Aligning Infinite-Dimensional Covariance Matrices in Reproducing Kernel Hilbert Spaces for Domain Adaptation

Zhen Zhang, Mianzhi Wang, Yan Huang, Arye Nehorai
Washington University in St. Louis
{zhen.zhang, mianzhi.wang, yanhuang640, nehorai}@wustl.edu

Abstract

Domain shift, which occurs when there is a mismatch between the distributions of training (source) and testing (target) datasets, usually results in poor performance of the trained model on the target domain. Existing algorithms typically solve this issue by reducing the distribution discrepancy in the input spaces. However, for kernel-based learning machines, performance highly depends on the statistical properties of data in reproducing kernel Hilbert spaces (RKHS). Motivated by these considerations, we propose a novel strategy for matching distributions in RKHS, which is done by aligning the RKHS covariance matrices (descriptors) across domains. This strategy is a generalization of the correlation alignment problem in Euclidean spaces to (potentially) infinite-dimensional feature spaces. In this paper, we provide two alignment approaches, for both of which we obtain closed-form expressions via kernel matrices. Furthermore, our approaches are scalable to large datasets since they can naturally handle out-of-sample instances. We conduct extensive experiments (248 domain adaptation tasks) to evaluate our approaches. Experiment results show that our approaches outperform other state-of-the-art methods in both accuracy and computationally efficiency.

1. Introduction

Standard supervised learning algorithms rely on the assumption that the training data and the testing data are drawn from an identical distribution. The validity of this assumption guarantees that the trained model can generalize well to the testing set. However, in real world applications, data are sensed by various types of acquisition devices and in different situations. For example, in computer vision tasks, images may be taken by cameras with different resolutions or under different light conditions. In such cases, a distribution mismatch or domain shift usually occurs, and consequently traditional statistical learning methods tend to perform poorly. Therefore, how to handle the statistical heterogeneity among data becomes a fundamental problem, called the domain adaptation problem.

A domain adaptation problem usually involves two domains: the source domain and the target domain. The source domain is composed of labeled data $\{\boldsymbol{X}^s, \vec{\boldsymbol{l}}_X\} = \{(x_i, l_{x_i})\}_{i=1}^{N_s}$, which can be used to train a reliable classifier. The target domain is composed of unlabeled data $\boldsymbol{Y}^t = \{y_j\}_{j=1}^{N_t}$, whose statistical properties are different. The main objective is to adapt the model (*e.g.*, classifier) trained on the source domain to the target domain.

Many works have considered this problem. One class of algorithms [4, 14, 26] reduces the distribution discrepancies across domains by pointwise re-weighting. Another widely investigated paradigm finds domain-invariant feature representations. Typical algorithms include domain invariant projection (DIP) [1], transfer component analysis (TCA) [23], and joint distribution alignment (JDA) [20]. However, sometimes, the statistical distributions across domains are very different, and even their supports are significantly mismatched. In such cases, it is difficult to find suitable weights for matching, or to identify domain-invariant features. More recently, to solve the above issue, another line of algorithms, which consider "moving" the source data to the target domain so as to make their distributions closer, has been proposed. In [7], Courty *et al.* borrowed a concept from optimal transport theory [28], making use of the optimal transport plan (map) to "transport" source data. But their method suffers from a drawback: It can be applied only in transductive settings. That is, when new data are available, one need to recompute a new optimization problem. In [27], Sun *et al.* used a linear map to transform source data to align covariance matrices across domains. However, this method considers only the linear correlation of data, which limits its applications on datasets with complex nonlinear correlation structures.

All the above works attempt to tackle domain shift in the input spaces, which is probably not optimal for kernel-based learning machines. Moreover, there exist various data representations, such as strings [19], graphs [18], lattices [5], manifolds [30], and proteins [3], which are not represented as vectors in Euclidean spaces. Instead they can be well characterized by kernel functions. Hence, for such datasets, it is not straightforward (and sometimes not even possible) to apply the above algorithms.

Motivated by all these considerations, we propose a novel and conceptually intuitive framework for domain adaptation, which is done by aligning infinite-dimensional covariance matrices (descriptors) across domains. More specifically, we first map the original features to a RKHS, and then use a linear operator in the result space to "move" the source data to the target domain such that the RKHS covariance descriptors of the transformed data and target data are close. Computing the pairwise inner product with the transformed and target samples, we obtain a new domain-invariant kernel matrix with the closed-form expression, which can be used in any kernel-based learning machine. In this paper, we provide two types of linear transformations (operators) in RKHS: the kernel whitening-coloring map and the kernel optimal transport map, each of which corresponds to a new domain-invariant kernel.

As we will show, our approaches have several advantages: (1) They support various data representations, as long as the kernel functions are well-defined. (2) They can naturally handle out-of-sample patterns. (3) They can align distributions with large shifts. (4) Exploiting the principal eigenstructures of RKHS covariance descriptors, our approaches are computationally efficient.

Organization. In Section 2, we describe the correlation alignment problem in Euclidean spaces $\mathbb{R}^n$, and introduce two solutions. In Section 3, we discuss methods of empirically estimating covariance descriptors in RKHS. Section 4 and Section 5 form the core of our paper, where we generalize correlation alignment to infinite dimensional settings, and develop domain-invariant kernel machines. In Section 6, we report our experimental results on cross-domain visual object recognition and document classification. In the supplementary material, we provide proofs of all mathematical results in the paper and more discussion on the experimental results.

2. Correlation alignment in $\mathbb{R}^n$

Given two positive definite covariance matrices $\boldsymbol{\Sigma}_s$ and $\boldsymbol{\Sigma}_t \in \mathbb{R}^{n\times n}$ obtained from source and target samples respectively, we aim at finding a linear transformation $\boldsymbol{T} : \mathbb{R}^n \to \mathbb{R}^n$, such that

$$\boldsymbol{T}\boldsymbol{\Sigma}_s\boldsymbol{T}^T = \boldsymbol{\Sigma}_t. \qquad (1)$$

Eq (1) corresponds to matching two centered Gaussian distributions. That is, given a random vector $\vec{\boldsymbol{x}} \sim N(\vec{\boldsymbol{0}}, \boldsymbol{\Sigma}_s)$, we need to ensure that after transformation, the new random vector $\vec{\boldsymbol{y}} = \boldsymbol{T}\vec{\boldsymbol{x}}$ follows the distribution $N(\vec{\boldsymbol{0}}, \boldsymbol{\Sigma}_t)$.

There may exist many solutions of (1). In our paper, we investigate two typical ones and generalize them to infinite-dimensional settings. One solution is the so-called "frustratingly easy" whitening-coloring map [27], and the other one is the optimal transport map between Gaussian distributions [10]. In the next two subsections, we discuss the mechanism of these two solutions in aligning correlations. Furthermore, we consider the rank-deficient case (*i.e.*, $\boldsymbol{\Sigma}_s$ and $\boldsymbol{\Sigma}_t$ are not invertible), providing the modified solutions and the corresponding sufficient conditions under which "complete matching", characterized by (1), still holds.

