-0066. Mingsheng Long and Jianmin Wang are the corresponding authors.

References

[1] V. Vapnik, *Statistical Learning Theory*. Hoboken, NJ, USA: Wiley, 1998.

[2] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, pp. 1345–1359, Oct. 2010.

[3] F. Perronnin, J. Sánchez, and Y. Liu, "Large-scale image categorization with explicit data embedding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 2297–2304.

[4] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2066–2073.

[5] L. Duan, I. Tsang, D. Xu, and S. Maybank, "Domain transfer SVM for video concept detection," in *Proc. IEEE 22nd Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1375–1381.

[6] L. Duan, D. Xu, I. W.-H. Tsang, and J. Luo, "Visual event recognition in videos by learning from web data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1667–1680, Sep. 2012.

[7] J. Blitzer, R. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2006, pp. 120–128.

[8] S. J. Pan, X. Ni, J.-T. Sun, Q. Yang, and Z. Chen, "Cross-domain sentiment classification via spectral feature alignment," in *Proc. 19th Int. Conf. World Wide Web*, 2010, pp. 751–760.

[9] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Scholkopf, and A. J. Smola, "A kernel two-sample test," *J. Mach. Learn. Res.*, vol. 13, pp. 723–773, 2012.

[10] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Scholkopf, "Correcting sample selection bias by unlabeled data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 601–608.

[11] S. Si, D. Tao, and B. Geng, "Bregman divergence-based regularization for transfer subspace learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 7, pp. 929–942, Jul. 2010.

[12] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.

[13] X. Cao, D. Wipf, F. Wen, and G. Duan, "A practical transfer learning algorithm for face verification," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 3208–3215.

[14] S. J. Pan, J. T. Kwok, and Q. Yang, "Transfer learning via dimensionality reduction," in *Proc. 22nd Nat. Conf. Artif. Intell.*, 2008, pp. 677–682.

[15] L. Duan, I. W. Tsang, and D. Xu, "Domain transfer multiple kernel learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 465–479, Mar. 2012.

[16] K. Zhang, V. W. Zheng, Q. Wang, J. T. Kwok, Q. Yang, and I. Marsic, "Covariate shift in hilbert space: A solution via surrogate kernels," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 388–395.

[17] K. Zhang, I. W. Tsang, and J. T. Kwok, "Improved nyström low-rank approximation and error analysis," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 1232–1239.

[18] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.

[19] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Proc. 14th Int. Conf. Comput. Vis.*, 2013, pp. 2200–2207.

[20] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York, NY, USA: Springer, 2009.

[21] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, "Deep canonical correlation analysis," in *Proc. 30th Int. Conf. Mach. Lang.*, 2013, pp. 1247–1255.

[22] J. Blitzer, "Domain adaptation of natural language processing systems," Ph.D. dissertation, Dept. Comput. Inf. Sci., Univ. Pennsylvania, Philadelphia, PA, USA, 2008.

[23] P. Yang and W. Gao, "Multi-view discriminant transfer learning," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 1848–1854.

[24] B. Zhang and Z.-Z. Shi, "Classification of big velocity data via cross-domain canonical correlation analysis," in *Proc. IEEE Int. Conf. BigData*, 2013, pp. 493–498.

[25] R. Johnson and T. Zhang, "Graph-based semi-supervised learning and spectral kernel design," *IEEE Trans. Inf. Theory*, vol. 54, no. 1, pp. 275–288, Jan. 2008.

[26] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet, "Simplemkl," *J. Mach. Learn. Res.*, vol. 9, pp. 2491–2521, 2008.

[27] J. Zhuang, I. W. Tsang, and S. C. Hoi, "A family of simple non-parametric kernel learning algorithms," *J. Mach. Learn. Res.*, vol. 12, pp. 1313–1347, 2011.

[28] C. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 682–688.

[29] X. Zhu, J. Kandola, Z. Ghahramani, and J. Lafferty, "Nonparametric transforms of graph kernels for semi-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 1641–1648.

[30] S. C. H. Hoi, M. R. Lyu, and E. Y. Chang, "Learning the unified kernel machines for classification," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 187–196.

[31] Q. Mao and I. W. Tsang, "Parameter-free spectral kernel learning," in *Proc. 26th Conf. Uncertainty Artif. Intell.*, 2010, pp. 350–357.

[32] R. Jin, T. Yang, M. Mahdavi, Y.-F. Li, and Z.-H. Zhou, "Improved bounds for the Nyström method with application to kernel classification," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6939–6949, Oct. 2013.

