Pattern Analysis and Applications (2023) 26:1425–1439
https://doi.org/10.1007/s10044-023-01170-y

THEORETICAL ADVANCES

RKHS subspace domain adaption via minimum distribution gap

Yanzhen Qiu[1] · Chuangfeng Zhang[1] · Chenkui Xiong[1] · Zhengming Ma[1] · Shaolin Liao[1]

Received: 24 April 2022 / Accepted: 19 April 2023 / Published online: 17 May 2023

Abstract

Subspace learning of Reproducing Kernel Hilbert Space (RKHS) is most popular among domain adaption applications. The key goal is to embed the source and target domain samples into a common RKHS subspace where their distributions could match better. However, most existing domain adaption measures are either based on the first-order statistics that can't accurately qualify the difference of distributions for non-Guassian distributions or complicated co-variance matrix that is difficult to be used and optimized. In this paper, we propose a neat and effective RKHS subspace domain adaption measure: Minimum Distribution Gap (MDG), where the rigorous mathematical formula can be derived to learn the weighting matrix of the optimized orthogonal Hilbert subspace basis via the Lagrange Multiplier Method. To show the efficiency of the proposed MDG measure, extensive numerical experiments with different datasets have been performed and the comparisons with four other state-of-the-art algorithms in the literature show that the proposed MDG measure is very promising.

Keywords Domain adaption · RKHS · Maximum mean difference (MMD) · Lagrange multiplier method (LMM) optimization

1 Introduction

Usually, since the distributions of samples from the source and target domain are different from each other, directly applying the classifiers trained on source domain samples to target domain samples would lead to poor classification performance. It's unwise to retrain a new classifier on target domain samples due to the deficiency of labeled samples. And domain adaption can address this classification problem, because it can transfer the knowledge learned from source domain to target domain [1–11]. For example, with the help of domain adaption, the classifier trained on labeled source domain that consists of ID photos under controlled condition stored in police stations can work well on the unlabeled target domain that consists of target photos captured by some video monitors [12, 13]. At present, a common way of domain adaption based on distribution difference is that source and target domains are transformed into a Reproducing Kernel Hilbert Space (RKHS) subspace shared by the domains, which should be optimized so that their distributions are as close as possible [2, 5–8]. It can be seen that the distribution difference metric named as domain adaption measure is vital for RKHS subspace learning. The Maximum Mean Difference (MMD) is the most representative domain adaption measure. Many related studies [4–7, 10, 11, 14] used the MMD measure to judge the distribution gap between different domains. Generally, the MMD measure between the source domain data $\left\{x_i^s \mid i = 1 \cdots n_s\right\}$ and the target domain data $\left\{x_j^t \mid j = 1, \dots n_t\right\}$ can be written as

$$\operatorname*{argmin}_{H_s} \left\| \frac{1}{n_s} \sum_{i=1}^{n_s} y_i^s - \frac{1}{n_t} \sum_{j=1}^{n_t} y_j^t \right\|_H^2 ,$$

where H_s is a RKHS subspace and $\|\cdot\|_H$ is the RKHS norm; also, $\left\{y_i^s \mid i = 1, \dots n_s\right\}$ and $\left\{y_j^t \mid j = 1, \dots n_t\right\}$ represent the RKHS subspace projections of the original data $\left\{x_i^s \mid i = 1 \cdots n_s\right\}$ and $\left\{x_j^t \mid j = 1, \dots n_t\right\}$, respectively.

✉ Zhengming Ma
issmzm@mail.sysu.edu.cn

✉ Shaolin Liao
liaoshlin@mail.sysu.edu.cn

Yanzhen Qiu
qiuyzh7@mail2.sysu.edu.cn

Chuangfeng Zhang
zhangchf8@mail2.sysu.edu.cn

Chenkui Xiong
xiongchk@mail2.sysu.edu.cn

[1] Sun Yat-Sen University, Guangzhou, China

Although MMD measure is simple and easy to implement, it has its theoretical defect in terms of measuring the distribution difference between two domains: (1) MMD only considers their mean values but ignores their higher-order moments, such as variances; (2) domains with diverse distributions maybe have the same mean value. So, the MMD measure is unable to accurately measure the distribution discrepancy.

In addition, co-variance matrix measure based on the second-order moment proposed by Li [8] and Sun [15] is also commonly used to measure the distribution distance between source and target domains. And the co-variance matrix measure of source domain data and target domain data is

$$\arg\min_{H_s} \left\| \sum{}_s - \sum{}_t \right\|_F^2,$$

where $\|\cdot\|_F$ is Frobenius norm and the definition of co-variance matrices is

$$\sum{}_s = \frac{1}{n_s}\left(y_i^s - \bar{\mu}^s\right)\left(y_i^s - \bar{\mu}^s\right)^T,$$

$$\sum{}_t = \frac{1}{n_t}\left(y_i^t - \bar{\mu}^t\right)\left(y_i^t - \bar{\mu}^t\right)^T,$$

with

$$\bar{\mu}^s = \frac{1}{n_s}\sum_{i=1}^{n_s} y_i^s, \quad \bar{\mu}^t = \frac{1}{n_t}\sum_{i=1}^{n_t} y_i^t.$$

From the perspective of distribution matching, MMD-based or co-variance-based domain adaption methods aim to align the mean (MMD measure) and covariance (co-variance matrix measure) of different domains to align the distributions of domains, which are suitable for the domains obeying Guassian distribution. However, in real world, domains usually obey complex non-Guassian distributions. So, the MMD measure and co-variance matrix measure cannot fully display the performance of domain adaption based on the RKHS subspace learning. In addition, the complicated co-variance matrix measure has large computational costs, because it needs iterative optimization.

To solve the above limitations, we propose a new domain adaption measure MDG. The distributions of the source data X_s and target data X_t in subspace can be matched better by MDG measure, which enhance the transferability of the models trained on source domain. And the MDG measure of the source data X_s and target data X_t is as followed:

$$\mathrm{MDG}\left(X_s, X_t\right) = E\left[\left\|X_s - X_t\right\|^2\right]. \tag{1}$$

Our main contributions are as follows, (1) we prove that the MDG measure is effective for the RKHS subspace classification; (2) the optimized RKHS subspace has been analytically derived through the Lagrange Multiplier Method (LMM); and (3) the results of extensive experiments on different dataset have verified the advantages of the proposed MDG measure, compared to the approaches based on the MMD measure and co-variance matrix measure.

The rest of this paper is organized as follows: In Sect. 2, we briefly review partial-related works on traditional domain adaption based on RKHS subspace learning and deep domain adaption-based neural network; In Sect. 3, we introduce some necessary background of second-order random variable, the related definitions of RKHS and RKHS subspace learning; In Sect. 4, we give the proof of the transformation validity of RKHS subspace, propose the MDG measure and apply it into RKHS subspace classification. In addition, the optimization problem, algorithm and computational complexity analysis of MDG measure are added in Sect. 4. In Sect. 5, the experiments show the validity of MDG measure from the aspects of classification accuracy, running time and RKHS subspace dimension stability; And the conclusion is made in Sect. 6.

2 Related work

Domain adaption [16] aims to transfer the knowledge learned from the well-labeled source domain to help the poor-labeled target domain. The domain adaption based on RKHS subspace learning [2] is the very popular among domain adaption methods, which learn a latent RKHS subspace for source and target domains to reduce their distribution difference. So, the key problem of RKHS subspace is how to measure the distribution gap of two domains. Gretton et al. [14] proposed the MMD to measure the distribution distance of two domains, which simply takes the two means of two domains in RKHS as their distributions, respectively. Currently, the MMD-based methods is the most common among the RKHS subspace learning. For instance, TCA proposed by Pan et al. [17] learned a shared and latent RKHS subspace by using the MMD to reduce distribution divergence and preserving the data properties as much as possible, where the distribution of target domain can align the source domain better. Therefore, the trained models on source domain could apply and perform well on target domain. What's more, Pan et al. put forward a semi-supervised TCA (SSTCA) [17], which considers the label information in subspace learning. IGLDA [6] not only uses MMD to measure the distribution distance of two domains but also retains the local geometry of the labeled source domain data to unearth a suitable subspace, where the distributions could be as much as similar. In 2017, the proposed MIDA [18] reduces the distribution gap between the source domain and target domain by minimizing the MMD distance of them, in the meantime, keeps the maximum independence of the domain features. The MMD-based TIT [5] and LPJT [19] extend domain adaption into the heterogeneous domain adaption [20, 21] which handles domains with arbitrary features and dimensionalities by learning different transformations for different domains. In addition to MMD, the co-variance matrix measure based on second-order moment proposed by DACoM model is used to measure distribution gap to match the distributions of domains. And in DACoM model [8], the local geometric structure and discriminative information are preserved simultaneously.