2.1. Whitening-coloring map

Write (1) as $(\boldsymbol{T}\boldsymbol{\Sigma}_s^{\frac{1}{2}})(\boldsymbol{T}\boldsymbol{\Sigma}_s^{\frac{1}{2}})^T = (\boldsymbol{\Sigma}_t^{\frac{1}{2}})(\boldsymbol{\Sigma}_t^{\frac{1}{2}})^T$, and set $\boldsymbol{T}\boldsymbol{\Sigma}_s^{\frac{1}{2}} = \boldsymbol{\Sigma}_t^{\frac{1}{2}}$. Then we can obtain an immediate solution

$$\boldsymbol{T}_{\mathrm{WC}} = \boldsymbol{\Sigma}_t^{\frac{1}{2}}\boldsymbol{\Sigma}_s^{-\frac{1}{2}}. \qquad (2)$$

The solution $\boldsymbol{T}_{\mathrm{WC}}$ can be explained in a two-step procedure: The source samples are first whitened by $\boldsymbol{\Sigma}_s^{-\frac{1}{2}}$, and then recolored by $\boldsymbol{\Sigma}_t^{\frac{1}{2}}$.

However, in practice, the estimated covariance matrices are usually not invertible, because the dimension of features may be larger than the sample numbers, and samples may concentrate on just a low-dimensional subspace. As a result, the solution $\boldsymbol{T}_{\mathrm{WC}}$ is ill-defined. We consider the ad-hoc modification, *i.e.*,

$$\hat{\boldsymbol{T}}_{\mathrm{WC}} = \boldsymbol{\Sigma}_t^{\frac{1}{2}}(\boldsymbol{\Sigma}_s^{\frac{1}{2}})^{\dagger}, \qquad (3)$$

where "†" denotes the Moore-Penrose pseudoinverse. Different from the full-rank situation, the validity of (1) under the transformation $\hat{\boldsymbol{T}}_{\mathrm{WC}}$ depends on the eigenstructures of the source and target covariance matrices.

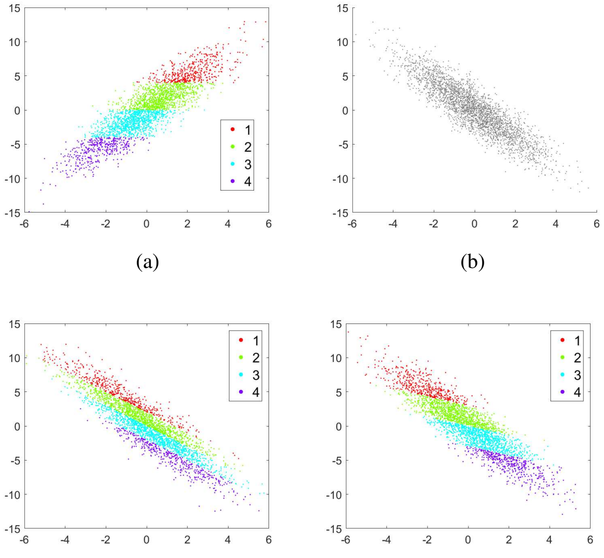
Theorem 1 *If* $\mathrm{Im}(\boldsymbol{\Sigma}_t) \subseteq \mathrm{Im}(\boldsymbol{\Sigma}_s)$, *then* $\hat{\boldsymbol{T}}_{\mathrm{WC}}\boldsymbol{\Sigma}_s\hat{\boldsymbol{T}}_{\mathrm{WC}}^T = \boldsymbol{\Sigma}_t$.

Remark 1 *Given any matrix* $\boldsymbol{A}$, $\mathrm{Im}(\boldsymbol{A})$ *denotes its range space, i.e.,* $\mathrm{Im}(\boldsymbol{A}) = \{\boldsymbol{A}\vec{\boldsymbol{x}}, \vec{\boldsymbol{x}} \in \mathbb{R}^n\}$. *The condition in the proposition requires that the source subspace where the source samples concentrate contains the target subspace.*

2.2. Optimal transport map

Given two distributions, μ_s and μ_t, there are infinitely many maps $\mathcal{T}$ (including nonlinear ones) that can transform μ_s to μ_t, denoted as $\mathcal{T}_{\#}\mu_s = \mu_t$. Monge's optimal transport problem [28] is to find the most efficient map, in the sense of minimizing the total transportation cost. The problem is formulated as

$$\begin{aligned} &\min_{\mathcal{T}} \int_{\mathbb{R}^n} c(\vec{\boldsymbol{x}}, \mathcal{T}(\vec{\boldsymbol{x}}))\mathrm{d}\mu_s \\ &\text{s.t.} \quad \mathcal{T}_{\#}\mu_s = \mu_t, \end{aligned} \qquad (4)$$



(a)

(b)

(c)

(d)

Figure 1: (a) The source samples $\boldsymbol{X}_s = [\vec{\boldsymbol{x}}_1, \vec{\boldsymbol{x}}_1, ..., \vec{\boldsymbol{x}}_N]$. Different colors represent different classes. (b) The target domain. (c) The results of transforming the source samples by the whitening-coloring map (*i.e.*, $\vec{\boldsymbol{x}}_i \to \boldsymbol{T}_{\mathrm{WC}}(\vec{\boldsymbol{x}}_i)$). (d) The results of transforming the source samples by the optimal transport map (*i.e.*, $\vec{\boldsymbol{x}}_i \to \boldsymbol{T}_{\mathrm{OT}}(\vec{\boldsymbol{x}}_i)$).

where $c(\vec{\boldsymbol{x}}, \vec{\boldsymbol{y}})$ defines the cost of moving unit mass from location $\vec{\boldsymbol{x}}$ to location $\vec{\boldsymbol{y}}$, and the cost is usually chosen as the squared distance function, *i.e.*, $c(\vec{\boldsymbol{x}}, \vec{\boldsymbol{y}}) = \|\vec{\boldsymbol{x}} - \vec{\boldsymbol{y}}\|_2^2$.

If μ_s and μ_t are two regular Gaussian distributions, i.e., $\mu_s = N(\vec{\mathbf{0}}, \boldsymbol{\Sigma}_s)$, $\mu_t = N(\vec{\mathbf{0}}, \boldsymbol{\Sigma}_t)$, and $\boldsymbol{\Sigma}_s$ and $\boldsymbol{\Sigma}_t$ are invertible, then the optimizer of (4) is a symmetric and linear transformation [10], denoted as $\boldsymbol{T}_{\mathrm{OT}}$, and its expression is

$$\boldsymbol{T}_{\mathrm{OT}} = \boldsymbol{\Sigma}_t^{\frac{1}{2}} (\boldsymbol{\Sigma}_t^{\frac{1}{2}} \boldsymbol{\Sigma}_s \boldsymbol{\Sigma}_t^{\frac{1}{2}})^{-\frac{1}{2}} \boldsymbol{\Sigma}_t^{\frac{1}{2}}. \tag{5}$$

Note that in the literature, $\boldsymbol{T}_{\mathrm{OT}}$ is often written as $\boldsymbol{T}_{\mathrm{OT}} = \boldsymbol{\Sigma}_s^{-\frac{1}{2}} (\boldsymbol{\Sigma}_s^{\frac{1}{2}} \boldsymbol{\Sigma}_t \boldsymbol{\Sigma}_s^{\frac{1}{2}})^{\frac{1}{2}} \boldsymbol{\Sigma}_s^{-\frac{1}{2}}$. In the supplementary material, we show the equivalence between these two expressions. As we mentioned before, the transformation between Gaussian distributions is equivalent to the alignment of covariance matrices. So the optimal transport map $\boldsymbol{T}_{\mathrm{OT}}$ is another solution of (1).

$\boldsymbol{T}_{\mathrm{OT}}$, developed from the optimal transport theory, attempts to avoid long-distance "transportation". Therefore, compared with the $\boldsymbol{T}_{\mathrm{WC}}$, $\boldsymbol{T}_{\mathrm{OT}}$ can avoid distorting the intrinsic structure of data. A toy example for comparing $\boldsymbol{T}_{\mathrm{WC}}$ and $\boldsymbol{T}_{\mathrm{OT}}$ is shown in Fig. 1.