[33] R. Chattopadhyay, Q. Sun, W. Fan, I. Davidson, S. Panchanathan, and J. Ye, "Multisource domain adaptation and its application to early detection of fatigue," *ACM Trans. Knowl. Discovery Data*, vol. 6, no. 4, pp. 18:1–18:26, Dec. 2012.

[34] L. Duan, D. Xu, and I. W. Tsang, "Domain adaptation from multiple sources: A domain-dependent regularization approach," *IEEE Trans. Neural Netw.*, vol. 23, no. 3, pp. 504–518, Mar. 2012.

[35] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu, "Co-clustering based classification for out-of-domain documents," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2007, pp. 210–219.

[36] F. Zhuang, P. Luo, Z. Shen, Q. He, Y. Xiong, Z. Shi, and H. Xiong, "Mining distinction and commonality across multiple domains using generative model for text classification," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 11, pp. 2025–2039, Jun. 2011.

[37] M. Long, J. Wang, G. Ding, S. J. Pan, and P. S. Yu, "Adaptation regularization: A general framework for transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 5, pp. 1076–1089, May 2014.

[38] B. Chen, W. Lam, I. W. Tsang, and T.-L. Wong, "Discovering low-rank shared concept space for adapting text mining models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1284–1297, Jun. 2013.

[39] J. Gao, W. Fan, J. Jiang, and J. Han, "Knowledge transfer via multiple model local structure mapping," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2008, pp. 283–291.

[40] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, pp. 213–226.

[41] W. Li, L. Duan, D. Xu, and I. W. Tsang, "Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 6, pp. 1134–1148, Aug. 2013.

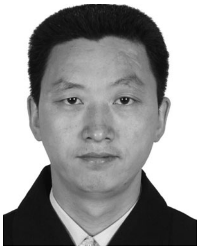
[42] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, 2006.

[43] X. Ling, W. Dai, G.-R. Xue, Q. Yang, and Y. Yu, "Spectral domain-transfer learning," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2008, pp. 488–496.

[44] H. Daumé III, "Frustratingly easy domain adaptation," in *Proc. 45th Annu. Meeting Assoc. Comput. Linguistics*, 2007, pp. 256–263.

[45] J. Yang, R. Yan, and A. G. Hauptmann, "Cross-domain video concept detection using adaptive SVMs," in *Proc. 15th Int. Conf. Multimedia*, 2007, pp. 188–197.

Mingsheng Long graduated from Tsinghua University, China, in 2014, where he received the BE degree in electrical engineering and the PhD degree in computer science. He is currently a postdoctoral researcher in the School of Software, Tsinghua University. He is concurrently a visiting scholar with AMPLab, UC Berkeley. His research interests include machine learning, data mining, and databases.



Jianmin Wang graduated from Peking University, China, in 1990, and received the ME and PhD degrees in computer software from Tsinghua University, China, in 1992 and 1995, respectively. He is a full professor in the School of Software, Tsinghua University. His research interests include unstructured big data management, workflow and BPM technology, and large-scale data analytics. He has published 100 papers in major journals and conferences, including *TKDE*, *DMKD*, *WWWJ*, SIGMOD, VLDB, ICDE, SIGKDD, SIGIR, AAAI, CVPR, and ICCV. He led to develop a product data/lifecycle management system, which has been implemented in hundreds of enterprises in China. He leads to develop an unstructured big data management system, LaUDMS.

Jiaguang Sun received the BS degree in automation science from Tsinghua University, Beijing, China, in 1970. He is a professor with Tsinghua University, where he is also the director of the School of Information Science and Technology and of the Tsinghua National Laboratory for Information Science and Technology. He is dedicated in teaching and R&D activities in computer graphics, computer-aided design, formal verification of software, and database systems. He has been an academician of the Chinese Academy of Engineering since 1999.

Philip S. Yu received the PhD degree in EE from Stanford University. He is a distinguished professor in computer science at the University of Illinois at Chicago and is the Wexler chair in Information Technology. His research interest is on big data, including data mining, data stream, database and privacy. He is the editor-in-chief of the *ACM Transactions on Knowledge Discovery from Data*. He was the editor-in-chief of the *IEEE Transactions on Knowledge and Data Engineering* (2001-2004). He received a Research Contributions Award from IEEE International Conference on Data Mining (2003) and a Technical Achievement Award from IEEE Computer Society (2013). He is a fellow of the IEEE.