Deep domain adaption integrates domain adaption into the neural networks to learn more transferable features, which conducts to adapt models trained on source domain to a different but related target domain. For instance, DDC [22] proposed a new CNN architecture that introduces an domain adaption layer and an additional domain adaption loss term based on MMD measure to learn domain invariant representation. Therefore, DDC improves the problem of domain shift between source domain and target domain. In order to further reduce the distribution discrepancy between source domain and target domain, DAN [23] proposed multi-kernel MMD measure(MK-MMD), and then applied MK-MMD measure into pre-trained AlexNet model. Benefiting from CNN and MK-MMD measure, the DAN is likely to learn features that work well on the target domain. In 2017, Deep CORAL [15] extended co-variance matrix measure into the deep neural network, that is, co-variance measure between the source and target feature activation's was added as a domain adaption loss term. Joint training with co-variance loss and classification loss, Deep CORAL could enhance the transferability of feature representation. In addition to combining domain adaption and neural network for classification, Liang et al. [24] applied MK-MMD measure into CNNs and proposed a transferable reconstruction neural network for the compressed signal (CTCS), which applied MK-MMD measure to fine-tuning the pre-trained network. Therefore, the reconstruction capability on target domain signals can be achieved by only fine-tuning the network trained on source domain signals.

3 Preliminary

In this section, some related background knowledge are introduced. First of all, we give the definition of the second-order moment random variable and the necessary and sufficient condition for two second-order moment random variables to be equal. Next, we review some basic concept of RKHS. Finally, we introduce the framework of the RKHS subspace learning. The notions appeared in this paper is collected in Table 1.

Table 1 The table of symbols used in the paper

Ω	Instance space
L^2	Hilbert space composed of quadratic integrable variables
H	Reproducing kernel Hilbert space (RKHS) composed of integrable functions
$H_s \subset H$	RKHS subspace
$\mathbb{R}$	Real number space
$\mathbb{R}^d$	d-dimensional real vectors space
$(\cdot,\cdot)_{L^2}$	Inner product defined in L^2 space
$\langle\cdot,\cdot\rangle_H$	Inner product defined in H space
x_i, x_i^s, x_i^t	An ith general data sample, an ith source data sample, and an ith target data sample in the space Ω
y_i, y_i^s, y_i^t	An ith general data sample, an ith source data sample, and an ith target data sample in the space H_s
$\sum_s, \sum_t$	co-variance matrix of source and target instances
$\bar{\mu}_s, \bar{\mu}_t$	Mean vector of source and target instances
$X, Y, Z, Y_i, Y_i^s, Y_i^t$	Second-order moment random variable
$g : \Omega \rightarrow \mathbb{R}$	A function in Hilbert space H
φ	The mapping from Ω to H
$K, \tilde{k}_i$	Kernel matrix and the ith column vector of K
$k(\cdot,\cdot)$	Kernel function
Θ, ϑ_i	A set of basis of the subspace of H, and the ith orthonormal basis
W, w_{ij}	The coefficient matrix, and the element of W
δ	Parameter of the RBF kernel function
k	Parameter of k-Nearest neighbor
N	The number of all instances
n_s, n_t	The number of source instances and target instances

3.1 Second-order moment random variable

Given a random variable X which obeys a distribution $p(x)$, it becomes a second-order moment random variable if the condition $E\left[|X|^2\right] = \int_\Omega x^2 p(x)\mathrm{d}x < +\infty$ is satisfied. From a physical point of view, a second-order random variable is the limited-energy random signal. And in real life, all signals have limited energy. So, the source and target domain data in original space can be treated as the samplings from two second-order random variables with different distributions.

Assuming that a set of random variables satisfies $\left\{X\middle|E\left[|X|^2\right] < +\infty\right\}$, it is called as a L^2 space that is a Hilbert space and its inner product is defined as [25, 26]

$$(X, Y)_{L^2} = E\left[XY^*\right],$$

where $\forall X, Y \in L^2$, the star denotes the complex conjugate, and the inner product specified by round brackets on L^2 space. Besides, the norm in L^2 is defined as [27]

$$\|X\|_{L^2} = \sqrt{(X,X)_{L^2}}.$$

In light of the positive definiteness of inner product defined in Hilbert space L^2, the necessary and sufficient condition for two second-order random variables to be equal is that the mean squared error between them is zero, which can be formulated as follows (see details in “Appendix A”),

$$\begin{aligned} X_1 = X_2 \Leftrightarrow \|X_1 - X_2\|_{L^2}^2 &= \left(X_1 - X_2, X_1 - X_2\right)_{L^2} \\ &= E\left[\left|X_1 - X_2\right|^2\right] = 0. \end{aligned}$$

where X_1 and X_2 all are second-order random variables from the L^2 space.

3.2 Reproducing kernel Hilbert space

Similarly, the continuous square integrable function space H is given by [28]

$$H : \left\{f\middle| f : \Omega \to \mathbb{R}, \int_\Omega |f(x)|^2 \mathrm{d}x < +\infty\right\},$$

H is a Hilbert space and the inner product of H space is [9],

$$\langle f, g\rangle_H = \int_\Omega f(x) g^*(x)\mathrm{d}x,$$

where the star denotes the complex conjugate.

In particular, a Hilbert space is called a RKHS space if its kernel $k(x', x) : \Omega \times \Omega$ satisfies the following [10, 25, 26]:

For $\forall f \in H$, it can be reproduced through the RKHS inner product of the function itself and the feature vector $k(\cdot, x)$:

$$f(x) = \langle f, k(\cdot, x)\rangle_H.$$

from which the following also holds,

$$\left\langle k(\cdot, x'), k(\cdot, x)\right\rangle_H = k(x', x).$$

3.3 The RKHS subspace learning framework

In domain adaption applications, $X_s = \left\{x_1^s, \dots, x_{n_s}^s\right\}$ and $X_t = \left\{x_1^t, \dots, x_{n_t}^t\right\}$ are from the source and target domains respectively and obey different distributions. Domain adaption based on RKHS subspace learning tries to find a better RKHS subspace to minimize their distribution difference.

First, the kernel transformation $\varphi(x) = k(\cdot, x)$ maps the data samples $X = X_s \cup X_t = \left\{x_1^s, \dots, x_{n_s}^s, x_1^t, \dots, x_{n_t}^t\right\} = \left\{x_1, \dots, x_N\right\} \subseteq \Omega$ into the RKHS space H. And the new orthogonal basis ϑ_i of RKHS subspace H_s can be constructed through linear combination of these non-orthogonal feature vectors:

$$\vartheta_i = \sum_{j=1}^{N} w_{ji}\varphi\left(x_j\right), \quad i = 1, \dots, d, \tag{2}$$

which can be cast into the matrix form as follows

$$\Theta = \Phi W, \tag{3}$$

with

$$W = \begin{bmatrix} w_{11} & \cdots & w_{1d} \\ \vdots & \ddots & \vdots \\ w_{N1} & \cdots & w_{Nd} \end{bmatrix}, \quad \Theta = \left[\vartheta_1 \ \dots \ \vartheta_d\right],$$

$$\Phi = \left[\varphi\left(x_1\right) \ \dots \ \varphi\left(x_N\right)\right].$$

The orthogonality of the new basis Θ satisfies the following condition

$$\begin{bmatrix} \langle \vartheta_1, \theta_1\rangle_H & \cdots & \langle \vartheta_1, \vartheta_d\rangle_H \\ \vdots & \ddots & \vdots \\ \langle \vartheta_d, \vartheta_1\rangle_H & \cdots & \langle \vartheta_d, \vartheta_d\rangle_H \end{bmatrix} = \Theta^T\Theta = I_d. \tag{4}$$

Substituting Eq. (3) into Eq. (4), the following is obtained:

$$\begin{bmatrix} \langle \vartheta_1, \theta_1\rangle_H & \cdots & \langle \vartheta_1, \vartheta_d\rangle_H \\ \vdots & \ddots & \vdots \\ \langle \vartheta_d, \vartheta_1\rangle_H & \cdots & \langle \vartheta_d, \vartheta_d\rangle_H \end{bmatrix} = W^T K W = I_d, \tag{5}$$

where K is the kernel matrix given by

$$K = \begin{bmatrix} k\left(x_1, x_1\right) & \cdots & k\left(x_1, x_N\right) \\ \vdots & \ddots & \vdots \\ k\left(x_N, x_1\right) & \cdots & k\left(x_N, x_N\right) \end{bmatrix} \tag{6}$$

Then, a certain domain adaption measure is used to achieve the optimal RKHS subspace H_s with basis Θ characterized by its weighting matrix W.