Still, if $\boldsymbol{\Sigma}_s$ and $\boldsymbol{\Sigma}_t$ are non-invertible, we replace the matrix inverse "-" with the pseudoinverse "†", *i.e.*,

$$\hat{\boldsymbol{T}}_{\mathrm{OT}} = \boldsymbol{\Sigma}_t^{\frac{1}{2}} (\boldsymbol{\Sigma}_t^{\frac{1}{2}} \boldsymbol{\Sigma}_s \boldsymbol{\Sigma}_t^{\frac{1}{2}})^{\dagger\frac{1}{2}} \boldsymbol{\Sigma}_t^{\frac{1}{2}}. \tag{6}$$

In the next theorem, we provide the sufficient condition for "complete matching" (1) under the transformation $\hat{\boldsymbol{T}}_{\mathrm{OT}}$.

Theorem 2 *If* $\mathrm{Ker}(\boldsymbol{\Sigma}_s) \cap \mathrm{Im}(\Sigma_t) = \{\vec{\mathbf{0}}\}$, *then we have* $\hat{\boldsymbol{T}}_{\mathrm{OT}} \boldsymbol{\Sigma}_s \hat{\boldsymbol{T}}_{\mathrm{OT}}^T = \boldsymbol{\Sigma}_t$.

Remark 2 *Roughly speaking, because* $\mathrm{Ker}(\boldsymbol{\Sigma}_s) \perp \mathrm{Im}(\boldsymbol{\Sigma}_s)$*, the condition in above theorem implies that there should be substantial overlap between* $\mathrm{Im}(\boldsymbol{\Sigma}_s)$ *and* $\mathrm{Im}(\boldsymbol{\Sigma}_t)$*. In addition, this condition is milder than that in Theorem 1, since* $\mathrm{Im}(\boldsymbol{\Sigma}_t) \subseteq \mathrm{Im}(\boldsymbol{\Sigma}_s) \implies \mathrm{Ker}(\boldsymbol{\Sigma}_s) \cap \mathrm{Im}(\Sigma_t) = \{\vec{\mathbf{0}}\}$.

3. Covariance descriptor estimation in RKHS

In RKHS, covariance descriptors, which are (potentially) infinite dimensional, are the generalization of covariance matrices in $\mathbb{R}^n$. Let $\mathcal{X}$ be any nonempty set. Let k be a positive definite kernel on $\mathcal{X} \times \mathcal{X}$. Let $\mathcal{H}_{\mathcal{K}}$ be the reproducing kernel Hilbert space generated by k, and let $\phi : \mathcal{X} \to \mathcal{H}_{\mathcal{K}}$ be the corresponding feature map. Given the samples $\boldsymbol{X} = [x_1, x_2, ..., x_N]$, we introduce two covariance descriptor estimation methods in the following discussion.

3.1. Maximum likelihood estimation (MLE)

Let $\Phi_{\boldsymbol{X}} = [\phi(x_1), \phi(x_2), ..., \phi(x_N)]$ be the RKHS data matrix, then the MLE of the covariance descriptor is

$$MC = \Phi_{\boldsymbol{X}} \boldsymbol{J}_N \boldsymbol{J}_N^T \Phi_{\boldsymbol{X}}^T, \tag{7}$$

where $\boldsymbol{J}_N = \frac{1}{\sqrt{N}}(\boldsymbol{I}_N - \frac{1}{N}\vec{\mathbf{1}}_N \vec{\mathbf{1}}_N^T)$ is the centering matrix. The regularized version is

$$MC = \Phi_{\boldsymbol{X}} \boldsymbol{J}_N \boldsymbol{J}_N \Phi_{\boldsymbol{X}}^T + \rho I_{\mathcal{H}_{\mathcal{K}}}. \tag{8}$$

The form (7) is the natural analogy of the MLE of the covariance matrix for the Gaussian model in $\mathbb{R}^n$. This form has been widely applied in kernel ridge regression [17] and in covariance-descriptor based image classification [24].

3.2. Computationally efficient estimation (CEE)

We assume that the RKHS data $\Phi_{\boldsymbol{X}}$ are sampled from the factor analysis model:

$$\phi(x) = \mu + \mathcal{A}\vec{z} + \epsilon, \tag{9}$$

where $\vec{z}$ is a d-dimensional latent variable and $\vec{z} \sim N(\vec{\mathbf{0}}, \boldsymbol{I}_d)$, and $\epsilon \sim N(0_{\mathcal{H}_{\mathcal{K}}}, \rho I_{\mathcal{H}_{\mathcal{K}}})$. Then the covariance of $\phi(x)$ is $C = \mathcal{A}\mathcal{A}^T + \rho I_{\mathcal{H}_{\mathcal{K}}}$. As shown in [31], the estimation of $\mathcal{A}$ is $\Phi_X \boldsymbol{W}_{\boldsymbol{X}}$, where

$$\boldsymbol{W}_{\boldsymbol{X}} = \boldsymbol{J}_N \boldsymbol{V}_d (\boldsymbol{I}_d - \rho \boldsymbol{\Lambda}_d^{-1})^{\frac{1}{2}} \tag{10}$$

is an $N \times d$ matrix, and $\boldsymbol{V}_{\boldsymbol{d}}$ and $\boldsymbol{\Lambda}_d$ store the top d eigenpairs of $\boldsymbol{C}_{XX} = \boldsymbol{J}_N^T \boldsymbol{K}_{\boldsymbol{XX}} \boldsymbol{J}_N$. Now we can obtain a new estimated covariance descriptor:

$$EC = \Phi_{\boldsymbol{X}} \boldsymbol{W}_{\boldsymbol{X}} \boldsymbol{W}_{\boldsymbol{X}}^T \Phi_{\boldsymbol{X}}^T + \rho I_{\mathcal{H}_{\mathcal{K}}}. \tag{11}$$

In the subsequent computation, it will be seen that if the dimension d of the latent variable is small enough, *i.e.*, $d \ll N$, the total time complexity will be significantly reduced. Therefore, we say that $\mathrm{E}C$ is a computationally efficient estimator. More discussions on this estimator can be found in [16, 31].

4. Covariance descriptors alignment in RKHS

With estimated covariance descriptors in $\mathcal{H}_{\mathcal{K}}$, we can generalize correlation alignment to infinite dimensional settings. In this section, we provide computable expressions of the corresponding whitening-coloring map and the optimal transport map (termed the "kernel whitening-coloring map" and the "kernel optimal transport map", respectively) via kernel matrices. The derivation procedures are provided in the supplementary material.

