SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

September 2012 Vol. 55 No. 9: 1983–2007
doi: 10.1007/s11432-012-4611-x

A kernel learning framework for domain adaptation learning

TAO JianWen[1,3]*, CHUNG FuLai[2] & WANG ShiTong[1,2]

[1]*School of Digital Media, Jiangnan University, Wuxi* 214122*, China;*
[2]*Department of Computing, Hong Kong Polytechnic University, Hong Kong, China;*
[3]*School of Information Engineering, Zhejiang Business Technology Institute, Ningbo* 315012*, China*

Received February 5, 2012; accepted April 14, 2012; published online June 22, 2012

Abstract Domain adaptation learning (DAL) methods have shown promising results by utilizing labeled samples from the source (or auxiliary) domain(s) to learn a robust classifier for the target domain which has a few or even no labeled samples. However, there exist several key issues which need to be addressed in the state-of-the-art DAL methods such as sufficient and effective distribution discrepancy metric learning, effective kernel space learning, and multiple source domains transfer learning, etc. Aiming at the mentioned-above issues, in this paper, we propose a unified kernel learning framework for domain adaptation learning and its effective extension based on multiple kernel learning (MKL) schema, regularized by the proposed new minimum distribution distance metric criterion which minimizes both the distribution mean discrepancy and the distribution scatter discrepancy between source and target domains, into which many existing kernel methods (like support vector machine (SVM), v-SVM, and least-square SVM) can be readily incorporated. Our framework, referred to as kernel learning for domain adaptation learning (KLDAL), simultaneously learns an optimal kernel space and a robust classifier by minimizing both the structural risk functional and the distribution discrepancy between different domains. Moreover, we extend the framework KLDAL to multiple kernel learning framework referred to as MKLDAL. Under the KLDAL or MKLDAL framework, we also propose three effective formulations called KLDAL-SVM or MKLDAL-SVM with respect to SVM and its variant μ-KLDALSVM or μ-MKLDALSVM with respect to v-SVM, and KLDAL-LSSVM or MKLDAL-LSSVM with respect to the least-square SVM, respectively. Comprehensive experiments on real-world data sets verify the outperformed or comparable effectiveness of the proposed frameworks.

Keywords domain adaptation learning, support vector machine, multiple kernel learning, maximum mean discrepancy, maximum scatter discrepancy

Citation Tao J W, Chung F L, Wang S T. A kernel learning framework for domain adaptation learning. Sci China Inf Sci, 2012, 55: 1983–2007, doi: 10.1007/s11432-012-4611-x

1 Introduction

The conventional machine learning methods usually assume that the training and test data are drawn from identically and independently distribution (i.i.d.). Constructing mining and learning algorithms for data that may not be i.i.d. is one of the newly emergent research topics in data mining and machine learning [1,2]. For example, the key challenge of text classification is that accurately-labeled task-specific

*Corresponding author (email: jianwen_tao@yahoo.com.cn)

data are scarce while task-relevant data are abundant. In these cases, it is very expensive or even impossible to re-define the needed training data and reconstruct the learning models. Hence, it is very important and indispensable to reduce the need and effort to re-define the training data. Recently, there has been increasing research interest in developing new transfer learning (or domain adaptation) methods which can learn robust classifiers with a few or even no labeled patterns from the target domain by leveraging a large amount of labeled training data from other auxiliary domains. In domain adaptation learning (DAL) terminologies, one or more auxiliary domains are identified as the source domains of knowledge transfer, and the domain of interest is known as the target domain. DAL with non-i.i.d. data can help us construct more accurate learning models to perform new learning tasks in target domain for connecting samples to their true labels, thus simplifying the expensive data collection process for exploring the knowledge discovery process [3,4]. Domain adaptation has attracted more and more attention in the recent years [1–13].

In general, previous domain adaptation methods can be classified into two categories [8,4]: instance based methods and feature-based methods. Instance based methods assume a common relationship between the class label and samples and use weighting or sampling strategies to correct differences between training and testing distributions. In feature based methods, shared feature structure is learned in order to transfer knowledge from training data to testing data. Interested readers may refer to [8] for the more complete survey of DAL methods. However, there are several key issues which need to be addressed in the state-of-the-art DAL methods as follows.

1) Effective distribution discrepancy metric criterion. As we may know well, mean (or expectation) and variance (or scatter) are two main features characterizing the distribution of samples which measure order one and order two statistics, respectively. We claim that it is indispensable to consider both mean and variance (or scatter) of data distribution in order to efficiently measure the distribution discrepancy between source and target domains. Recently, from a novel feature reduction point of view, ref. [2] also pointed out that it is not enough to measure the distribution distance between two domains to some extent by only considering the mean of the distribution of samples. According to the aforementioned analysis, the state-of-the-art DAL methods based on maximum mean discrepancy (MMD) [14] (e.g. [1,13,15–17]) only focused on the order one statistic of the data distributions, instead of considering both the order one and order two statistics of the data distributions, simultaneously, thus limiting the generalization capacity of these models learned for specific domain adaptation learning problems to some extent.

2) Effective kernel learning. Most of the state-of-the-art DAL methods are either variants of SVM or in tandem with SVM or other kernel methods [13,18]. The prediction performances of these kernel methods heavily depend on the choice of the kernel. Unfortunately, the most suitable kernel for a particular task is often unknown in advance. Moreover, exhaustive search on a user-defined pool of kernels will be quite time-consuming when the size of the pool becomes large [18]. Hence, it is crucial to learn an appropriate kernel efficiently to make the performance of the employed kernel-based DAL method robust or even improved. Very recently, several multiple kernel learning (MKL) methods [19–22] have been proposed. However, these methods commonly assume that both training data and test data are drawn from the same domain. As a result, these MKL methods cannot learn the optimal kernel with the combined data from the source and target domains for the DAL problem. Therefore, the training data from the source domain may degrade the performance of MKL algorithms in the target domain. Nevertheless, as demonstrated in [13,18], in some constrained condition (e.g. distribution distance minimization between different domains), MKL can significantly improve the performance of DAL in some extent. Because of the central role of the kernel function in DAL as mentioned above, a good choice of the kernel is imperative to the success of DAL method. So, in this paper, we will borrow multiple kernel learning (MKL) as well as use a single, carefully selected kernel technique to the construction of our framework.

3) Multiple source domains transfer. As we may know, the brute-force domain transfer without the selection of source domains, some of which may be useless for DAL, may degrade the classification performance of DAL [16,17,23], which is a well-known open problem termed as negative transfer [23]. Hence, recently, several multiple source domain adaptation methods [16,17,23–27] were proposed to learn robust classifiers with training data from multiple source domains. As demonstrated in these previous

works, the DAL classifiers trained with the data from one or more source domains can effectively improve the learning performance in target domain. However, there is still no specific metric criterion used to minimize the scatter distribution mismatch between the source and target domains in these methods. Hence, in this paper, we will focus on the issues about both single and multiple source domains transfer, with respect to our proposed distribution distance metric criterion between domains.

Aiming at the mentioned-above issues, in this paper, we propose a unified kernel learning framework for domain adaptation learning (KLDAL) and its effective extension based on MKL framework. In some RKHS, KLDAL addresses the non-i.i.d. data learning problem by learning an optimal kernel and a low complexity decision function that well separates the domains data, regularized by both the complexity risk of the function and the distribution discrepancy between domains, measured by simultaneously considering both means and variances of the two domains. The idea is to in effect find an optimal kernel space in which the means and variances of the training and test data distributions are brought to be consistent, so that the labeled training data can be used to learn a model for the test data. In particular, we aim to obtain a linear kernel classifier based on the Representer Theorem [28], in an optimal reproducing kernel Hilbert space such that it achieves a trade-off between the maximal margin between classes and the minimal discrepancy between the training and test distributions. Hence, the main contributions of this paper include:

1) To deal with the considerable change between feature distributions of different domains, KLDAL learning an optimal kernel and a low complexity decision function that well separates the domain data, regularized by both the complexity risk of the function and the distribution discrepancy between domains, measured by simultaneously considering both means and variances of the two domains. In practice, KLDAL provides a unified framework to simultaneously learn an optimal kernel function as well as a robust classifier. Thus, many existing kernel methods, including SVM [29], v-SVM [30], TSVM [31], Least Squares SVM (LS-SVM) [32], and so on, can be incorporated into this framework to tackle DAL problems.

2) The distribution discrepancy of different domains can be tuned to diminish smoothly by introducing a tuned parameter γ controlling the kernel band width.

3) Using multiple kernel learning technique, we extend the proposed framework to multiple kernel learning framework for DAL, referred to as MKLDAL.

4) In addition, under the framework of KLDAL or MKLDAL, we also propose three effective formulations called KLDAL-SVM (or MKLDAL-SVM) with respect to SVM and its variant μ-KLDALSVM (or μ-MKLDALSVM) with respect to v-SVM, and KLDAL-LSSVM (or MKLDAL-LSSVM) with respect to the least-square SVM (LS-SVM), respectively.

2 Related works

The key idea of our method is to find a feature transform such that the distance between the testing and training data distributions, based on some distribution distance measure, is minimized, while at the same time maximizing a class separation distance or classification performance criterion for the training data. There has also been work describing how to measure the distance between distributions. One popular distribution distance measure is the Kullback-Leibler divergence [4], based on the entropy concept. However, in terms of our methods, we try to find a nonparametric method in a reproducing kernel Hilbert space (RKHS), which can efficiently compute its corresponding optimization formulation as well as obtain a satisfied distribution distance measure. One method is actually rooted at the maximum mean discrepancy (MMD) measure [14] that has recently been shown to be both efficient and effective for estimating the distance between two distributions in an RKHS. This measure is deduced from computing the distribution distance by finding a kernel function from a given class of kernel functions restricted to a unit ball in some RKHS. Additionally the particular form of this measure fits quite well into our learner formulation, as shown in next section. The basic idea of LMPROJ [1] is to minimize the distribution mean distance between source and target domain data by finding a feature translation in an RKHS based on empirical risk minimization principle, thus implementing transfer learning with cross-domains. Besides,

there have existed several other methods based on MMD, including DTSVM [13], DTMKL [18] TCA [2], FastDAM [16] and KMM [15].

Besides, very recently, Bruzzone et al. [4] propose the domain adaptation support vector machine (DASVM), which extends Transductive SVM (TSVM) to label unlabeled target samples progressively and simultaneously remove some auxiliary labeled samples. Cross-domain SVM (CD-SVM) proposed by Jiang et al. [33] uses the k-nearest neighbors from the target domain to define a weight for each auxiliary sample, and then the SVM classifier may be trained with the re-weighted auxiliary sample.

Instead of directly learning the kernel matrix, several efficient multiple kernel learning (MKL) methods [19–21] have been proposed to learn the kernel function in which the kernel function is assumed to be a linear combination of multiple predefined kernel functions (referred to as base kernel functions). And these methods simultaneously learn the decision function as well as the kernel. In practice, MKL has been successfully employed in many computer vision applications [13,18,22].

Recently, the work in [17] and its journal extension [16] also focus on the setting with multiple source domains and the domain adaptation machine (DAM) algorithm was specifically proposed for multiple source domain adaptation problems such as visual video detection. Besides, Yang et al. [34] proposed the adaptive support vector machine (A-SVM) to learn a new SVM classifier $f^t(x)$ for the target domain, which is adapted from an existing classifier $f^s(x)$ trained with the instances from multiple source domains.

Some researchers also theoretically studied the domain adaptation problem [5,11,35–37]. For more details on the theory of the domain adaptation problems, the interested readers may refer to [37].

All the methods stated above, including transfer learning and domain adaptation, are closely related to multi-task learning and may be viewed as a special case of semi-supervised learning where unlabeled samples are used to improve the learning of a decision function [1,4]. The difference exists in the fact that there is an assumed bias between training and testing samples in transfer learning [4]. A recent survey of semi-supervised learning can be found in [38]. A discussion of possible sample bias in a multi-task learning framework can be seen in [15].

3 RKHS embedding and metrics on probabilistic distribution

Kernel methods are broadly used as an effective way of constructing nonlinear algorithms from linear ones by embedding data sets into some higher dimensional reproducing kernel Hilbert spaces (RKHSs) [39]. A generalization of this idea is to embed probabilistic distributions into RKHS, giving us a linear method for dealing with higher order statistics [40,41]. Let a complete inner product space H of functions F, and for $g \in F$, $g : \boldsymbol{X} \to \boldsymbol{R}$, where $\boldsymbol{X}$ is a nonempty compact set. If the linear dot function mapping $g \to g(\boldsymbol{x})$ exists for all $\boldsymbol{x} \in \boldsymbol{X}$, we call H a reproducing kernel Hilbert space (RKHS). Under the aforementioned conditions, $g(x)$ can be denoted by an inner product: $g(\boldsymbol{x}) = \langle g, \varphi(\boldsymbol{x})\rangle_H$, where $\varphi : \boldsymbol{X} \to H$ denotes the feature space projection from $\boldsymbol{x}$ to H. And the inner product of the images of any points $\boldsymbol{x}$ and $\boldsymbol{x}'$ in feature space is called kernel $k(\boldsymbol{x}, \boldsymbol{x}') = \langle \varphi(\boldsymbol{x}), \varphi(\boldsymbol{x}')\rangle_H$. It is pointed out in [41] that the RKHS with Gaussian kernel is universal.

Definition 1 (Integral probability metric on RKHS embedding distributions [40])**.** Given the set Θ of all Borel probabilistic measures defined on the topological space M, and the RKHS (H,k) of functions on M with k as its reproducing kernel, for any $P \in \Theta$, denote $Pk := \int_M k(., x)\mathrm{d}P(\boldsymbol{x})$. If k is measurable and bounded, then we may define the embedding of P in H as $Pk \in H$. Then, the RKHS embedding distributions distance between two such mappings associated with $P, Q \in \Theta$ is defined as

$$\gamma_k(P, Q) = ||Pk - Qk||_H, \tag{1}$$

Gaussian kernel mapping can provide us an effective RKHS embedding skill for the consistency estimation of the probability distribution distance between different domains [40,41]. Hence, in the sequel, we adopt the Gaussian kernel function $k_\sigma(\boldsymbol{x}, \boldsymbol{z}) = \exp(-\frac{1}{2\sigma^2}||\boldsymbol{x} - \boldsymbol{z}||^2)$, where $\boldsymbol{x}, \boldsymbol{z} \in \boldsymbol{X}$, and σ denotes the kernel bandwidth, as the reproducing kernel in Hilbert space in this paper. It is worthy to note that instead of

using a fixed and parameterized kernel, one can also use a finite linear combination of kernels to compute γ_k.