Finally, the feature vector $\varphi(x_i)$ is projected onto the RKHS subspace H_s that satisfies the constraint formula of Eq. (5). According to the subspace projection theorem in Hilbert space [27], the coordinates y_i of the feature vector $\varphi(x_i)$ in the RKHS subspace basis H_s with Θ is given by

$$y_i = \begin{bmatrix} y_{i1} \\ \vdots \\ y_{id} \end{bmatrix} = \begin{bmatrix} \left\langle \varphi\left(x_i\right), \vartheta_1 \right\rangle_H \\ \vdots \\ \left\langle \varphi\left(x_i\right), \vartheta_d \right\rangle_H \end{bmatrix} \in \mathbb{R}^d. \tag{7}$$

where d is dimension of the RKHS subspace H_s.

4 RKHS subspace classification with MDG

In this section, we first introduce the proposed MDG measure; second, we confirm the mapping validity of RKHS subspace, that is, a second-order moment random variable in the original data space is still a second-order moment random variable when is transformed into RKHS subspace; then, we apply our MDG measure for the RKHS classification and derive its optimized formula via the LMM; at last, we analysis the algorithm of MDG-based RKHS subspace learning and its computational cost.

4.1 Minimum distribution gap

Suppose there are two second-order moment variables, that is, source domain $X_s \sim p(x)$ and target domain $X_t \sim q(x)$ where $p(x) \neq q(x)$. In order to achieve the goal of aligning the different distributions, we propose an effective MDG measure to reduce the discrepancy between X_s and X_t, as shown in Eq. (1).

In real application, the exact joint probability density functions of X_s and X_t are unknown, and only the sampling data sets from X_s and X_t are available, namely $X_s = \left\{x_1^s, \ldots, x_{n_s}^s\right\}$ and $X_t = \left\{x_1^t, \ldots, x_{n_t}^t\right\}$. So, Eq. (1) can be rewritten as:

$$\begin{aligned} \mathrm{MDG}\left(X_s, X_t\right) &= E\left[\left|X_s - X_t\right|^2\right] \\ &= \int_{\Omega\times\Omega} \left\|x^s - x^t\right\|^2 p\left(x^s, x^t\right) \mathrm{d}x^s \mathrm{d}x^t \\ &\approx \frac{1}{n_s n_t} \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} \left\|x_i^s - x_j^t\right\|^2, \end{aligned} \tag{8}$$

where $p(x^s, x^t)$ is the joint probability density function of X_s and X_t and it is replaced by the uniform distribution.

4.2 The mapping validity of RKHS subspace

Here, we give a proof for the mapping validity of RKHS subspace. In other words, a second-order moment variable is still second-order moment through the transformation of RKHS subspace. And this proof is essential for the MDG measure to be further extended into the RKHS subspace.

For a second-order moment random variable $X \in \Omega$, we get a random variable Y in light of the projection theorem Eq. (7):

$$Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_d \end{bmatrix} = \begin{bmatrix} \langle \varphi(X), \vartheta_1 \rangle_H \\ \vdots \\ \langle \varphi(X), \vartheta_d \rangle_H \end{bmatrix},$$

which represents the projection of $\varphi(X)$ in subspace H_s with the orthogonal basis $\vartheta_i (i = 1 \cdots d)$. Now, we prove Y is a second-order moment random variable by proving that each component Y_i of Y meets the condition $E\left[\left|Y_i\right|^2\right] = \int_\Omega y_i^2 p\left(y_i\right) \mathrm{d}y_i < +\infty$. In fact, we have

$$\begin{aligned} E\left[\left|Y_i\right|^2\right] &= E\left[\left|\langle \varphi(X), \vartheta_i \rangle_H\right|^2\right] \\ &= \int_\Omega \left|\langle \varphi(x), \vartheta_i \rangle_H\right|^2 p(x) \mathrm{d}x \\ &\le \int_\Omega \left|\langle \varphi(x), \vartheta_i \rangle_H\right|^2 \mathrm{d}x \\ &= \int_\Omega \left|\left\langle \varphi(x), \sum_{j=1}^N \omega_{ji} \varphi\left(x_j\right) \right\rangle_H\right|^2 \mathrm{d}x \\ &= \int_\Omega \left|\sum_{j=1}^N \omega_{ji} \left\langle \varphi(x), \varphi\left(x_j\right) \right\rangle_H\right|^2 \mathrm{d}x \\ &= \int_\Omega \left|\sum_{j=1}^N \omega_{ji} k\left(x, x_j\right)\right|^2 \mathrm{d}x \\ &\le \sum_{p=1}^N \sum_{q=1}^N \left|\omega_{pi}\omega_{qi}\right| \left|\int_\Omega k\left(x, x_q\right) k\left(x, x_p\right) \mathrm{d}x\right| \\ &\le \sum_{p=1}^N \sum_{q=1}^N \left|\omega_{pi}\omega_{qi}\right| \sqrt{\int_\Omega k^2\left(x, x_q\right) \mathrm{d}x} \\ &\quad \sqrt{\int_\Omega k^2\left(x, x_p\right) \mathrm{d}x} < +\infty, \end{aligned}$$

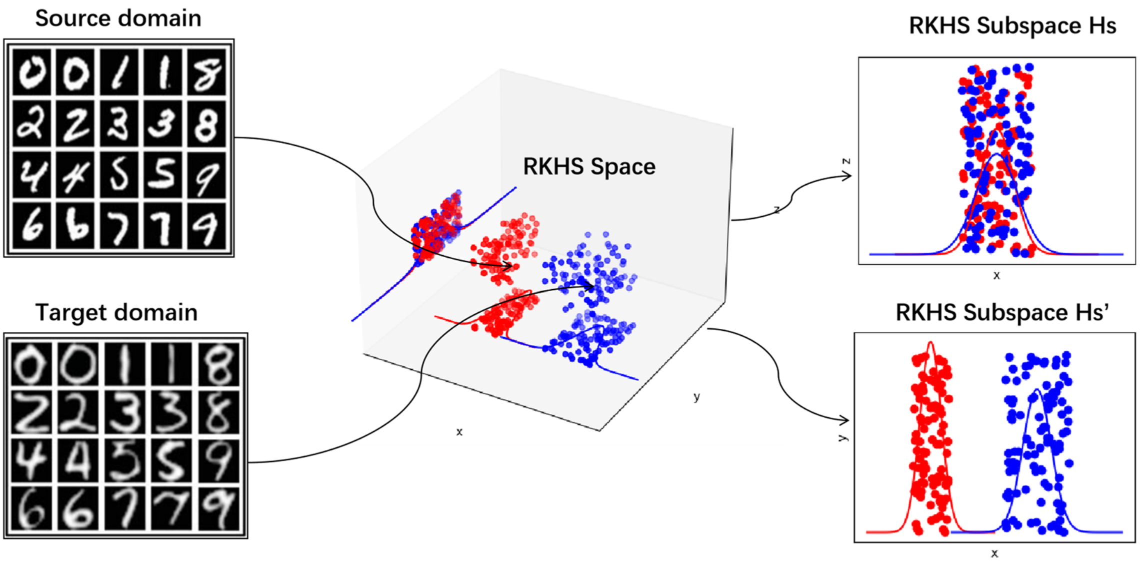


Fig. 1 The illustration of RKHS subspace domain adaption via MDG. Firstly, we map the instances from two domains into the RKHS space (the red dots and blue dots represent instances from source and target domains respectively). Then, we project these mapped instances into RKHS subspace H_s and RKHS subspace $H_s^{'}$ respectively, where H_s is the optimal subspace learned by minimizing the MDG measure proposed in this paper and $H_s^{'}$ is non-optimal. Obviously, the distribution gap of the two domains data has been minimized more dramatically in the optimal RKHS subspace H_s than in the non-optimal RKHS subspace $H_s^{'}$

where the X follows the probability density function $0 \leq p(x) \leq 1$ and $\varphi(x) = k(\cdot, x)$ is an absolutely integrable function.

From the above derivation, we can make a useful conclusion that mapping second-order moment variables into RKHS subspace, these variables still are second-order moment. In light of this conclusion, we can apply the MDG measure to the domain adaption based on RKHS subspace learning.