Given the source samples $\boldsymbol{X}_s = [x_1, x_2, ..., x_{N_s}]$ and the target samples $\boldsymbol{Y}_t = [y_1, y_2, ..., y_{N_t}]$, let $\Phi_{\boldsymbol{X}} = [\phi(x_1), \phi(x_2), ..., \phi(x_{N_s})]$ and $\Phi_{\boldsymbol{Y}} = [\phi(y_1), \phi(y_2), ..., \phi(y_{N_t})]$ be the corresponding RKHS data matrices, and let $\boldsymbol{K}_{XX}$, $\boldsymbol{K}_{XY}$, and $\boldsymbol{K}_{YY}$ be the kernel matrices, which are respectively defined by $(\boldsymbol{K}_{XX})_{ij} = \langle \phi(x_i), \phi(x_j) \rangle$, $(\boldsymbol{K}_{XY})_{ij} = \langle \phi(x_i), \phi(y_j) \rangle$, and $(\boldsymbol{K}_{YY})_{ij} = \langle \phi(y_i), \phi(y_j) \rangle$. To satisfy the eigenstructure conditions proposed in Theorem 1 and 2, we make use of the regularized covariance descriptor for the source domain data, which corresponds to the artificial assumption that the source samples are dispersed in the whole Hilbert space. That is, for MLE, we have

$$\mathrm{M}C_s = (\Phi_X \boldsymbol{J}_{N_s})(\Phi_X \boldsymbol{J}_{N_s})^T + \rho I_{\mathcal{H}_{\mathcal{K}}} \tag{12a}$$

$$\mathrm{M}C_t = (\Phi_Y \boldsymbol{J}_{N_t})(\Phi_Y \boldsymbol{J}_{N_t})^T, \tag{12b}$$

and for CEE, we have

$$\mathrm{E}C_s = (\Phi_X \boldsymbol{W}_X)(\Phi_X \boldsymbol{W}_X)^T + \rho I_{\mathcal{H}_{\mathcal{K}}} \tag{13a}$$

$$\mathrm{E}C_t = (\Phi_Y \boldsymbol{W}_Y)(\Phi_Y \boldsymbol{W}_Y)^T. \tag{13b}$$

We can see that the expressions of MLE and CEE share the same structure, which leads to similar subsequent derivations. Therefore, for brevity, we use (12) for the kernel whitening-coloring map, and use (13) for the kernel optimal transport map.

4.1. Kernel whitening-coloring map

Proposition 1 *With the maximum likelihood estimators* (12)*, the kernel whitening-coloring map is given by*

$$\begin{aligned} \mathrm{k}\hat{T}_{\mathrm{WC}} =& (\mathrm{M}C_t)^{\frac{1}{2}} (\mathrm{M}C_s)^{\dagger \frac{1}{2}} \\ =& \Phi_Y \boldsymbol{J}_{N_t} \boldsymbol{C}_{YY}^{\dagger \frac{1}{2}} \left[\boldsymbol{C}_{YX} \boldsymbol{A} \boldsymbol{J}_{N_s} \Phi_X^T + \frac{1}{\sqrt{\rho}} \boldsymbol{J}_{N_t} \Phi_Y^T \right], \end{aligned} \tag{14}$$

where $\boldsymbol{C}_{YY} = \boldsymbol{J}_{N_t}^T \boldsymbol{K}_{YY} \boldsymbol{J}_{N_t}$ and $\boldsymbol{C}_{YX} = \boldsymbol{J}_{N_t}^T \boldsymbol{K}_{YX} \boldsymbol{J}_{N_s}$ are centered kernel matrices, and $\boldsymbol{A} = \sum_{k=1}^{r} \frac{1}{\lambda_k} (\frac{1}{\sqrt{\lambda_k + \rho}} - \frac{1}{\sqrt{\rho}}) \vec{\boldsymbol{v}}_k \vec{\boldsymbol{v}}_k^T$, and $\{\lambda_k, \vec{\boldsymbol{v}}_k\}_{k=1}^r$ are positive eigenpairs of $\boldsymbol{C}_{XX}$.

4.2. Kernel optimal transport map

Proposition 2 *With the computationally efficient estimators* (13)*, the kernel optimal transport map is given by*

$$\begin{aligned} \mathrm{k}\hat{T}_{\mathrm{OT}} &= (\mathrm{E}C_t)^{\frac{1}{2}} \left[(\mathrm{E}C_t)^{\frac{1}{2}} (\mathrm{E}C_s) (\mathrm{E}C_t)^{\frac{1}{2}} \right]^{\dagger \frac{1}{2}} (\mathrm{E}C_t)^{\frac{1}{2}} \\ &= \Phi_Y \boldsymbol{W}_Y \left[\boldsymbol{C}_{YX}^w \boldsymbol{C}_{XY}^w + \rho(\boldsymbol{\Lambda}_Y - \rho \boldsymbol{I}_d) \right]^{\dagger \frac{1}{2}} \boldsymbol{W}_Y^T \Phi_Y^T, \end{aligned} \tag{15}$$

where $\boldsymbol{C}_{YX}^w = \boldsymbol{W}_Y^T \boldsymbol{K}_{YX} \boldsymbol{W}_X$ and $\boldsymbol{C}_{XY}^w = (\boldsymbol{C}_{YX}^w)^T$, and $\boldsymbol{\Lambda}_Y$ is the diagonal matrix storing the top d eigenvalues of $\boldsymbol{C}_{YY}$.

Note that both (14) and (15) are computable expressions. Take $\mathrm{k}\hat{T}_{\mathrm{WC}} : \mathcal{H}_{\mathcal{K}} \to \mathcal{H}_{\mathcal{K}}$ for example, $\forall f \in \mathcal{H}_{\mathcal{K}}$, we can immediately obtain $\mathrm{k}\hat{T}_{\mathrm{WC}}(f)$, *i.e.*, $\mathrm{k}\hat{T}_{\mathrm{WC}}(f) = \sum_{i=1}^{N_t} b_i \phi(y_i)$, where b_i are entries of vector $\vec{\boldsymbol{b}} = \boldsymbol{J}_{N_t} \boldsymbol{C}_{YY}^{\dagger \frac{1}{2}} [\boldsymbol{C}_{YX} \boldsymbol{A} \boldsymbol{J}_{N_s} \vec{\boldsymbol{F}}_X + \frac{1}{\sqrt{\rho}} \boldsymbol{J}_{N_t} \vec{\boldsymbol{F}}_Y]$, and $\vec{\boldsymbol{F}}_X = [f(x_1), f(x_2), ..., f(x_{N_s})]^T$ and $\vec{\boldsymbol{F}}_Y = [f(y_1), f(y_2), ..., f(y_{N_t})]^T$ due to the reproducing property.

5. Algorithms

Now, based on the computable expressions (14) and (15), we can perform nonlinear correlation alignment in RKHS. The idea behind our algorithm is rather intuitive: We first "move" the centered source data $\Psi_s = \sqrt{N_s} \Phi_X \boldsymbol{J}_{N_s}$ to the target domain by $\mathrm{k}\hat{T}_{\triangle}$ ($\triangle = \mathrm{WC}$ or OT), *i.e.*,

$$\Psi_s \to \Psi_{s \to t} = \mathrm{k}\hat{T}_{\triangle}(\Psi_s), \tag{16}$$

which guarantees well-matching between the covariance descriptors of the transformed source samples, $\Psi_{s \to t}$, and the centered target samples, $\Psi_t = \sqrt{N_t} \Phi_Y \boldsymbol{J}_{N_t}$. Then we use the transformed source samples to train a classifier (model). We sketch the procedure in Fig. 2.