Specifically, by Definition 1, we can have the following definition for domain adaptation learning problems.

Definition 2 (Projected mean distance metric on RKHS embedding domain distributions)**.** Let $p, q \in \Theta$ and linear function f: $f(x) = \langle w, \varphi(x)\rangle$, where w is a projection vector. Then the mean distance metric on RKHS embedding domain distributions is defined as

$$\gamma_{KM}(p,q)^2 = \left|\left| \int_{\boldsymbol{X}^s} f_{\boldsymbol{x}\sim p}(x)\mathrm{d}p - \int_{\boldsymbol{X}^t} f_{\boldsymbol{z}\sim q}(\boldsymbol{z})\mathrm{d}q \right|\right|^2 \triangleq \gamma_{KM}(p_n, q_m)^2, \tag{2}$$

where$x \in \boldsymbol{X}^s, \boldsymbol{z} \in \boldsymbol{X}^t$ and $\gamma_{KM}(p_n, q_m)$ is an empirical estimator of $\gamma_{KM}(p,q)$ defined as

$$\gamma_{KM}(p_n, q_m)^2 = \left|\left| \int_{\boldsymbol{X}^s} \boldsymbol{w}^{\mathrm{T}}\varphi(x)\mathrm{d}p_n - \int_{\boldsymbol{X}^t} \boldsymbol{w}^{\mathrm{T}}\varphi(\boldsymbol{z})\mathrm{d}q_m \right|\right|^2 \triangleq \boldsymbol{w}^{\mathrm{T}} \left|\left| \frac{1}{n}\sum_{i=1}^{n}\varphi(\boldsymbol{x}_i) - \frac{1}{m}\sum_{j=1}^{m}\varphi(\boldsymbol{z}_j) \right|\right|^2 \boldsymbol{w}, \tag{3}$$

where $\boldsymbol{x}_i \in \boldsymbol{X}^s$, $\boldsymbol{z}_j \in \boldsymbol{X}^t$.

Definition 3 (Projected scatter distance metric on RKHS embedding domain distributions)**.** Let $p, q \in \Theta$ and linear function f: $f(x) = \langle \boldsymbol{w}, \varphi(\boldsymbol{x})\rangle$, where w is a projection vector. Then the scatter distance metric on RKHS embedding domain distributions is defined as

$$\gamma_{KS}(p,q) = \left| \int_{\boldsymbol{X}^s} f_{x\sim p}(x) f_{\boldsymbol{x}\sim p}(x)^{\mathrm{T}}\mathrm{d}p - \int_{\boldsymbol{X}^t} f_{\boldsymbol{z}\sim q}(\boldsymbol{z}) f_{\boldsymbol{z}\sim q}(\boldsymbol{z})^{\mathrm{T}}\mathrm{d}q \right| \triangleq \gamma_{KS}(p_n, q_m), \tag{4}$$

where $\boldsymbol{x} \in \boldsymbol{X}^s, \boldsymbol{z} \in \boldsymbol{X}^t$ and $\gamma_{KS}(p_n, q_m)$ is an empirical estimator of $\gamma_{ks}(p,q)$ defined as

$$\gamma_{KS}(p_n, q_m) = \left| \int_{\boldsymbol{X}^s} \boldsymbol{w}^{\mathrm{T}}\varphi(\boldsymbol{x})\varphi(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{w}\mathrm{d}p_n - \int_{\boldsymbol{X}^t} \boldsymbol{w}^{\mathrm{T}}\varphi(\boldsymbol{z})\varphi(\boldsymbol{z})^{\mathrm{T}}\boldsymbol{w}\mathrm{d}q_m \right|. \tag{5}$$

Definition 4 (Distribution distance metric on RKHS embedding domain distributions)**.** Distribution distance metric on RKHS embedding domain distributions with probabilistic distribution $p, q \in P$ is defined as

$$\gamma_{KMS}(p,q) = (1-\lambda)\gamma_{KM} + \lambda\gamma_{KS} \triangleq (1-\lambda)\gamma_{KM}(p_n, q_m) + \lambda\gamma_{KS}(p_n, q_m), \tag{6}$$

where $\lambda \in [0,1]$ and when $\lambda = 0$, $\gamma_{KMS} = \gamma_{KM}$. The parameter λ is treated as a trade-off between probabilistic distribution mean and scatter (or variance). When λ increases, γ_{KMS} is biased in favour of preserving the distribution scatter consistency between both domainsand contrarily γ_{KMS} is biased in favour of preserving the distribution mean consistency between both domains. Hence, the proposed method can preserve both the distributions consistency between domains and the discriminative information in both domains

It can be guaranteed by the following theorem that the probabilistic distributions discrepancy between both domains can be measured sufficiently.

Theorem 1 (See [40])**.** Let F be a unit ball defined in some universal RKHS H with a kernel $k(\cdot,\cdot)$, which are all defined in a compact metric space. And let $\boldsymbol{X}$ be a compact subset in the metric space with Borel probability metrics p and q. Then $\gamma_{KMS}(F, p, q) = 0$ if and only if $p = q$.

4 Kernel learning framework for domains adaptation

4.1 Concepts and problem formulation

For a pattern classification problem, given a domain D with a distribution $P(x, y)$, $x \in \boldsymbol{X}$, $y \in \boldsymbol{Y}$, which is the true underlying distribution for the investigated classification problem, where $\boldsymbol{X}$ and $\boldsymbol{Y}$ denote all

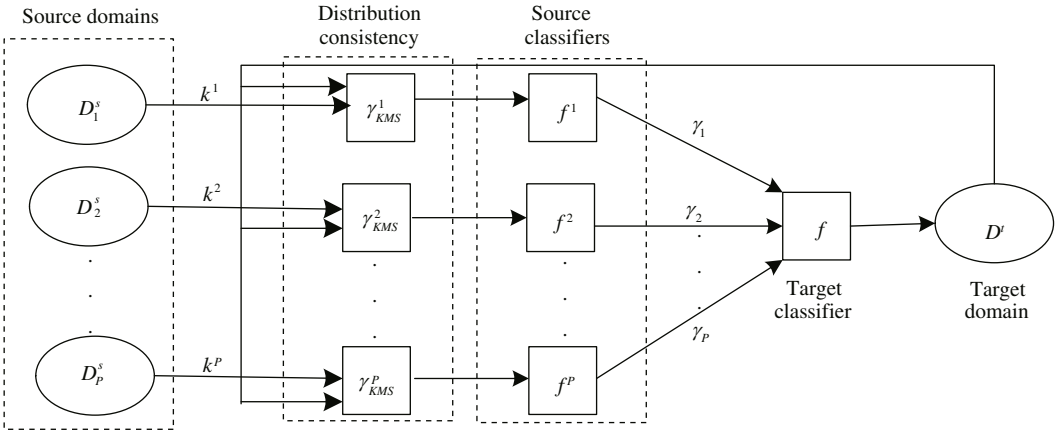

Figure 1 The kernel learning framework for domains adaptation.

possible instances and the corresponding class labels for the considered problem, respectively. A classifier is a function $f(\boldsymbol{x}) : \boldsymbol{X} \to \boldsymbol{Y}$ which maps data $\boldsymbol{x} \in \boldsymbol{X}$ to label set $\boldsymbol{Y}$. For DAL, unlabeled test patterns $\boldsymbol{X}^t = \{\boldsymbol{x}_i^t\}_{i=1}^m, \boldsymbol{X}^t \subset \boldsymbol{X}$ are drawn from a target domain D^t different from the source domain D^s of training samples $\boldsymbol{X}^s = \{(x_i^s, y_i^s)\}_i, x_i^s \in \boldsymbol{X}, y_i^s \in \boldsymbol{Y}$. This may happen when the available labeled data are out of date, whereas the test data are obtained from fast evolving information sources, or when series of data acquired at different times should be classified, while training samples collected only at once are available such as web query classification and web news classification tasks. In this sense, let $P^s(\boldsymbol{x}, y) = P^s(y|\boldsymbol{x}) \cdot P^s(\boldsymbol{x})$ and $P^t(\boldsymbol{x}, y) = P^t(y|\boldsymbol{x}) \cdot P^t(x)$ be the true underlying distributions for the source and target domains, respectively. The key idea is to reduce the distribution distance between $P^t(\boldsymbol{x}, y)$ and $P^s(\boldsymbol{x}, y)$ by some distribution transform technique. If $P^t(y|\boldsymbol{x})$ does not deviate a lot from $P^s(y|\boldsymbol{x})$, domain adaptation learning may become necessary. In the framework of domain adaptation, most of the learning methods are inspired by the idea that, although different, these two considered domains are highly correlated [1,3,4].

In this paper, we focus on the setting with one or multiple source domains, which is referred to as one or multiple source domains adaptation learning. Let us represent the instances from the target domain as $\boldsymbol{X}^t = (x_i^t, y_i^t)_{i=1}^m$, where y_i^t is the label of x_i^t. We also define $\boldsymbol{X}_p^s = (x_i^p, y_i^p)_{i=1}^n$ as the dataset from the pth source domain, where $p = 1, 2, \ldots, P$ and P is the total number of source domains. Also, we assume the dimension of each instance x to be d. In the sequel, the transpose of vector/matrix is denoted by the superscript T.

4.2 Proposed framework

The key goals of our framework are to find an optimal kernel space such that the mean and variance distances between the distributions of the testing and training data are minimized sufficiently, while at the same time maximizing the class margin or certain classification performance criterions for the training data, thus learning a robust classification model to effectively make prediction for target domain. Hence, our proposed Kernel learning framework for DAL (KLDAL) aims to find a linear ensemble decision function by employing one or multiple source domains in a universal reproduced kernel Hilbert space for the target domain

$$f(x) = \sum_{p=1}^{P} \gamma_p f^p(x), \tag{7}$$

as well as the kernel functions $k^p (1 \leqslant p \leqslant P)$ simultaneously (see Figure 1 for illustration), where $\gamma_p \in [0, 1]$ is the weight of each source classifier $f^p(x)$, which measures the distribution relevance between the pth source domain and the target domain, $\sum_{p=1}^P \gamma_p = 1$, $f^p(x) = (\boldsymbol{w}^p)^{\mathrm{T}} \varphi(\boldsymbol{x}^p) + b^p$, and $w^p = \sum_{i=1}^n \alpha_i^p \varphi(x_i^p)$ is a linear projection vector, b^p is the bias term and α_i^p's are the coefficients of the kernel

expansion for the decision function $f^p(x)$ using Representer Theorem [28]. In practice, great efforts have been made to minimize the distribution discrepancy between the pth source domain and the target domain and to reduce the empirical risk of the classification decision function as much as possible, thus implementing cross-domain learning. The proposed KLDAL, for the pth source domain, can then be formulated as

$$[k^p, f^p] = \arg\min_{k^p, f^p} \frac{1}{2}\|\boldsymbol{w}^p\|_{k^p}^2 + \gamma_{KMS}^p(p,q) + C\sum_{i=1}^{n} V(\boldsymbol{x}_i^p, y_i^p, f^p), \quad 1 \leqslant p \leqslant P, \tag{8}$$

where $\boldsymbol{x}_i^p \in \boldsymbol{X}_p^s$ is a set of training data, $y_i^p \in \boldsymbol{Y}_p^s$ is the class label corresponding to $\boldsymbol{x}_i^p$, V measures the fitness of the function in terms of predicting the class labels for the training data and is called the risk function, and$C$ is a trade-off parameter to balance the distribution discrepancy of two domains and the structural risk functional V.

In order to solve the primal in (8) effectively, we first introduce the following theorem.

Theorem 2. The primal of KLDAL in (8) can be reformulated as

$$[k^p, f^p] = \arg\min_{k^p, f^p} \frac{1}{2}(\boldsymbol{\beta}^p)^{\mathrm{T}}\Omega_1^p\boldsymbol{\beta}^p + C\sum_{i=1}^{n} V(\boldsymbol{x}_i^p, y_i^p, f^p), \tag{9}$$

where $\boldsymbol{x}_i^p \in \boldsymbol{X}_p^s$, $\boldsymbol{x}_j^p \in \boldsymbol{X}_p^s \cup \boldsymbol{X}^t$, $\boldsymbol{\Omega}_1^p \in \mathbb{R}^{(n+m)\times(n+m)}$ is a positive semi-definite kernel matrix.