4.3 MDG for RKHS subspace classification

In this paper, the MDG as domain adaption measure is used for the domain adaption shown in Fig. 1. Specifically, we first transform the source domain $X_s = \left\{x_1^s, \ldots, x_{n_s}^s\right\}$ and target domain $X_t = \left\{x_1^t, \ldots, x_{n_t}^t\right\}$ into the RKHS subspace H_s to get $Y_s = \left[y_1^s, \ldots, y_{n_s}^s\right]$ and $Y_t = \left[y_1^t, \ldots, y_{n_t}^t\right]$, which represent the coordinates of the corresponding projection on the orthogonal basis Θ of subspace H_s. According to the proof in Sect. 4.2, Y_s and Y_t are second-order moment variables.

Then, we minimize the MDG between Y_s and Y_t to learn a optimal RKHS subspace H_s so that their distributions are as close as possible. So, our goal is to minimize the following problem:

$$\begin{aligned} &\arg\min_{W} \left\|Y_s - Y_t\right\|^2 = E\left[\left|Y_s - Y_t\right|^2\right], \\ &\quad \text{s.t.} \quad W^T K W = I_d. \end{aligned} \tag{9}$$

With the help of Sects. 4.1 and 4.2, Eq. (9) can be derived as follows

$$\begin{aligned} \left\|Y_s - Y_t\right\|^2 &= \frac{1}{n_s n_t} \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} \left\|y_i^s - y_j^t\right\|^2 \\ &= \frac{1}{n_s n_t} \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} \left\|W^T\left(K(:,i) - K(:,n_s+j)\right)\right\|^2 \\ &= \frac{1}{n_s n_t} \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} \left\|W^T \varphi_{ij}\right\|^2 \\ &= \frac{1}{n_s n_t} \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} tr\left(W^T \varphi_{ij} \varphi_{ij}^T W\right) \\ &= \mathrm{tr}\left(W^T\left(\frac{1}{n_s n_t} \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} \varphi_{ij} \varphi_{ij}^T\right) W\right) \\ &= \mathrm{tr}\left(W^T \Psi W\right), \end{aligned} \tag{10}$$

where $\varphi_{ij} = K(:,i) - K(:,n_s+j)$, and

$$\Psi = \frac{1}{n_s n_t} \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} \varphi_{ij} \varphi_{ij}^T. \tag{11}$$

Fig. 2 Examples of four domains in Office-Caltech10 dataset

Finally, optimization problem of Eq. (9) reduces to

$$\arg\min_{W}\mathrm{tr}\left(W^T\Psi W\right), \quad \text{s.t.} \quad W^TKW = I_d. \tag{12}$$

4.4 Optimization problem

Next, we explain in detail how Eq. (12) is solved by LMM: Because K is a SPD matrix, it can be factorized through its eigenvalues and eigenvectors matrices as follows

$$K = U\Lambda U^T = U\Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}}U^T, \tag{13}$$

where $UU^T = I$ and Λ is a diagonal matrix.

Denoting $\Lambda^{\frac{1}{2}}U^T$ by L, the following are obtained

$$K = L^TL, \quad W^TKW = W^TL^TLW = I_d, \tag{14}$$

from which the matrix trace of Eq. (12) is given by

$$tr\left(W^T\Psi W\right) = tr\left(W^TL^T\left(L^T\right)^{-1}\Psi L^{-1}LW\right).$$

Now, denoting $G = LW$, the optimization problem of Eq. (12) is transformed into

$$\arg\min_{G} tr\left(G^TAG\right), \quad \text{s.t.} \quad G^TG = I_d, \tag{15}$$

with

$$A = \left(L^T\right)^{-1}\Psi L^{-1}, \tag{16}$$

which can be solved through the LMM with the Lagrangian function given by [29]

$$L(G,Z) = \mathrm{tr}\left(G^TAG\right) - tr\left(\left(G^TG - I_d\right)Z\right), \tag{17}$$

where Z is a symmetric matrix and z_{ij} are Lagrange multipliers.

Equation (17) can be solved as follows

$$\begin{cases} \frac{\partial L(G,Z)}{\partial G} = 2AG - 2GZ = 0 \\ \frac{\partial L(G,Z)}{\partial Z} = \left(G^TG - I_d\right)^T = 0 \end{cases} \Rightarrow \begin{cases} AG = GZ; \\ G^TG = I_d. \end{cases} \tag{18}$$

When Z is a diagonal matrix, if G is the eigenvectors matrix of A, then Eq. (18) is satisfied and the minimization problem of Eq. (15) can be achieved by selecting the smallest d eigenvalues and the corresponding eigenvectors.

When Z is not a diagonal matrix but symmetric, it can be factorized in terms of its eigenvalues matrix Σ and eigenvectors matrix V as follows,

$$Z = V\Sigma V^T. \tag{19}$$

Substituting Eq. (19) into Eq. (18), the following is obtained,

$$\begin{cases} AG = GV\Sigma V^T \Rightarrow AGV = GV\Sigma \Rightarrow A\tilde{G} = \tilde{G}\Sigma, \\ G^TG = I_d \Rightarrow V^TG^TGV = I_d \Rightarrow \tilde{G}^T\tilde{G} = I_d, \end{cases} \tag{20}$$

where $\tilde{G} = GV$ and we have used the orthogonality relation of the eigenvectors matrix $V^TV = I_d$.

It's clear that the eigenvalues matrix Σ is a diagonal matrix and the minimization problem of Eq. (15) can be achieved when $\tilde{G}$ is formed with d smallest eigenvectors of A.

According to the above analysis, the minimization problem of Eq. (15) can be achieved by selecting the smallest d eigenvectors of A.

Finally, we can get the optimized weighting matrix W of the original optimization problem of Eq. (12) as follows

$$W = L^{-1}G = \left(\Lambda^{\frac{1}{2}}U^T\right)^{-1}G = U\Lambda^{-\frac{1}{2}}G. \tag{21}$$

4.5 Algorithm of RKHS classification with MDG

The procedure for the solution of MDG is summarized in Algorithm 1, which is explained as follows:

```
Algorithm 1 Calculate $W$ under MDG measure
  Inputs:
  Two different instance set $X_s$, $X_t$;
  Dimension of subspace $d$;
  Kernel function $k(\cdot,\cdot)$.
  Output:
  Coefficient matrix $W$.
1: $n_s \leftarrow size(X_s, 2)$.
2: $n_t \leftarrow size(X_t, 2)$.
3: Construct the RKHS kernel matrix $K$ from $X_s$ and $X_t$ according to Eq. (6), and matrix $\Psi = \frac{1}{n_s n_t}\sum_{i=1}^{n_s}\sum_{j=1}^{n_t}\varphi_{ij}\varphi_{ij}^T$ according to Eq. (10).
4: Calculate eigenvectors matrix $U$ and eigenvalues matrix $\Lambda$ of $K$ according to Eq. (13).
5: Calculate $L \leftarrow \Lambda^{\frac{1}{2}}U^T$ according to Eq. (14) and form the matrix $A \leftarrow \left(L^T\right)^{-1}\Psi L^{-1}$ according to Eq. (16).
6: Select the $d$ smallest eigenvectors of the matrix $A$ to construct matrix $G$ according to Eq. (18) or Eq. (20).
7: Finally, obtain the weighting matrix of the RKHS subspace $W \leftarrow L^{-1}G$ according to Eq. (21).
```

The input of the algorithm are samples from X_s and X_t, the kernel function $k(\cdot,\cdot)$, and the RKHS subspace dimension d; and the output of the algorithm are the weighting matrix W that characterizes the orthogonal basis Θ of the RKHS subspace.

The algorithm takes the samples from both X_s and X_t to form the joint RKHS subspace kernel matrix K; then its eigenvectors matrix U is calculated; after that, the intermediate matrix L and A are calculated from Eqs. (14) and (16), respectively; and finally, the weighting matrix W that characterizes the orthogonal basis of the RKHS subspace is obtained by selecting the d smallest eigenvectors of the intermediate matrix A according to Eq. (21).

After having the weighting matrix W that characterizes the orthogonal basis of the RKHS subspace, the unknown labels of instances X_t can be obtained as follows:

1. The data samples set $X = X_s \cup X_t$ in H_s can be projected to the RKHS subspace as $Y = W^T K$: the samples set from source domain X_s are projected to get $Y_s = Y\left(:, n_s\right)$, and the data samples set X_t are projected to get $Y_t = Y\left(:, n_s + 1 : n_s + n_t\right)$;
2. Train the classifier with the projection samples Y_s;
3. Use the trained classifier to label the projection samples Y_t.