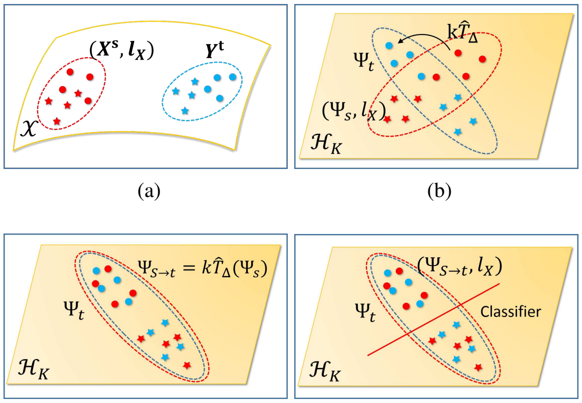
Similar to [29], we obtain a domain-invariant kernel matrix $\tilde{\boldsymbol{K}}$ by computing the pairwise inner product with transformed and target samples:

$$\tilde{\boldsymbol{K}} = \begin{bmatrix} \triangle\tilde{\boldsymbol{K}}_{ss} & \triangle\tilde{\boldsymbol{K}}_{ts}^T \\ \triangle\tilde{\boldsymbol{K}}_{ts} & \triangle\tilde{\boldsymbol{K}}_{tt} \end{bmatrix} = \begin{bmatrix} \Psi_{s \to t}^T \Psi_{s \to t} & \Psi_{s \to t}^T \Psi_t \\ \Psi_t^T \Psi_{s \to t} & \Psi_t^T \Psi_t \end{bmatrix}. \tag{17}$$

Using the kernel whitening-coloring map (14), we get[1]

$$\begin{aligned} \Psi_{s \to t} &= \mathrm{k}\hat{T}_{\mathrm{WC}}(\Psi_s) = \sqrt{N_s} \Phi_Y \boldsymbol{J}_{N_t} \boldsymbol{C}_{YY}^{\dagger \frac{1}{2}} \boldsymbol{B} \\ \mathrm{WC}\tilde{\boldsymbol{K}}_{ss} &= N_s \boldsymbol{B}^T \boldsymbol{B} \\ \mathrm{WC}\tilde{\boldsymbol{K}}_{ts} &= \sqrt{N_s N_t} \boldsymbol{C}_{YY}^{\frac{1}{2}} \boldsymbol{B} = \sqrt{N_s N_t} \boldsymbol{U}_Y \boldsymbol{\Lambda}_Y^{\frac{1}{2}} \boldsymbol{U}_Y^T \boldsymbol{B}, \end{aligned} \tag{18}$$

[1] We provide detailed derivation procedures in the supplementary material.



(a) (b)

(c) (d)

Figure 2: (a) The labeled dataset $(\boldsymbol{X}^s, \vec{\boldsymbol{l}}_X)$ in the source domain and the unlabeled dataset $\boldsymbol{Y}^t$ in the target domain. Dots and stars represent different classes. (b) The centered source data, Ψ_s, and centered target data, Ψ_t, in RKHS. (c) Transform the source samples by the map $\mathrm{k}\hat{T}_\triangle$ ($\triangle = \mathrm{WC}$ or OT), and the resultant data are $\Psi_{s\to t} = \mathrm{k}\hat{T}_\triangle(\Psi_s)$. (d) Train a classifier with $\Psi_{s\to t}$.

where $\boldsymbol{B} = \boldsymbol{C}_{YX}(\boldsymbol{C}_{XX} + \rho\boldsymbol{I}_{N_t})^{-\frac{1}{2}}$, and $(\boldsymbol{U}_Y, \Lambda_Y^{\frac{1}{2}})$ stores the top d eigenpairs of $\boldsymbol{C}_{YY}$. Note that, in practice, in order to exploit the principal components and reduce the computational complexity, we artificially select d to be a small integer, *i.e.*, $d \ll N_t$.

Using the kernel optimal transport map (15), we get

$$\begin{aligned} \Psi_{s\to t} &= \mathrm{k}\hat{T}_{\mathrm{OT}}(\Psi_s) = \sqrt{N_s}\Phi_Y\boldsymbol{W}_Y\boldsymbol{D} \\ \mathrm{OT}\tilde{\boldsymbol{K}}_{ss} &= N_s\boldsymbol{D}^T(\boldsymbol{\Lambda}_Y - \rho\boldsymbol{I}_d)\boldsymbol{D} \qquad (19) \\ \mathrm{OT}\tilde{\boldsymbol{K}}_{ts} &= \sqrt{N_sN_t}\boldsymbol{J}_{N_t}\boldsymbol{K}_{YY}\boldsymbol{W}_Y\boldsymbol{D}, \end{aligned}$$

where $\boldsymbol{D} = \left[\boldsymbol{C}^w_{YX}\boldsymbol{C}^w_{XY} + \rho(\boldsymbol{\Lambda}_Y - \rho\boldsymbol{I}_d)\right]^{\dagger\frac{1}{2}}\boldsymbol{W}_Y^T\boldsymbol{K}_{YX}\boldsymbol{J}_{N_s}$.

The kernel matrix $\triangle\tilde{\boldsymbol{K}}_{tt}$ of the centered target samples remains unchanged in both cases, *i.e.*,

$$\mathrm{WC}\tilde{\boldsymbol{K}}_{tt} = \mathrm{OT}\tilde{\boldsymbol{K}}_{tt} = \Psi_t^T\Psi_t = N_t\boldsymbol{C}_{YY}. \qquad (20)$$

5.1. Domain-invariant kernel machine

The new learned kernel (17) after (nonlinear) correlation alignment can be used in any kernel-based algorithm. For example, in kernel ridge regression, the predicted labels for the target dataset $\boldsymbol{Y}^t$ is

$$\vec{\boldsymbol{l}}_Y = (\triangle\tilde{\boldsymbol{K}}_{ts})(\triangle\tilde{\boldsymbol{K}}_{ss} + \gamma\boldsymbol{I}_{N_s})^{-1}\vec{\boldsymbol{l}}_X. \qquad (21)$$

For the kernel support vector machine, after training a classifier on the source partition $(\triangle\tilde{\boldsymbol{K}}_{ss}, \vec{\boldsymbol{l}}_X)$, we can predict labels of the target by

$$\vec{\boldsymbol{l}}_Y = (\triangle\tilde{\boldsymbol{K}}_{ts})(\vec{\boldsymbol{\alpha}} \odot \vec{\boldsymbol{l}}_X) + \vec{\boldsymbol{b}}, \qquad (22)$$

where $\vec{\boldsymbol{\alpha}}$ is the Lagrangian multiplier, $\odot$ is the Hadamard product, and $\vec{\boldsymbol{b}}$ is the bias.

5.2. Out-of-sample prediction

Our algorithms can naturally generalize to out-of-sample patterns. That is, when new target data $\bar{\boldsymbol{Y}}^t$ come, we can directly obtain the inner product matrix between the new centered samples, $\Psi_{\bar{Y}}$, and the transformed source samples, $\Psi_{s\to t}$, that already exist, instead of recomputing the total model.

5.3. Time complexity

For the case where we use $\mathrm{k}\hat{T}_{\mathrm{WC}}$ to "move" source data, it takes $O(N_s^3) + N_tN_s^2$ time to compute $\boldsymbol{B}$, $N_tN_s^2$ time to compute $\mathrm{WC}{-}\tilde{\boldsymbol{K}}_{ss}$, and $O(dN_t^2)$[2] $+ dN_t^2 + dN_tN_s$ time to compute $\mathrm{WC}{-}\tilde{\boldsymbol{K}}_{ts}$.

For the case where we use $\mathrm{k}\hat{T}_{\mathrm{OT}}$ to "move" source data, the total time complexity is $O(dN_s^2) + O(dN_s^2) + O(dN_sN_t) + O(d^3)$. Thanks to the "efficient" estimation of covariance descriptors that exploits only the principal eigenstructure, we can avoid large-scale matrix inversion and multiplication. Hence, although the expressions in (19) are complicated, the computational time complexity is low.

6. Experiments

In this section, we apply our approaches to two real-world problems: visual objects recognition and document classification. We first compare our approaches with other state-of-the-art domain adaptation algorithms in a transductive setting, which means that all target samples are used for estimating covariance descriptors and evaluating the trained model. We next conduct experiments to measure the ability of our approaches to deal with out-of-sample patterns.