Proof. Given a nonempty data matrix $\boldsymbol{X}_p = (\{\boldsymbol{x}_i^p\}_{i=1}^n, \{\boldsymbol{z}_j\}_{j=1}^m)$, $\boldsymbol{x}_i^p \in \boldsymbol{X}_p^s$, $\boldsymbol{z}_j \in \boldsymbol{X}_t$, let us consider a nonlinear function $\varphi : \boldsymbol{x} \to \varphi(x)$ mapping $\boldsymbol{x}$ in the primal input space into $\varphi(\boldsymbol{x})$ in the feature space. Then the data matrix X_p in the input space can be represented as $\varphi(\boldsymbol{X}_p) = (\{\varphi(\boldsymbol{x}_i^p)\}_{i=1}^n, \{\varphi(\boldsymbol{z}_j)\}_{j=1}^m)$ in the feature space. From the analysis of the normal weight vector $\boldsymbol{w}^p$ in the linear function $f^p(\boldsymbol{x}) = (\boldsymbol{w}^p)^{\mathrm{T}}\varphi(\boldsymbol{x})$ in the kernel space, we know that $\boldsymbol{w}^p$ is related to both source domain samples and target domain samples. In terms of Representer Theorem, $\boldsymbol{w}^p$ in the feature space can be formulated as

$$\boldsymbol{w}^p = \sum_{i=1}^{n}\beta_i^p\varphi(\boldsymbol{x}_i^p) + \sum_{j=1}^{m}\beta_j\varphi(\boldsymbol{z}_j),$$

where $\boldsymbol{\beta}^p = (\beta_1^p, \ldots, \beta_n^p, \beta_{n+1}, \ldots, \beta_{m+n})^{\mathrm{T}}$ denotes the weight vector. Hence $\boldsymbol{w}^p = \varphi(\boldsymbol{X}_p)\boldsymbol{\beta}^p$. Thereby,

$$\frac{1}{2}\|\boldsymbol{w}\|^2 = (\boldsymbol{\beta}^p)^{\mathrm{T}}(\varphi(\boldsymbol{X}_p))^{\mathrm{T}}\varphi(\boldsymbol{X}_p)\boldsymbol{\beta}^p = (\boldsymbol{\beta}^p)^{\mathrm{T}}\boldsymbol{K}_1^p\boldsymbol{\beta}^p, \tag{10}$$

where, $\boldsymbol{K}_1^p = \{k^p(x_i^p, x_j^p)\}_{i,j}^{n+m}$ is a $(n+m)\times(n+m)$ symmetrical positive semi-definite kernel matrix, and $x_i^p, x_j^p \in \boldsymbol{X}_p$. And Eq. (3) can be formulated as

$$\begin{aligned}\gamma_{KM}^p(p_n, q_m)^2 &= (\boldsymbol{w}^p)^{\mathrm{T}}\left\|\frac{1}{n}\sum_{i=1}^{n}\varphi(\boldsymbol{x}_i^p) - \frac{1}{m}\sum_{j=1}^{m}\varphi(\boldsymbol{z}_j)\right\|^2\boldsymbol{w}^p \\ &= \left\|\frac{1}{n}\sum_{j=1}^{n+m}(\beta_j^p)^{\mathrm{T}}\varphi(\boldsymbol{x}_j^p)^{\mathrm{T}}\sum_{i=1}^{n}\varphi(\boldsymbol{x}_i^p) - \frac{1}{m}\sum_{i=1}^{n+m}(\beta_i^p)^{\mathrm{T}}\varphi(\boldsymbol{x}_i^p)^{\mathrm{T}}\sum_{j=1}^{m}\varphi(\boldsymbol{x}_j)\right\|^2 = (\boldsymbol{\beta}^p)^{\mathrm{T}}\Sigma_1^p\boldsymbol{\beta}^p,\end{aligned} \tag{11}$$

where Σ_1^p is a $(n+m)\times(n+m)$symmetrical positive semi-definite kernel matrix defined as

$$\Sigma_1^p = \frac{1}{n^2}\boldsymbol{K}_s^p[1]^{n\times n}(\boldsymbol{K}_s^p)^{\mathrm{T}} + \frac{1}{m^2}\boldsymbol{K}_t[1]^{m\times m}\boldsymbol{K}_t^{\mathrm{T}} - \frac{1}{nm}(\boldsymbol{K}_s^p[1]^{n\times m}\boldsymbol{K}_t^{\mathrm{T}} + \boldsymbol{K}_t[1]^{m\times n}(\boldsymbol{K}_s^p)^{\mathrm{T}}), \tag{12}$$

where $\boldsymbol{K}_s^p$ is a $(n+m)\times n$ kernel matrix for the training data from the pth source domain, $\boldsymbol{K}_t$ is a $(n+m)\times m$ kernel matrix for the testing data from target domain, and $[\boldsymbol{1}]^{k\times l}$ is a $k\times l$ matrix of all ones.

By the same way, Eq. (5) can be further formulated as

$$\begin{aligned}
\gamma_{KS}^{p}(p_n, q_m) &= \left| \int_{\boldsymbol{X}_p^s} (w^p)^{\mathrm{T}} \varphi(\boldsymbol{x}^p) \varphi(\boldsymbol{x}^p)^{\mathrm{T}} w^p \mathrm{d}p_n - \int_{\boldsymbol{X}^t} (w^p)^{\mathrm{T}} \varphi(z) \varphi(\boldsymbol{z})^{\mathrm{T}} w^p \mathrm{d}q_m \right| \\
&= \left| \int_{\boldsymbol{X}_p^s} \sum_{j,k=1}^{n+m} (\beta_j^p)^{\mathrm{T}} \varphi(\boldsymbol{x}_j^p)^{\mathrm{T}} \varphi(\boldsymbol{x}^p) \varphi(\boldsymbol{x}^p)^{\mathrm{T}} \beta_k^p \varphi(\boldsymbol{x}_k^p) \mathrm{d}p_n \right. \\
&\quad \left. - \int_{\boldsymbol{X}^t} \sum_{j,k=1}^{n+m} (\beta_j^p)^{\mathrm{T}} \varphi(\boldsymbol{x}_j^p)^{\mathrm{T}} \varphi(\boldsymbol{z}) \varphi(\boldsymbol{z})^{\mathrm{T}} \beta_k^p \varphi(\boldsymbol{x}_k^p) \mathrm{d}q_m \right| \\
&= \left| \sum_{j,k=1}^{n+m} (\beta_j^p)^{\mathrm{T}} \beta_k^p \int_{\boldsymbol{X}_p^s} k_\sigma(\boldsymbol{x}_j^p, \boldsymbol{x}^p) k_\sigma(\boldsymbol{x}^p, \boldsymbol{x}_k^p) \mathrm{d}p_n - \sum_{j,k=1}^{n+m} (\beta_j^p)^{\mathrm{T}} \beta_k^p \int_{\boldsymbol{X}^t} k_\sigma(\boldsymbol{x}_j^p, \boldsymbol{z}) k_\sigma(\boldsymbol{z}, \boldsymbol{x}_k^p) \mathrm{d}q_m \right| \\
&\triangleq \left| \frac{1}{n} \sum_{j,k=1}^{n+m} (\beta_j^p)^{\mathrm{T}} \beta_k^p \sum_{i=1}^{n} k_\sigma(\boldsymbol{x}_j^p, \boldsymbol{x}_i^p) k_\sigma(\boldsymbol{x}_i^p, \boldsymbol{x}_k^p) - \frac{1}{m} \sum_{j,k=1}^{n+m} (\beta_j^p)^{\mathrm{T}} \beta_k^p \sum_{i=1}^{m} k_\sigma(\boldsymbol{x}_j^p, \boldsymbol{z}_i) k_\sigma(\boldsymbol{z}_i, \boldsymbol{x}_k^p) \right| \\
&= (\boldsymbol{\beta}^p)^{\mathrm{T}} \left| \frac{1}{n} \boldsymbol{K}_s^p (\boldsymbol{K}_s^p)^{\mathrm{T}} - \frac{1}{m} \boldsymbol{K}_t (\boldsymbol{K}_t)^{\mathrm{T}} \right| \boldsymbol{\beta}^p = (\boldsymbol{\beta}^p)^{\mathrm{T}} (\Sigma_1^p)' \boldsymbol{\beta}^p, \qquad (13)
\end{aligned}$$

where $(\Sigma_1^p)'$ is a $(n+m) \times (n+m)$ symmetrical positive semi-definite kernel matrix, which is defined as $(\Sigma_1^p)' = |\frac{1}{n} \boldsymbol{K}_s^p (\boldsymbol{K}_s^p)^{\mathrm{T}} - \frac{1}{m} \boldsymbol{K}_t (\boldsymbol{K}_t)^{\mathrm{T}}| \in \mathbb{R}^{(n+m)\times(n+m)}$. With Eqs. (10), (12), (13), let $\Omega_1^p = \boldsymbol{K}_1^p + (1 - \lambda)\Sigma_1^p + \lambda(\Sigma_1^p)'$. Then we can have the result of Theorem 2 with respect to Eq. (6) and Eq. (8).

By Eqs. (12) and (13), the proposed framework KLDAL measures the distribution discrepancy of cross-domains using several algebraic operations of kernel functions in the kernel space by mapping the input space into the kernel space, thus reducing the computational complexity of the distribution discrepancy in cross-domains to certain extent.

Without loss of generality, in the sequel, we only focus on the case of the pth source domain transfer learning ($1 \leqslant p \leqslant P$), and also for the simplicity, we omit all the marks p in the aforementioned equations. Finally, we can readily ensemble P source classifiers into the final target classifier for target domain learning.

4.3 KLDAL using hinge loss

The hinge loss function is a commonly used risk function in the form of $V = (1 - y_i f(x_i))_+$ [42] in which $(x)_+ = x$ if $x \geqslant 0$ and zero otherwise. Then, the structural risk functional becomes classical SVM, which is the first formulation coined as KLDAL-SVM in this paper. Thus, according to the formulation of primal SVM, (9) can be reformulated as

$$\min_{k,\beta,\xi} \frac{1}{2} \boldsymbol{\beta}^{\mathrm{T}} \Omega_1 \beta + C \sum_{i=1}^{n} \xi_i. \qquad (14)$$

Then, the corresponding constrained optimization problem in (6) can be rewritten as

$$\begin{aligned}
&\min_{k,\beta,b,\xi} \tfrac{1}{2} \beta^{\mathrm{T}} \Omega_1 \boldsymbol{\beta} + C \sum_{i=1}^{n} \xi_i, \\
&\text{s.t.} \quad y_i \Big(\sum_{j=1}^{n+m} \beta_j k_\sigma(x_i, x_j) + b \Big) \geqslant 1 - \xi_i, \ i = 1, \ldots, n,
\end{aligned} \qquad (15)$$

where, $x_i \in \boldsymbol{X}^s$.

Theorem 3. The dual of the primal in Eq. (15) can be formulated as

$$\min_{\alpha} \frac{1}{2} \boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{H}^{\varphi} \boldsymbol{\alpha} - 1^{\mathrm{T}} \boldsymbol{\alpha},$$

$$\text{s.t.} \quad 0 \leqslant \alpha_i \leqslant C, \quad i = 1, \ldots, n, \quad \sum_{i=1}^{n} \alpha_i y_i = 0,$$

where $\boldsymbol{H}^{\varphi} = \widetilde{\boldsymbol{Y}}\boldsymbol{K}_s^{\mathrm{T}}(\Omega_1)^{-1}\boldsymbol{K}_s\widetilde{\boldsymbol{Y}}$, and $\widetilde{\boldsymbol{Y}} = \mathrm{diag}(y_1, \ldots, y_n)$, $y_i \in \boldsymbol{Y}^s$.

Proof. We can obtain the result by the same way of Theorem 1 in [43]. More details can be seen in [43].

By the same way of the classical SVM, the biased variable b^{ϕ} in the kernel space can be formulated as

$$b^{\varphi} = -\frac{1}{2}\Big(\frac{1}{|\boldsymbol{X}_{s+}|}\sum_{x\in\boldsymbol{X}_{s+}}\sum_{j=1}^{n+m}\beta_j k_\sigma(\boldsymbol{x}_j, \boldsymbol{x}) + \frac{1}{|\boldsymbol{X}_{s-}|}\sum_{\boldsymbol{x}\in\boldsymbol{X}_{s-}}\sum_{j=1}^{n+m}\beta_j k_\sigma(\boldsymbol{x}_j, \boldsymbol{x})\Big).$$

As we know, the v-support vector machine (v-SVM) [30] is a typical variant of primal SVM for classification in which Schölkopf et al. introduced a new parameter v instead of C in SVM to control the number of support vectors and the training errors. Hence, as a variant of KLDAL-SVM based on v-SVM, μ-KLDALSVM can be formulated as

$$\min_{\beta,\xi,b} f = \frac{1}{2}\boldsymbol{\beta}^{\mathrm{T}}\Omega_1\beta - \mu\rho + \frac{1}{N}\sum_{i=1}^{n}\xi_i, \tag{16}$$

$$\text{s.t.} \quad y_i\Big(\sum_{j=1}^{N}\beta_j k_\sigma(\boldsymbol{x}_i, \boldsymbol{x}_j) + b\Big) \geqslant \rho - \xi_i, \quad i = 1, \ldots, n, \tag{17}$$

where the variables $N = n + m$, $\rho \geqslant 0, \mu > 0$ and $\xi_i \geqslant 0$ have the same meaning as in v-SVM. Similar to v-SVM, the dual of the primal in Eq. (16)–Eq. (17) can be formulated as

$$\min_{\alpha} \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{H}^{\varphi}\boldsymbol{\alpha}$$

$$\text{s.t.} \quad 0 \leqslant \alpha_i \leqslant \frac{1}{N} i = 1, \ldots, n, \quad \sum_{i=1}^{n}\alpha_i y_i = 0, \quad \sum_{i=1}^{n}\alpha_i \geqslant \mu,$$

where $\boldsymbol{H}^{\varphi} = \widetilde{\boldsymbol{Y}}\boldsymbol{K}_s^{\mathrm{T}}(\Omega_1)^{-1}\boldsymbol{K}_s\widetilde{\boldsymbol{Y}}$, and $\widetilde{\boldsymbol{Y}} = \mathrm{diag}(y_1, \ldots, y_n)$, $y_i \in \boldsymbol{Y}^s$.

It is worthy to note that the significance of μ in μ-KLDALSVM is similar to that of v in v-SVM. Compared with the dual of KLDAL-SVM, the dual of μ-KLDALSVM has two differences. First, there is an additional constraint $\sum_{i=1}^{n}\alpha_i \geqslant \mu$. Second, the linear term $\sum_{i=1}^{n}\alpha_i$ no longer appears in the dual of μ-KLDALSVM.