4.6 Computational complexity

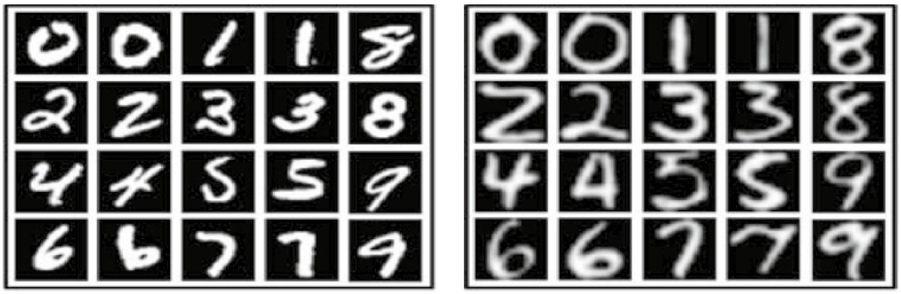
According to the Algorithm 1, the computation costs of our MDG-based RKHS subspace learning consists of three major parts:

1. The Construction the kernel matrix K in step 3, and it costs $\mathcal{O}\left(mn^2\right)$ for computing (m is the dimension of samples)
2. The construction of matrix Ψ in step 3, and it costs $\mathcal{O}\left(n_s n_t n^2\right)$ for computing ($n = n_s + n_t$)
3. The optimization of coefficient matrix W in step 4 and step 7, which costs $\mathcal{O}\left(dn^2\right)$

So, the overall computational complexity of Algorithm 1 would be $\mathcal{O}\left(mn^2 + n_s n_t n^2 + dn^2\right)$.

5 Experiments

In this section, we conduct two kinds of experiments to verify the classification effectiveness of our MDG measure: one is the comparison with the MMD and co-variance measures; the another is to apply our MDG measure into the four domain adaption algorithms to replace their original



(a) Samples from MNIST Dataset (b) Samples from USPS Dataset

Fig. 3 Examples of 0–9 digits in handwritten digits dataset

Table 2 Classification accuracy comparison of the Office-Caltech10 dataset

$k = 1$	MDG	MMD	Cov	$k = 3$	MDG	MMD	Cov
$A \to C$	**0.2048**	0.0971	0.1238	$A \to C$	**0.2012**	0.1211	0.1300
$A \to D$	**0.1529**	0.0510	0.0892	$A \to D$	**0.1529**	0.0701	0.0828
$C \to A$	**0.1670**	0.0793	0.1378	$C \to A$	**0.1733**	0.1096	0.1315
$D \to A$	**0.1858**	0.0835	0.1002	$D \to A$	**0.1806**	0.0866	0.1023
$D \to C$	**0.1523**	0.0825	0.0908	$D \to C$	**0.1478**	0.0935	0.1264
$D \to W$	**0.3966**	0.0915	0.0915	$D \to W$	**0.2949**	0.0983	0.0847
$W \to A$	**0.1795**	0.0825	0.0908	$W \to A$	**0.1983**	0.0929	0.0971
$W \to C$	**0.1273**	0.0935	0.1051	$W \to C$	**0.1407**	0.0971	0.1140
$k = 5$	MDG	MMD	Cov	$k = 7$	MDG	MMD	Cov
$A \to C$	**0.2208**	0.1282	0.1443	$A \to C$	**0.2315**	0.1273	0.1434
$A \to D$	**0.1529**	0.0764	0.1019	$A \to D$	**0.1529**	0.0892	0.0764
$C \to A$	**0.1587**	0.1013	0.1106	$C \to A$	**0.1618**	0.0971	0.1388
$D \to A$	**0.1754**	0.0887	0.1065	$D \to A$	**0.1743**	0.0981	0.1033
$D \to C$	**0.1434**	0.0962	0.1256	$D \to C$	**0.1514**	0.0971	0.1238
$D \to W$	**0.2881**	0.1322	0.0949	$D \to W$	**0.2475**	0.1593	0.1186
$W \to A$	**0.2077**	0.0939	0.0905	$W \to A$	**0.1983**	0.0971	0.1033
$W \to C$	**0.1532**	0.0944	0.1113	$W \to C$	**0.1621**	0.0944	0.1104

distribution discrepancy measures to evaluate MDG measure's performance. In addition, we conduct the experiment to verify the insensitivity of MDG measure to RKHS subspace dimension.

5.1 The real-world datasets

We assess the performance of the proposed MDG measure on four popular datasets: Office-Caltech10 dataset, handwritten digits dataset, text dataset and VLSIC dataset. The data that support the findings of this study are available from this website.[1] Next, the four datasets are introduced, respectively.

1. *Office-Caltech10 dataset.* Office-Caltech10 dataset consists of four domains: Amazon (A, collected from Amazon), DSLR (D, shot by SLR camera), webcam (W, collected by webcam), and Caltech (C, collected by Caltech) [30]. Each domain contains 10 classes, such as backpack, monitor, headphone and so on. Examples of headphones from A, D, W, and C domains are shown in Fig. 2. And each domain is used as source domain and target domain repeatedly.
2. *Text dataset.* This dataset comes from Reuters-21,578 dataset[2] including 21,578 documents and 672 categories. In fact, we use a pre-processed dataset, which are divided into three categories: orgs, places, and people, with each category containing two sub-classes [6]. We regard these three categories as three domains and select

[1] https://github.com/jindongwang/transferlearning/tree/master/data.

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

Table 3 Classification accuracy comparison of the text dataset

Orgs → places	MDG	MMD	Cov
$k = 1$	**0.5465**	0.4612	0.5177
$k = 3$	**0.5523**	0.5062	0.5091
$k = 5$	**0.5446**	0.4976	0.5110
$k = 7$	**0.5638**	0.5072	0.5283

Table 4 Accuracy comparison of the handwritten digits dataset

	MDG	MMD	Cov
MNIST → USPS			
$k = 1$	**0.6489**	0.1739	0.4267
$k = 3$	**0.6517**	0.1844	0.4150
$k = 5$	**0.6683**	0.1878	0.4156
$k = 7$	**0.6728**	0.2100	0.4167
USPS → MNIST			
$k = 1$	**0.3775**	0.2120	0.1000
$k = 3$	**0.3865**	0.2450	0.1435
$k = 5$	**0.3795**	0.2555	0.1410
$k = 7$	**0.3675**	0.2530	0.1625

randomly two categories as source and target domain respectively on each classification task.

3. *Handwritten digits dataset.* The handwritten digits dataset consists of MNIST[3] and USPS[4] dataset with different distributions, which include handwritten 10 digits from 0 to 9. MNIST dataset contains 70,000 sheets of 28 × 28 gray images, and the USPS dataset contains 11,000 sheets of 16 × 16 gray images. Since the large amount of samples in this dataset and the limited processing power of our device, the subset of handwritten digits dataset are used in following experiments, which consists of 2000 images from MNIST and 1800 images from USPS that are all randomly selected. Then, some data preparation are done for this subset, which contains the uniformly scaling these gray images to 16 × 16 images, and then flattening each image into 256 dimensional vector. Some examples of the handwritten digits dataset are shown in Fig. 3. And MNIST and USPS dataset are taken as source and target domain by turns.
4. *VLSIC dataset.* The VLSIC dataset consists of 5 domains: VOC2007(V), LabelMe(L), SUN09(S), ImageNet(I) and Caltech101(C) from different distributions. Since the original data have very high dimension, we firstly applied PCA [31] to reduce the dimension of original data from 4096 into 300. And then we selected the 5 classes shared by the five domains to conduct the experiments.

5.2 The comparison with MMD and co-variance measures

In this subsection, we conduct the comparison with MMD and co-variance measures on the above four dataset. Specially, the MMD measure is the most popular among domain adaption algorithms, and co-variance measure has recently been used in domain adaption algorithms [8, 15]. For simplicity, co-variance measure be denoted as cov in Tables 2, 3, 4 and 5. In this subsection experiments, the used parameters are set up to:

1. The Gaussian Radial Basis function (RBF) kernel is chosen as the reproducing kernel of RKHS [9]: $k\left(x_1, x_2\right) = e^{-\frac{\|x_1 - x_2\|^2}{2\delta^2}}, \delta = 10.$
2. The dimension of the RKHS subspace H_s has been set to $d = 30$ for the handwritten digits dataset and $d = 100$ for the other three datasets.
3. k-Nearest Neighbor method (knn) [32] is used for classification, and experiments are carried out on $k = [1, 3, 5, 7]$. The calculation of classification accuracy is as followed:

$$\text{accuracy} = \frac{\sum_{x_t \in X_t} \left\{x_t \in X_t \cap \text{knn}\left\{x_t\right\} = label\left\{x_t\right\}\right\}}{\text{num}\left\{X_t\right\}},$$

   where X_t is target domain samples set and $\text{num}\left\{X_t\right\}$ is the number of samples in X_t, $\text{knn}\left\{x_t\right\}$ is the label predicted by knn method for a target data x_t and label $\left\{x_t\right\}$ is the ground truth label of x_t.
4. The number of iterations of the co-variance domain adaption measure is set to 10.