6.1. Datasets

Four benchmark datasets, *i.e.*, COIL20, Office-Caltech, 20-Newsgroups, and Reuters-21578 are used in the experiments.

The **COIL20** [22] dataset contains $1,440$ grayscale images of 20 classes of objects. The images of each object were taken at a pose interval of 5 degrees. As a result, each object has $360^o/5^o = 72$ images. Each image is 32×32 pixels with 256 gray levels. We follow the procedure in [20] to construct two domains. That is, the whole dataset is partitioned into two subsects, COIL1 and COIL2. The images taken in the directions $[0^o, 85^o] \cup [180^o, 265^o]$ are contained in COIL1, and the images taken in the directions $[90^o, 175^o] \cup [270^o, 355^o]$ are contained in COIL2. The data distributions in COIL1 and COIL2 should be different. There are two domain adaptation tasks, *i.e.*, C1 → C2 and C2 → C1.

[2] Extracting top d eigenvectors of an $N_t \times N_t$ matrix requires $O(dN_t^2)$ time [25].

The **Office-Caltech** [15] dataset contains the images of ten classes of objects from four domains: 958 images downloaded from the **A**mazon website, $1,123$ images gathered from a web image search (**C**altech), 157 images taken with a **D**SLR camera, and 295 images from **W**ebcams. They form 12 domain adaptation tasks: A → C, A → D,...,W → D. We consider two types of features: the SURF features [2] and the DeCAF6 features [9]. The SURF features represent each image with an 800-bin normalized histogram, which is subsequently standardized by $z-$score. The DeCAF6 features represent each image with a 4,096-dimensional vector, which is extracted from the 6th layer of a convolutional neural network.

The **20-Newsgroups** [3] dataset has approximately $20,000$ documents, which are categorized in a hierarchical structure. The top categories are *comp*, *rec*, *sci*, and *talk*, each of which has four subcategories. Data drawn from different subcategories and under the same top categories are considered as different but related domains. The task is to predict the top category to which the sample belongs. Following the procedures in [8] and [21], we generate the cross-domain datasets. For each of the two top categories P and Q, we select two subcategories from each of them to form the source domain, and use the rest subcategories of P and Q as the target data. Therefore, given top categories P and Q, we can construct $C_4^2 C_4^2 = 36$ domain adaptation tasks, denoted as "P vs Q". There are $C_4^2 = 6$ possible combinations for top categories. In total, we have $6 \times 36 = 216$ tasks. For every 36 tasks, we report the average performance. More detailed description is given in [8] and [21]. We adopt the preprocessed version of 20-Newsgroups, which contains $15,033$ documents represented by $25,804$-dimensional features.

The **Reuters-21578**[4] dataset has three top categories, *i.e.*, *orgs*, *places* and *people*, each of which has many subcategories. Still, samples that belong to different subcategories are treated as ones drawn from different domains. Based on this, we can construct 6 cross-domain text datasets: *orgs vs people*, *people vs orgs*, *orgs vs places*, *places vs orgs*, *people vs places*, and *places vs people*. For every pair of datasets, *e.g.*, *orgs vs people*, *people vs orgs*, we report the average performance. We adopt the preprocessed version of Reuters-21578, which contains $3,461$ documents represented by $4,771$-dimensional features.

In summary, we have constructed $2+12\times 2+216+6 = 248$ domain adaptation tasks.

6.2. Transductive experiments setup

We first design experiments in the transductive setting. We employ the domain-invariant SVM describe in (22) to conduct classification. We use "KWC" and "KOT" to denote our approaches. We compare our approaches with many state-of-the-art algorithms[5]: (1) support vector machine without adaptation (SVM), (2) transfer component analysis (TCA) [23], (3) subspace alignment (SA) [12], (4) surrogate kernel matching (SKM) [29], (5) geodesic flow kernel (GFK) [13], (6) transfer kernel learning (TKL) [21], (7) correlation alignment (CORAL) [27], (8) transfer multiple kernel learning (TMKL) [11], and (9) joint distribution optimal transportation (JDOT) [6]. We use SVM as the final classifier for all above methods.

On the visual adaptation datasets, we use the radial basis function (RBF) kernel for all kernel-based algorithms, *i.e.*, SVM, TCA, SKM TKL, and TMKL. On the document datasets, we use both the linear kernel and the RBF kernel, and conduct comparisons in both settings. The width of the RBF kernel is set to be the mean value of the squared distances between all training samples.

All the methods mentioned above have hyperparameters. As in [7, 11, 21, 23, 29], we randomly select a small subset of target samples as validation sets to tune parameters and evaluate on the remaining target samples. Note that some algorithms (*e.g.*, GFK and SA) have heuristic methods to determine parameters and some (*e.g.*, TKL) have default parameters, but in some datasets, these parameters perform rather poorly. In order to fairly compare all the methods, we consider both the heuristic (or default) and manually selected parameters, and report the best results.

For algorithms requiring dimension d, we select d from $\{2, 4, 5, 10, 15, 20, ..., 50\}$. For algorithms requiring regularization parameter γ, we select γ from $\mathcal{R} = \{0.01, 0.1, 1, 5, 10, 50, 100\}$. For TKL, we search for the damping factor ζ in $\{1.1, 1.2, ..., 2\}$. For TMKL and SVM, we select the tradeoff parameter θ and C from $\mathcal{R}$. For our method KWC, we search ρ in $\frac{1}{Ns}\mathcal{R}$. For our method KOT, since $(\boldsymbol{I}_d - \rho\boldsymbol{\Lambda}_d^{-1})$ (see (10)) should be positive definite, we first write $\rho = \lambda_{\min}\gamma$, and then search γ in $\{0.01, 0.1, 0.2, ..., 0.9\}$, where $\lambda_{\min} = \mathrm{Min}\{\lambda_d^X, \lambda_d^Y\}$, and λ_d^X and λ_d^Y are the dth eigenvalues of C_{XX} and C_{YY}, respectively.

6.3. Experimental results

The experimental results on these four datasets are reported in Table 1, 2, 3, 4, and 5. The best results are highlighted in bold. For almost all the tasks, our approaches KWC and KOT significantly outperform the standard SVM classifier. Especially on the document datasets, *i.e.*, 20-Newsgroups and Reuters-21578, the average performance improvements are more than 10 percent, which demonstrates the power of aligning RKHS covariance descriptors in tackling the domain shift issue. Compared with other state-of-the-art algorithms, our approaches achieve superior or comparable results on all the object recognition and document classification tasks. For some cross-domain datasets,

[3] http://qwone.com/ jason/20Newsgroups/

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

[5] We use the codes published by the corresponding authors.

Table 1: Recognition accuracy (in %) on the COIL20 dataset.

Task	SVM	TCA	GFK	SKM	SA	CORAL	TKL	KWC	KOT
C1 → C2	87.50	82.64	83.75	87.50	84.86	82.22	86.25	88.33	**90.42**
C2 → C1	86.81	87.50	85.69	90.00	87.22	81.67	88.89	89.03	**91.11**
Mean	87.16	85.07	84.72	88.75	86.04	81.95	87.57	88.68	**90.77**

Table 2: Recognition accuracy (in %) on the Office-Caltech dataset with SURF features.