4.4 KLDAL using least square loss

In this subsection, we propose another formulation coined as KLDAL-LSSVM inspired by the idea of LS-SVM [32], which can be formulated as

$$\arg\min_{k,f} \gamma_{KMS}(p, q) + \frac{C}{2}\sum_{i=1}^{n}\xi_i^2. \tag{18}$$

Along the same line of KLDAL-SVM, the constrained problem formulation of Eq. (18) is defined as

$$\min_{\beta,\xi,b} f = \frac{1}{2}\boldsymbol{\beta}^{\mathrm{T}}\Omega_1\beta + \frac{C}{2}\sum_{i=1}^{n}\xi_i^2 \tag{19}$$

$$\text{s.t.} \quad \sum_{j=1}^{n+m}\beta_j k_\sigma(\boldsymbol{x}_i, \boldsymbol{x}_j) + b = y_i - \xi_i, \ i = 1, \ldots, n. \tag{20}$$

Theorem 4 (Analytic solution to binary class)**.** Given parameter $\lambda \in [0, 1]$, for a binary classification problem, the optimal solution of Eqs. (19) and (20) is equivalent to the linear system of equations with respect to variable $\boldsymbol{\alpha}$ as follow:

$$\begin{bmatrix} 0 & \mathbf{1}_n^T \\ \mathbf{1}_n & \widetilde{\Omega} \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{Y}^s \end{bmatrix}. \tag{21}$$

Proof. The Lagrange equation of the optimal problem in Eqs. (19) and (20) can be formulated as

$$L_\lambda(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) = \frac{C}{2}\sum_{i=1}^{n}\xi_i^2 + \frac{1}{2}\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\Omega}_1\boldsymbol{\beta} - \sum_{i=1}^{n}\alpha_i\Big(\sum_{j=1}^{n+m}\beta_j k_\sigma(\boldsymbol{x}_j, \boldsymbol{x}_i) + b + \xi_i - y_i\Big), \tag{22}$$

where α_i is Lagrange multiplier. We compute the corresponding partial derivatives with respect to optimal variables respectively and set them to 0, thus eliminating variables $\boldsymbol{\beta}$ and ξ_i. We have

$$\begin{bmatrix} 0 & \mathbf{1}_n^{\mathrm{T}} \\ \mathbf{1}_n & \widetilde{\boldsymbol{\Omega}} \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{Y}^s \end{bmatrix}, \tag{23}$$

where $\mathbf{1}_n = [1, \ldots, 1]^{\mathrm{T}}$, $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_n]^{\mathrm{T}}$, $\boldsymbol{Y}^s = [y_1, \ldots, y_n]^{\mathrm{T}}\widetilde{\boldsymbol{\Omega}} = \boldsymbol{K}_s^{\mathrm{T}}(\boldsymbol{\Omega}_1)^{-1}\boldsymbol{K}_s + \frac{\boldsymbol{I}_n}{C}$, $\boldsymbol{I}_n$ is an n-dimensional identity matrix.

As for multi-class classification problems, the traditional skills are to separate a multi-class classification problem into several binary classification problems in OAO (one against one) or OAA (one against all) way. However, the main drawbacks of these skills are high computational complexity and imbalance between classes. Hence, here we introduce the vector labeled outputs into the solution of KLDAL-LSSVM, which can make the corresponding computational complexity independent of the number of classes and require no more computation than a single binary classifier [44]. Furthermore, Szedmak and Shawe-Taylor [44] pointed out that this technique does not reduce the classification performance of a learning model but in some cases can improve it, with respect to OAO and OAA. Therefore, we represent the class labels according to the one-of-$c$rule, namely, if training sample x_i $(i = 1, \ldots, n)$ belongs to the kth class, then the class labels of x_i are

$$Y_i = [\underbrace{0, \ldots, 1}_{k}, \ldots, 0]^{\mathrm{T}} \in \mathbb{R}^c,$$

where the kth element is 1 and all the other elements are 0. Hence, for some multi-class classification problem, the optimal problem of KLDAL-LSSVM can be formulated as

$$\min_{\boldsymbol{\beta}, \boldsymbol{\xi}, b} f = \frac{1}{2}\widetilde{\boldsymbol{\beta}}^{\mathrm{T}}\boldsymbol{\Omega}_1\widetilde{\boldsymbol{\beta}} + \frac{C}{2}\sum_{i=1}^{n}\xi_i^2, \quad \text{s.t.} \quad \widetilde{\boldsymbol{\beta}}^{\mathrm{T}}\boldsymbol{K}_s + b = \boldsymbol{Y}_i - \xi_i, i = 1, \ldots, n, \tag{24}$$

where $\widetilde{\boldsymbol{\beta}} \in \mathbb{R}^{n\times c}$, $b \in \mathbb{R}^c$.

Theorem 5 (Analytic solution to multi-class)**.** Given parameter $\lambda \in [0, 1]$, for a multi-class classification problem, the optimal solution of Eq. (24) is equivalent to the linear system of the following equation:

$$\begin{bmatrix} b & \boldsymbol{\alpha} \end{bmatrix} \begin{bmatrix} 0 & \mathbf{1}_n^{\mathrm{T}} \\ \mathbf{1}_n & \widetilde{\Omega} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_c & \widetilde{\boldsymbol{Y}^s} \end{bmatrix}, \tag{25}$$

where, $\mathbf{0}_c = [0, \ldots, 0]^{\mathrm{T}}$, $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1, \ldots, \alpha_n]^{\mathrm{T}}$, $\widetilde{\boldsymbol{Y}^s} = [\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_n]^{\mathrm{T}}$, $\widetilde{\Omega}$ is the same as in Theorem 7.

Proof. The procedures of this proof are the same as in Theorem 4.

Theorems 4 and 5 actually provide us the KLDAL-LSSVM versions for both binary and multi-class classification problems, respectively. It is clearly shown from Eqs. (21) and (25) that KLDAL-LSSVM keeps the same solution framework for both binary and multi-class cases.

5 Extension: multiple kernel learning framework for DAL

Multiple kernel learning (MKL) refers to the process of learning a kernel machine with multiple kernel functions or kernel matrices. Recent research efforts on MKL have shown that learning SVMs with multiple kernels not only increases the accuracy but also enhances the interpretability of the resulting classifiers [13,18,22]. Our MKL formulation is to find an optimal way to linearly combine the given kernels. Suppose we have a set of base kernel functions $\{k_h\}_{h=1}^M$ (or base kernel matrices $\{\boldsymbol{K}_h\}_{h=1}^M$). An ensemble kernel function k (or ensemble kernel matrix $\boldsymbol{K}$) is then defined by $k(x_i,x_j)=\sum_{h=1}^M \beta_h k_h(x_i,x_j), \beta_h \geqslant 0$ (or $\boldsymbol{K}=\sum_{h=1}^M \beta_h \boldsymbol{K}_h, \beta_h \geqslant 0$). Consequently, an often-used MKL model from binary-class data $\{(x_i, y_i \in \pm 1)\}_{i=1}^n$ is

$$f(x)=\sum_{i=1}^n \alpha_i y_i k(x_i,x)+b=\sum_{i=1}^n \alpha_i y_i \sum_{h=1}^M \beta_h k_h(x_i,x)+b.$$

Optimizing over both the coefficients $\{\alpha_i\}_{i=1}^n$ and $\{\beta_h\}_{h=1}^M$ is one particular form of the MKL problems. Our framework KLDAL utilizes such an MKL optimization to yield a more flexible DAL scheme, referred to as MKLDAL, in which an appropriate kernel in the form of a convex combination of some given kernels can be automatically determined during the optimization process.

Let us consider the use of a convex combination of M kernels $\boldsymbol{K}_1, \boldsymbol{K}_2, \ldots, \boldsymbol{K}_M$, with the corresponding kernel-induced feature maps $\phi_1, \phi_2, \ldots, \phi_M$ [20,22]. Then, the multiple kernel matrix $\boldsymbol{K}_M^s$ for the data from source domain, $\boldsymbol{K}_M^t$ for the data from target domain and $\boldsymbol{K}_M$ for the data from both source and target domains can be three convex combinations of M kernels $\{\boldsymbol{K}_h^s\}_{h=1}^M$, $\{\boldsymbol{K}_h^t\}_{h=1}^M$ and $\{\boldsymbol{K}_h\}_{h=1}^M$, respectively, where $\boldsymbol{K}_h^s$, $\boldsymbol{K}_h^t$ and $\boldsymbol{K}_h$ are defined as the same as $\boldsymbol{K}_s$, $\boldsymbol{K}_t$ and $\boldsymbol{K}_1$ in the framework KLDAL, respectively. Using the MKL formulation in [20] and [21], we can have $\boldsymbol{K}_M^s=\sum_{h=1}^M \mu_h \boldsymbol{K}_h^s$, $\boldsymbol{K}_M^t=\sum_{h=1}^M \mu_h \boldsymbol{K}_h^t$, and $\boldsymbol{K}_M=\sum_{h=1}^M \mu_h \boldsymbol{K}_h$, $\mu_h \geqslant 0$, $\sum_{i=1}^M \mu_i=1$. Hence, the single kernel learning framework for DAL in (9) can be extended to

$$[k,f]=\arg\min_{k,f} \frac{1}{2}\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\Omega}_M\beta+C\sum_{i=1}^n V(x_i,y_i,f), \tag{26}$$

where, $\boldsymbol{\Omega}_M=\boldsymbol{K}_M+(1-\lambda)\boldsymbol{\Sigma}_M+\lambda\boldsymbol{\Sigma}'_M$, $\Sigma_M=\frac{1}{n^2}\boldsymbol{K}_M^s[1]^{n\times n}(\boldsymbol{K}_M^s)^{\mathrm{T}}+\frac{1}{m^2}\boldsymbol{K}_M^t[1]^{m\times m}(\boldsymbol{K}_M^t)^{\mathrm{T}}-\frac{1}{nm}(\boldsymbol{K}_M^s[1]^{n\times m}(\boldsymbol{K}_M^t)^{\mathrm{T}}+\boldsymbol{K}_M^t[1]^{m\times n}(\boldsymbol{K}_M^s)^{\mathrm{T}})$, and $\boldsymbol{\Sigma}'_M=\left|\frac{1}{n}\boldsymbol{K}_M^s(\boldsymbol{K}_M^s)^{\mathrm{T}}-\frac{1}{m}\boldsymbol{K}_M^t(\boldsymbol{K}_M^t)^{\mathrm{T}}\right|$. By Eq. (26), when $M=1$, MKLDAL degrades to KLDAL. Hence, KLDAL is a special case of MKLDAL.

Thus, the constrained formulations based on Hinge loss and least square loss, referred to as MKLDAL-SVM (or μ-MKLDALSVM) and MKLDAL-LSSVM respectively, can be respectively formulated using MKL as follows.

$$\begin{array}{ll}
\min\limits_{k,\beta,b,\xi} \frac{1}{2}\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\Omega}_M\beta+C\sum\limits_{i=1}^n \xi_i \quad \text{s.t.} & \min\limits_{k,\beta,b,\xi} \frac{1}{2}\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\Omega}_M\beta-\mu\rho+C\sum\limits_{i=1}^n \xi_i \quad \text{s.t.} \\
y_i\big(\sum\limits_{j=1}^{n+m}\beta_j\sum\limits_{l=1}^M \mu_l k_l(\boldsymbol{x}_j,\boldsymbol{x}_i)+b\big)\geqslant 1-\xi_i, \quad \text{or} & y_i\big(\sum\limits_{j=1}^{n+m}\beta_j\sum\limits_{l=1}^M \mu_l k_l(\boldsymbol{x}_j,\boldsymbol{x}_i)+b\big)\geqslant \rho-\xi_i, i=1,\ldots,n, \\
\mu_l\geqslant 0, \quad \sum\limits_{l=1}^M \mu_l=1 & \mu_l\geqslant 0, \quad \sum\limits_{l=1}^M \mu_l=1
\end{array} \tag{27}$$

$$\begin{array}{ll}
\min\limits_{\boldsymbol{\beta},\boldsymbol{\xi},b} f=\frac{1}{2}\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\Omega}_M\beta+\frac{C}{2}\sum\limits_{i=1}^n \xi_i^2 \quad \text{s.t.} & \sum\limits_{j=1}^{n+m}\beta_j\sum\limits_{l=1}^M \mu_l k_l^{(}\boldsymbol{x}_j,\boldsymbol{x}_i)+b=y_i-\xi_i,\ i=1,\ldots,n, \\
& \mu_l\geqslant 0, \quad \sum\limits_{l=1}^M \mu_l=1
\end{array} \tag{28}$$

where, $\boldsymbol{K}_l$ is an $(n+m)\times n$ kernel matrix for the training data from source domain and $[\boldsymbol{K}_l]_{ji}=k_l(x_j,x_i)$,$x_i\in X_s$, and $x_j\in X_s\cup X_t$.

From Theorem 3, we can easily derive the dual formulations of (27) and (28). Due to the space limitation, we omit the procedure of the derivation. Besides, Eqs. (27) and (28) have a similar optimization form as described in [20]; thus they can be straightforwardly solved using the existing MKL solver software packages like SimpleMKL [20]. More details can be seen in [20,22].

6 Discussion

6.1 Singular matrix problem

It is worthwhile to note that the matrix $\Omega_M (M \geqslant 1)$ in the proposed framework aforementioned may possibly be singular, i.e., the so-called singular matrix problem. If this case happens, the inverse matrix of Ω_M can not be obtained, and thus the proposed algorithm becomes unfeasible. Recently, there have been several feasible techniques proposed to solve the singular matrix problem. The popular ones include singular value decomposition (SVD), QR-decomposition and principle component analysis (PCA), and so on. However, the main drawback of these techniques is their high computational complexities. Hence, for the so-called singular matrix problem, in order not to increase the computational cost of the proposed algorithm, we only regularize matrix Ω_M with a small identity matrix with the same dimension as $\Omega_M = \boldsymbol{K}_M + (1-\lambda)\Sigma_M + \lambda\Sigma'_M + \lambda_0 \boldsymbol{I}$, where $\lambda_0 \geqslant 0$ is a tuned parameter and $\boldsymbol{I}$ is an $(n+m)\times(n+m)$ unit matrix.

6.2 On the kernel bandwidth

Theorem 6 (See [41]). Given a class of Gaussian kernel functions $K_g = \{e^{-||x-z||_2^2/2\sigma^2}, \boldsymbol{x}, \boldsymbol{z} \in \mathbb{R}^d : \sigma \in [\sigma_0, \infty)\}$, where $\sigma_0 > 0$, for any $k_\sigma, k_\tau \in K_g$ and $0 < \tau < \sigma < \infty$, $\gamma_{k\sigma}(P,Q) \geqslant \gamma_{k\tau}(P,Q)$.