And the specific classification task arrangement are as followed:

1. *Office-Caltech10 dataset classification.* According to IGLDA [6], the SURF (Speed Up Robust Features) [33] of the dataset are first extracted; then the features are normalized and z-scored so that their means are zero and the standard deviations are set up to one. In total, we carried out six tasks: $A \rightarrow C$, $A \rightarrow D$, $C \rightarrow A$, $D \rightarrow A$, $D \rightarrow C$, $D \rightarrow W$, $W \rightarrow A$, $W \rightarrow C$. In detail, $A \rightarrow C$

[3] http://yann.lecun.com/exdb/mnist/index.html.

[4] http://www-i6.informatik.rwth-aachen.de/.

Table 5 Accuracy comparison of the VLSIC dataset

$k = 1$	MDG	MMD	Cov	$k = 3$	MDG	MMD	Cov
$C \rightarrow L$	**0.4620**	0.2771	0.2364	$C \rightarrow L$	**0.4635**	0.2677	0.2003
$C \rightarrow S$	**0.3827**	0.2367	0.1999	$C \rightarrow S$	**0.3851**	0.1987	0.1496
$C \rightarrow V$	**0.4437**	0.2707	0.2145	$C \rightarrow V$	**0.4437**	0.2556	0.1807
$I \rightarrow C$	**0.3781**	0.1929	0.2007	$I \rightarrow C$	**0.3816**	0.1731	0.1816
$I \rightarrow V$	**0.3353**	0.1842	0.1928	$I \rightarrow V$	**0.3326**	0.1505	0.1431
$V \rightarrow L$	**0.3823**	0.3008	0.3200	$V \rightarrow L$	**0.3923**	0.3313	0.3343
$k = 5$	MDG	MMD	Cov	$k = 7$	MDG	MMD	Cov
$C \rightarrow L$	**0.4646**	0.2944	0.2101	$C \rightarrow L$	**0.4654**	0.3309	0.2161
$C \rightarrow S$	**0.3851**	0.2188	0.1755	$C \rightarrow S$	**0.3851**	0.2282	0.1755
$C \rightarrow V$	**0.4437**	0.2823	0.1899	$C \rightarrow V$	**0.4437**	0.3089	0.1931
$I \rightarrow C$	**0.2678**	0.1767	0.1908	$I \rightarrow C$	**0.2707**	0.1710	0.1830
$I \rightarrow V$	**0.1525**	0.1517	0.1466	$I \rightarrow V$	**0.1540**	0.1327	0.1437
$V \rightarrow L$	**0.4040**	0.3566	0.3611	$V \rightarrow L$	**0.3938**	0.3859	**0.3938**

Table 6 Accuracy comparison of the Office-Caltech10: TIT versus TIT_MDG

Source → Target	TIT	TIT_MDG	Source → Target	TIT	TIT_MDG
$A \rightarrow C$	0.5314	**0.5527**	$D \rightarrow A$	0.6761	**0.6823**
$A \rightarrow D$	0.5143	**0.5238**	$D \rightarrow C$	0.5367	**0.5474**
$A \rightarrow W$	0.6294	**0.6447**	$D \rightarrow W$	0.7970	**0.8122**
$C \rightarrow A$	0.6792	**0.6948**	$W \rightarrow A$	0.6181	**0.6275**
$C \rightarrow D$	0.5429	**0.5524**	$W \rightarrow C$	0.5060	**0.5154**
$C \rightarrow W$	0.5228	**0.5787**	$W \rightarrow D$	0.7810	**0.8095**

Table 7 Accuracy comparison of the Office-Caltech10: IGLDA versus IGLDA_MDG

Source → Target	IGLDA	IGLDA_MDG	Source → Target	IGLDA	IGLDA_MDG
$A \rightarrow C$	0.3108	**0.3215**	$D \rightarrow A$	0.3486	**0.3977**
$A \rightarrow D$	0.2866	**0.3949**	$D \rightarrow C$	0.3019	**0.3224**
$A \rightarrow W$	0.2169	**0.3627**	$D \rightarrow W$	**0.7390**	0.7153
$C \rightarrow A$	0.3591	**0.3998**	$W \rightarrow A$	0.3727	**0.3862**
$C \rightarrow D$	0.2166	**0.3057**	$W \rightarrow C$	0.3072	**0.3455**
$C \rightarrow W$	0.2712	**0.3356**	$W \rightarrow D$	0.6624	**0.7261**

means that Amazon is the source domain and Caltech is the target domain.

2. *Text dataset classification.* We set up only one classification task: orgs → places.
3. *Handwritten digits dataset classification.* For the handwritten digits dataset, we set two tasks, MNIST → USPS and USPS → MNIST, where MNIST → USPS means that the MNIST dataset are selected as the source domain and USPS dataset are target domain.
4. *VLSIC dataset classification.* Six tasks are set up on this dataset: $C \rightarrow L, C \rightarrow S, C \rightarrow V, I \rightarrow C, I \rightarrow V, V \rightarrow L$.

5.3 Comparisons with state-of-the-art domain adaption algorithms

In this subsection, we compare the proposed MDG measure with TIT [5], IGLDA [6], LPJT [19] and MIDA [18] algorithms in the literature to show its performance. It's noted that the above algorithms consist of not only domain

Table 8 Accuracy comparison of the handwritten digits dataset: LPJT versus LPJT_MDG

Source → Target	LPJT	LPJT_MDG
MNIST → USPS	0.7439	**0.7911**
USPS → MNIST	0.5605	**0.5800**

Table 9 Accuracy comparison of the Text dataset: MIDA vs MIDA_MDG

Source → target	MIDA	MIDA_MDG
Orgs → people	0.5828	**0.6035**
Orgs → places	0.5542	**0.6069**
People → places	0.5227	**0.5525**

adaption measures, but also other regularization terms to ensure the classification performance. For the objective assessment of our MDG, we replace the domain adaption measure used in each algorithm with the MDG measure so that we get four nearly-new domain adaption algorithms, namely, TIT_MDG, IGLDA_MDG, LPJT_MDG and MIDA_MDG. For example, TIT_MDG is obtained by replacing the domain adaption measure used in the TIT algorithm with MDG method, and the original regularization of TIT algorithm remains unchanged. And IGLDA_MDG, MIDA_MDG and LPJT_MDG are generated alike. We totally have four comparison tasks: TIT vs TIT_MDG, IGLDA vs IGLDA_MDG, MIDA vs MIDA_MDG and LPJT vs LPJT_MDG. Since the four original algorithms all apply SVM to classify, the very common SVM classifiers with different kernels are used to classify the target domain samples. Besides, the dimension of RKHS subspace is 100.

1. *TIT versus TIT_MDG.* In this experiment, the Office-Caltech10 dataset are used and twelve tasks are set up totally. We randomly select two domains samples for each task, and one is as source domain and the other as target domain. In addition, we use the SVM classifier with RBF kernel.
2. *IGLDA versus IGLDA_MDG.* We conduct 12 tasks on Office-Caltech10 dataset to compare the IGLDA_MDG with IGLDA, and tasks setup are as above. In addition, the SVM classifier based on linear kernel is used.
3. *LPJT versus LPJT_MDG.* In this experiment, we conduct two tasks on handwritten digits dataset, that is, MNIST→USPS and USPS → MNIST. And we select the RBF kernel-based SVM classifier to classify.
4. *MIDA versus MIDA_MDG.* Here, we compare the combined algorithm MIDA_MDG with MIDA to verify the effectiveness of our MDG measure on handwritten digits dataset. And SVM classifier based on linear kernel is used.