Task	SVM	TCA	GFK	SKM	SA	CORAL	DTMKL	KWC	KOT
A → C	43.90	43.18	41.49	42.48	41.23	45.1	45.01	**45.95**	44.88
A → D	45.22	38.85	36.30	35.03	36.91	39.5	40.85	**47.77**	43.95
A → W	38.30	41.69	32.20	40.00	40.27	**44.4**	36.94	41.69	43.73
C → A	53.34	55.43	**55.63**	52.19	50.61	52.1	54.33	53.24	52.92
C → D	48.40	47.13	48.81	47.13	44.68	45.9	44.74	49.68	**52.23**
C → W	44.75	44.40	42.68	43.39	41.41	46.4	42.04	**47.80**	45.76
D → A	30.17	39.14	40.29	37.58	36.43	37.7	34.03	**41.65**	38.94
D → C	29.56	34.73	35.00	34.11	35.35	33.8	32.10	**39.36**	38.02
D → W	62.37	83.05	80.68	83.73	85.42	84.7	81.69	**85.76**	**85.76**
W → A	32.15	38.41	36.64	37.47	36.63	36.0	36.53	**39.04**	36.85
W → C	25.65	33.83	28.85	29.83	33.25	33.7	32.50	**36.42**	34.02
W → D	84.71	81.53	80.25	84.71	81.34	86.6	**88.85**	82.80	84.71
Mean	44.88	48.45	46.57	47.30	46.96	48.8	47.47	**50.93**	50.15

like D → C in Table 2, Comp vs Sci and Rec vs Sci in Table 4, and People vs Places in Table 5, both KWC and KOT largely outperform the best competitive methods. A possible explanation is that our strategy of "moving" the source samples to the target domain allows us to align distributions with large shifts. Note that although CORAL uses a similar strategy, it does not consider high-order (or nonlinear) correlations. As a result, the performances of CORAL on COIL20 (see Table 1) and Office-Caltech with the DeCAF6 features (see Table 3) are less competitive.

6.4. Out-of-sample generalization

In this subsection, we measure our approaches' ability to generalize out-of-sample patterns. We conduct experiments on the office-caltech dataset with SURF and DeCAF6 features. To train the model, we randomly select half labeled samples from the source domain and half unlabeled samples from the target domain. We test the model on the remaining unlabeled samples in the target domain. We repeat the above procedures 500 times, and report the average accuracies and standard errors. We compare our approaches with SVM, TCA and GFK. In Table 6 and 7, the experimental results show that our approaches KWC and KOT outperform the baseline methods with statistical significance.

6.5. Empirical time complexity

In this subsection, we empirically compare the computational time of our approaches with other algorithms. We implement all the algorithms using the Matlab on an Intel i7-5500U, 2.40 GHz CPU. We test them on the Comp vs Rec dataset of 20-Newsgroups, which contains 36 cross-domain adaptation tasks. For every task, both the source and the target domain have approximately $4,000$ samples of dimension $25,804$. For fair comparison, we set the same dimension $d = 5$ for TCA, KWC, and KOT. We report the average running times in Table 8. It can be seen that SKM is most expensive, which may be due to the fact that it does not consider the low-rank approximations. TCA is relatively time-consuming. Our approach KOT is extremely efficient, which demonstrates our theoretical analysis of its time complexity in the Section 5.3.

7. Conclusion and future work

In this paper, we presented a mathematical and computational framework for domain adaptation by aligning infinite-dimensional covariance matrices in RKHS. We proposed two alignment strategies: the kernel whitening-coloring map and the kernel optimal transport map, and derived their closed-form expressions via kernel matrices. We further obtained two domain-invariant kernel matrices that can be used in any kernel-based algorithm. Empirically, we applied our framework to numerous domain adaptation tasks, and achieved promising results on both visual and document datasets. Moreover, our approaches possess out-of-sample generalizability and computational efficiency, which enable it scale to large datasets.

In the future, we plan to take the geometry of RKHS data into account, expecting to get further performance improvement by jointly aligning statistical distributions and geometric structures of data across domains.

References

[1] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann. Unsupervised domain adaptation by domain invariant projection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 769–776, 2013.

Table 3: Recognition accuracy (in %) on the Office-Caltech dataset with DeCAF6 features.

Task	SVM	TCA	GFK	SKM	SA	CORAL	TKL	JDOT	KWC	KOT
A → C	83.44	83.79	84.59	83.62	85.13	83.70	86.11	85.22	**87.27**	86.02
A → D	84.71	84.71	84.71	84.71	**89.17**	84.71	87.26	87.90	87.26	87.26
A → W	74.92	81.69	85.08	75.59	**85.42**	75.59	84.75	84.75	84.75	85.08
C → A	91.54	93.21	92.17	92.17	90.50	91.54	92.69	91.54	**93.53**	91.69
C → D	87.90	88.54	**92.99**	89.81	85.99	87.90	91.08	89.81	91.08	91.08
C → W	80.00	86.44	87.12	80.34	81.02	80.68	83.39	**88.81**	86.44	**88.81**
D → A	79.65	90.92	88.52	86.85	83.19	84.24	88.93	88.10	89.77	**90.40**
D → C	66.25	78.81	74.09	76.14	81.75	72.04	80.05	84.33	**85.22**	83.53
D → W	98.31	97.29	98.31	98.31	88.47	**98.98**	90.85	96.61	97.29	97.29
W → A	72.65	84.86	86.74	81.94	81.42	72.44	89.56	**90.71**	89.04	89.35
W → C	64.92	76.31	80.68	75.51	75.78	67.76	80.68	82.64	**83.17**	82.46
W → D	**100**	99.36	99.36	99.36	94.90	**100**	**100**	98.09	99.36	99.36
Mean	82.02	87.16	87.86	85.36	85.23	83.30	87.95	89.04	**89.52**	89.36

Table 4: Recognition accuracy (in %) on the 20-Newsgroups dataset.

Task	Linear kernel					RBF kernel				
	SVM	TCA	SKM	KWC	KOT	SVM	TCA	SKM	KWC	KOT
Comp vs Rec	87.51	95.12	92.03	96.58	**96.75**	87.65	94.99	91.77	96.67	**96.82**
Comp vs Sci	75.38	77.95	78.03	**88.78**	88.67	75.09	82.64	79.11	**88.95**	88.72
Comp vs Talk	95.44	97.18	**97.87**	96.80	97.20	95.45	95.96	97.12	97.07	**97.13**
Rec vs Sci	73.82	83.10	77.34	**92.40**	92.23	74.28	83.70	74.21	**92.41**	92.39
Rec vs Talk	83.27	93.35	86.69	**94.77**	94.59	83.18	93.10	88.56	94.68	**94.87**
Sci vs Talk	76.85	79.95	79.21	**84.54**	82.80	76.98	80.53	78.31	**83.74**	82.91
Mean	82.05	87.78	85.20	**92.31**	92.04	82.11	88.49	84.85	**92.25**	92.14

Table 5: Recognition accuracy (in %) on the Reuters-21578 dataset.

Task	Linear kernel					RBF kernel				
	SVM	TCA	TKL	KWC	KOT	SVM	TCA	TKL	KWC	KOT
Orgs vs People	79.44	82.87	83.76	83.59	**84.89**	79.00	83.94	84.71	84.82	**84.93**
Orgs vs Places	66.98	75.69	**80.85**	77.95	79.92	67.48	73.96	79.84	**79.89**	79.18
People vs Places	59.28	60.30	68.48	71.73	**72.51**	60.73	64.33	68.02	**74.66**	71.36
Mean	68.57	72.95	77.70	77.76	**79.11**	69.07	74.08	77.52	**79.79**	78.49

Table 6: Out-of-Sample prediction on the Office-Caltech dataset with SURF features.