Theorem 6 shows that the larger the kernel bandwidth is, the larger the distance of RKHS embedding domain distributions will become, thus decreasing the convergence rate of KLDAL (or MKLDAL). In order to investigate the performance influence of the kernel bandwidth on KLDAL (or MKLDAL), we parameterize the Gaussian kernel bandwidth, namely, the Gaussian kernel function is generalized as

$$k_{\sigma/\gamma}(x, x_i) = \exp\left(-\frac{||x - x_i||^2}{2(\sigma/\gamma)^2}\right), \tag{29}$$

where γ is a tunable parameter. In terms of the following experimental results, as γ increases, samples in intra-domain exhibit strong cohesion, thus leading to the overlapping of samples from different classes to some extent, which makes against the pattern classification. On the other hand, as γ decreases, the convergence rate of KLDAL (or MKLDAL) may decrease to some extent. Hence, we constrain the parameter γ as $\gamma \in [1, \gamma_0]$, where γ_0 is a tunable threshold. The matrix $\Omega_M (M \geqslant 1)$ can be reformulated as

$$\widetilde{\Omega} = \boldsymbol{\Omega}_M^{(\sigma/\gamma)}, \tag{30}$$

where $\omega_M^{(\eta)}$ denotes the kernel matrix ω_M with kernel bandwidth η. Eq. (30) shows that the kernel matrix Ω_1 (or Ω_M) in KLDAL (or MKLDAL) can be tuned by parameter γ, thus further improving the adaptation capacity of the proposed frameworks.

According to the above analysis, the scatter distance metric on RKHS embedding domain distributions can preserve the scatter consistency of both domains and reduce the scatter in intra-domain in certain range of kernel bandwidth as well, thus accelerating the convergence rate of the proposed algorithm and improving the effectiveness of the proposed methods.

6.3 Comparisons with related works

As for distribution distance measure for cross-domain learning, most existing cross-domain learning algorithms (e.g. [3], [4], [6], [7], [33], and [34], etc.) do not explicitly consider any specific criterion in measuring the distribution mismatch of samples between different domains. Even though explicitly considering some distribution measure criterions, several methods, such as [1], [13], [15–18], [35], [41], etc., just consider minimizing the distribution mean mismatch between the source and target domains. However, to be different with the methods aforementioned above, for DAL problems, we propose a novel distance metric criterion, inspired by the idea of MMD but essentially different from that of MMD, on RKHS embedding domain distributions, by simultaneously minimizing both distribution mean and scatter.

Particularly, three recently proposed DAL methods LMPROJ [1], DTSVM [13], A-SVM [34], FastDAM [16], and DASVM [4] are the most related with our frameworks based methods. To be essentially different from these existing methods, our methods are a general (multiple) kernel learning framework based on the proposed new distribution discrepancy metric criterion, originally generalized from MMD by taking both maximum mean discrepancy and maximum scatter discrepancy measure in an optimal RKHS. Thus, our methods, based on our new (multiple) kernel learning framework, significantly outperform those methods based on MMD (e.g. LMPROJ and DTSVM) for DAL problems, which can be confirmed by the following experiments. Besides, our kernel learning frameworks are also essentially different from DASVM with respect to the formulations of them. As mentioned in Section 2, DASVM is a progressive DAL method, which adopts three steps to iteratively learn an optimal classifier for target domain. One main drawback of DASVM is that the initial SVM of DASVM completely ignores the distribution discrepancy between different domains, which may degrade the overall classification performance of DASVM when the distribution distance of different domains is relatively large. However, our methods, based on a unified (multiple) kernel learning framework, try to learn an optimal SVM classifier for source domains just in one step by considering both the distribution mean and the distribution scatter discrepancy between two domains, simultaneously, thus implementing cross-domain learning for target domain.

In summary, in contrast to the state-of-the-art methods mentioned above, our frameworks are unified cross-domain kernel learning framework, in which the robust kernel classifier is learned in some RKHS by explicitly minimizing both mean and scatter distribution mismatch between the source and target domains by using labeled patterns from source domains and target domain. Most importantly, three kernel learning machines (e.g., SVM, v-SVM, and LS-SVM) have been readily embedded into our frameworks to solve several DAL problems.

7 Experiments

To evaluate the effectiveness of the proposed framework KLDAL and its extension MKLDAL for domain adaptation learning problems, we systematically compare them with several state-of-the-art algorithms on several domain adaptation related applications: 1) document retrieval, 2) face recognition, and 3) video concept detection. For all of the data sets, true labels are available, for instances, from source domains. All of labeled samples from source domain and unlabeled samples from target domain are selected as training data and testing data, respectively, except in the trials on TRECVID dataset, in which a few labeled samples from target will also be randomly extracted as training data.

We construct two small-scale text datasets trials to show the single and multiple source domains adaptation learning performance and parameters influence of the methods KLDAL-SVM and μ-KLDALSVM under the condition that the proposed techniques preserve the consistency of both distribution mean and variance between source and target domains. Moreover, we design several single or multiple source cross-domain face classification tasks to show the outperformed performance of the proposed method KLDAL-LSSVM in the area of multi-class classification problems with high dimensional data. Last but not the least, we also construct a large-scale visual video concepts detection tasks on TRECVID datasets to further evaluate the domain adaptation learning performance of the method μ-MKLDALSVM based on the proposed multiple kernel framework MKLDAL.

Throughout this experimental part, we also use standard Gaussian kernel function as $k_\theta(\boldsymbol{x}, \boldsymbol{z}) = \exp(-\gamma||\boldsymbol{x} - \boldsymbol{z}||^2)$ for several related kernel methods such as SVM, TSVM, DASVM, KMM, LMPROJ, A-SVM, Multi-KMM [48] and FastDAM [16], where γ is set to $1/d$ (d is the feature dimension). For multiple kernel learning in DTSVM, according to the setting in [13], we use kernel parameters as $1.2^\delta\gamma$, where δ is set to $\{0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5\}$ for the Gaussian kernel function, thus constructing eight base kernels for DTSVM.

For our KLDAL based methods, we use the parameterized Gaussian kernel as $k_{\sigma/\gamma}(x, x_i) = \exp(-\frac{\|x-x_i\|^2}{2(\sigma/\gamma)^2})$, where the kernel parameter σ can be obtained by minimizing MMD to get the most conservative test, which follows the setting in [14]. Empirically, we first select σ as the square root of the mean norm of the training data for binary classification and $\sigma\sqrt{c}$ (where c is the number of classes) for

Table 1 Description of the cross domain text datasets on single source domain

	Task	Data sets	Number of training samples		Number of testing samples	
			Positive class	Negative class	Positive class	Negative class
1	20NG	Comp vs. Sci	1958	1972	2923	1977
2		Rec vs. Talk	1993	1568	1984	1658
3		Rec vs. Sci	1984	1977	1993	1972
4		Sci vs. Talk	1971	1403	1978	1850
5		Comp vs. Rec	2916	1993	1965	1984
6		Comp vs. Talk	2914	1568	1967	1685
7	Email spam	User1 vs. User2	User1's emails		User2's emails	
8		User2 vs. User3	User2's emails		User3's emails	
9		User3 vs. User1	User3's emails		User1's emails	

multi-class classification. For our MKLDAL based methods, for fair comparison, we also use kernel parameters as $1.2^{\delta}\sigma^2$ according to the setting in [45], where δ is set the same as in DTSVM, for the parameterized Gaussian kernel function, thus constructing eight base kernels for our MKLDAL-based methods.

In the context, SVMs (such as SVM or v-SVM, and TSVM) is implemented by the state-of-the-art software package such as LIBSVM [13] and the other algorithms are implemented by MATLAB 2009b with respect to LIBSVM and SimpleMKL software package [20].

7.1 Experiments on real-world small-scale datasets

In this subsection, we demonstrate the efficiency and effectiveness of the proposed methods under the framework KLDAL on two different classes of real-world domain adaptation tasks. The first class of tasks is the cross-domain text classification on the 20Newsgroups and email spam. The second one is a multi-class domain adaptation learning problem in YALE and ORL face databases. Tables 1 and 2 summarize the text datasets and give the indices to some of which we will refer in our experimental results.

7.1.1 *Description of data sets*

1) 20Newsgroups dataset. The 20Newsgroups (20NG) dataset[1)] contains 18774 documents, and has a hierarchical structure with 6 main categories and 20 subcategories. Each set of sub-categories represents a different domain in which different words will be more common. Features are given by converting the documents into bag-of-word representations which are then transformed into feature vectors using the term frequency. For more details about the sub-categories, see [46]. Table 1 shows some more detailed information about the experimental datasets drawn from 20NG.

Besides, we also choose the instances from three main categories with at least four subcategories and generate three settings for evaluating multiple source domain adaptation algorithms. For each setting, we consider one main category as the positive class and use another one as the negative class, and employ all the labeled instances from two subcategories (i.e., one from the positive class and the other from the negative class) to construct one domain. In the experiments, we have three source domains and one target domain (see Table 2 for the detailed settings). The training dataset comprises all the labeled samples from the source domains. The samples in the target domain are used as the unlabeled training data and the test data.

We repeat the experiments 10 times with different randomly sampled training instances from each source domains and report the means and the standard deviations.

2) Email spam dataset. In email spam datasets[2)], there are three email subsets (denoted by User1, User2 and User3, respectively) annotated by three different users. In this trial, the task is to classify spam and non-spam emails. Since the spam and non-spam emails in the subsets have been identified by

1) Available at: http://people.csail.mit.edu/jrennie/20Newsgroups/
2) Available at: http://www.ecmlpkdd2006.org/challenge.html

Table 2 Description of the cross domain text datasets on multiple source domains

	Task	Data sets	Source domains	Target domain
10	20NG	rec versus sci	rec.autos & sci.crypt rec.motorcycles & sci.electronics rec.sport.baseball & sci.med	rec.sport.hockey & sci.space
11		comp versus rec	comp.graphics & rec.autos comp.os.ms-windows.misc & rec.motorcycles comp.sys.ibm.pc.hardware & rec.sport.baseball	comp.sys.mac.hardware & rec.sport.hockey
12		sci versus comp	sci.crypt & comp.graphics sci.electronics & comp.os.ms-windows.misc sci.med & comp.sys.ibm.pc.hardware	sci.space & comp.sys.mac.hardware
13		Email spam	User1 User2 User3	Public mail set

different users, the data distributions of the three subsets are different but correlative. Each subset has 2500 emails, of which half are non-spam (labeled as 1) and the other half are spam (labeled as –1). On this data set, in terms of Ref. [18], we consider three settings for single source domain adaptation learning: (i) User1 (source domain) & User2 (target domain); (ii) User2 (source domain) & User3 (target domain) and (iii) User3 (source domain) & User1 (target domain). More detailed settings information can be found in Table 1. For each setting, the training data set contains all labeled samples from the source domain. And the samples in the target domain are used as the unlabeled test ones. Again, the word-frequency feature is used to represent each document.

Again, for the case of multiple source domains adaptation learning, we consider the three user-annotated sets as three source domains, and employ the publicly available email set as the target domain (see Table 2 for more details). The training dataset comprises all the labeled samples from the source domain. The samples in the target domain are used as the unlabeled training data and also as the test data. We repeat the experiments 10 times and report the means and the standard deviations.

3) Face data sets. In this part, in order to assess the effectiveness of the proposed frameworks on multi-class classification problems with high feature dimension, we investigate the performance of the proposed method KLDAL-LSSVM with vector labeled outputs for face recognition on two benchmark databases: YALE and ORL face databases [47]. There are 165 images about 15 individuals in YALE face datasets, where each person has 11 images. The images demonstrate variations in lighting condition, facial expression, and with or without glasses. Each image is cropped at a size of 32 $\times$32 pixels in our experiment; The ORL database contains 400 images grouped into 40 distinct subjects with 10 different images for each. The images are captured at different times, and for some subjects, the images may vary in facial expressions and facial details. All the images are taken against a dark homogeneous background with the tolerance for some side movement of about 20. The original images are all sized 112$\times$92 pixels with 256 gray levels per pixel, which are further down-sampled into 32 $\times$32 pixels in our experiment. Figures 2 (a) and (c) show the cropped images of one person in YALE and ORL face database, respectively.

For the case of single source domain adaptation learning, we randomly select 8 images of each individual from YALE and ORL, respectively, to construct the source domain dataset. The target domain datasets are generated by rotating anticlockwise the original source domain dataset 3 times by 10, 30, and 50 degrees, respectively. Due to rotation, source and target-domain data exhibit different distributions. Particularly, the greater the rotation angle is, the more complex the resulting domain adaptation problem becomes [4]. Thus we construct 3 single source domain adaptation learning problems for each face database. Figures 2 (b) and (d) illustrate the face samples with rotation angle 10 degrees.

In addition, for the case of multiple source domains adaptation learning, we consider the three anticlockwise rotating face datasets from YALE and ORL, respectively, as three source domains, and employing the original face datasets, randomly selected as mentioned-above, as the target domain. The training

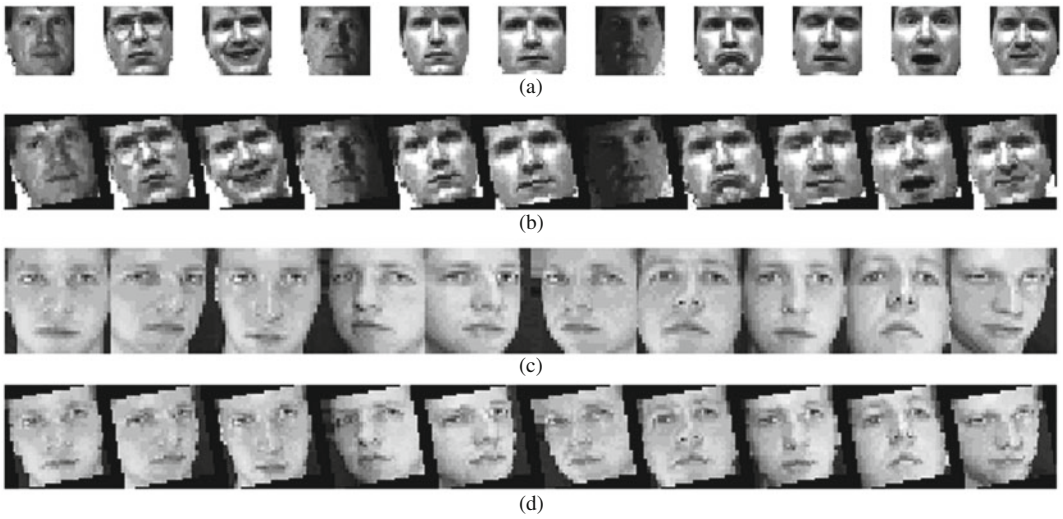

Figure 2 Face examples from the face databases Yale and ORL. (a) The processed Yale faces for an object; (b) the processed Yale faces for an object with rotation angle 10 degree; (c) the processed ORL faces for an object; (d) the processed ORL faces for an object with rotation angle 10 degree.

dataset comprises all the labeled samples from the source domains. The unlabeled samples from the target domain are used as test data.