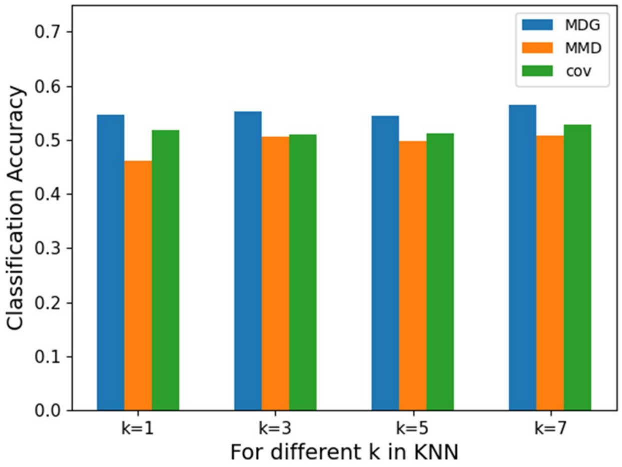
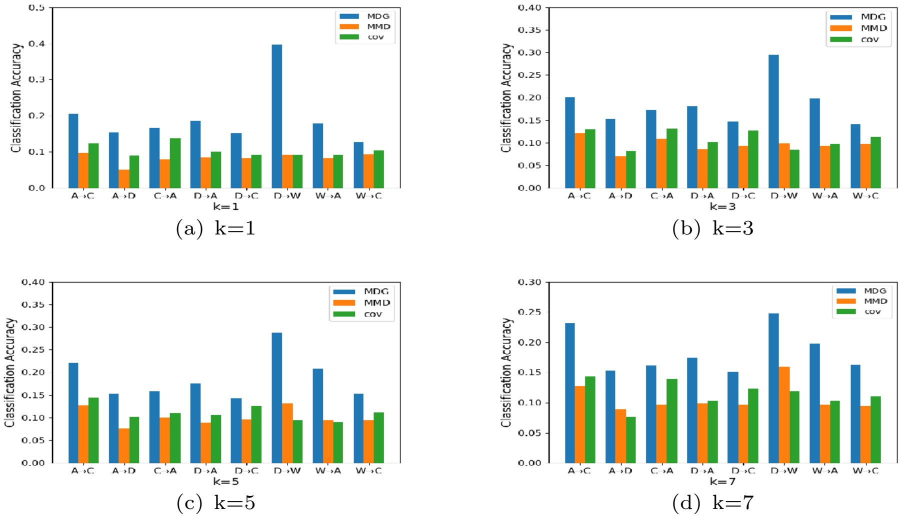


Fig. 5 The classification accuracy of MDG, MMD, co-variance measure in different *k* on text dataset

5.4 Classification results

Under the experiment setting of Sects. 5.2 and 5.3, we get the all classification results and report them in

Fig. 4 The classification accuracy of MDG, MMD, co-variance measure in different *k* on Office-Caltech10 dataset



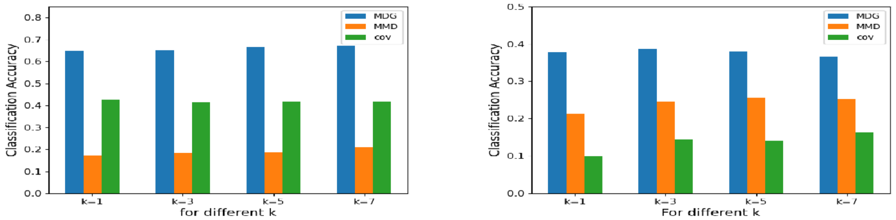
(a) k=1

(b) k=3

(c) k=5

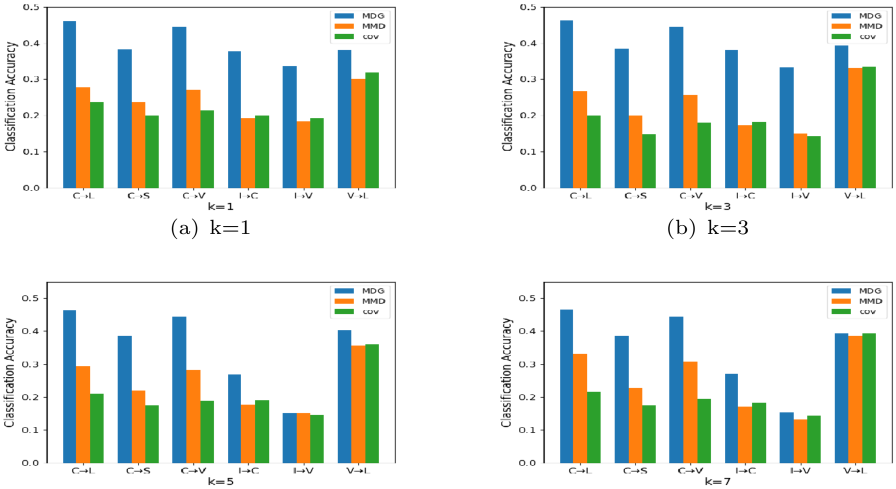
(d) k=7

Fig. 6 The classification accuracy of MDG, MMD, co-variance measure in different k on handwritten digits dataset



(a) MNIST → USPS (b) USPS → MNIST

Fig. 7 The classification accuracy of MDG, MMD, co-variance measure in different k on VLSIC Datasets



(a) k=1 (b) k=3

(c) k=5 (d) k=7

Tables 2, 3, 4, 5, 6, 7, 8 and 9 and the best result in each classification task is bolded for convenience.

The classification results of Sect. 5.2 on the four real-world dataset are all collected in Tables 2, 3 and 4 and visualize with Figs. 4, 5, 6 and 7, respectively. Among these domain adaption measures, the MDG measure works much better than the MMD measure and the un-optimized co-variance measure, which delivers more decent classification accuracy. It show that MDG measure can learn a good common subspace for source and target domain samples, where their distributions match better than in the subspace learned by MMD or co-variance. For 80% tasks of this subsection, co-variance measure achieves higher classification accuracy than MMD measure. The reason is that MMD measure use the first-order moment statistical information of domain, while the second-order statistical information are used in co-variance. And the generally low classification accuracy of tasks in the subsection is due to the fact that the domain adaption measure only focus on global information—inter-domain distribution difference, but ignores the local information such as intra-class distance within domain, the local geometric structure, and discriminative information [5, 6, 8, 11, 16, 18, 19]. So, current domain adaption algorithms all consider the global and local information at the same time. However, since the innovation of this paper is to propose a neat and effective MDG measure to align the different distributions, the local information is not considered for the time being.

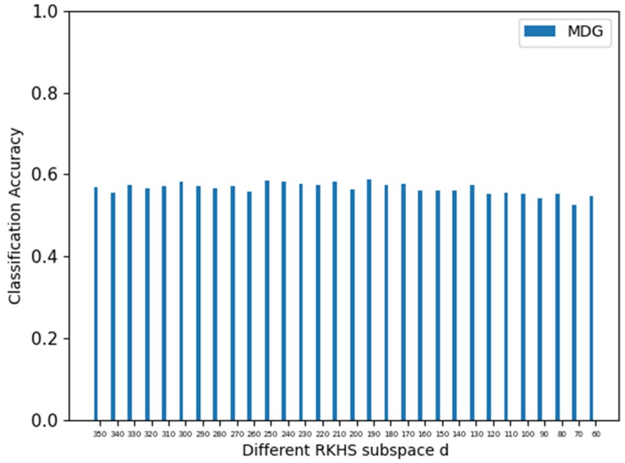


Fig. 8 The classification accuracy of orgs → places task with RKHS subspace dimension d from 350 to 60

Tables 6, 7, 8 and 9 show that the proposed MDG works well on the four dataset, which transforms the source and target data into a great latent RKHS subspace where the distribution gap is smaller than the original algorithms so that it enhances the ability to classify.

In addition, we compared their running speed on the domain adaption task orgs → places. The codes of MMD, co-variance and MDG were written in MATLAB R2018a, and no parallel computing was used. The running times of MMD, co-variance and MDG were 0.5 s, 96.3 s and 29.1 s, respectively. Although the running time of our measure is not the shortest, it is acceptable compared with the co-variance method. And considering the classification results, MDG is more practical to use than MMD and co-variance measures.

5.5 RKHS subspace dimension sensitivity analysis

The RKHS is an infinite linear space, so its subspace dimension can be arbitrary or even infinite. Therefore, it is difficult or even impossible to realize RKHS subspace learning by computer. For domain adaption methods based on RKHS subspace learning, the subspace is constructed by the linear combination of the transformed samples in RKHS. According to Sect. 3.3, the upper limit of the subspace dimension d is the rank of the kernel matrix K, that is N. In practice, d is often selected adaptively according to the input data.

We perform the experiment on tuning $d(d < N)$ to show that the proposed RKHS subspace learning based on MDG measure is robust on the parameter d, namely the classification accuracy remains stable when d changes over a large range. Keeping other parameters unchanged, we constantly adjust the dimension d of the subspace from 350 to 60, and conduct a classification task every 10 dimension on orgs → places. And the results of different d is showed in Fig. 8

From Fig. 8, we can see that the classification results remain robust even d changes over a large range.