Task	SVM	TCA	GFK	KWC	KOT
A → C	41.5(1.9)	38.6(2.7)	39.0(2.5)	**43.6(2.2)**	41.9(2.2)
A → D	40.0(4.8)	33.4(5.3)	36.4(5.1)	**40.0(5.0)**	36.4(5.2)
A → W	33.2(3.6)	36.3(4.3)	36.7(4.5)	**37.5(4.1)**	36.1(4.1)
C → A	49.5(2.3)	40.0(3.4)	50.0(2.3)	**51.7(2.6)**	50.6(2.7)
C → D	**46.5(4.9)**	41.1(6.6)	44.3(4.7)	45.1(5.2)	43.5(5.2)
C → W	40.9(3.9)	40.7(6.9)	42.7(4.7)	**44.1(4.5)**	**44.1(4.3)**
D → A	24.7(2.9)	31.5(3.3)	35.1(2.9)	35.7(2.8)	**35.9(3.1)**
D → C	24.6(2.6)	31.1(3.0)	31.1(2.4)	**34.8(2.5)**	34.2(2.6)
D → W	44.6(7.4)	64.1(6.1)	70.3(4.5)	**72.3(4.9)**	70.1(4.7)
W → A	28.0(3.0)	34.6(3.2)	33.8(2.8)	**36.5(2.4)**	35.4(2.9)
W → C	22.4(2.7)	32.0(2.5)	27.8(2.3)	**34.4(2.2)**	32.2(2.3)
W → D	72.3(5.0)	**74.6(4.9)**	74.3(4.6)	70.7(4.8)	69.7(5.2)
Mean	39.0	41.5	43.5	**45.5**	44.2

Table 7: Out-of-Sample prediction on the Office-Caltech dataset with DeCAF6 features.

Task	SVM	TCA	GFK	KWC	KOT
A → C	81.3(1.6)	81.5(1.5)	81.9(1.6)	**86.8(1.2)**	84.3(1.4)
A → D	81.8(3.4)	82.5(4.0)	84.6(3.4)	**86.2(3.3)**	85.5(3.4)
A → W	73.0(3.1)	75.3(3.2)	73.8(3.8)	84.4(2.9)	**84.6(3.0)**
C → A	90.7(0.9)	90.3(1.2)	89.4(1.7)	**92.8(0.9)**	91.5(1.2)
C → D	85.4(2.8)	81.1(4.6)	87.2(4.6)	**88.6(3.9)**	87.5(3.5)
C → W	78.5(2.6)	76.5(4.0)	80.8(3.9)	**83.8(3.3)**	82.0(3.4)
D → A	67.7(6.0)	83.7(4.2)	83.2(2.8)	87.7(2.8)	**88.0(2.5)**
D → C	57.7(6.3)	73.3(3.2)	71.8(3.1)	**81.8(2.4)**	78.8(2.3)
D → W	90.1(4.3)	92.8(3.0)	95.3(2.5)	**95.6(2.5)**	92.8(3.3)
W → A	69.8(5.2)	82.0(2.2)	81.6(2.3)	**86.8(2.9)**	86.0(2.6)
W → C	61.9(3.3)	73.6(3.9)	75.0(2.4)	80.6(2.0)	**80.7(2.0)**
W → D	98.5(1.3)	97.6(2.3)	**99.2(1.2)**	98.2(1.7)	97.6(2.0)
Mean	78.0	82.5	83.7	**87.8**	86.6

[2] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. *Computer vision–ECCV 2006*, pages 404–417, 2006.

[3] A. Ben-Hur and W. S. Noble. Kernel methods for predicting protein–protein interactions. *Bioinformatics*, 21(suppl_1):i38–i46, 2005.

[4] L. Bruzzone and M. Marconcini. Domain adaptation problems: A dasvm classification technique and a circular validation strategy. *IEEE transactions on pattern analysis and machine intelligence*, 32(5):770–787, 2010.

Table 8: Time complexity comparison on the Comp vs Rec dataset

Algorithms	SVM	TCA	SKM	KWC	KOT
Time(linear)	3.45s	93.39s	273.32s	23.76s	9.09s
Time(RBF)	7.31s	95.65s	291.65s	27.16s	12.21s

[5] C. Cortes, P. Haffner, and M. Mohri. Lattice kernels for spoken-dialog classification. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 1, pages I–628. IEEE, 2003.

[6] N. Courty, R. Flamary, A. Habrard, and A. Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. *arXiv preprint arXiv:1705.08848*, 2017.

[7] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy. Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1853–1865, 2017.

[8] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu. Co-clustering based classification for out-of-domain documents. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 210–219. ACM, 2007.

[9] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655, 2014.

[10] D. Dowson and B. Landau. The fréchet distance between multivariate normal distributions. *Journal of multivariate analysis*, 12(3):450–455, 1982.

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

[12] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the IEEE international conference on computer vision*, pages 2960–2967, 2013.

[13] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2066–2073. IEEE, 2012.

[14] A. Gretton, A. J. Smola, J. Huang, M. Schmittfull, K. M. Borgwardt, and B. Schölkopf. Covariate shift by kernel mean matching. 2009.

[15] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. 2007.

[16] M. Harandi, M. Salzmann, and F. Porikli. Bregman divergences for infinite dimensional covariance matrices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1003–1010, 2014.

[17] Z. John Lu. The elements of statistical learning: data mining, inference, and prediction. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 173(3):693–694, 2010.

[18] R. I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *ICML*, volume 2, pages 315–322, 2002.

[19] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. Text classification using string kernels. *Journal of Machine Learning Research*, 2(Feb):419–444, 2002.

[20] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu. Transfer feature learning with joint distribution adaptation. In *Proceedings of the IEEE international conference on computer vision*, pages 2200–2207, 2013.

[21] M. Long, J. Wang, J. Sun, and S. Y. Philip. Domain invariant transfer kernel learning. *IEEE Transactions on Knowledge and Data Engineering*, 27(6):1519–1532, 2015.

[22] S. A. Nene, S. K. Nayar, H. Murase, et al. Columbia object image library (coil-20). 1996.

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

[24] M. H. Quang, M. San Biagio, and V. Murino. Log-hilbert-schmidt metric between positive definite operators on hilbert spaces. In *Advances in Neural Information Processing Systems*, pages 388–396, 2014.

[25] D. C. Sorensen. Implicitly restarted arnoldi/lanczos methods for large scale eigenvalue calculations. In *Parallel Numerical Algorithms*, pages 119–165. Springer, 1997.

[26] M. Sugiyama, S. Nakajima, H. Kashima, P. V. Buenau, and M. Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in neural information processing systems*, pages 1433–1440, 2008.

[27] B. Sun, J. Feng, and K. Saenko. Return of frustratingly easy domain adaptation. In *AAAI*, volume 6, page 8, 2016.

[28] C. Villani. *Topics in optimal transportation*. Number 58. American Mathematical Soc., 2003.

[29] K. Zhang, V. Zheng, Q. Wang, J. Kwok, Q. Yang, and I. Marsic. Covariate shift in hilbert space: A solution via sorrogate kernels. In *International Conference on Machine Learning*, pages 388–395, 2013.

[30] Z. Zhang, M. Wang, Y. Xiang, and A. Nehorai. Geometry-adapted gaussian random field regression. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 6528–6532. IEEE, 2017.

[31] S. K. Zhou and R. Chellappa. From sample similarity to ensemble similarity: Probabilistic distance measures in reproducing kernel hilbert space. *IEEE transactions on pattern analysis and machine intelligence*, 28(6):917–929, 2006.