7.1.2 *Experimental setup*

1) Detailed setup for text datasets (20NG and Email Spam). For the case of single source domain adaptation learning, besides baseline methods of the standard support vector machine (SVM) and the transductive support vector machine (TSVM), we choose for comparison three recent state-of-the-art algorithms from KDD'08 [6] that showed impressive results, out-performing baseline methods and some previous transfer learning methods in their experiments: (i) Cross domain spectral classifier (CDSC) [7] (out-performing the methods in [6] in their experiments).; (ii) Locally-weighted ensemble (LWE) classifier in [6]; and (iii) LMPROJ in [1]. We implemented their method in Matlab directly following the algorithm as presented in the paper. Moreover, for the multiple source domains adaptation applications, we compare our KLDAL based methods with three multiple source domains adaptation learning algorithms such as A-SVM, FastDAM, and Multi-KMM [48].

2) Detailed setup for face datasets. According to Theorem 5, we test the performance of KLDAL-LSSVM, in comparison with CDSC, LWE, LMPROJ, and DASVM. And for a comprehensive comparison, we also perform the baseline method LS-SVM for face recognition with different distributions. Note that for these multi-class classification tasks, KLDAL-LSSVM can use traditional OAO (one against one) multi-class separation strategy, or the vector labeled outputs strategy discussed in Subsection 4.4. We then refer to KLDAL-LSSVM in the above two cases as KLDAL-LS and KLDAL-LSSVM, respectively. Besides, DASVM, CDSC, LWE, LS-SVM and LMPROJ all adopt OAO (one against one) multi-class separation strategy to finish the corresponding multi-class classification tasks. Besides, we also compare KLDAL-LSSVM with A-SVM, FastDAM, and Multi-KMM for the case of the multiple source domains adaptation learning, in which we train OAA (One against All) SVM classifiers in A-SVM, FastDAM, and Multi-KMM.

For two baseline methods SVM and TSVM, we vary the regularization parameter C and report the best result of each method with the optimal C, where $C \in \{0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100\}$.

For LWE method in [6], we use the same three methods that they used in their experiments for the ensemble, namely the Winnow algorithm from the SNoW Learning Architecture, a logistic regression algorithm from the BBR package and the LIBSVM implementation of a support vector machine classifier [6]. We obtained parts of the code for their algorithm from the author's website http://ews.uiuc.edu/∼jinggao3/kdd08transfer.htm and implemented the rest following the algorithm. There are two important parameters in LWE, i.e., the number of clusters c' in the test set and the selection threshold τ to filter the predictions with low confidence. According to [14], we select $c' = 2$, and $\tau \in [0.5, 1]$.

Table 3 Average classification accuracies (%) of all methods on the text datasets for single source domain adaptation learning

Methods	Data sets								
	20NG						Email spam		
	1	2	3	4	5	6	7	8	9
SVM	72.53*	70.10*	75.40*	78.00*	83.80	92.70*	96.08	96.89	91.7*
TSVM	76.75*	73.40*	83.90	81.20*	85.24	88.74*	96.21	97.0	91.80*
CDSC	69.80*	**82.92**	64.00*	70.84*	82.72*	90.20*	83.28*	92.14*	90.02*
LWE	85.24	78.60*	87.20	75.32*	**88.30**	94.00*	93.51*	98.74	88.78*
LMPROJ	82.52*	79.30	86.34*	84.68	85.40*	93.43*	93.21*	94.0*	88.79*
DASVM	82.91	81.10	**87.83***	84.55	87.00*	94.73*	96.89	**97.65**	94.50
KLDAL-SVM	83.73	81.40	86.71	84.92	86.20	94.80	96.49	97.25	93.20
μ-KLDALSVM	**84.04**	82.34	86.71	**85.05**	86.84	**95.23**	**97.19**	97.25	**93.85**

$*$ The performance of μ-DAKSVM is statistically significant compared with other classifiers at p-value$\leqslant$0.05.

In terms of [7], for CDSC method, we use a cosine similarity measure $k(x, y) = \langle x, y\rangle / \|x\| \|y\|$, commonly used in text mining. We use the same initialization and parameters in [7]. According to [7], we select the optimal parameters in CDSC as $\lambda = 0.025$, and $\beta = 15$.

For DASVM method, LIBSVM is used to train both the supervised SVMs in the first step, and with proper modifications, the proposed DASVMs. As pointed out in [4], we fixed the parameters τ =0.5 and β=0.03.

We empirically set the parameter α in Multi-KMM at 1. The Multi-KMM classifier is finally learned by employing the shifted samples from the source domains.

In A-SVM, FastDAM and our KLDAL based methods (for the case of multiple source domains adaptation), we need to determine the weight γ_p for the pth source classifier. For fair comparison, we set $\gamma_p = \exp(-\omega\gamma_{KM}(D^s, D^t))$ for A-SVM and FastDAM, but $\gamma_p = \exp(-\omega\gamma_{KMS}(D^s, D^t))$ for our KLDAL based methods, $\omega \in \{0, 1, 10, 100, 1000, 10000\}$ is the bandwidth parameter to control the spread of $\gamma_{KM}(D^s, D^t)$ and $\gamma_{KMS}(D^s, D^t)$, where $\gamma_{KM}(D^s, D^t)$ and $\gamma_{KMS}(D^s, D^t)$ are the maximum distribution mean distance (MMD) and maximum distribution distance as defined in Definition 4, respectively, on RKHS Embedding domain distributions between source domain D^s and target domain D^t. In the experiments, we further normalize the sum of the weights γ_p's as 1 and empirically set $\omega = 100$. Besides, we set regularization parameter C to default value 1 for A-SVM and FastDAM, and set $\lambda_L = \lambda_D = 1$ for FastDAM.

For our KLDAL based methods, there are three tunable parameters C, λ, γ. In practice, we first fix $\lambda = 1$, and $C = 100$. Then, we tune the parameters to obtain the optimal ones in terms of the best test accuracy. The tunable parameter γ can be set by minimizing γ_{KMS} to get the most optimal target test. We also perform detailed parameter sensitivity analysis to show how the performance is affected by each of the parameters in our methods in Subsection 7.3.

7.1.3 *Experimental results*

Tables 3–6 show the overall classification accuracy rate of different classifiers on the 20NG, email spam, and Face datasets in the case of single and multiple source domains adaptation learning, respectively. From these results, we can make several interesting observations as follows.

1) The baseline classifier has the worst performance on almost all learning tasks among all classifiers in Tables 3 and 5. It is worth noting that we obtain a little better results on SVM and TSVM than that typically reported in the previous literatures (e.g. [1–3] and [16]) on the same datasets used in our trials. This is because in our trials instead of selecting a default parameter on the training data to be performed and in order to allow a fair comparison with the domain adaptation learning methods we reported the best results over a set of parameters for these baseline methods.

2) In Table 3, we can observe that six classifiers, i.e, CDSC, LWE, LMPROJ, DASVM, KLDAL-SVM and its variation μ-KLDALSVM, exhibit comparable recognition capacity on both 20NG and Email spam datasets. Moreover, it is worthy to note that the proposed method KLDAL-SVM and its variant

Table 4 Average classification accuracies (%) of all methods on the text datasets for multiple source domains adaptation learning

Datasets		Methods				
		A-SVM	FastDAM	Multi-KMM	KLDAL-SVM	μ-DAKSVM
20NG	10	95.84	97.72*	97.78*	**98.02**	**98.02**
	11	97.58*	98.83	**98.91***	98.76	98.80
	12	92.16	94.52*	92.00	93.86	**95.30**
Email spam	13	70.13*	83.42*	73.44*	85.29	**85.52**

$*$ The performance of μ-DAKSVM is statistically significant compared with other classifiers at p-value$\leqslant$0.05.

Table 5 Average classification accuracy (%) of all methods on face datasets for single source domain adaptation learning

Faces data		Methods						
		LS-SVM	LMPROJ	LWE	CDSC	DASVM	KLDAL-LS	KLDALLSSVM
YALE	10 degree	61.78	68.45	63.78	62.47	68.68	69.93	**70.24**
	30 degree	58.37	64.13	61.66	60.70	65.28	66.20	**66.47**
	50 degree	52.29	62.08	58.78	60.20	63.18	**63.70**	63.00
ORL	10 degree	76.30	85.94	80.90	84.64	85.84	84.18	**86.28**
	30 degree	70.72	82.00	79.33	83.71	**84.02**	83.4	83.10
	50 degree	65.70	78.65	72.22	**79.91**	79.85	79.74	79.25

μ-KLDALSVM demonstrate significantly high classification accuracy in most cases, which validates that it is more stable than other classifiers. Besides, the results in Table 3 also show that the proposed method KLDAL-SVM and its variant μ-KLDALSVM perform somewhat better than LMPROJ in almost all datasets, which justifies that the only emphasis on minimizing distribution mean discrepancy between both domains is far from sufficient for domain adaptation learning. Hence, we should introduce more underlying information, such as distribution scatter discrepancy minimization, into the regularization framework of the classifier to further enhance the classification performance. Although DASVM also obtained comparable classification accuracy with respect with our methods on some datasets in some extent, in practice, we found that DASVM possessed relatively higher time complexity than other classifiers. A possible explanation is that the circular validation strategy in DASVM increased the running time of DASVM, thus degrading its convergence performance. In addition, form Table 3, we can see that μ-KLDALSVM keeps obviously superior capacity over KLDAL-SVM in classification accuracy for almost all datasets, which demonstrates that parameter μ can be used to enhance the generalization capability of KLDAL-SVM. Therefore, we use μ-KLDALSVM (or μ-MKLDALSVM) instead of KLDAL-SVM (or MKLDAL-SVM) for the performance evaluation hereinafter.

3) The overall accuracy of LS-SVM is lower than any other classifier on all domain adaptation learning tasks, which is consistent with SVM. With the increase of rotation angle, the classification performance of all classifiers declines gradually. However, DASVM and KLDAL-LSSVM (or KLDAL-LS) seem to decrease more slowly than other methods. Exceptionally, CDSC exhibits competitive performance to some extent with other methods, particularly on more complex data sets. A possible explanation is that spectral technique can improve the DAL performance in certain extent.

4) As shown in Table 5, the KLDAL-LS, KLDAL-LSSVM, and DASVM methods deliver more stable results across all the datasets than other classifiers. However, KLDAL-LSSVM obtains the best classification accuracy more frequently than any other methods. Throughout the trials in this part, we found that KLDAL-LSSVM took less time than KLDAL-LS, and DASVM on a majority of the datasets. Thus, we can assume that KLDAL-LSSVM is competitive with the best method for a majority of all datasets with respect to performance and computational complexity. Hence, as discussed in Subsection 4.4, KLDAL-LSSVM possesses overall domain adaptation learning advantages over other methods in computational complexity and classification accuracy, which further verifies that KLDAL-LSSVM with vector labeled outputs not only decreases the computational complexity for multiple-class classification but also

Table 6 Average classification accuracy (%) of all methods on face datasets for multiple source domains adaptation learning

Datasets	Methods			
	A-SVM	FastDAM	Multi-KMM	KLDAL-LSSVM
YALE	72.14	77.36	74.58	**78.08**
ORL	88.00	92.50	90.76	**92.84**

improves the classification performance in some cases. Table 5 also shows that although KLDAL-LSSVM seems to have overall advantage over KLDA-LS in classification accuracy, KLDAL-LS is actually considerably comparable to KLDAL-LSSVM in some extent.

5) For the multiple source domains adaptation learning in Tables 4 and 6, we can observe that A-SVM achieves good results by only using the labeled instances from the source domains, possibly because some source domains are highly relevant to the target domain. This conjecture is also supported by measuring the distances between the source domains and the target domain with the MMD criterion. Multi-KMM are generally better than A-SVM and FastDAM, which demonstrates that Multi-KMM can successfully shift the means of source domains toward the target domain on these datasets. However, our KLDAL based methods outperform other algorithms in most cases when there are no labeled target samples.

6) The results in Tables 3–6 clearly demonstrate that our KLDAL based methods can learn a robust target classifier for domain adaptation by leveraging a set of source classifiers. Hence, we can conclude that our KLDAL based methods using the source classifiers can improve the performance of domain adaptation learning.

7) In order to examine whether the proposed methods under the framework KLDAL are significantly better than the other methods, we performed the paired two-tailed t-test [29] on the classification results of the 10 runs to calculate the statistical significance of the proposed method μ-KLDALSVM. The smaller the p-value, the more significant the difference of the two average results and a p-value of 0.05 is a typical threshold which is considered to be statistically significant. Thus, in Tables 3 and 4, if the p-value of each dataset is less than 0.05, the corresponding results will be denoted by "*". Therefore, in Tables 3–6, we can clearly find that the proposed method μ-KLDALSVM possesses significantly better classification performance than other classifiers in most datasets. This just verifies the consistency with our conclusions obtained above.

7.2 Experiments on large-scale datasets

In this subsection, we demonstrate the efficiency and effectiveness of the proposed framework MKLDAL based method on a large-scale dataset, i.e., TRECVID dataset.