6 Conclusion

In this paper, we study a neat and effective MDG measure for RKHS subspace domain adaption classification problem. The MDG measure optimizes the RKHS subspace, where distribution difference between the source-domain data and the target-domain data are as small as possible. Compared to the first-order moment MMD measure and the second-order moment co-variance, the MDG measure has the advantage of capturing the higher-order moments of the distribution. Also, compared to the complicated co-variance measure, it has the advantage of easy to use and can be optimized analytically: rigorous mathematical formula has been derived for the weighting matrix of the optimized orthogonal Hilbert subspace basis, via the LMM optimization. At last, extensive experiments with four image dataset have been carried out. Comparisons with other four state-of-the-art domain adaption algorithms in the literature with both the MMD and co-variance measures show that the RKHS subspace based on MDG measure approach does achieve better classification performance in general.

And according to Sect. 2, some recent works have applied MMD and co-variance into deep neural network as additional loss term to enhance the transferability of feature representation. Hence, in our future work, we will consider extending MDG measure into the deep learning architecture.

Appendix A: Identical random variables

Two second-order moment random variables are identical if and only if their statistical mean square error is zero,

$$E\left[\left|Y - Y'\right|^2\right] = 0 \Leftrightarrow Y = Y'. \tag{22}$$

To demonstrate this, the variance of Eq. (22) can be expressed as follows

$$E\left[\left|Y - Y'\right|^2\right] = \int\int_{\Omega(Y,Y')} \left(y - y'\right)^2 p(y, y')dydy'. \tag{23}$$

Because both $\left(y - y'\right)^2$ and $p(y, y')$ are semi-definite or non-negative, Eq. (23) is zero when one of the following two conditions are met for all points in the probability domain Ω,

$$\begin{cases} \left(y - y'\right)^2 = 0; \\ p(y, y') = 0. \end{cases} \tag{24}$$

It can be shown that Eq. (24) is equivalent to the joint probability $p(y, y') = f(y)\delta(y - y')$,

$$\begin{aligned} p(y) &= \int_{y'} p(y, y')dy' = \int_{y'} f(y)\delta(y - y')dy' = f(y), \\ p(y') &= \int_{y} p(y, y')dy = \int_{y} f(y)\delta(y - y')dy = f(y), \end{aligned} \tag{25}$$

from which the marginal probabilities of Y and Y' are identical and Eq. 22 is proved.

Funding No funding was received to assist with the preparation of this manuscript.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

References

1. Bruzzone L, Marconcini M (2010) Domain adaptation problems: a DASVM classification technique and a circular validation strategy. IEEE Trans Pattern Anal Mach Intell 32(5):770–787. https://doi.org/10.1109/TPAMI.2009.57
2. Gopalan R, Li R, Chellappa R (2014) Unsupervised adaptation across domain shifts by generating intermediate data representations. IEEE Trans Pattern Anal Mach Intell 36(11):2288–2302. https://doi.org/10.1109/TPAMI.2013.249
3. Zhang Y, Deng B, Tang H, Zhang L, Jia K (2020) Unsupervised multi-class domain adaptation: theory, algorithms, and practice. IEEE Trans Pattern Anal Mach Intell https://doi.org/10.1109/TPAMI.2020.3036956
4. Chen B, Lam W, Tsang IW, Wong TL (2013) Discovering low-rank shared concept space for adapting text mining models. IEEE Trans Pattern Anal Mach Intell 35(6):1284–1297. https://doi.org/10.1109/TPAMI.2012.243
5. Li J, Lu K, Huang Z, Zhu L, Shen HT (2019) Transfer independently together: a generalized framework for domain adaptation. IEEE Trans Cybern 49(6):2144–2155. https://doi.org/10.1109/TCYB.2018.2820174
6. Jiang M, Huang W, Huang Z, Yen GG (2017) Integration of global and local metrics for domain adaptation learning via dimensionality reduction. IEEE Trans Cybern 47(1):38–51. https://doi.org/10.1109/TCYB.2015.2502483
7. Pan SJ, Tsang IW, Kwok JT, Yang Q (2011) Domain adaptation via transfer component analysis. IEEE Trans Neural Netw 22(2):199–210. https://doi.org/10.1109/TNN.2010.2091281
8. Li L, Zhang Z (2019) Semi-supervised domain adaptation by covariance matching. IEEE Trans Pattern Anal Mach Intell 41(11):2724–2739. https://doi.org/10.1109/TPAMI.2018.2866846
9. Steinwart I, Hush D, Scovel C (2006) An explicit description of the reproducing kernel Hilbert spaces of Gaussian RBF kernels. IEEE Trans Inf Theory 52(10):4635–4643. https://doi.org/10.1109/TIT.2006.88171
10. Zhang Z, Wang M, Nehorai A (2020) Optimal transport in reproducing kernel Hilbert spaces: theory and applications. IEEE Trans Pattern Anal Mach Intell 42(7):1741–1754. https://doi.org/10.1109/TPAMI.2019.2903050
11. Deng WY, Lendasse A, Ong YS, Tsang IWH, Chen L, Zheng QH (2019) Domain adaption via feature selection on explicit feature map. IEEE Trans Neural Netw Learn Syst 30(4):1180–1190. https://doi.org/10.1109/TNNLS.2018.2863240
12. Feng Y, Yuan Y, Lu X (2021) Person reidentification via unsupervised cross-view metric learning. IEEE Trans Cybern 51(4):1849–1859. https://doi.org/10.1109/TCYB.2019.2909480
13. Tao D, Jin L, Wang Y, Li X (2015) Person reidentification by minimum classification error-based KISS metric learning. IEEE Trans Cybern 45(2):242–252. https://doi.org/10.1109/TCYB.2014.2323992
14. Gretton A, Borgwardt K, Rasch M, Schölkopf B, Smola A (2006) A kernel method for the two-sample-problem. Adv Neural Inf Process Syst 19:513–520
15. Sun B, Saenko K (2016) Deep coral: correlation alignment for deep domain adaptation. In: European conference on computer vision. Springer, pp 443–450
16. Pan SJ, Yang Q (2010) A survey on transfer learning. IEEE Trans Knowl Data Eng 22(10):1345–1359. https://doi.org/10.1109/TKDE.2009.191
17. Pan SJ, Tsang IW, Kwok JT, Yang Q (2011) Domain adaptation via transfer component analysis. IEEE Trans Neural Netw 22(2):199–210. https://doi.org/10.1109/TNN.2010.2091281
18. Yan K, Kou L, Zhang D (2018) Learning domain-invariant subspace using domain features and independence maximization. IEEE Trans Cybern 48(1):288–299. https://doi.org/10.1109/TCYB.2016.2633306
19. Li J, Jing M, Lu K, Zhu L, Shen HT (2019) Locality preserving joint transfer for domain adaptation. IEEE Trans Image Process 28(12):6103–6115. https://doi.org/10.1109/TIP.2019.2924174
20. Tsai YHH, Yeh YR, Wang YCF (2016) Learning cross-domain landmarks for heterogeneous domain adaptation. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR), pp 5081–5090
21. Li J, Lu K, Huang Z, Zhu L, Shen HT (2019) Heterogeneous domain adaptation through progressive alignment. IEEE Trans Neural Netw Learn Syst 30(5):1381–1391. https://doi.org/10.1109/TNNLS.2018.2868854
22. Tzeng E, Hoffman J, Zhang N, Saenko K, Darrell T (2014) Deep domain confusion: maximizing for domain invariance. arXiv:1412.3474
23. Long M, Cao Y, Wang J, Jordan M (2015) Learning transferable features with deep adaptation networks. In: International conference on machine learning. PMLR, pp 97–105
24. Liang J, Li L, Zhao C (2021) A transfer learning approach for compressed sensing in 6G-IoT. IEEE Internet Things J 8(20):15276–15283
25. Saitoh S, Sawano Y (eds) (2016) Theory of reproducing kernels and applications. Springer, Singapore
26. Gori F, Martínez-Herrero R (2021) Reproducing kernel Hilbert spaces for wave optics: tutorial. JOSA A 38(5):737–748
27. Yosida K et al (1965) Functional analysis. Springer, Berlin
28. Paulsen VI, Raghupathi M (2016) An introduction to the theory of reproducing kernel Hilbert spaces. Cambridge University Press, Cambridge
29. Fukunaga K (2013) Introduction to statistical pattern recognition. Elsevier, Amsterdam
30. Gong B, Shi Y, Sha F, Grauman K (2012) Geodesic flow kernel for unsupervised domain adaptation. In: IEEE conference on computer vision and pattern recognition, vol 2012. IEEE, pp 2066–2073
31. Wold S, Esbensen K, Geladi P (1987) Principal component analysis. Chemom Intell Lab Syst 2(1–3):37–52
32. Peterson LE (2009) K-nearest neighbor. Scholarpedia 4(2):1883
33. Bay H, Tuytelaars T, Van Gool L (2006) Surf: speeded up robust features. In: European conference on computer vision. Springer, pp 404–417