7.2.1 *Description of data set*

The TRECVID video corpus[3)] is one of the largest annotated video benchmark data set for research purposes. The TRECVID 2005 data set contains 61901 key-frames extracted from 108 hours of video programs from six broadcast channels, and the TRECVID 2007 data set contains 21532 key-frames extracted from 60 hours of news magazine, science news, documentaries and educational programming videos [13,33]. As shown in [33], TRECVID data set is a challenge for DAL methods due to the large difference between TRECVID 2007 data set and TRECVID 2005 data set in terms of program structure and production values. 36 semantic concepts are chosen from the LSCOM-lite lexicon [49], which covers 36 dominant visual concepts present in broadcast news videos. The 36 concepts have been manually annotated to describe the visual content of the key-frames in both TRECVID 2005 and 2007 data sets. Three low-level global features grid color moment (225 dimension), Gabor texture (48 dimension) and edge direction histogram (73 dimension) are extracted to represent the diverse contents of key-frames, because of their consistent good performances reported in TRECVID [33,34]. Moreover, the three types

3) http://www-nlpir.nist.gov/projects/trecvid

of global features can be efficiently extracted, and the previous works [33,34] also show that the cross-domain issue exists when using these global features. We further concatenate the three types of features to form a 346-dimensional feature vector for each key-frame.

7.2.2 *Experimental setup*

We systematically compare our proposed method μ-MKLDALSVM with the baseline SVM, and other cross-domain learning algorithms including A-SVM, cross-domain SVM (CD-SVM), DTSVM, and kernel mean matching (KMM). Note that μ-MKLDALSVM and the standard SVM can use the labeled training data set D^s from the source domain, or the combined training data set $D^s \cup D_l^t$ from both source and target domains, where D_l^t is the labeled training data set from target domain. We then refer to μ-MKLDALSVM and SVM in the above two cases as μ-MKLDAL_A, μ-MKLDAL_AT, SVM_A and SVM_AT, respectively. The cross-domain learning methods A-SVM, CD-SVM, KMM, and DTSVM also make use of the combined training data set $D^s \cup D_l^t$ for model learning. 4000 unlabeled samples from the target domain are randomly selected as the unlabeled training data set D_u^t for model learning in CD-SVM, KMM, DTSVM, and our MKLDAL based methods.

DTSVM and our MKLDAL based methods can make use of multiple base kernels. For fair comparison, we use the same kernels for other methods including SVM_A, SVM_AT, A-SVM, CD-SVM and KMM. Note that we make use of the unlabeled target training data D_u^t in KMM, LMPROJ, DTSVM and our MKLDAL based methods for distribution distance measure. The labeled and unlabeled training data are employed to measure the data distribution mismatch between two domains for KMM, and DTSVM using the MMD criterion and for our MKLDAL based methods using distribution distance metric criterion on RKHS embedding domain distributions as in Definition 4.

We give the best performance for each method over a range of parameters, and for the A-SVM, CD-SVM and KMM methods we center this range on the best performing parameters reported in their respective papers. For KMM, the parameter B is empirically set at 0.99. For all methods, We fix the regularization parameter C at the default value 1 in LIBSVM [45] for the large scale TRECVID data set, because it is time-consuming to run the experiments multiple times using different C. For DTSVM, the parameter θ in DTSVM needs to be determined beforehand. According to [13], we empirically set $\theta = 1$. And for our methods, the parameters λ, γ in our methods need to be selected beforehand. In practice, we first fix $\lambda = 1$. Then, we tune the parameter λ to obtain the optimal ones in terms of the best test accuracy. The tunable parameter γ can be set by minimizing γ_{KMS} to get the most optimal target test.

7.2.3 *Experimental results*

For performance evaluation, we use non-interpolated average precision (AP) [13] which has been used as the official performance metric in TRECVID since 2001. AP is related to the multi-point average precision value of a precision-recall curve, and incorporates the effect of recall when AP is computed over the entire classification results.

We compare our MKLDAL based methods with other algorithms on the challenging TRECVID data set for the video concept detection task. Similarly as in [13], we group 36 concepts into three categories according to the frequency of positively labeled samples in the TRECVID 2007 data set. The first group consists of 12 concepts with high positive frequency (more than 0.05), the second group consists of 11 concepts with moderate positive frequency ($0.01 \leqslant$positive frequency$\leqslant 0.05$), and the third group consists of the remaining 13 concepts with low positive frequency (less than 0.01). In Figure 3, we use three rows to show the per-concept AP for the three groups. Table 7 gives the Mean Average Precision (MAP) of the concepts of three groups and all 36 concepts, referred to as Group-1, Group-2, Group-3 and Group-ALL, respectively. From Figure 3 and Table 7, we have the following interesting observations:

1) SVM_A is much worse than other classifiers according to the MAPs over all the 36 concepts, which demonstrates that the SVM classifier learned with the training data from the source domain performs poorly on the target domain. The explanation is that the data distributions of TRECVID data sets collected in different years are quite different. It is interesting to observe that SVM_AT outperforms

Table 7 Performance comparison of MKLDAL-based methods with other methods in terms of mean AP from concepts of three groups including 36 concepts

	SVM_A	SVM_AT	A-SVM	CD-SVM	KMM	DTSVM	μ-MKLDAL_A	μ-MKLDAL_AT
Group-1	37.02%	39.79%	40.85%	40.06%	40.22%	44.98%	45.33%	**45.99%**
Group-2	12.32%	12.91%	12.62%	12.60%	12.83%	15.29%	15.36%	**15.94%**
Group-3	13.20%	14.95%	14.88%	13.18%	12.93%	16.91%	17.05%	**17.51%**
Group-All	20.85%	22.55%	22.72%	21.95%	21.99%	25.73%	25.91%	**26.48%**

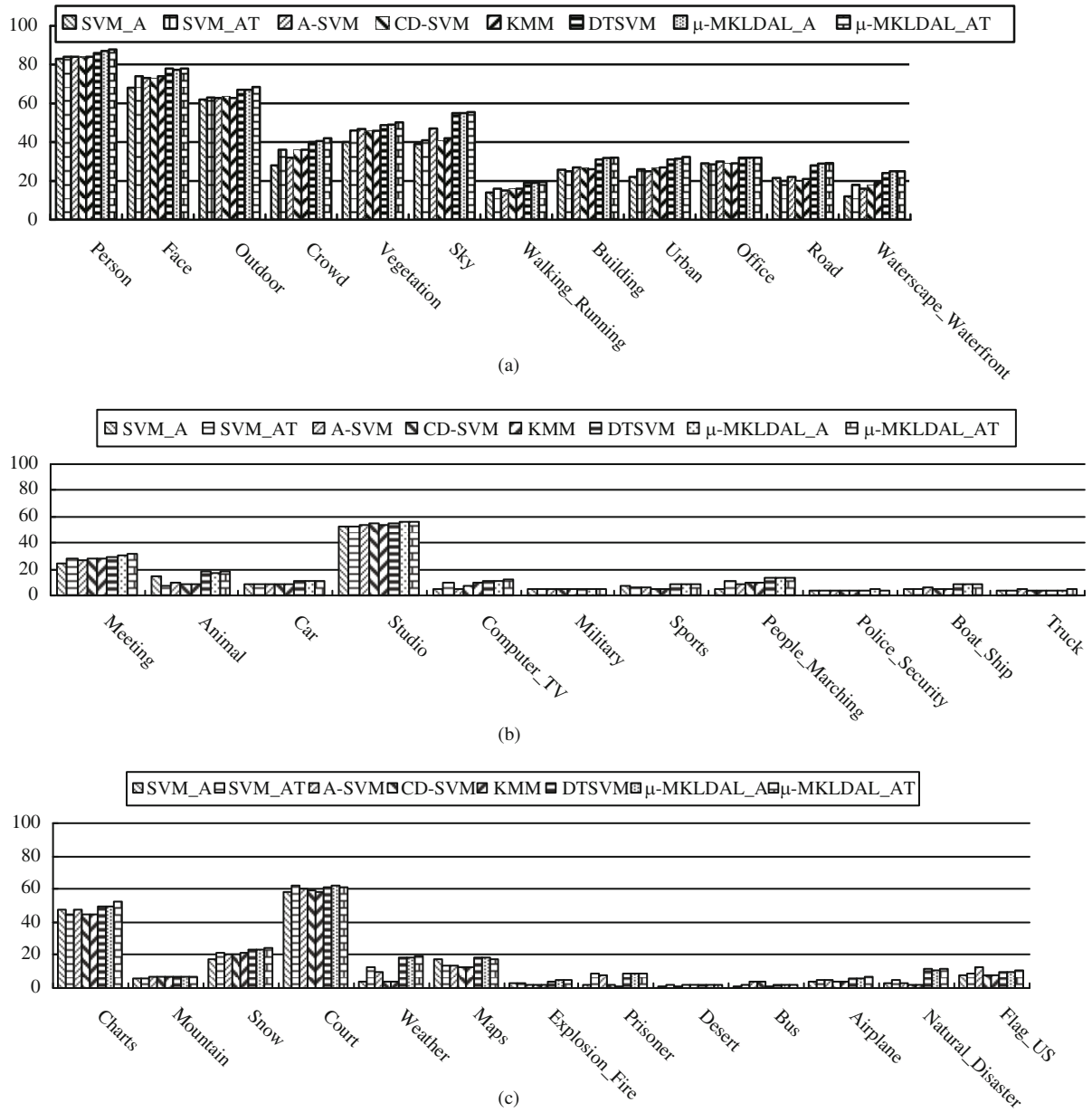

Figure 3 Performance comparison of MKLDAL-based methods with other methods on all 36 concepts. (a) Group-1; (b) Group-2; (c) Group-3.

A-SVM, CD-SVM, and KMM in terms of Group-1 in some extent, but A-SVM, CD-SVM, and KMM are somewhat better than SVM_AT in terms of Group-3. The explanation is that the concepts in Group-1 generally have a large number of positive patterns in both source and target domains. Intuitively, when sufficient positive samples exist in both domains, the samples distribute densely in the feature space. In this case, the distributions of samples from two domains may overlap [13], and thus, the data from the

source domain may be helpful for video concept detection in the target domain. On the other hand, for the concepts in Group-3, the positive samples from both domains distribute sparsely in the feature space. It is more likely that there is less overlap between the data distributions of two domains. Therefore, for the concepts in Group-3, the data from the source domain may degrade the performance for video concept detection in the target domain.

2) SVM_AT outperform SVM_A in terms of MAPs from all the three groups, which demonstrates that the information from the target domain can be effectively used in SVM to improve the classification performance in the target domain. We also observe that KMM and CD-SVM are slightly worse than SVM_AT in terms of Group-All. A possible explanation is that in CD-SVM, k-nearest neighbors from the target domain are used to define the weights for the source patterns. When the total number of positive training samples in the target domain is very limited (e.g., 10 positive samples per concept in this work), the learned weights for the source patterns are not reliable, which may degrade the performance of CD-SVM. Similarly, KMM learns the weights for the source samples in an unsupervised setting without using any label information, which may not be as effective as other cross-domain learning methods.

3) DTSVM, μ-MKLDAL_A, and μ-MKLDAL_AT outperform all other methods in terms of MAPs from all the three groups and achieve the best results in almost all of 36 concepts. These results clearly demonstrate that DTSVM and our MKLDAL based methods can successfully minimize the data distribution mismatch between two domains and the structural risk functional through effective combination of multiple base kernels. However, μ-MKLDAL_A and μ-MKLDAL_AT are better than DTSVM in terms of Group-ALL, because of the additional consideration of the scatter discrepancy between two domains in our methods. Moreover, μ-MKLDAL_AT is a little better than μ-MKLDAL_A in terms of MAP from all the three groups. The reason is the same as that for SVM_AT vs. SVM_A in 1).

7.3 Experiments on parameter sensitivity

In this subsection, in order to explicitly explain the parameters influence on the classification performance of the proposed method KLDAL-SVM, we give the experimental results about parameter sensitivity in Figure 4 for the accuracy criterion and the four parameters $C, \lambda, \gamma, \omega$ in the case of multiple source domains adaptation learning. In our experiments, we take $\gamma_0 = 10$ in terms of Theorem 6. For each plot in Figure 4, three parameters are fixed at the best values while the fourth parameter is varied to generate the plots. Here we show representative results on a couple of text datasets in Table 1, including the third, seventh, tenth, and thirteenth datasets. In Figures 4 (a), (b), and (c), we show the sensitivity results of the parameters C, λ, γ over the first two datasets and in Figure 4(d) we illustrate the sensitivity result of the parameter ω on the last two datasets. In Figure 4, we can observe several results as follows.

1) As shown in Figure 4 (a), the proposed method, which is based on structure risk minimization model, is considerably sensitive to regularization parameter C for a wide range of values. And as C varies smoothly, the accuracy of the proposed method significantly changes accordingly, which verifies the importance of C to be tuned.

2) In Figure 4 (b), it is shown that when λ=0, i.e., ignoring of distribution scatter discrepancy between source and target domains, the proposed method cannot achieve the optimal performance. As λ increases, the performance of the proposed method will become a little better, and levels off to a maximum for a wide range of parameters. However, when λ=1, i.e., ignoring of distribution mean discrepancy between source and target domains, the performance significantly declines. From the above analysis, we can conclude that in domain adaptation, learning a classifier by only minimizing the distribution mean distance or variance distance between two different domains may not be enough, and only when simultaneously considering both the distribution mean distance and variance distance between two different domains, can we obtain the optimal classification performance.

3) In Figure 4 (c), we see that as explained in Theorem 6, smaller values of γ (e.g. $\gamma \in [1, 2)$), i.e. larger values of Gaussian kernel bandwidth, tend to decrease the convergence rate of the distribution scatter between two different domains due to the increase of the distribution scatter of intra-domains, while larger values of γ (e.g. $\gamma \in [6, +\infty)$), i.e. smaller values of Gaussian kernel bandwidth, lead to the class overlapping of intra-domains due to the high cohesion of data distributions of intra-domains. Both these

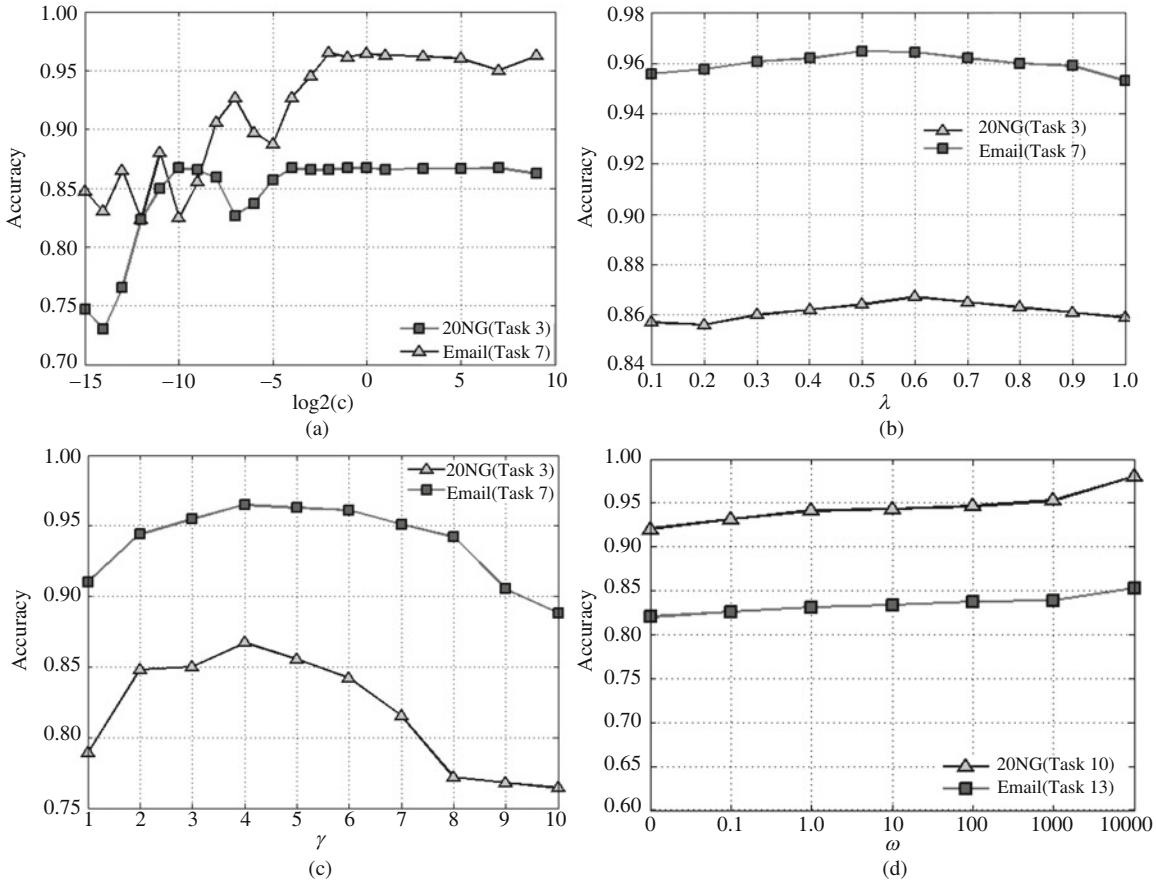

Figure 4 Parameters sensitivity. (a) Parameter C sensitivity; (b) parameter λ sensitivity; (c) parameter γ sensitivity; (d) parameter ω sensitivity.

cases can lead to a degradation in the classification performance of the proposed method. Only in some moderate range of values (e.g. $\gamma \in [2, 6)$), dose the proposed method achieve relatively high performance.

4) Recall that ω is the bandwidth parameter for the calculation of γ_p. In Figure 4 (d), we show the performance variations of KLDAL-SVM with respect to ω. When setting ω=0, we have equal weights for all source domains (i.e., $\gamma_p = 1/P$, $\forall p = 1, 2, \ldots, P$). From Figure 4 (d), we observe that our method achieves better performances with ω=10000 than with ω=0, which demonstrates that it is beneficial to adopt the minimum distribution discrepancy metric criterion, as defined in Definition 4, to measure the distribution relevance between each source domain and the target domain.

8 Conclusions and future work

In this paper, we attempt to address domain adaptation learning problems by the proposed framework KLDAL and its extension MKLDAL, which borrows the multiple kernel learning framework, using regularization with the goal of structure risk minimization of a classifier while at the same time minimizing both mean and scatter discrepancy between two distributions of source and target domains. The proposed framework KLDAL (or MKLDAL) based methods extend the principle of SVMs to the domain adaptation framework by taking into account the fact that unlabeled test samples are drawn from a target domain D^t different from the source domain $D^s$of training samples. It is worth noting that the proposed KLDAL (or MKLDAL) based methods are designed to address a problem conceptually different from those faced by transductive and semi-supervised SVMs, which have been defined for handling problems where labeled and unlabeled data are drawn from the same domain. Thus, they are ineffective in domain adaptation, where training data are assumed to be available only for a source domain different (even if related) from the target domain of the (unlabeled) test samples. With extensive experimental study on toy and real-world data sets we demonstrate the effectiveness of the proposed methods, comparing them

with some recent state-of-the-art methods. Our results demonstrate the effectiveness of this viewpoint of using such regularization as mentioned above to find a decision function that brings the source and target domain distributions together so that the source domain data can be effectively exploited.

We plan to continue our research work in this direction in the future, by pursuing several promising avenues. First, we plan to validate the power of the proposed methods more extensively through other kernel functions. Second, in this paper, we only investigate the case where the source, target and auxiliary data sources share the same feature space. We plan to extend and apply our methods to enable it to deal with heterogeneous transfer learning [3]. In the framework of domain adaptation, due to the absence of prior information for target-domain, traditional statistical validation strategies proposed in the previous literature somewhat cannot be used for assessing the effectiveness of the resulting classifier. Hence, in near future, we also expect to investigate an effective validation strategy to validate the solutions consistent with D^s and D^t.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China (Grants Nos. 60903100, 60975027, 2011NSFC), Natural Science Foundation of Jiangsu Province (Grants Nos. BK2009067, 2011NSF_JS), Natural Science Foundation of Ningbo City (Grant No. 2009A610080) and 2011 Postgraduate Student's Creative Research Fund of Jiangsu Province.

References

1 Quanz B, Huan J. Large margin transductive transfer learning. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM). New York: ACM , 2009. 1327–1336
2 Pan S J, Tsang I W, Kwok J T, et al. Domain adaptation via transfer component analysis. IEEE Trans Neural Netw, 2011, 22: 199–210
3 Xiang E W, Cao B, Hu D H, et al. Bridging domains using world wide knowledge for transfer learning. IEEE Trans Knowl Data Eng, 2010, 22: 770–783
4 Bruzzone L, Marconcini M. Domain adaptation problems: A DASVM classification technique and a circular validation strategy. IEEE Trans Pattern Anal Mach Intell, 2010, 32: 770–787
5 Blitzer J, Crammer K, Kulesza A, et al. Learning bounds for domain adaptation. In: Proceedings of The Neural Information Processing Systems (NIPS). Cambridge: MIT Press, 2007. 129–136
6 Gao J, Fan W, Jiang J, et al. Knowledge transfer via multiple model local structure mapping. In: Proceedings of the 14th ACM SIGKDD conference on Knowledge Discovery and Data Mining. New York: ACM, 2008
7 Ling X, Dai W, Xue G, et al. Spectral domain transfer learning. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2008
8 Pan S J, Yang Q. A survey on transfer learning. IEEE Trans Knowl Data Eng, 2010, 22 : 1345–1359
9 Blitzer J, McDonald R, Pereira F. Domain adaptation with structural correspondence learning. In: Proceedings of 2006 Conference Empirical Methods Natural Lang, Sydney, 2006. 120–128
10 Blitzer J, Dredze M, Pereira F. Boom-Boxes and Blenders: domain adaptation for sentiment classification. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL '07), Pereira, 2007. 440–447
11 Mansour Y, Mohri M, Rostamizadeh A. Domain adaptation: learning bounds and algorithms. In: The 22nd Annual Conference on Learning Theory (COLT 2009), Montreal, 2009
12 Daume' III H. Frustratingly easy domain adaptation. In: Proceedings of Annual Meeting Association for Computational Linguistics, Prague, 2007. 256–263
13 Duan L X, Tsang I W, Xu D, et al. Domain transfer SVM for video concept detection. In: Proc IEEE Int'l Conf Computer Vision and Pattern Recognition, Miami, 2009. 1375–1381
14 Gretton A, Harchaoui Z, Fukumizu K, et al. A fast, consistent kernel two-sample test. In: Lafferty J, Williams C K I, Shawe-Taylor J, et al, eds. Advances in Neural Information Processing Systems, Vancouver, 2010. 673–681
15 Huang J, Smola A, Gretton A, et al. Correcting sample selection bias by unlabeled data. In: Proceedings of Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, 2006
16 Duan L X, Xu D, Tsang I W. Domain adaptation from multiple sources: a domain-dependent regularization approach. IEEE Trans Neural Netw Learn Syst, 2012, 23: 504–518
17 Duan L X, Tsang I W, Xu D, et al. Domain adaptation from multiple sources via auxiliary classifiers. In: Proceedings of the 26th International Conference on Machine Learning (ICML 2009), Montreal , 2009

18 Duan L X, Tsang I W, Xu D. Domain transfer multiple kernel learning. IEEE Trans Pattern Anal Mach Intell, 2012, 34: 465–479

19 Bach F R, lanckriet G R G, Jordan M. Multiple kernel learning, conic duality, and the SMO algorithm. In: Proc Int'l Conf Machine Learning. Banff: IEEE Press, 2004

20 Rakotomamonjy A, Bach F R, Canu S, et al. SimpleMKL. Mach Learn Res, 2008, 9: 2491–2521

21 Sonnenburg S, Ra tsch G, Scha fer C, et al. Large scale multiple kernel learning. J Mach Learn Res, 2006, 7: 1531–1565

22 Hu M Q, Chen Y Q, Kwok J T Y. Building sparse multiple-kernel SVM classifiers. IEEE Trans Neural Netw, 2009, 20

23 Rosenstein M T, Marx Z, Kaelbling L P. To transfer or not to transfer. In: Advances in Neural Information Processing Systems. Cambridge: MIT Press, 2005

24 Seah C W, Tsang I W, Ong Y S, et al. Predictive distribution matching SVM for multi-domain learning. In: Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2010), Barcelona, 2010

25 Luo P, Zhuang F, Xiong H, et al. Transfer learning from multiple source domains via consensus regularization. In: Proc. ACM Conf. Inf. Knowledge. Management, Napa Valley, 2008. 103–112

26 Mansour Y, Mohri M, Rostamizadeh A. Domain adaptation with multiple sources. In: Advances in Neural Information Processing Systems 21. Cambridge: MIT Press, 2009. 1041–1048

27 Crammer K, Kearns M, Wortman J. Learning from multiple sources. J Mach Learn Res, 2008, 9: 1757–1774

28 Scholkopf B, Herbrich R, Smola A J. A generalized representer theorem. In: Proc. COLT'2001. Amsterdam: Springer Press, 2001. 416–426

29 Vapnik V. Statistical Learning Theory. New York: John Wiley and Sons, 1998

30 Schölkopf B, Smola A J, Williamson R, et al. New support vector algorithms. Neural Comput, 2000, 12: 1207–1245

31 Joachims T. Transductive inference for text classification using support vector machines. In: Bratko I, Dzeroski S, eds. Proceedings of ICML-99, 16th International Conference on Machine Learning. San Francisco: Morgan Kaufmann Publishers, 1999. 200–209

32 Suykens J A K, Vandewalle J. Least squares support vector machine classifiers. Neural Process Lett, 1999, 9: 293–300

33 Jiang W, Zavesky E, Chang S F, et al. Cross-domain learning methods for high-level visual concept classification. In: Proc IEEE Int'l Conf Image Processing, San Diego, 2008. 161–164

34 Yang J, Yan R, Hauptmann A G. Cross-domain video concept detection using adaptive SVMs. In: Proc ACM Int'l Conf Multimedia, Augsburg, 2007. 188–197

35 Ben-David S, Blitzer J, Crammer K, et al. Analysis of representations for domain adaptation. In: The Neural Information Processing Systems, Cambridge: MIT Press, 2007

36 Ben-David S, Luu T, Lu T, et al. Impossibility theorems for domain adaptation. J Mach Learn Res, 2010, 9: 129–136

37 Ben-David S, Blitzer J, Crammer K, et al. A theory of learning from different domains. Mach Learn, 2010, 79: 151–175

38 Zhu X. Semi-supervised learning literature survey. Madison: Department of Computer Science, University of Wisconsin. Technical Report. 2008

39 Hofmann T, Schölkopf B, Smola A J. Kernel methods in machine learning. Annal Stat, 2007, 36: 1171–1220

40 Sriperumbudur B K, Gretton A, Fukumizu K, et al. Hilbert space embeddings and metrics on probability measures. J Mach Learn Res, 2010, 11: 1517–1561

41 Sriperumbadur B K, Fukumizu K, Gretton A, et al. Kernel choice and classifiability for RKHS embeddings of probability distributions. In: Advances in Neural Information Processing Systems 21. Cambridge: MIT Press, 2010. 1750–1758

42 Wu Y, Liu Y. Robust truncated hinge loss support vector machines. J Am Stat Assoc, 2007, 102: 974–983

43 Tao J W, Wang S T. Locality-preserved maximum information variance v-support vector machine. Acta Autom Sin, 2012, 38: 97–108

44 Szedmak S, Shawe-Taylor J. Muticlass learning at one-class complexity. Southampton: School of Electronics and Computer Science. Technical Report No: 1508. 2005

45 Chang C C, Lin C J. Training v-support vector classifiers: Theory and algorithms. Neural Comput, 2001, 13: 2119-2147

46 Dai Q Y W, Xue G R, Yu Y. Co-clustering based classification for out-of-domain documents. In: Proceedings of the Thirteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, 2007. 210–219

47 Cai D, He X F, Han J W. Orthogonal Laplacianfaces for face recognition. IEEE Trans Image Process, 2006, 15: 3608–3614

48 Schweikert G, Widmer C, Schölkopf B, et al. An empirical analysis of domain adaptation algorithms for genomic sequence analysis. In: Advances in Neural Information Processing Systems 21. Cambridge: MIT Press, 2009. 1433–1440

49 Naphade M R, Smith J, Tesic J, et al. Large-scale concept ontology for multimedia. IEEE Multimed Mag, 2006,13: 86–91